University of M'Hamed Bougara, Boumerdes

Faculty of Sciences

ICCSAITCS '2022

The First International Conference On Cyber Security, Artificial Intelligence and Theoretical Computer Science

Proceeding of the conference

December 19 – 20, 2022, Boumerdes, Algeria

Honorary General Chairs

Pr. Mostepha YAHI, Rector of Univ. Boumerdes, Algeria

Pr. Amar RIBA, Dean of the Faculty of Sciences, Univ. Boumerdes, Algeria

Conference General Chair

Pr. Mohamed Amine RIAHLA, Univ. Boumerdes, Algeria

Conference Co-Chairs

Pr. Mohamed MEZGHICHE, Univ. Boumerdes, Algeria

Pr. M'hamed HAMADOUCHE, Univ. Boumerdes, Algeria

Pr. Menouar BOULIF, Univ. Boumerdes, Algeria

Pr. Ali BERRICHI, Univ. Boumerdes, Algeria

Dr. Rabah IMACHE , Univ. Boumerdes, Algeria

Preface

The M'Hamed Bougara university is very pleased to welcome you in Boumerdes at the The First International Conference On Cyber Security, Artificial Intelligence and Theoretical Computer Science (ICCSAITCS'22) held in the Faculty of Sciences.

The ICCSAITCS'22 aims for presenting new advances and research results in the fields of:

- Cyber Security and networking,
- Artificial Intelligence and. optimization techniques
- Formal methods for software and hardware.

The conference is expected to provide researchers, lecturers, engineers, and scientists around the world the opportunity to interact and present their latest advanced research in these fields.

We deeply thank the organizing committee and all those who by their help, devotion, competence, and good mood, allowed the good progress of these two days of the ICCSAITCS'22.

We also thank all the participants who insured the animation of the conference, and allowed fruitful exchanges and discussions profitable to everybody

Organizing Committee

1. Pr. Mohamed Amine RIAHLA
2. Pr. Mhamed HAMMADOUCHE
3. Pr. Ali Berrichi
4. Pr. BOULIF Menouar
5. Dr. Rabah IMACHE
6. Dr. Youcef YAHIATENE
7. Dr. Abdellah REZOUG
8. Dr. Fayçal TOUAZI
9. Dr. Dhia Eddine SALHI
10. Dr. Hocine MOKRANI
11. Dr. Rachid DJERBI
12. Dr. Mohamed BENNAI
13. Dr. Zahira CHOUIREF
14. Dr. Razika LOUNAS
15. Dr. Asma KHOUDI
16. Mrs. Besma ALOUANE
17. Dr. Selma DJEDDAI
18. Mrs. Safia GUENADIZ
19. Dr. Ibtihel BADDARI
20. Mrs. Wahiba OTMANINE
21. Mrs. Safia Guenadiz
22. Mrs. Drifa Hadjidj
23. Dr. Hamadouche Samiya
24. Dr. Mesbah Abdelhak
25. Dr. KHELIFI Abdelkarim
26. Dr. SIHEM GOUMIRI
27. Dr. ABET Rami

Scientific Committee

1. Pr. MOHAMED AMINE Riahla
2. Dr. BADACHE Ismail
3. Dr. BELGACEM Ali
4. Dr. BELKACEM Samia
5. Dr. BERRICHI Ali
6. Dr. BOUDANE Fatima
7. Dr. BOUDJELABA Hakim
8. Dr. BOUKELLOUZ Wafa
9. Pr. BOULIF Menouar
10. Dr. CHAOUCHE Ali
11. Dr. CHEKKAI Nassira
12. Dr. Chouiref ZAHIRA
13. Dr. DJEDDAI Selma
14. Dr. DJERBI Rachid
15. Dr. FERRAHI Ibtissam
16. Dr. GUERBAI Yasmine
17. Dr. HADJIDJ Drifa
18. Dr. HAMADOUCHE Samiya
19. Dr. HARRAR Khaled
20. Dr. IMACHE Rabah
21. Dr. ISHAKBOUSHAKI Saida
22. Dr. KEDJAR Saadia
23. Dr. KHOUDI Asmaa
24. Dr. LEBBAH Fatima zahra
25. Dr. MAHDI Ismahane
26. Dr. MAOUCHE Amin riadh
27. Pr. Ammar Mohamed
28. Dr. MOKRANI Hocine
29. Dr. NEKKAA Messaouda
30. Dr. OUKAS Nourredine
31. Dr. RAZIKA Lounas

32. Dr. REZZOUG Abdellah
33. Dr. SALHI Dhaieddine
34. Dr. SOUAD Kherroubi
35. Dr. Tiberkak ALLAL
36. Dr. TOUAZI Fayçal
37. Dr. YAHIATENE Youcef

Summary

A New Branch and Bound Algorithm for Mining Frequent Conceptual Links in Social Networks

Hadjer Djahnit[1][0000-0002-8071-7057] and Malika Bessedik[1][0000-0002-1007-9096]

[1]Laboratoire des Méthodes de Conception de Systèmes (LMCS),
Ecole nationale Supérieure d'Informatique (ESI),
BP 68M -16 270 Oued Smar, Alger, Algérie.
{h_djahnit,m_bessedik}@esi.dz

Abstract. The frequent conceptual links is a descriptive data mining task which aims at describing a social network in term of the most connected type of nodes. This is done by grouping nodes into clusters or groups according to their attributes and checking the number of links between the nodes of each couple of groups, if this number is greater than a predefined threshold, the set of links is referred to as a frequent conceptual link (FCL). Although relatively recent, this task has received a number of research, chiefly in order to optimize the exploration of the search process. Indeed, the problem is defined as NP-hard, where the search process depends on the size of the network, the number of attributes and the set of their possible values whose combination can explode quickly. In this paper, we propose a new algorithm for mining the frequent conceptual links in a social network based on the technique of the branch and bound. In addition to defining an upper bound for the potential patterns in the search space, the algorithm implements other techniques which improve significantly the performance of the search process and allows to fix the shortcomings constated in the previous implementations.

Keywords: Frequent conceptual links, data mining, Branch and Bound, social network analysis, frequent pattern mining.

1 Introduction

During the last decades, data mining has constituted one of the domains which receives a massive amount of research within various topics and tasks. In fact, tasks like classification [1, 14], clustering [1, 4, 14], link prediction [15] or frequent patterns (5, 20, 22, 23) are all an integral part of the data mining techniques with a common goal of extracting latent and interesting knowledge from data.

Furthermore, regarding the role that these tasks fill in the studied topics, they are characterized as descriptive or predictive techniques [15]; while the first one aim to " summarize the data by identifying some relevant features in order to describe how

things organize and actually work", [13] the goal of the later is to " analyze current and historical facts to make predictive assumptions about future or unknown events" [13].

It is in this context that the problem of mining frequent conceptual links in a network is addressed. Indeed, the frequent conceptual links mining task is a descriptive data mining technique that provides knowledge about the most connected type of nodes in a network [12]. More precisely, in a social network, a conceptual link includes all links connecting two groups of nodes, such that the nodes in each group share common attributes. The frequency of a link is defined with respect to a predefined threshold called the minimum support threshold and when the number of links connecting two groups of nodes is greater than this threshold, they are considered as a frequent conceptual link (FCL).

Obviously this pattern gathers at least two of the most interesting tasks of data mining: clustering and frequent pattern mining and inherits all of their benefits. On the one hand, using both the link and attribute information of the network offers more meaningful patterns by the mining task [16]. On the other hand, considering the patterns above a predefined threshold indicates how often the pattern occurs in the dataset and consequently measures the generality and the significance of the pattern [1]. Finally, the FCL mining approach is distinguished by its ability to synthetize the acquired knowledge in one simple and content rich visualization called the conceptual view, [6] which may be of great benefit for the researchers as they can read directly the most connected features of a network and apply it in many real world applications like designing marketing strategies or the landscape planning [2, 6].

The FCL problem has been proven NP-hard [3], the size of search space depends on the number of attributes and the set of their possible values, which makes the overall process very time consuming or even impossible on a large network. Hence, different implementations have been proposed: (i): FLMin [8], which extracts frequent conceptual links by exploiting the downward closure property[1]; (ii): MFCLMin [7], which extracts the maximum FCLs by exploiting the properties of downward closure and frequency; (iii): H-MFCLMin [10] which adds to the two previous properties, a new threshold called the filtration threshold and eliminates groups with a number of nodes below this threshold (iv): D-MFCLMin [11] which is an improvement of the H-MFCLMin that exploits the dependency property in order to prune the search space; (v): the Bin-MFCLMin [17], which implements a compressed binary structure of the input network and finally (vi): PALM [12] which implements the parallelism applied to the concept lattice formed by potential frequent patterns in order to simultaneously explore several parts of the search space. Varying between sequential or parallel implementations, exhaustive or heuristics solutions; each of these implementations reached to exploit some properties of the problem or to use some technical tools in order to optimize the exploration of the search space and gain consequently in the execution time.

In the same direction, this work proposes a Branch and Bound algorithm for mining frequent conceptual links in a social network. Indeed, we propose an upper bound for the potential frequent patterns support, a depth first traversal of the search space and a

[1] See the next section for details about the properties cited in this paragraph

candidate generation technique by augmentation to improve the search process performances. The rest of this paper is organized as follows: section 2 introduces the main preliminaries for the comprehension of the problem, section 3 exposes the proposed approach, finally, we finish with the experimental results and a conclusion.

2 Problem Statement

In order to model our problem, we define a social network S with two sets, N the set of nodes and E the set of edges linking the nodes. We further define a set of attributes A, and a set of attribute values V where every node in N has a value vij for each attribute ai in A.

The couple (attribute, value) is called an item and a set of items is called an itemset. Hence, every node in the network is described by an itemset. For instance, the couple (gender=M) is an item, while the set (gender=M, age<=30) is an itemset which describes all the men under thirty years old in the network.

We note I the set of all the itemsets that may be formed from the attribute list A and their values V. If I1, I2 are two itemsets included in I, then the set of links between the group of nodes satisfying I1 and the group of nodes satisfying I2, is called a conceptual link:

(I1, I2) a conceptual link = {e=(n1, n2) ; e ∈E ; n1, n2 ∈N ; I1, I2 ∈I ; n1 satisfies I1 and n2 satisfies I2} (1)

In the last formula, I1 is called the left itemset and I2 is called the right itemset. Going back to the last example, the conceptual link defined by ((gender=F), (gender=M, age<=30)) gathers all the links which connect between women and men under thirty years old in the network.

Support of a Conceptual Link. The support of a conceptual link is the ratio between the number of links of the conceptual link and the total number of links in the network.

$$supp(I1, I2) = \frac{|\{e=(n1,n2)\ ;\ e \in E\ ;\ n1,n2 \in N\ ;\ I1,I2 \in I\ ;\ n1\ satisfies\ I1\ and\ n2\ satisfie\ \ I2|}{|E|} \quad (2)$$

Frequent Conceptual Link. For a predefined support threshold β, a conceptual link (I1, I2) is frequent if its support is greater than β .

$$(I1,I2)frequent \equiv Support\ (I1,I2) \geq \beta \quad (3)$$

Sub and Super-Conceptual Links. If I1 is an itemset constituted of k1 attribute-values couples (a1=v11, …, ak1 = v1k1) and I2 is an itemset constituted of k2 attribute-values couples (a1=v21, …, ak2 = v2k2). If k1<=k2 and all the items of I1 are included in I2, we note I1 ⊆ I2 and say that I1 is a sub-itemset of I2 while I2 is a super-itemset of I1.

Let (IL1, IR1) and (IL2, IR2) be two conceptual links where IL1, IR1 are respectively the left and right itemsets of the first conceptual link and IL2, IR2 are respectively

the left and right itemsets of the second conceptual link. If IL1 ⊆ IL2 and IR1 ⊆ IR2, we say that (IL1, IR1) is a sub-conceptual link of (IL2, IR2) while (IL2, IR2) is a super-conceptual link of (IL1, IR1).

Maximum Frequent Conceptual Link (MFCL). A frequent conceptual link is said to be maximum if it is not a sub-conceptual link of any other frequent conceptual link.

Downward Closure Property. According to the downward closure property, all the sub-conceptual links of a frequent conceptual link are frequent. Furthermore, all the super-conceptual links of a non-frequent conceptual link are non-frequents. This property is very useful in pruning the search space particularly when performing an iterative search process which creates candidates conceptual links from frequent sub-conceptual links. A detailed proof is given in [5].

3 The Proposed Approach

Our objective in this work is to improve the performances of the frequent conceptual links extraction process. In fact, although the results obtained in [17] are much better than the previous ones, we can identify two shortcomings:

1. The impact of the input network compression cannot be seen for low support values and the algorithm shows almost the same performances of the original solution.
2. The candidate generation is still a heavy task, as it is based on a join operation and it checks for every new candidate that it is not already generated.

Hence, in order to overcome these two constraints, we propose to tackle the problem of FCL extraction using the Branch & Bound technique, the next section gives the detail of our contribution.

3.1 Branch & Bound for FCL Mining

Branch & Bound is an algorithm design paradigm for discrete and combinatorial optimization problems. It consists on implicitly enumerating all solutions of a solution space S by examining subsets of S while exploiting certain properties of the considered problem. This technique succeeds in eliminating partial solutions that certainly do not lead to the optimal solution, thus in many cases obtaining the solution in a reasonable time. The design of a typical Branch and Bound algorithm requires: A bounding function, a problem-specific branching rule and a technique for exploring the search space.

In order to apply the branch & bound technique to our problem, we have defined two search trees, each associated with a part of the conceptual link; remember, at this level, that a conceptual link is made up of two itemsets (a left itemset and a right itemset). In addition, each tree is browsed separately in order to determine the frequent itemsets before taking pairs of frequent itemsets (left and right) and extracting the frequent conceptual links.

Depth First Search . The first optimization we propose in the new MFCL extraction process consists of parsing the search space in a depth first manner, i.e. when an itemset candidate is known to be frequent, the next candidate to be checked will be its first super-itemset. If this later is also frequent, the process moves to its first super-itemset and so on, until all the super-itemsets are exhausted or a non-frequent itemset is encountered, in which case the process go back and check the next super-itemset. During the scanning of the search tree, the pruning of potential solutions is based on the Apriori principle, according to which when an itemset is frequent, all its sub-itemsets are retained since they are necessarily frequent, and when an itemset is infrequent, all of its super-itemsets are pruned since they are infrequent [9].

Branching by Itemsets Augmentation . From the frequent 1-itemsets, the longer itemsets candidates are generated by augmentation, i.e. every itemset of size k is augmented successively by all the items that are superior to its last item, according to a predefined order. For instance: Assume that we have four items considered in the following order : i1, i2, i3, i4. The list of the itemsets created from these items is depicted in figure 1.

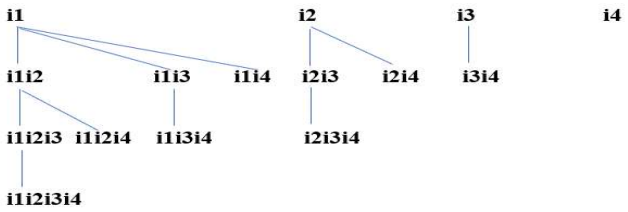


Fig. 1. Itemset augmentation

Property . Every new candidate itemset generated in this way is unique.

Proof. The new candidate itemsets are created either from the same itemset by using different augmentation items or from different itemsets using the same or different augmentation items which prevents the same candidate itemset from being created twice in both cases.

The creation of itemsets by augmentation generates the itemsets only once and avoids the operations of joining the itemsets and duplicate checking which will have an impact on the search process performances.

The Upper Bound. The efficiency of a branch and bound method is based largely on a good function to bound the potential solutions. The more the bound is close to the optimal solution the more is the number of the pruned solutions from the search space.

The upper bound we define for the potential solutions in the search space is applied when a candidate FCL of size k is defined to be non-frequent in which case a backtracking is made in order to check the frequency of the next (k-1)-FCL. The support of this later can be bounded using that of its predecessor and its successor, if it is found that the upper limit is lower than the predefined threshold, the candidate in question can

be pruned and there is no longer any need to check its frequency or that of its super-conceptual links.

For instance, let C, C1, C2 be three conceptual links which share the same left itemset LI and constituted of the following right itemsets RI, RI1, RI2 respectively which are three itemsets of size k, k+1, k+2 (k>=1) respectively, where (k+1)-itemset is obtained by augmentation of the (k)-itemset and the (k+2)-itemset is obtained by augmentation of the (k+1)-itemset. Assume that the C is frequent, the C1 is frequent and the C2 is non frequent, than the process go back to the next (k+1)-FCL candidate to check its frequency which may be bounded using the support of the k-FCL and the previous (k+1)-FCL according to the following formula:

Supp((k+2)-FCL) <= Supp((k+1)-FCL candidate) <= Supp((k+2)-FCL) + Supp(k-FCL) – Supp(Prev(k+1)-FCL) (4)

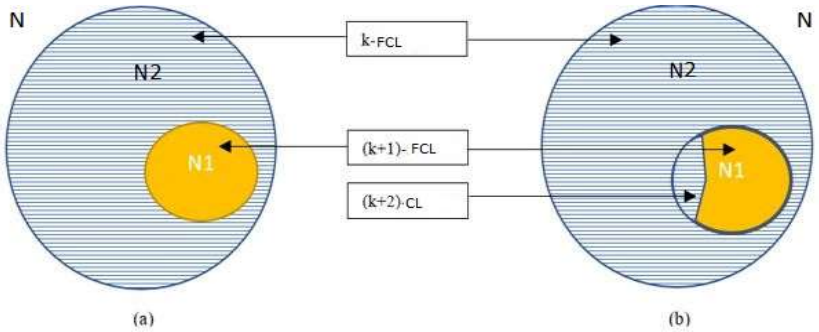


Fig. 2. relation between the k, k+1 and the k+2 FCL. N2 corresponds to the region with blue stripes. a: N1 and N2 are disjoint, b: N1∩N2 ≠Φ

Proof . The arguments can be given by illustrating the relation between the sets of links corresponding to every conceptual link.

Let N, N1, N2, N3 be four conceptual links defined by :

- N = {e = (a, b), e ∈ E, a satisfies LI and b satisfies RI}
- N1 = {e = (a, b), e ∈ E, a satisfies LI and b satisfies RI1}
- N2 = {e = (a, b), e ∈ E, a satisfies LI and b satisfies RI2}
- N3 = {e = (a, b), e ∈ E, a satisfies LI and b satisfies RI3}

Where LI is the left itemset shared by the four conceptual links, RI is the right k-itemset of the conceptual link N, RI1 is the right (k+1)-itemset of the conceptual link N1 and is obtained by augmenting the k-itemset RI with a frequent item a. RI2 is the right (k+1)-itemset obtained by augmenting the k-itemset RI with a frequent item b and finally RI3 is the right (k+2)-itemset obtained by augmenting the (k+1)-itemset RI1 with the frequent item b. hence, N3 corresponds to the links between the set of nodes satisfying the left itemset LI and both the (k+1)-itemsets RI1 and RI2.

If N1 and N2 are disjoint, i.e. N1∩N2 = Φ, then, N2 may cover all the links between the set of nodes satisfying the itemset LI and the set of nodes satisfying RI but not satisfying RI1 which correspond to the blue striped region in figure 2a, hence N2 ⊆ N-N1.

Else if N1∩N2 ≠ Φ, then, N2 may cover all the links between the set of nodes satisfying the itemset LI and the set of nodes satisfying the itemset RI and not satisfying the itemset RI1 and the set of nodes satisfying the itemset RI3, which corresponds to the blue striped region in figure 2b, so N2 ⊆ (N-N1) ∪ N3;

From the last relation, we deduce that the size of N2 is smaller than the size of the set (N-N1) ∪ (N3). As N3 corresponds to the links between the set of nodes satisfying the left itemset LI and the set of nodes satisfying both (k+1)-itemsets (RI1, RI2) obtained by augmenting the k-itemsets RI with the frequent items a and b, this set corresponds to a (k+2)-itemset composed of the k-itemset and the two items a and b. we finally get |N2| <= |N|-|N1| + |N3| which correspond exactly to :

Supp((k+2)-FCL) <= Supp((k+1)-FCL candidate) <= Supp((k+2)-FCL) + Supp(k-FCL) – Supp(Prev(k+1)-FCL)

The last formula may also be applied to the itemsets in the left and right trees as follow:

Supp((k+2)-itemset) <= Supp((k+1)-itemset) <= Supp((k+2)-itemset) + Supp(k-itemset) – Supp(Prev(k+1)-itemset)

3.2 Support Counting of the Last Itemset

When considering the gender attribute in a network, the number of outgoing links from nodes verifying the itemset (gender=female), is equal to the total number of outgoing links minus the number of outgoing links from nodes verifying the itemset (gender=male). Similarly if we consider the marital status attribute which can take the values: single, married, divorced and widowed, the number of outgoing links from nodes verifying the itemset (gender=female, status = widows) is equal to the number of outgoing links from nodes matching the itemset (gender=female) minus the number of outbound links from females in other categories. In general, the number of links outgoing from a group of individuals which satisfies an itemset I1 of size k obtained by augmenting an itemset I2 of size k-1 with an item li, such that li is the last item in the list of items corresponding to an attribute t, is equal to the number of outgoing links from the group of individuals which satisfies the itemset I2 minus the number of outgoing links from all the groups of individuals which satisfies the itemsets constructed from itemset I2 augmented by all possible values of the attribute t except for the last item li. This technique allows us to save a database scan for every attribute used in an augmentation and will have an important impact on the execution time.

Property. Let I be an itemset of size k; and T an attribute with d possible values; domain(T) = {v1, v2,…, Vd}. We have

$$support(I) = \sum_{i=1}^{d} support(I + vi) \quad (5)$$

Proof. If d=2, domain (T) = {v1, v2}

Let N be the set of nodes satisfying the k-itemset I, N1 be the set of nodes from N which satisfy the (k+1)-itemset obtained by augmenting the k-itemset I with v1. Because each node has one and only one value for each attribute, then N2 which corresponds to the set of nodes satisfying v2 is equal to the difference between N and N1 N2 = N-N1 and consequently support (I) = support(I+v1) + support(I+v2)
If d>2, domain (T) = {v1, v2, …, vd}
Assume that N1 corresponds to the set of nodes satisfying (I+v1) and N2 = N-N1 corresponds to the set of nodes satisfying the other values of T. then, support(I) = support(I+v1) + support(I+vj); j∈2, …, d. applying the last decomposition to the set N2 until j∈{d-1, d} we get $support(I) = \sum_{i=1}^{d} support(I + vi)$
Using this formula, we can obtain the support of the (k+1)-itemset obtained by augmenting the k-itemset by the last value vd of an attribute T, from the support of the k-itemset and the support of the (d-1) previous (k+1)-itemsets.
Support(last(k+1)-itemset) = support(I+vd) = support(k-itemset) - $\sum_{i=1}^{d-1} support(I + vi)$
With this formula, we can gain a database pass for every attribute, because the support of the last itemset is always counted directly from the previous ones.

3.3 Pseudocode of BB-MFCLMin Algorithm

The pseudocode of the BB-MFCLMin algorithm is given in the listings 1 through 3. As it is depicted in the main program [Algorithm 1], the process consists of finding the list of one frequent conceptual links, the list of one-left itemsets and that of one-right itemsets at first [lines 9-24], then using them to create the left and right trees of frequent itemsets of size T (T>1) [lines 25, 28]. Finally, the frequency of each candidate conceptual link constituted of a left and right frequent itemset is checked in order to extract the frequent ones [lines 30 to 38]. At this stage, whenever a frequent conceptual link is retained, the list of all its frequent sub-conceptual links are deleted in order to return only the maximum FCLs [lines 37-38].

```
1. Algorithm 1: The BB-MFCLMin algorithm.
2. Input: network data in a binary compressed representa-
tion
4.    Minimum support threshold β
5.    Number of Network links S
6. Output: List of maximum FCLs ListMFCL
Begin
//Generation of the one-frequent conceptual links
9.  LeftFreqItemsetcand ← list of 1-itemsets m where
support (m) >= β * S
11. RightFreqItemsetcand ← list of 1-itemsets m where
support (m) >= β * S
13. For each leftItem in LeftFreqItemsetcand
14.   For each rightItem in RightFreqItemsetcand
```

```
15.     Support = countConceptualLinkSupport(leftItem,
rightItem)
17.       If(Support >= β * S)
18.         Add the conceptual link (leftItem, rightItem)
to listMFCL
20.         Add leftItem to LeftFreqItemset
21.         Add rightItem to RightFreqItemset
22.       End if
23.     End for
24. End for
25. leftTree←CreateFrequentItemsetTree(LeftFreqItemset,
LEFT)
27. rightTree←CreateFrequentItemsetTree(RightFreqItemset,
RIGHT)
29. ReverseLeftTree()
//generation of the T-conceptual links
30. for leftItemset in leftTree
31.   for rightItemSet in rightTree
32.     Supp ← CountConceptualLinkSupport(leftItemset,
rightItemset)
34.     If(supp>= β*S)
35.       listMFCL ← listMFCL.add(leftItemset, rightItem-
set)
37.       Remove all sub-conceptual links of the newly
added frequent conceptual link
39. Return listMFCL
40.END
```

The detail of the CreateFrequentItemsetTree function is given in the listing titled [Algorithm 2]. According to the direction parameter value (LEFT/RIGHT), this function takes the list of left or right frequent one-itemsets and creates a list of all the frequent itemsets ordered as if they are parsed in a depth first manner. Before starting the processing, all the one-itemsets are marked explored frequent (EF) [line 6] which means that their support threshold is calculated and is above the minimum support threshold. Then, the itemsets are taken one by one, if an itemset is marked EF, it is augmented by the AugmentItemset function in order to generate larger candidate itemsets from it [line 9]. The list of the new candidates obtained are marked not explored (N) [line 10] and inserted just after the itemset in question which allows a depth first parsing for the list of the processed itemsets [line 11]. Last, if the itemset in question is marked not explored (N), then we scan the dataset in order to check its frequency [line 14], if it is found greater than the support threshold, then it is marked frequent as well as all its sub-itemsets [lines 20-21], otherwise, it is removed from the processed list as well as all its super-itemsets since they cannot be frequent [lines 16-17].

```
1 ALGORITHM 2: CreateFrequentItemsetTree
2 Input : List of oneFrequentItemsets listFreqItemsets
```

```
          direction : left/right
4 Output: Frequent Itemset Tree
5 Begin
6 Mark all the itemsets in listFreqItemsets with [EF]
7 For itemset in listFreqItemsets
    If(itemset is marked EF)
      listNewCandidateItemset←AugmentItemset(itemset)
10    Mark all the itemsets in newCandidateItemset with [N]
11     Insert the listNewCandidateItemset in listFreqItem-
sets just after itemset
13   Else If(itemset is marked N)
14     Supp = CountItemSetSupport(itemset)
15    If(supp < β * S)
16        Remove itemset from listFreqItemsets
17        Remove all the super-itemsets of itemset from
listFreqItemsets
19    Else
20        Mark itemset with [EF]
21        Mark  all the sub-itemsets of itemset with [F]
         listNewCandidateItemset←AugmentItemset(itemset)
23       Mark all the itemsets in newCandidateItemset with
[N]
25       Insert  the  listNewCandidateItemset  in  list-
FreqItemsets just after itemset
27 Return listFreqItemsets
End
```

The CountConceptualLinkSupport function calculates the support of an FCL candidate with the same method implemented in the Bin-MFCLMin algorithm [17], i.e. using the binary compressed input data structure and the bitwise operators. When a non-frequent super-conceptual link is founded, the support of the FCL candidate is upper bounded according to the formula 4 [lines 8-14]. If this bound is under the minimum support threshold, the support is returned in order to remove the FCL candidate as well as all its super-conceptual links. Otherwise, the database is parsed and the exact value of the support is returned.

```
1. Algorithm 3: CountConceptualLinkSupport()
2. Input: candidate conceptual link c=( leftItemset,
rightItemset) of size k+1
4.         Minimum support threshold β
5.         Number of Network links S
6. Output: support or support bound
Begin
8. If(c is sub-conceptual link of non-FCL c1):
9.  Supp-k-FCL = support of the k-FCL used to generate c
```

```
10. Support-(k+1)-FCL = support of a (k+1)-FCL generated
from c
12. Support-(k+2)-FCL = support(c1)
13. Support-bound = Support-(k+2)-FCL + Supp-k-FCL - Sup-
port-(k+1)-FCL
15. If(Support-bound < β*S)
16.   Return Support-bound
17. Else
18.   Support = parse the compressed input structure and
get the exact value of support
20.   Return support
21. End
```

4 Experimental Results

In this section, we present the results of experiments conducted to test the designed algorithm performance in terms of run time, since the algorithm returns optimal solution. We evaluated the performances of our algorithm on the Pockec social network [18] which is also the dataset used to validate the Bin-MFCLMin [17] and the D-MFCLMin [11] algorithms. Table 1 gives the characteristics of the assessed network.

Table 1. Characteristics of the Pockec social network

Network size (number of links and nodes)	961 431 links 223 291 nodes
Number of attributes	5
The domain value of the attributes	13
Type of the network	Uni-partite, directed

The algorithm will be compared to the last contribution, namely the Bin-MFCLMin algorithm. For the reasons cited in the "proposed approach" section which motivates this contribution, the new algorithm will also be compared to the MFCLMin algorithm, the focus will be on the low support values, as the number of accepted patterns will be greater. Thus, we will check that the new improvements are able to the fix the lack observed for the low support values. The comparison is done according to the execution time and the number of candidate pruned.

Figures 3 plots the amount of time spent by BB-MFCLMin algorithm against different values of support threshold. While the new algorithm performs nearly as well as the Bin-MFCLMin for high support values, the gain is more substantial for lower values. for instance, while the Bin-MFCLMin achieved a gain of 2% for the 0.2% support value comparing to the MFCLMin algorithm, the BB-MFCLMin reaches up to 62% for the same value (figure 3b). Furthermore, comparing to the MFCLMin algorithm , we can clearly see that the BB-MFCLMin keeps the improvements achieved by the input data compression already implemented in the Bin-MFCLMin algorithm as the gain for high support values reaches up to 94% of the execution time (figure 3a).

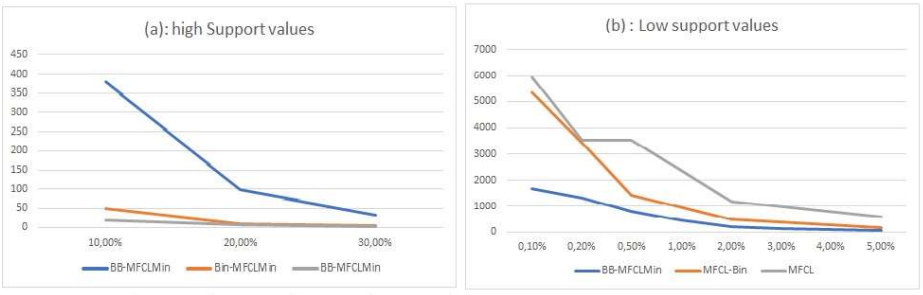


Fig. 3. Execution time of the MFCLMin, Bin-MFCLMin and the BB-MFCLMin algorithms on high and low support values

On the other hand, the percentage of itemsets candidates pruned with the upper bound varies between 0 and 14% according to the total number of itemsets(figure 4), the effect of the upper bound starts from a support threshold of 2% and the number of candidate pruned without any need to a database scan, increases with the support values until it petered out another time when the number of candidates and patterns decreases for high support values. On the other hand, the same figure shows the number of FCL candidate pruned with the upper bound, the pruning starts in this case from the 0.1% support threshold and increases with the support values where it reaches up to 3% for 15% of the support threshold.

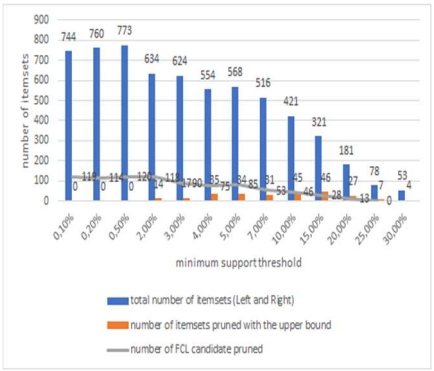


Fig. 4. Number of candidate pruned with the upper bound

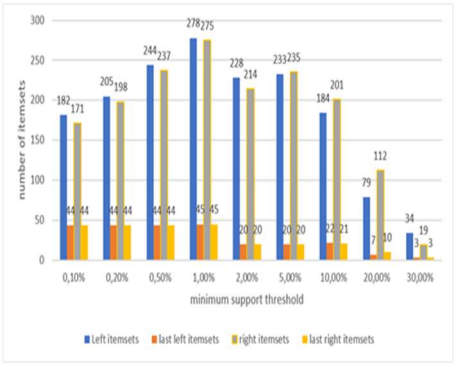


Fig. 5. Support counting of the last itemset

The last optimization technique which concerns the support counting of the last itemset, performs a database scan gain for every attribute, as the support of the candidate itemset obtained with an augmentation using its last item is always counted directly from its origin itemset and the other items of the same attribute. Figure 5 plots the number of database scan saved with the support counting of the last itemset for both left and right trees. The gain varies between 8% and 25% of the total number itemsets which impacts considerably the computation time.

5 Conclusion

We examined the optimization of the process of extracting the frequent conceptual link patterns from large networks. This problem, even though, is proven NP-hard, may be resolved exhaustively for large networks and low support values. Like earlier implemented techniques such as parallelism or input network compression, the use of the Branch and Bound technique allows us to significantly optimize the search process and tackle larger networks. Indeed, we have proposed an upper bound for the support of potential patterns and an augmentation-based candidate generation technique to significantly improve the performance of the search process and save up to 90% of the execution time.

In the future, the implementation of a parallel version to this solution will make the whole process faster. Furthermore, the use of approximate approaches such as metaheuristics could also be an interesting alternative to tackle larger networks.

References

1. Vazirgiannis, M., Halkidi, M., Gunopulos, D.: Uncertainty Handling and Quality Assessment in Data Mining. Springer-Verlag London (2003)
2. Stattner, E., Collard, M.: Frequent links: An approach that combines attributes and structure for extracting frequent patterns in social networks. 16th East-European Conference on Advances in Databases and Information Systems (2012).
3. Stattner, E., Collard, M.: Social-based conceptual links: Conceptual analysis applied to social networks. International Conference on Advances in Social Networks Analysis and Mining (2012a).
4. Fortunato, S.: Community detection in graphs. ArXiv (2009).
5. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the 20th International Conference on Very Large Data Bases (1994).
6. E. Stattner. : Contributions à l'étude des réseaux sociaux : propagation, fouille, collecte de données. Thèse pour obtenir le titre de Docteur en Sciences de l'Université des Antilles et de la Guyane (2012b).
7. Stattner, E., Collard, M.: Social-based conceptual links: Conceptual analysis applied to social networks ». International Conference on Advances in Social Networks Analysis and Mining (2012c).
8. Stattner, E., Collard, M.: FLMin: An Approach for Mining Frequent Links in Social Networks. International Conference on Networked Digital Technologies (2012d).
9. Nagiza, F.S., Hendrix, W., Jenkins, J., Padmanabhan, K, Chakraborty, A.: Practical graph mining with R. CRC Press (2014).
10. Stattner, E., Collard, M.: Towards a hybrid algorithm for extracting maximal frequent conceptual links in social networks. IEEE International Conference on Research Challenges in Information Science (2013).
11. Tabatabaee, H.: DMFCLMin: A New Algorithm for Extracting Frequent Conceptual Links from Social Networks. International Journal of Advanced Computer Science and Applications (2017).
12. Stattner, E., Eugenie, R., Collard, M.: PALM: A Parallel Mining Algorithm for Extracting Maximal Frequent Conceptual Links from Social Networks. International Conference on Database and Expert Systems Applications (2017).

13. Stattner, E., Collard, M.: Descriptive Modeling of Social Networks, Procedia Computer Science, Volume 52, 226-233 (2015).
14. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson (2005)
15. Getoor, L., Diehl, C.: Link Mining : a survey. SIGKDD Explor (2005).
16. Fani, H., Bagheri, E.: Community Detection in Social Networks. World Scientific Publishing Company (2017).
17. Djahnit, H., Bessedik, M.: Exhaustive Solution for Mining Frequent Conceptual Links in Large Networks using a binary compressed representation. 13th international conference on knowledge discovery and information retrieval. http://www.kadir.ic3k.org (2021).
18. Takac, L., Zabovsky, M.: Data Analysis in Public Social Networks, International Scientific Conference & International Workshop Present Day Trends of Innovations, Lomza, Poland (2012).
19. Leskovec, J., Adamic, L., Adamic, B.: The Dynamics of Viral Marketing. ACM Transactions on the Web (ACM TWEB) (2007).
20. Luna, JM., Fournier-Viger, P., Ventura, S. : Frequent itemset mining: A 25 years review. WIREs Data Mining Knowl Discov (2019).
21. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the 20th International Conference on Very Large Data Bases (1994).
22. Aggarwal, C., Han, J. (eds.) : Frequent Pattern Mining, Springer International Publishing Switzerland (2014).
23. Cafaro, M., Pulimeno, M.: Frequent Itemset Mining. Springer Nature Switzerland (2019).

bAHA: binary Artificial Hummingbird Algorithm for feature selection: Covid-19 as a case study

Adel Got[1], Djaafar Zouache[2], and Abdelouahab Moussaoui[3]

[1] Computer Science Departement, University of Science and Technology Houari Boumedienne, Algiers, Algeria
[2] Computer Science Departement, University of Mohamed El bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
{adeladl34,djaafarzouache}@yahoo.fr
[3] Computer Science Departement, University of Ferhat Abbas, Setif, Algeria
moussaoui.abdel@gmail.com

Abstract. In machine learning, the quality of the managed data plays an important to provide a good classification accuracy. However, real-world datasets as they include relevant features, they include also irrelevant and redundant features, which is undesirable in classification problems. Feature selection is one of the most powerful techniques to identify the most informative and relevant features. In this paper, and by exploiting the advantages offered by swarm intelligence algorithms in exploring the feature space, a novel binary Artificial Hummingbird Algorithm (bAHA) is proposed to perfom an effective feature selection task. The efficiency of the method is compared against four well-known feature selection algorithms on eight benchmarking datasets. Moreover, it is applied to a Covid-19 dataset for patient health prediction to verify its applicability in more recent real-world dataset. The experimental results show the capability of the proposed algorithm to select few number of features with high classification accuracy compared to its competitors, whether in the used benchmarking datasets, or in the Covid-19 dataset.

Keywords: Feature selection · Optimal feature subset · Artificial Hummingbird Algorithm · Classification accuracy.

1 Introduction

With the huge and the fast increase in the data amount related to the most real-world applications, Machine Learning (ML) as a traditional subfield of Artificial Intelligence (AI), is becomes today an important mean to deal with this growing in data. It helps the decision maker to extract valuable knowledge from huge information amount by using different techniques. Under this context, the classification is one of these techniques which aims to classify unknown data to its appropriate target class. The accuracy of this task is related to the learning model trained on a set of data with known classes [4]. Furthermore, the quality of the learning model itself depends on the relevance of certain information usually called features. However, in the most cases, the datasets of real-world problems

include not only relevant features, but they include also non-informative, irrelevant and redundant features which can degrade the classification performance [3]. In such situations, identifying the most useful features is an intuitive and effective manner to improve the classification accuracy of learning algorithms. For this reason, the so-called Feature Selection (FS) is usually invoked as a preprocessing step in ML to find the optimal subset of features. Logically speaking, the optimal subset is the one that includes a few number of features with high classification accuracy (or minimum error rate) [6]. Furthermore, Feature selection can be viewed as an optimization problem in which the objective is to minimize both number of features and the error rate (or maximize the classification accuracy). Therefore, and as the most optimization problems, FS is considered as an NP-complete problem in which it is too hard and even impossible to perform an exhaustive search in reasonable time. Indeed, for N features, $(N^2 - 1)$ possible feature subsets should be evaluated and compared, and this is too computationally expensive especially in huge datasets. In this case, performing a global search seems to be an effective choice to solve FS problem.

Population-based metaheuristics such as Swarm Intelligence (SI) are considered, without a doubt, the most popular global search techniques to solve optimization problems [13, 8]. As their name indicates, they use a population of search agents during the optimization process, which make them able to explore effectively the search space in acceptable computational cost. Particle Swarm Optimization (PSO) [10], Grey Wolf Optimizer (GWO) [12], Whale Optimization Algorithm (WOA) [11], Manta Ray Foraging Optimizer (MRFO) [17], etc. are some examples of Swarm Intelligence algorithms. Further, the remarkable success of these metaheuristics in solving different kinds of optimization problems has motivated many researchers to hire them in FS problem. For instance, we can quote the Binary Particle Swarm Optimization (BPSO) [1], feature selection using GWO algorithm for Arabic text classification [2], Wrapper-based feature selection using binary Whale Optimization Algorithm (bWOA) [9], Boosted Harris hawks optimizer for wrapper-based feature selection (IHHO) [15], Manta Ray Foraging Optimizer [5, 7], etc.

Recently, a new Swarm Intelligence metaheuristic namely Artificial Hummingbird Algorithm (AHA) has been developed [16] for solving continuous optimization problems. In this study, a new binary Artificial Hummingbird Algorithm is introduced for solving FS problem. bAHA method is compared with four well-regarded FS-techniques on eight datasets. Then, it is applied to a real Covid-19 dataset. Experimental results show the efficiency of the proposed method in acquiring few features with high accuracy.

The rest of paper is organized as follows. Section 2 describes the Artificial Hummingbird Algorithm. Section 3 presents the proposed method. Experimental results are outlined and analysed in Section 4. Finally, Section 5 concludes the paper.

2 Artificial Hummingbird Algorithm

AHA algorithm is a Swarm Intelligence technique designed for solving global optimization problems. It is inspired from the social behavior of Hummingbirds in nature [16], and it is based on three sophisticated foraging strategies and three flight techniques. Hence, its mathematical model is desribed in the following subsections:

2.1 Initialization

Like all existing population-based metaheuristics, the initial population is set as follows:

$$x_i = Lb + rand(0,1)(Ub - Lb) \quad i = 1, 2, ..., N \tag{1}$$

Where, Lb and Ub are the lower and upper boundaries, respectively. x_i^t is the current position of the i^{th} hummingbird. In the same time, the initial visit table is set as follows:

$$VT_{i,j} = \{\, 0 \quad if \quad i \neq jNull \quad i = ji = 1, ..., N \quad and \quad j = 1, ..., N \tag{2}$$

Where, $VT_{i,j} = Null$ means that a hummingbird has consumed its food from its position, whereas $VT_{i,j} = 0$ means that the i^{th} hummingbird has just visited the j^{th} food position.

2.2 Guided foraging

During this exploration procedure, a hummingbird performs three flights mathematically defined as below:

- **Axial flight**

$$D^i = \{\, 1 \quad if \quad i = randi([1, d])0 \quad elsei = 1, ..., d \tag{3}$$

- **Diagonal flight**

$$D^i = \{\, 1 \quad if \quad i = P(j),\ j \in [1, k],\ k \in [2, \lceil r_1 * (d-2) \rceil + 1]0 \quad else \tag{4}$$

  Where, $P = randperm(k)$ generates a random permutation from 1 to k, and r_1 is a random number between 0 and 1.
- **Omnidirectional flight**

$$D^i = 1, ..., d \tag{5}$$

Therefore, the guided foraging behavior can be modeled by:

$$x_i(t+1) = x_{i,tar}(t) + \alpha.D.[x_i(t) - x_{i,tar}(t)]; \quad \alpha \sim N(0,1) \tag{6}$$

Where, $x_{i,tar}(t)$ is the position of the the optimal solution obtained so far.

2.3 Territorial foraging

This strategy is considered as a local search, and it is formulated as follows:

$$x_i(t+1) = x_i(t) + \beta.D.x_i(t); \quad \beta \sim N(0,1) \tag{7}$$

2.4 Migration foraging

A hummingbird move, randomly, from the worst position to another position:

$$x_{wor}(t+1) = Lb + rand(0,1)(Ub - Lb) \tag{8}$$

Where, x_{wor} is the worst position in the population.

3 The proposed bAHA algorithm

AHA algorithm has been initially designed to solve continuous optimization problems. That is to say, the hummingbirds can update their positions in any coordinate in the search space. More precisely, the variables of a given individual can take real values. However, feature selection is considered as a discrete or combinatorial optimization problem. Indeed, in the feature space, the solutions are restricted to a binary representation, 0 or 1 [6]. Therefore, converting the continuous representation of hummingbird to its corresponding binary representation is a necessary act to make AHA algorithm able to deal with the binary nature of feature space. The intuitive way to ensure such action is the use of a transfer function. In the proposed bAHA algorithm, the hyperbolic tangent function is adopted as below:

$$T(x_i^d) = \frac{e^{-x_i^d} - 1}{e^{-x_i^d} + 1} \tag{9}$$

Where, x_i^d is the real-value of the i^{th} solution in the d^{th} dimension in the search space. Hence, its binary value is given by:

$$x_i^d = \{\, 1 \quad if \quad \mu < T(x_i^d) 0 \quad Otherwise \tag{10}$$

Where, $\mu \in [0,1]$ is a random number. $x_i^d = 1$ indicates that the d^{th} feature is selected, and $x_i^d = 0$ indicates that the corresponding feature is not selected.

On the other hand, define properly the objective function is an important aspect to get good results. In our case, the classification error rate and the number of features are combined together to construct the objective function. Such combination is realized by using a weighted sum function as follows:

$$Obj = w.Err + (1-w).\frac{S_f}{D} \tag{11}$$

Where, S_f and D are the number of selected features and the total number of features, respectively. The weight w is a number superior to 0.5. Finally, the error rate Err is evaluated using the KNN-classifier. Algorithm 1 outlines the pseudocode of bAHA algorithm.

Algorithm 1: Pseudocode of bAHA algorithm

```
Initialize the population POP with N hummingbirds
Evaluate the initial POP on the train set
while t < iter_max do
    for each individual do
        if rand() < 1/3 then
            Update D using Eq. 4 (Diagonal flight)
        else
            if rand() > 2/3 then
                Update D using Eq. 5 (Omnidirectional flight)
            else
                Update D using Eq. 3 (Axial flight)
            end
        end
        if rand() < 0.5 then
            Update the current position x_i using Eq. 6 (Guided
             foraging)
        else
            Update x_i using Eq. 7 (Territorial foraging)
        end
    end
    if mod(t, 2N) = 0 then
        Update the worst x_i using Eq. 8 (Migration foraging)
    end
    Convert POP to its binary representation using Eqs. 8 and 10
    Evaluate POP via the Eq. 11
    t = t+1.
end
Return the selected feature subset and its error rate
```

4 Experimental results and discussion

The performance of the proposed approach is validated on eight datasets (listed with their characteristics in Table 1). Each dataset is randomly divided as: 70% for training set, and 30% for testing set. The train set is used to evaluate the candidate features during the optimization procedure, while the test set is used to evaluate the set of features provided by the algorithm. Four well-regarded FS-algorithms namely BDE [14], BGWO [2], BPSO [1], and bWOA [9], are selected in the comparative study. For fair competition, the population size and the

number of iterations are set to 30 and 50, respectively. The comparison between algorithms is done based on the averaged results obtained over 15 independent runs.

Table 1. Characteristics of benchmarking datasets

Datasets	No. of samples	No. of features	No. of classes
Breast cancer	699	9	2
Musk	476	166	2
Segment	2310	19	7
Ionosphere	352	34	2
Semeion	1593	265	2
Spect	267	22	2
SpectF	267	44	2
Sports	1000	59	2

4.1 Results of bAHA on the selected datasets

Figure 1 shows the average of the best accuracies obtained by bAHA algorithm over the 15 runs compared to that obtained when using "ALL" available features in each dataset. We can remark from the figure that the proposed algorithm provides higher prediction accuracy than using all available features in almost all benchmarking datasets. Indeed, except in Breast cancer, Musk, and Semeion datasets, the proposed algorithm increased the classification accuracy of features in six datasets. These findings show the ability of the proposed algorithm to improve the performance of the learning model without the need of all features.

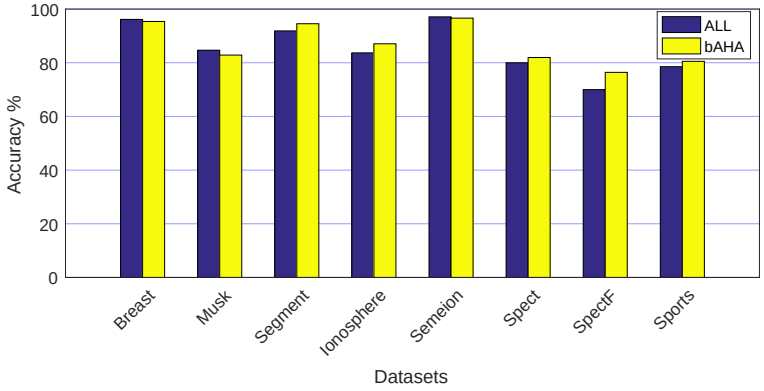


Fig. 1. Average accuracy of bAHA compared to that using "ALL" features

4.2 bAHA vs selected algorithms

In this section, the proposed algorithm is compared with the selected FS-techniques. The results are reported in Table 2, where the values in brackets presents the total number of features and the accuracy of KNN-classifier when using all features. The best results are marked in **boldface**.

Table 2. The results obtained by the compared algorithms

	Breast cancer (9, 96.15%)					Musk (166, 84.67%)				
	bAHA	BDE	BPSO	BGWO	bWOA	bAHA	BDE	BPSO	BGWO	bWOA
#feat	5.00	5.60	4.60	5.60	**4.90**	24.25	133.86	80.90	116.70	**22.90**
Best. Acc	96.15	**97.25**	96.21	**97.25**	**97.25**	**87.09**	84.67	86.29	83.67	82.25
Avg. Acc	95.38	**95.86**	95.15	95.65	95.21	**82.86**	80.37	82.01	81.04	79.35
	Segment (19, 91.87%)					Ionosphere (34, 83.69%)				
	bAHA	BDE	BPSO	BGWO	bWOA	bAHA	BDE	BPSO	BGWO	bWOA
#feat	**8.20**	11.20	7.70	11.80	9.40	3.30	19.66	12.10	19.00	**3.10**
Best. Acc	**96.18**	96.01	**96.18**	95.52	95.85	90.21	88.04	86.95	89.13	**91.30**
Avg. Acc	94.54	94.09	94.92	94.06	**95.42**	87.06	82.24	84.13	82.82	**89.02**
	Semeion (265, 97.11%)					Spect (22, 80%)				
	bAHA	BDE	BPSO	BGWO	bWOA	bAHA	BDE	BPSO	BGWO	bWOA
#feat	40.87	197.30	129.50	184.00	**40.40**	10.37	14.50	9.00	13.12	**8.70**
Best. Acc	97.11	96.63	97.11	**97.35**	96.39	84.28	**85.71**	84.28	84.28	81.42
Avg. Acc	**96.63**	96.25	96.44	96.46	95.33	**81.96**	80.71	**81.96**	80.00	77.28
	SpectF (44, 70%)					Sports (59, 78.54%)				
	bAHA	BDE	BPSO	BGWO	bWOA	bAHA	BDE	BPSO	BGWO	bWOA
#feat	5.70	29.00	18.75	26.60	**4.90**	8.50	43.10	28.70	42.20	**8.20**
Best. Acc	**80.00**	75.71	78.57	74.28	**80.00**	**83.14**	82.37	82.37	81.60	82.37
Avg. Acc	**76.42**	71.85	72.85	70.71	75.00	**80.53**	79.54	80.22	79.50	80.42

From Table 2, it is clear that the bAHA approach can effectively selects few number of features with remarkable improvement in the classiffication accuracy. For instance, in Segment dataset, bAHA selected on average of 8.20 from 19 features, which means around 43% of the initial feature set. In Semeion datasets, it selected an average of 40.87 from 265 features, which means around 15% of the original feature set, and it was able to select approximately 50% of the available features in Spect dataset (on average of 10.37 from 22 features). Regarding the prediction, it can be seen that bAHA algorithm can achieve higher (or equal in the worst case) accuracy than using all features in all datasets. For example, it was able to increase the accuracy in Musk dataset from 84.67% to 87.09%, in Ionosphere from 83.69% until 90.21%. Generally speaking, the results of Table 2 suggest that the proposed bAHA algorithm can provide the optimal feature subset which includes few features with better classification accuracy than using

all features. Therefore, bAHA algorithm can be successfully used in FS problem to reduce the dimensionality of dataset while increasing the prediction quality of the learning model. Compared to the selected algorithms, the results show that bAHA performs better in terms of both accuracy and number of features, especially against BDE, BPSO, and BGWO algorithms. Indeed, compared with BDE and BGWO, and except in Breast cancer dataset, bAHA generated superior accuracy and selected fewer features. Compared to BPSO, the proposed algorithm was able to provide better results in terms of both number of features and accuracy in five datasets (Musk, Ionosphere, Semeion, SpectF, and Sports). However, bWOA has the tendency to selects few features, but with a lower classification performance compared to bAHA algorithm. From these results, we can say that bAHA can effectively explore the feature space, and outperformed it competitors in terms of reduced dimensionality and classification performance.

Figure 2 illustrates the convergence speed of each algorithm. As it can be seen, bAHA algorithm has provided an acceptable convergence speed, but it suffer slightly from low convergence speed compared to the other algorithms. This can be explained by the used fitness function wich takes into account the number of features (described in Equation 11), which is not the case for the other algorithms because they consider only the error rate as a fitness function. Moreover, Table 3 presents the average execution time of algorithms over the 15 independent runs. From this table, it can be observed that the proposed algorithm run faster in three datasets, with very competitive results in the other five datasets.

Table 3. The average run time consumed by algorithms (unit: seconds).

	bAHA	BDE	BPSO	BGWO	bWOA
Breast	159.74	158.65	**152.30**	168.68	**152.30**
Musk	169.36	155.06	**151.85**	160.19	176.24
Segment	191.86	184.67	221.38	**183.71**	221.06
Ionosphere	**152.00**	162.30	158.20	156.94	193.18
Semeion	**185.35**	340.89	324.18	366.17	207.95
Spect	142.87	**135.69**	140.69	136.27	201.06
SpectF	**156.36**	165.98	160.49	168.16	158.91
Sports	173.65	175.67	176.20	**158.23**	168.53

4.3 Results on Covid-19 dataset

In this section, bAHA algorithm is applied to a real Covid-19 dataset for patient health prediction[4]. It includes 864 cases, 14 features, and 2 classes related to "death" and "recovery".

[4] https://github.com/Atharva-Peshkar/Covid-19-Patient-Health-Analytics

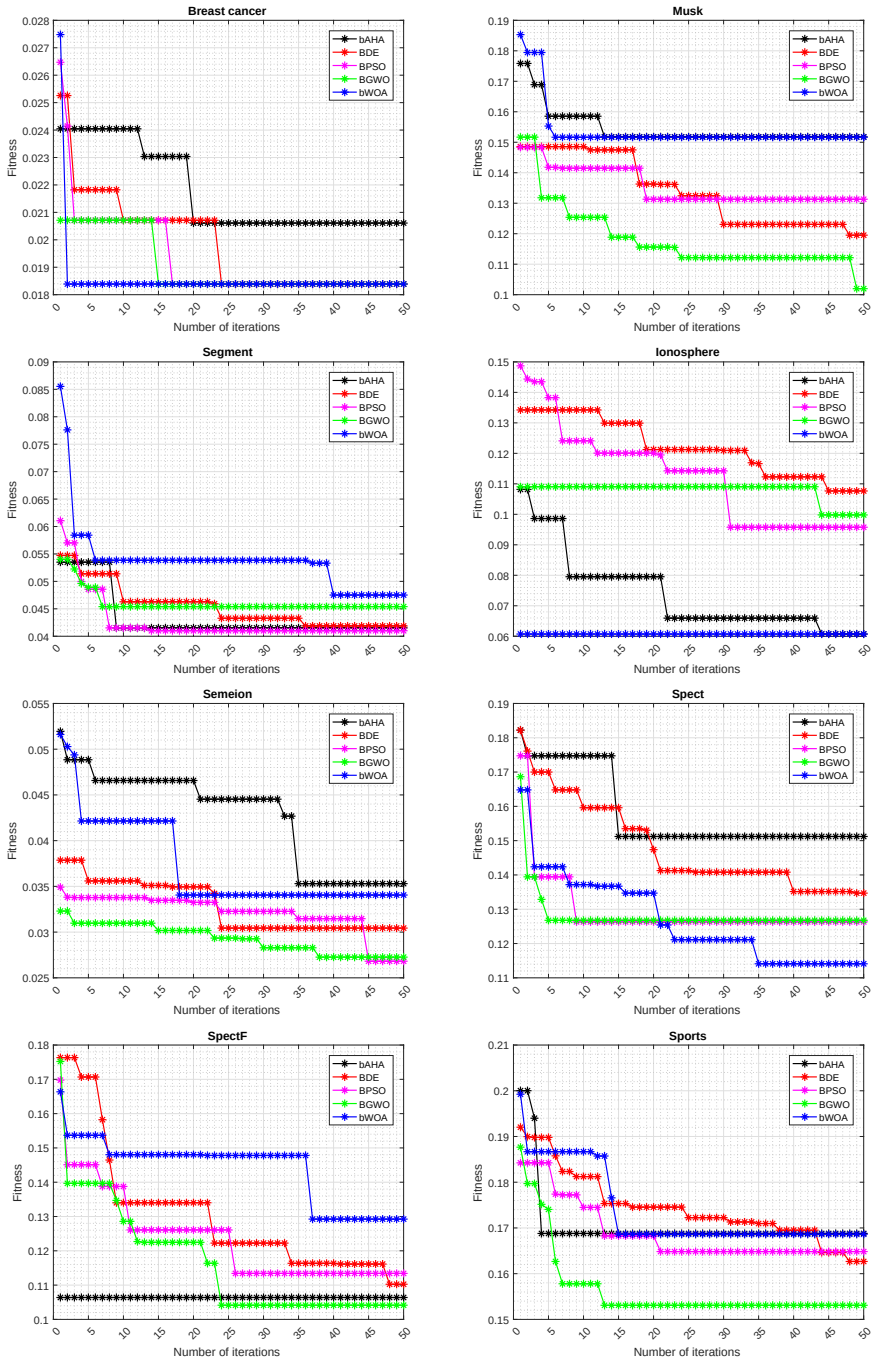


Fig. 2. The convergence speed over iterations

Table 4. Covid-19 dataset

No	Features	Description
1	Location	The location where patients belong to
2	Country	The country where patients belong to
3	Gender	The gender of patients
4	Age	The ages of patients
5	Sym_on	The date patients have the symptoms
6	Hosp_vis	The date patients visited the hospital
7	Vis_Wuhan	Whether the patients visited Wuhan, China
8	From_Wuhan	Whether the patients from Wuhan, China
9	Symptom1	Fever
10	Symptom2	Cough
11	Symptom3	Cold
12	Symptom4	Fatigue
13	Symptom5	Body pain
14	Symptom6	Malaise

Table 5 depicts the results of algorithms on Covid-19 datasets. From this table, it is can be releaved that bAHA algorithm has decreased the number of features from 14 to on average of 3 features. That is to say, a reduction rate of around 79% compared to the original dataset. In the same, bAHA algorithm was also able to increase the accuracy of patient health prediction from 91.11% to 92.44%. In comparison with the selected state-of-the-art algorithms, the significant superiority of the proposed approach is more evident.

Table 5. Comparison between algorithms on Covid-19 dataset

	Covid-19 (14, 91.11%)				
	bAHA	BDE	BPSO	BGWO	bWOA
AvgNB. selected features	**3.00**	7.60	6.75	7.50	3.75
Avg. Acc	**92.44**	90.75	91.44	91.55	92.11
Best. Acc	**92.88**	91.11	92.44	92.00	**92.88**

Figure 3 presents how many times a given feature is chosen during the 15 runs. The most selected features are F2 (Country), F3 (Gender), F4 (Age), and F5 (Sym_on). The results suggest that these features are the most informative features and they can contribute to enhance the performance of the learning model. Hence, they have a high impact in performing a good patient health prediction. In contrast, there are certains features, for example F1 (Location) and F9 (Symptom1), which have never been selected (or they have been selected

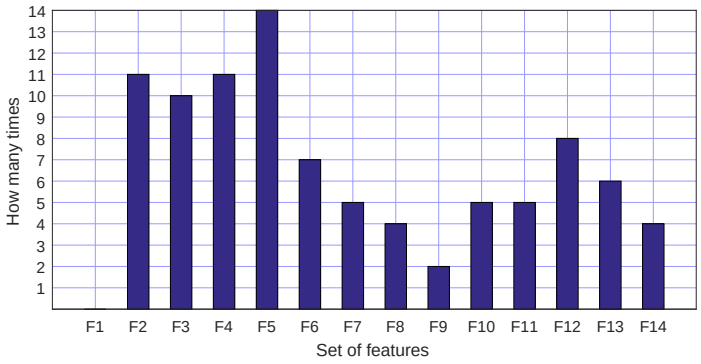


Fig. 3. Number of selection of each feature

in few times), which means that they haven't any impact in predicting the death or the recovered of the patient.

5 Conclusion

This paper presents a novel binary version of Artificial Hummingbird Algorithm for solving feature selection problem. For this reason, the hyperbolic tangent function is used to convert the continuous representation of hummingbirds towards their corresponding binary representation. The proposed bAHA algorithm benefits from a good exploration and exploitation of the feature space via three different foraging strategies. Furthermore, it takes into account both the classification accuracy and number of features during the optimization problem. The proposed bAHA algorithm was compared to four well-known algorithms on eight datasets. Additionally, it has been applied to a real Covid-19 dataset. The results have demonstrated the ability of the proposed algorithm to perform a good feature selection task. In fact, it was able to reduce the size of features with significant improvement in the classification accuracy of the final learning model. Furthermore, the proposed algorithm has shown very competitive behavior compared to the selected methods, and even outperformed them in the most cases.

However, despite these encouraging findings, the proposed algorithm has provided lower convergence speed, and this is due to the adopted objective function. Hence, the use of another objective function may be a part of future work. Also, we plan to improve the present algorithm to perform a feature selection and, simultaneously, tune the perameters of SVM-classifier.

References

1. Cervante, L., Xue, B., Zhang, M., Shang, L. (2012, June). Binary particle swarm optimisation for feature selection: A filter based approach. In 2012 IEEE Congress

on Evolutionary Computation (pp. 1-8). IEEE.
2. Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A. A., Aljarah, I., Faris, H. (2020). Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. Neural Computing and Applications, 32(16), 12201-12220.
3. Emary, E., Zawbaa, H. M., Hassanien, A. E. (2016). Binary ant lion approaches for feature selection. Neurocomputing, 213, 54-65.
4. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine, 17(3), 37-37.
5. Ghosh, K. K., Guha, R., Bera, S. K., Kumar, N., Sarkar, R. (2021). S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem. Neural Computing and Applications, 33(17), 11027-11041.
6. Got, A., Moussaoui, A., Zouache, D. (2021). Hybrid filter-wrapper feature selection using whale optimization algorithm: A multi-objective approach. Expert Systems with Applications, 183, 115312.
7. Hassan, I. H., Abdullahi, M., Aliyu, M. M., Yusuf, S. A., Abdulrahim, A. (2022). An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection. Intelligent Systems with Applications, 16, 200114.
8. Hussien, A. G., Hassanien, A. E., Houssein, E. H., Bhattacharyya, S., Amin, M. (2019). S-shaped binary whale optimization algorithm for feature selection. In Recent trends in signal and image processing (pp. 79-87). Springer, Singapore.
9. Hussien, A. G., Oliva, D., Houssein, E. H., Juan, A. A., Yu, X. (2020). Binary whale optimization algorithm for dimensionality reduction. Mathematics, 8(10), 1821.
10. Kennedy, J., Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.
11. Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.
12. Mirjalili, S., Mirjalili, S. M., Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.
13. Tawhid, M. A., Ibrahim, A. M. (2020). Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. International journal of machine learning and cybernetics, 11(3), 573-602.
14. Too, J., Abdullah, A. R., Mohd Saad, N. (2019). Hybrid binary particle swarm optimization differential evolution-based feature selection for EMG signals classification. Axioms, 8(3), 79.
15. Zhang, Y., Liu, R., Wang, X., Chen, H., Li, C. (2021). Boosted binary Harris hawks optimizer and feature selection. Engineering with Computers, 37(4), 3741-3770.
16. Zhao, W., Wang, L., Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. Computer Methods in Applied Mechanics and Engineering, 388, 114194.
17. Zhao, W., Zhang, Z., Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. Engineering Applications of Artificial Intelligence, 87, 103300.

Securing data warehouses against inferences using KNN

Fatima Zohra Benazza[1], Djamila Hamdadou[1], and Ilyes Khennak[2]

[1] Oran 1 Ahmed Ben Bella University, Oran, Algeria,
{benazza.fatima, hamdadou.djamila}@edu.univ-oran1.dz
[2] Laboratory for Research in Artificial Intelligence, USTHB, Algiers, Algeria,
ikhennak@usthb.dz

Abstract. Data Warehouse (DW) is currently one of the most powerful technologies to collect secret and sensitive corporate data of private lives of individuals. Nevertheless, it suffers from security shortcomings related to the management of access and the detection of possible inferences. On the other hand, Artificial Intelligence (AI) methods are nowadays one of the most used techniques to solve data warehouse security issues due to their ability to predict inferences. Therefore, in this work, we propose to use K-Nearest Neighbors (KNN), a well-known AI technique, to solve the problem of inferring hidden data from accessible data while considering security of operations. We extensively evaluate the proposed approach for data warehouse security using a query dataset. The results show that the proposed algorithm improves the performance of data warehouses security and demonstrates a substantial enhancement over the state-of-the-art.

Keywords: Data warehouse, Data security, Data inference, Machine learning, K-Nearest Neighbors.

1 Introduction

Business Intelligence (BI) is defined as the set of technologies that enable the processing, enhancement and presentation of data for the purposes of understanding and decision-making. It gives managers visibility into their company's performance in order to improve its ability to react more quickly than its competitors to new opportunities or market risks. BI relies on a specific information system called a Business Intelligence System (BIS), as opposed to transactional information systems. BI systems have several components that used to be summarized in a data warehouse. A data warehouse is a collection of integrated and historically recorded data that is used to make strategic decisions using analytical processing techniques. Most of the existing tools for data warehouse development focus on the data storage structure. Data warehouses integrate heterogeneous data and are used by executives to make strategic decisions. Being often proprietary, this data can be sensitive and must be controlled at access, hence the need to secure it [2].

Today the use of computer networks has become essential and our world has migrated to interconnected networks through the Internet [9][10], which makes them a target of attacks that are multiplying day by day. Hence the need to analyse the risks and implement a set of means to minimize the vulnerability of a system against these threats, while auditing the information system; in other words, it is necessary to adopt a security policy whose goal is to ensure: (i) integrity, (ii) confidentiality, (iii) availability and (iv) non-repudiation.

The study of related work on data warehouse security has allowed us to identify two classes of approaches: (i) approaches concerning the security of operations: these works make it possible to who has the right and what does he have the right to?, and (ii) approaches dealing with the prevention of inference problems; they answer the question of how to answer the question of how to prevent a user from inferring protected data from accessible protected data from accessible data?

The objective of this paper is to propose an approach against inferences that complements the On-Line Analytical Processing (OLAP) server functionalities. It controls and prohibits precise inferences made through queries using the Max, Min or Sum aggregation functions. More precisely, this approach exploits the KNN [12] against precise inferences. Compared to existing approaches based on data perturbation [14][4][5], our approach relies on query history; thus, it preserves the integrity of the data and avoids the heavy perturbation processing required after each data warehouse feed.

The rest of this paper is organized as follows. In the next section, we briefly survey and discuss related work on Data warehouse security. In Section 3 we present our proposed approach for solving the problem of inferences and experimental results are given in Section 4 . Finally, in Section 5, we provide some conclusions and outline areas for further research.

2 Related work

A multi-phase methodology for designing data warehouse security was proposed in [11]. The phases of the methodology are: preliminary analysis, design, logical modeling, physical modeling and implementation. Focusing on the preliminary analysis phase, the authors defined two categories of security requirements, basic requirements and advanced requirements. Basic requirements consist of hiding a cube, the faces of a cube, data details, and/or dimensions. While the advanced requirements consist in hiding the details of some faces of a cube, and/or defining security rules depending on the data itself. The requirements defined by the authors cover all the existing data in a data warehouse, Nevertheless, they did not propose an approach for their identification.

Soler et al. [13] proposed a Model Driven Approach (MDA) for the development of a secure data warehouse. This approach exploited the Query View Transformation (QVT) language to automate the passage from the conceptual level to the logical level. The MDA approach proposed by the authors deals with security at the conceptual level and at the logical level; however, these authors

did not provide a method to verify that the security constraints defined at the conceptual level are respected at the logical level.

A comparative study of some research works on warehouse security was presented in [15]. In this work, the authors detailed the security aspect of commercial OLAP tools. The study showed that the physical security (of infrastructures, servers, etc.) is insufficient to guarantee the security of a data warehouse. The authors have defined the security requirements of a data warehouse; however, they have not taken into account the case of inferences.

Sung et al. [14] defined the data security as a means to preserve the privacy of data in cells of a cube while providing query answers with high accuracy and meeting the three objectives of: (i) Security - sensitive data should not be disclosed, (ii) Accuracy - query results should have a high degree of accuracy, and (iii) Accessibility - restrictions should not prohibit legitimate queries. The zero-sum method, which they propose, only takes into account summation queries. It consists in adding random values to cells in order to alter their content. The sums of the random values per row and per column are equal to 0. The zero-sum method preserves the security of the data while answering queries with a high degree of accuracy; however, it requires a significant amount of computation time and only handles sum queries.

Cuzzocrea et al. [4] proposed a framework that allows to generate from a cube A a cube A' respecting the security constraints. These are based on metrics whose reliability is recognized by the data security field [5]. This framework allows to select: fraction of the dimensions, regions of the data with a skewed distribution, and data that satisfy the security constraints for each region. The framework does not require a large amount of computation time. One of the disadvantages of this approach is that it does not handle sum and avg queries, which are very common in data warehouses.

Elkhadir et al. [6] improved the robustness of the LDA method in detecting network intrusions by using the geometric mean vector of the class rather than the sample mean of the class. Experiments on KDDcup99 and NSL-KDD demonstrated the effectiveness of the proposed model, while showing its superiority over some Fisher LDA algorithms such as classical LDA, null-space LDA, median LDA, and direct LDA.

Almansob and Lomte [1] introduced an Intrusion Detection System (IDS) with large amount of data to address the challenges in various types of network attacks using machine learning techniques. On the other hand, a principal component analysis method was proposed to reduce the high dimensionality and characteristics of the data. The DARBAI dataset was used in this model and applied to the nearest neighbor method for classification.

Benaddi et al. [3] attempted to reduce the original features that represent all the connection records stored in a dataset with the aim of detecting intrusions. This fact that data mining gives hidden patterns and explores the data in a different way. The work proposed in this paper shows how they can extract relevant information using PCA-FC-KNN to build a robust IDS with maximum detection rate and minimum false alarms. The experimental results show that

with the increase of data size, the efficiency and accuracy of the intrusion detection algorithm gradually decrease. Compared with QR-OMPCA and Bayesian algorithms, the new algorithm in this paper still has better detection efficiency, especially when they used KNN classifier, they can detect all categories of attacks.

Three reduction techniques, namely Principal Component Analysis (PCA), Artificial Neural Network (ANN) and Non-Linear Principal Component Analysis (NLPCA), have been studied and analyzed in [8]. The performance of four classifiers, namely Decision Tree (DT), Support Vector Machine (SVM), K Nearest Neighbor (KNN) and Naïve Bayes (NB) were studied for current and reduced datasets. In addition, new performance measures, Classification Difference Measure (CDM), Specificity Difference Measure (SPDM), Sensitivity Difference Measure (SNDM), and F1 Difference Measure (F1DM) were defined and used to compare results on the actual and reduced data sets. Comparisons were performed using the new Coburg Intrusion Detection Data Set (CIDDS-2017) as well as the widely referenced NSL-KDD data set.

Xin and Wang [16] analysed the advantages and disadvantages of the existing intrusion detection algorithm, focuses on the in- depth study of the intrusion data feature xuan'ze based on deep learning, proposes a new feature selection method, and conducts experiments with the special intrusion detection dataset, and verifies the scientificity and practicability of the theoretical method by comprehensive comparative experiments and parameter analysis.

3 Proposal of an approach for securing against inferences

Our approach relies on a technique for preventing inferences of potential queries to be forbidden (Figure 1); the latter technique is a practical solution to provide a reasonable response time in the face of the processing required for prevention. Being query history based, it examines the query log to take into account data already accessed by the user. Inference prevention is based on KNN. It handles queries using the Min, Max or SUM functions. In this section, we detail the case of queries using the Max function; the treatment of queries using Min is identical. We start by introducing the basic concepts of KNN and some definitions useful for the presentation of our approach.

3.1 Basic concepts

In this framework, we have a training database consisting of N (input-output) pairs. To estimate the output associated to a new input x, the K nearest neighbors method consists in taking into account (in an identical way) the K training samples whose input is the closest to the new input x, according to a distance to be defined. Since this algorithm is based on distance, normalization can improve its accuracy. In pattern recognition, the K-nearest neighbors (KNN) algorithm is a non-parametric method used for classification and regression. In both cases,

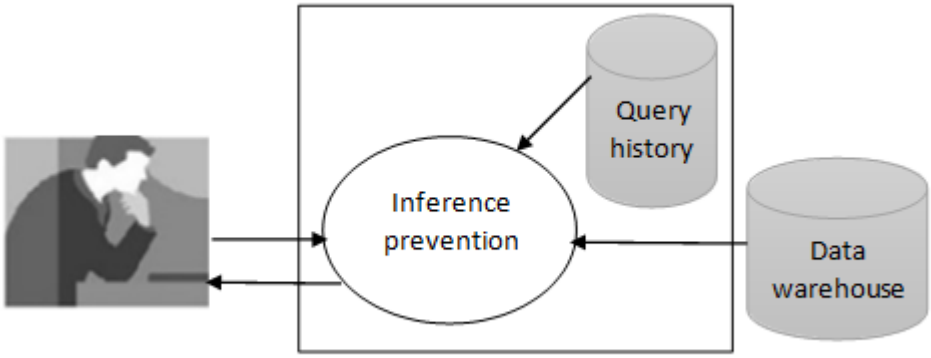


Fig. 1. Architecture of a secure warehouse against inferences

the goal is to classify the input into the category to which the K nearest neighbors in the space of features identified by learning belong. The result depends on whether the algorithm is used for classification or regression purposes.

In KNN classification, the result is a membership class. An input object is classified according to the majority result of the membership class statistics of its K nearest neighbors, (K is a positive integer usually small). If $K = 1$, then the object is assigned to the membership class of its nearest neighbor.

3.2 Definition

The KNN algorithm assumes that similar objects exist nearby in this space (nearest neighbors). In other words, similar things are close to each other. This notion of proximity can be formalized by calculating the distance between points on the graph. The most commonly used distance is the distance Euclidean [7]:

- Given a new text whose language we want to guess according to the frequency of appearance of the letter U and the letter H, let's call $I(XI; YI)$ the landmark associated to this unknown text. - Given a known text from our learning base, let's call $C(Xc; Yc)$ the landmark point associated with this text. - We can calculate the distance between our unknown text and our known text using Equation 1:

$$Distance(I, C) = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2} \tag{1}$$

- We need to compute this distance between point I and all points C_j in our learning base. Then select the K nearest neighbors of I. The value of K is to be defined (generally between 3 and 5 ; It is essential that it is an odd number).

3.3 The KNN algorithm

- Step 1: Select the number K of neighbors.
- Step 2: Calculate the distance.
- Step 3: Take the K closest neighbors according to the calculated distance.
- Step 4: Among these K neighbors, count the number of points belonging to each category.
- Step 5: Assign the new point to the most present category among these K neighbors.

Our database is the number of cases of COVID-19 in Algeria in the year 2020, 2021, 2022 and it is structured by city, day, month and number of cases. First of all, our work is based on the construction of the data set in order to use the KNN. The couples (x, y) in the data set represent the enumeration of the state by their numbers, the months by their indices and the number of cases remains as it is. Each point of the base we calculate its class based on the number of cases. the classes are defined as follows:

class 1 : number of cases is lower than the maximum number of cases /2.
class 2 : number of cases is higher than the maximum number of cases /2.

4 Experimental results

Prediction example:
A user can then try his luck by making a series of requests. He starts with :

Query 1: maximum number of cases in the year.
Query 2: maximum number of cases per month (January, February and December).

From the results of these two queries, it is possible to infer that the maximum number of covid 19 patients was obtained in December.

Query 3: Maximum number of cases per state.

From the result of the third query: it is possible to infer that the state that has marked the peak is ORAN.

Table 1. Result of Query 1.

Year	**Number of cases**
2021	2500

Let's now use our inference prevention approach to prevent the user from inferring the information obtained from the execution of the three queries in

Table 2. Result of Query 2.

Month	Number of cases
February	700
March	1000
December	2500

Table 3. Result of Query 3.

State	Number of cases
Oran	2500
Mostaganem	600
Tlemcen	500
Maascara	400
Sidi bel abbas	700

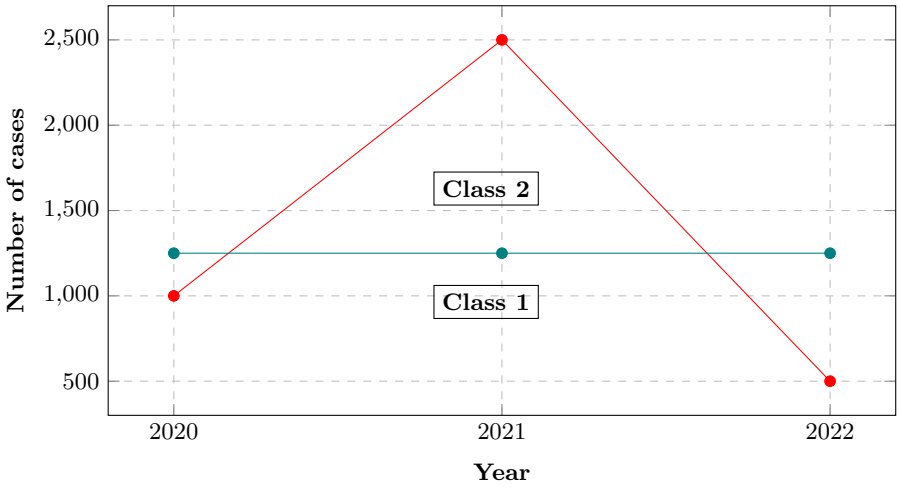


Fig. 2. Graph representing number of cases per year.

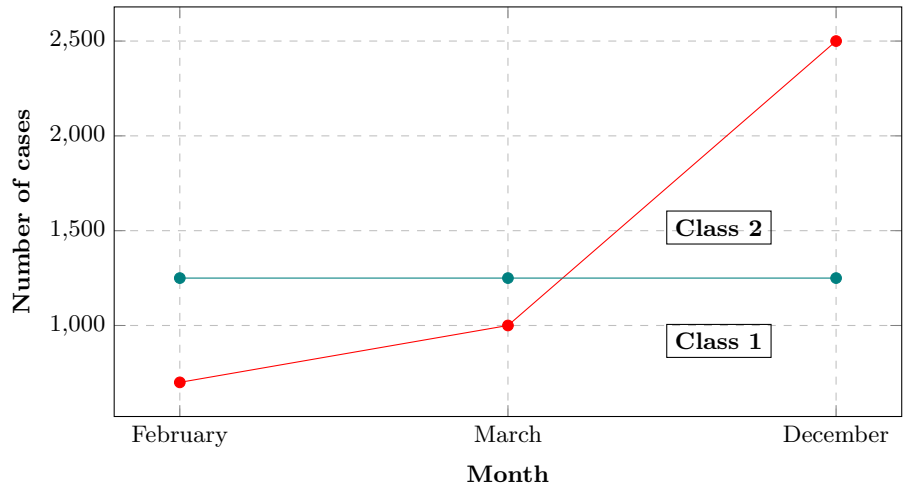


Fig. 3. Graph representing number of cases per month.

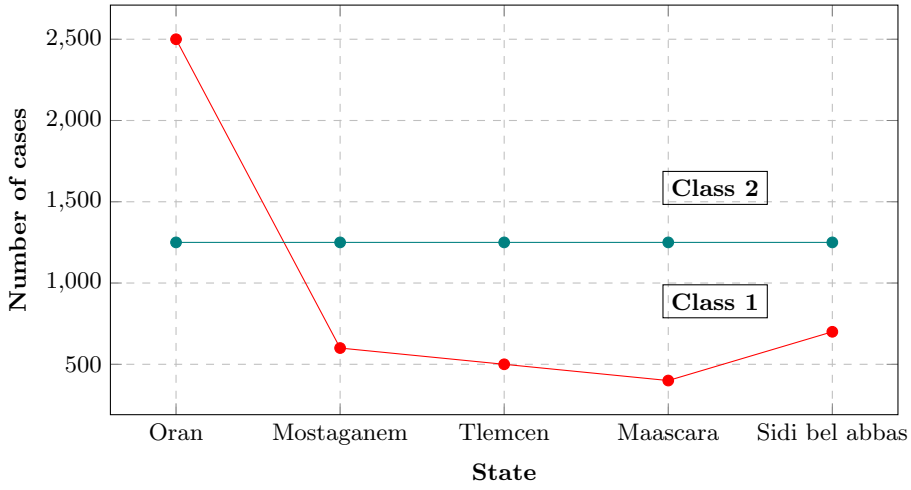


Fig. 4. Graph representing number of cases per city.

the previous example. We start by setting the threshold value to 3 and then calculate the class of each query The first query is in class 2 and so is the second and since the third query belongs to the same class as the two previous queries it is equal to the threshold 3. The result of the third query will not be delivered to the user. This example is only a small part of our database.

5 Conclusion

In this work, we use KNN to overcome the problem of inference detection in data warehouse security. We thoroughly evaluate the proposed approach for data warehouse security using KNN demonstrate that the proposed method succeeds in improving the responsiveness of the On-Line Analytical Processing (OLAP) server and produces a substantial improvement over other methods in terms of compared to other methods in terms of simplicity. And from the point of view of taking into consideration the max, min and sum requests at the same time. Based on the promising results presented in this research, future work will focus on hybridizing KNN with another artificial intelligence algorithm. Another possible research direction is to start SVM for data warehouse security management.

References

1. Almansob, S.M.H., Lomte, S.S.: Addressing challenges in big data intrusion detection system using machine learning techniques. International Journal of Computer Sciences and Engineering **5**(11), 127–130 (2017)
2. Arora, A., Gosain, A.: Intrusion detection system for data warehouse with second level authentication. International Journal of Information Technology **13**(3), 877–887 (2021)
3. Benaddi, H., Ibrahimi, K., Benslimane, A.: Improving the intrusion detection system for nsl-kdd dataset based on pca-fuzzy clustering-knn. In: 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6. IEEE (2018)
4. Cuzzocrea, A., Russo, V., Sacca, D.: A robust sampling-based framework for privacy preserving olap. In: International Conference on Data Warehousing and Knowledge Discovery, pp. 97–114. Springer (2008)
5. Dwork, C.: Differential privacy: A survey of results. In: International conference on theory and applications of models of computation, pp. 1–19. Springer (2008)
6. Elkhadir, Z., Chougdali, K., Benattou, M.: An effective cyber attack detection system based on an improved ompca. In: 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6. IEEE (2017)
7. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier (2011). DOI 10.1016/C2009-0-61819-5
8. Jain, M., Kaur, G.: A study of feature reduction techniques and classification for network anomaly detection. Journal of computing and information technology **27**(4), 1–16 (2019)
9. Khennak, I., Drias, H.: An accelerated pso for query expansion in web information retrieval: application to medical dataset. Applied Intelligence **47**(3), 793–808 (2017)

10. Khennak, I., Drias, H.: Strength pareto fitness assignment for pseudo-relevance feedback: application to medline. Frontiers of Computer Science **12**(1), 163–176 (2018)
11. Priebe, T., Pernul, G.: A pragmatic approach to conceptual modeling of olap security. In: International Conference on Conceptual Modeling, pp. 311–324. Springer (2001)
12. Russell, S.J.: Artificial intelligence a modern approach. Pearson Education, Inc. (2010)
13. Soler, E., Trujillo, J., Fernandez-Medina, E., Piattini, M.: A framework for the development of secure data warehouses based on mda and qvt. In: The Second International Conference on Availability, Reliability and Security (ARES'07), pp. 294–300. IEEE (2007)
14. Sung, S.Y., Liu, Y., Xiong, H., Ng, P.A.: Privacy preservation for data cubes. Knowledge and Information Systems **9**(1), 38–61 (2006)
15. Triki, S., Feki, J., Ben-Abdallah, H., Harbi, N.: Sécurisation des entrepôts de données: Etat de l'art et propo-sition d'une architecture. In: Quatrième Atelier sur les Systèmes Décisionnels, p. 29 (2009)
16. Xin, M., Wang, Y.: Research on feature selection of intrusion detection based on deep learning. In: 2020 International Wireless Communications and Mobile Computing (IWCMC), pp. 1431–1434. IEEE (2020)

Privacy preservation assessment of cancelable biometrics based on set strings templates

Rima Ouidad Belguechi

Laboratoire LMCS, Ecole nationale Supérieure d'Informatique ESI, Algiers, Algeria

Abstract. Over the last years, biometric recognition systems are widely deployed in different sectors. In spite of their popularity, biometric data is subject to the right of privacy preservation since biometric features expose personal sensitive information stored on the cloud server either on large-scale databases. As a result, feature transformation schemes known as cancelable biometrics (CB) have been proposed to prevent the storage of the unprotected template in the database as well as its direct usage in the comparison process. However, unexpected attacks like inversion attack can be conducted on CB to produce a pre-image of the original template, thereby threatening the subject privacy. Today, we lack a meaningful way to ensure privacy preservation in CB schemes. In this paper, we address this shortcoming by considering vulnerabilities in protected templates expressed as valued vectors set. We analyze some breakpoints like pre-image and impersonation attacks. Further, a genetic algorithm (GA) is used to understand the privacy concerns arising on such templates. For their compliance, fingerprints are taken as a case study. The experimental results attest a mitigated resistance to the reversing problem and highlight the weakness to synthetic data presentation attack.

Keywords: Privacy · Cancelable biometrics · Attacks · Genetic algorithm.

1 Introduction

The global biometric market is growing at a rapid pace to combat the increasing cases of security breaches, identity theft and data hacking [1]. Biometrics has no equal in establishing an absolute link between physical and digital identity; it is unique and immutable, ensuring access with high correctness. Contrary to passwords which to be effective, must be complex, frequently modified and unique to each application, biometric authentication is reputed easier to use. In the current context of mobility and strong connectivity while intrusion in network security is increasing, biometric recognition is almost present in IoT devices [2], web banking, blockchain frameworks and on cloud platforms as a developed security service [3]. Biometric systems are based on physiological or behavioural characteristics of individuals (e.g. fingerprint image, iris pattern, signature recognition). A classical verification system extracts a salient features

from the acquired sample, called biometric template, and compares it against the previously enrolled one. For the acceptance criteria, biometric data is considered personal and sensitive, so its collect or storage arise perennial concerns about privacy violation [4]. The assumption that the biometric template is sufficiently compact to not reveal information about the original biometric is wrong. It is now well established that reverse biometric engineering permits the reconstruction of the original biometric with high accuracy [5]. A prominent survey on inverse biometrics is given in [6]. Moreover, once an adversary recovers the stored template, it cannot be revoked and replaced due to its uniqueness. Today, one of the major challenges of biometrics is to guarantee the confidentiality of the biometric template. Cancelable biometrics (CB) has been devised to guarantee privacy by preventing inversion attacks.
CB techniques avoid storing original template, using a transformation function H, which depends on a user-specific key K.
Let X be a biometric template and Y its transformed version. A CB algorithm can be denoted by:

$$Y = H(X, K) \tag{1}$$

X is deleted and only Y will be saved. If Y is compromised, the revocability is ensured by the replacement of the key K. In Fig. 1, we illustrate the architecture of an authentication system by CB where the verification is done on the transformation space between $H(X, K)$ and $H(\grave{X}, K)$, X and $\grave{X}$ being respectively the probe and the query templates.

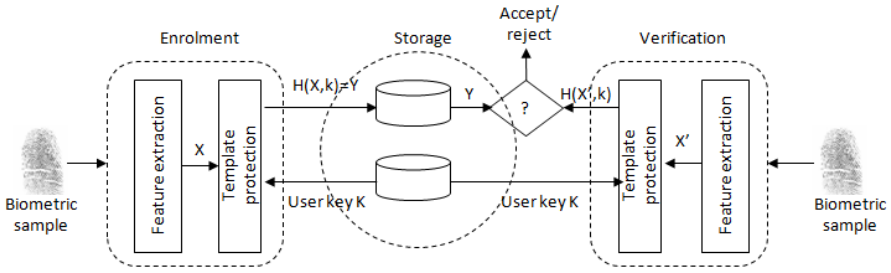


Fig. 1: An architecture of an authentication system based on cancelable biometrics

The ISO/IEC 24745 standard [7] identifies three criteria for biometric information protection : irreversibility, revocability and unlinkability. **Irreversibility** means that the retrieval of the original biometric data from stored protected template is infeasible. **Revocability** requires the system to be able to issue new protected model to replace the compromised one. **Unlinkability** means that several protected templates can be derived from the same original data without being matched. The security of one CB scheme is linked to the given criteria. Unification of an evaluation process is not fixed until now [8]. Moreover, the

definition of metrics remains the main element to quantify privacy leakage. Entropy, which is a common measure of security, is difficult to estimate regarding variations in the biometric data. So, there is still a considerable uncertainty to provide truthful estimation on security level of cancelable biometrics. To tackle the problem, we assess privacy in term of adversarial guessing effort for inversion attack. The contributions of the paper are as follows:

- We present a comprehensive principle of CB followed by the description of common CB schemes in literature.Regarding security and privacy perspectives, we establish a list of possible attacks on CB schemes.
- We propose a generic privacy analysis framework based on genetic algorithm (GA). A former GA based attack was proposed in [9] which only tackle the bit string format. The novelty here is that irreversibility criteria is generalised to all CB that handle unordered set of strings representing the biometric template expressed as $\{x_1, ..., x_m\} \in X$, each x_i is a valued-vector of length P.
- We apply this framework to a fingerprint-based cancelable system. The experimental results identify main weakness towards the design of an accurate CB system mainly in relation of the synthetic biometric sample generation.

The outline of the paper is organized as follows: In section 2, we provide a background about CB schemes. Section 3 presents a detailed GA framework for a generic CB scheme. In section 4, we present a case study on fingerprint-based cancelable system. Finally, we conclude the paper in section 5.

2 Background

In this section, we provide context to the work discussed in this paper.

2.1 CB schemes: principles and review

In cryptography, one-way hash functions are used to guarantee the confidentiality of passwords. Due to intra-class variability, it is not possible to apply them to biometrics. For two biometric templates $x_i, x_j \in X$ and their respective transform $y_i, y_j \in Y$, the matching in the new transformation space must preserve distances so it is a matter of finding the function H which satisfies the following relationship:

$$\parallel d - d_s \parallel < \epsilon, \exists \epsilon > 0$$

With $d = \parallel x_i - x_j \parallel$, the distance between two templates in the space of biometric templates and $d_s = \parallel y_i - y_j \parallel$ the distance between these two templates after transformation.
In mathematical term, this constraint can be modelled by Locality-Sensitive Hashing (LSH) [10] technique that hashes similar input into the same "buckets" with high probability about distance preservation. Building CB is considered as instantiating LSH principle. The transformation function H strongly depends

on the representation of the biometric template. Different modalities will give rise to different representations : continuous or discrete, ordered or unordered, vector or set.

However,in almost all CB schemes, vector or string representation of the input biometric template is useful and preferred, in fact :

In ***BioHashing*** proposed by Teoh et al. [11], the biometric feature is represented in a vector form $x \in R^P$. Using the key K as a seed, N orthogonal random vectors are generated $\{b_i \in R^P, i = 1..N\}$ and $N \leq P$ constituting the matrix B in PxN.The output protected template Y is the N-bit hashcode vector computed as :

$$y = Sig(\sum_i xb_i - \tau)$$

where $Sig(.)$ is the sign function, τ is an empirical threshold. xB is known as random projection (RP) with dimension reduction property. It has been shown in [12] that random mapping can preserve the distances in the sense that the inner product between the mapped vectors closely follows the inner product of the initial vectors. B as non-square matrix constitutes an underdetermined system of linear equations that having many solutions, so even if an impostor steals the key K and generate B, he cannot inverse the matrix B. Hence, the irreversbility of RP is preserved. To reinforce this many-to-one property, BioHashing adds a quantization step by the binarisation function $Sig(.)$.
This stolen-key attack in CB is a general main challenge and has been extensively addressed by researchers [13] :
The ***Partial Hadamard transform*** of Wang et al. [14], is also based on random projection $Y = xB$. However the projection matrix B is formed by selecting some number of rows from the full hadamard transform.
The ***Index-of-Max (IoM)*** technique proposed by Jin et al. [15] transforms the feature vector x into a hashcode of size N by collecting the max indexes generated by repeated random projections.
In ***bloom filters*** [16], the biometric template is devised into P equal size blocks and by using the bloom data structure, a cancelable bit-string is generated.
While bloom filters are more applied to face and iris, random projection is very used to fingerprints. The difficulty to apply random projection to fingerprints is in the extraction of free-alignment fixed-size vector for unordered set of minutiae which is hard to achieve in noisy and rotated images. Minutiae are a salient points representing a fingerprint. A lot of effort has been devoted to this task. In [17], pair-minutiae are quantized and bin indexed to generate a bit string of length 2^P. ***Minutia Cylinder Code (MCC)*** [18] is another effective and high-quality representation of local minutia structures in a set of bit-string. MCC template has been protected with the 2PMCC projection scheme [19] and in the more recent work of Bedari et al. [20]. This explains our interest to template strings representation for this research.

The presented schemes satisfy the CB criteria up to certain level. They also suffer from pitfalls and may be subject to some attacks threatening their security and privacy. We synthetize attacks on CB in the following.

2.2 Attacks on cancelable biometrics

Security of CB schemes is questioned when an impostor succeeded in impersonating legitimate user. He may put forward spoofing the scanner (by presenting an artificial biometric) or conducting some masquerade attacks :
In **Brute-force attack**, no a prior information is needed. The impostor sends randomly estimated values of the protected template Y. If Y is a m-bit length vector and the system has the decision threshold value T then the guessing complexity is $2^m(1-T)$. In **lost token attack**, the impostor knows the user key K and will try to apply this key on its own biometric. In **dictionary attack**, the impostor sends raw biometric data to the feature extractor module selected from a set of the most likely samples to succeed in order to impersonate the legitimate user. By knowing similarity scores, **hill-climbing optimisation attack** can be conducted by modifying the biometric template until the successful recognition.
In **Attack via record multiplicity**, an intruder tries to find correlation among multiple encoded templates created from the same biometric for accessing the original template.
In term of privacy leakage, **inversion attack** (also called pre-image or similarity based attack) is a serious issue that needs to be raised. When an intruder gains access to the templates stored in the database, then by a process of reverse engineering, by knowing the key of the user or not, he will try to generate the model containing the characteristics of original biometrics to break its confidentiality. The pre-image attack attempts to find biometric samples closely similar to the reference model so that after transformation the model will be accepted by the system even if the biometric characteristics are not alike.
There are few works surrounding the inversion attack. Among the most striking, we can cite [21, 9] that entail machine learning algorithms.

3 Privacy analysis framework model

There is still a lack of well-established methodology to properly analyze the cancelable system, no general agreement exists on metrics [8]. Based on literature review, the false acceptance rate (FAR) is a common way to evaluate the success of an attack: Let x_z and $\grave{x}_z$ represent the template and query biometric features of the user z, respectively. Let H be the transformation function. We denote m the dimension of $H(x_z, k_z)$ with k_z the user specific key. Let D_O, D_T be a similarity function in the original and transformed space, respectively. Hence :

$$FAR_T(\epsilon) = P(D_T(H(x_z, k_z), H(\grave{x}_z, k_z)) \geq T \tag{2}$$

Depending on the choice of the decision threshold T, $FAR_T(\epsilon)$ counts the number of acceptance when the person is an impostor.
In this paper, privacy leakage is analyzed considering inversion attack. We suppose the impostor has knowledge of the transformation function H, the user key K, the compromised template $H(X, K)$, and the fitness function $D_T(.)$. The subject of this attack is to estimate a biometric feature A that minimizes the

distance $D_T(H(X,K), H(A,K))$. A is called the pseudo-inverse of the protected template $H(X,K)$. If in addition $D_O(X,A) \leq \epsilon$, A is considered as the total inverse of $H(X,K)$. Since its a complex minimization problem, it can be treated with meta-heuristics. In this work, we use genetic algorithms (GA). GAs are a family of search algorithms inspired by the principles of evolution in nature that rely on the selection and survival of the fittest individuals by the mechanisms of : selection, crossover and mutation [22].

3.1 Problem constraints

In line with literature, CB schemes protect biometric feature expressed as valued-strings by generating bit-strings representation. In a generic way, we can express the problem as follows: The feature extraction module represents the raw data by a set of M fixed-size vectors ($M \geq 1$) :

$$template = \{m_i\}_{i=1}^{M}, m_i \text{ a vector of length } P \text{ (real, integer or bit valued)}$$

, The protection module converts each vector m_i into a binary vector of size N with $N < P$ using a user key K to enable revocability.

$$\text{protected template} = \{pm_i\}_{i=1}^{M}, pm_i \text{ a binary vector of length } N < P.$$

3.2 Search space codification

The search space is a set of possible biometric templates. Each template can be represented by a matrix (chromosome) of size MxP that we can consider as an integer matrix normalized in the range [0..255] (Fig. 2a). The population is a list of population_size chromosomes (Fig. 2b).

$$\begin{bmatrix} 0 & 235 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 55 & 200 \\ 36 & 56 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 68 & 176 \\ 172 & 77 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 87 & 56 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & 53 & \vdots \\ 46 & 87 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 22 & 44 \\ 133 & 134 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 45 & 231 \\ 0 & 65 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 78 & 40 \end{bmatrix}$$

(a) Chromosome

$$\text{Population} = \{ \begin{bmatrix} 123 & \cdots & 34 \\ \vdots & \ddots & \vdots \\ 44 & \cdots & 5 \end{bmatrix} \begin{bmatrix} 27 & \cdots & 41 \\ \vdots & \ddots & \vdots \\ 144 & \cdots & 40 \end{bmatrix} \ldots \begin{bmatrix} 12 & \cdots & 65 \\ \vdots & \ddots & \vdots \\ 86 & \cdots & 46 \end{bmatrix} \}$$

(b) Population

Fig. 2: Codification space for GA

3.3 Genetic algorithm

The main steps for the GA used to reverse the transformed template are in Algorithm 1. By having an accurate fitness function $f(.)$, the selection step can carry out a good evolution of the GA towards the optimum.

In order to compare two binary matrices Ch and Y, a global score denoting the number of paired row vectors from Ch end Y has to be obtained.

Algorithm 1 : Genetic algorithm based inversion attack

Impostor knowledge : the user key K, the compromised template Y, the transformation function $H(.)$ and the fitness function $f(.)$
Input : population_size, Crossover_probability P_c, Mutation_probability M_c, max_iteration, M,P,N and THlocal for the fitness function
Population initialization : Randomly generate population_size integer matrices of size MxP
While max_iteration not reached **do**
1: Using $H(.)$ and K, transform each matrix $Individual$ in the population to a binary matrix of size MxN. The binarization is intrinsic to the function $H(.)$ (i.e. BioHashing,...).
2 : Attribute to each individual represented by its chromosome Ch a *fitness value* by using the fitness function $f(Ch, Y)$ described in algorithm 2
Repeat
3: Select two parents with RouletteWheelSelection method
4 : Apply the crossover operator on the selected pairs with probability P_c else clone the parents for the next generation
Until The size of the new population is population_size
5 : Apply the mutation operator with probability P_m
Output : The best individual to have belonged to the population

As showed in Algorithm 2, we select pairs of vectors that display local similarity above a certain threshold that we call THlocal. We then form the pairs avoiding the repetition of the already paired ones. Local similarity between two binary vectors is calculated using Hamming distance (number of different bits).

Algorithm 2 : Fitness function $f(.,.)$

Input : matrix1, matrix2, M, N, THlocal
1 : Constitute matrix PV of size MxM as $PV[i;j] = 1$ if Hamming distance between binary vectors $matrix1[i,*]$ and $matrix2[j,*]$ is upper than THlocal, 0 elsewise
2 : Go through PV and reset all elements where two vectors i and j appear paired several times (keep only the first occurrence where $PV[i,j] = 1$).
3 : The number of pairs nb_pair is equal to number of 1s remaining in PV
4 : score = nb_pair * 100/ M
Output : score

The crossover operator between two parents is applied horizontally and vertically. An example is illustrated in fig. 3.

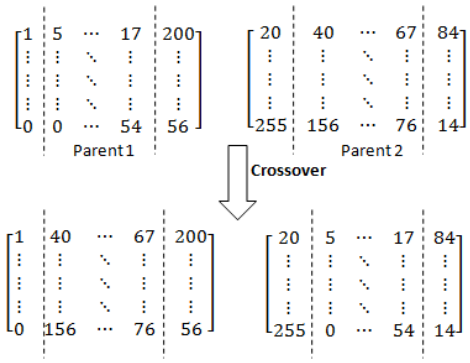


Fig. 3: Vertical crossing following a randomly chosen two points

4 Experimental results and analysis

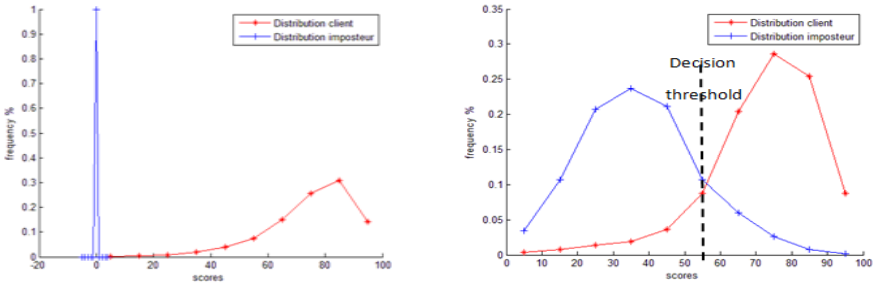
4.1 Attack accuracy

In this section, the GA proposed for attacking generic cancelable biometric system is applied to fingerprint template based on binary strings generated in [23] using a modified version of BioHashing transform.
The size of a chromosome is MxP=30x384, after protection, its MxN=30x180. M represents the average number of minutiae on a fingerprint and 384, the size of the vector that represents each minutiae that becomes 180 bits after BioHashing. We tested our attack design over the public database FVC2002-DB2 [24] composed of 8 fingerprints per individual for 100 individuals with reasonably good quality. The performance indice adopted in our tests is :

- **FAR@ET** : its the false acceptance rate where the threshold of the authentication system is set to the equal error rate (i.e. false acceptance equal to false rejection).

Parameters are set from the genuine and impostor distributions. The first sample image for each finger is used as the enrolled template, the rest are used as query templates, leading to 100 * 7 = 700 genuine scores and (100 x 99) /2 = 4950 impostor scores.
We conducted experiments to evaluate the performance of verification biometric system in three scenarios. The fig. 4a illustrates scores distribution with the protected template using a different key for each user on the system. Fig. 4b depicts the worst case of the lost token scenario. This was simulated by assigning the same K for all users in the dataset.

Table 1 reports the system performance with the fixed decision threshold T. As expected, we notice a performance degradation for the stolen key case.
The GA generates a random population composed of integer matrices and tries to find the individual that matches the compromised binary matrix. It converges



(a) Cancelable authentication system (b) Stolen key scenario

Fig. 4: Legitimate and impostor score distributions according usage scenarios

Table 1: Reference authentication biometric system performance

	Baseline system	Cancelable system	Stolen token case
FAR@ET	5.93%	0	10.68%
Decision threshold	30%	5%	60%

in a time varying between 1 and 30 min. In table 2, we summarize FAR@ET obtained according to the THlocal parameter of the fitness function (see algorithm2). This parameter represents the number of bits retrieved between one estimated protected vector and the original one for each minutiae. The ideal therefore is to recover the 180 binary elements of each row of the template. The FAR@ET metric estimates the average number of broken counts that is to say the success rate of the GA attack. We notice on the first columns of the table that the GA manages to estimate a biometric template which after the BioHashing phase could be accepted by the system in 100% of cases. But we note the weakness of the algorithm to estimate an acceptable model when the THlocal increases, that is to say by putting more constraints on the admissibility of an acceptable template. This is important because considering the Worst case scenario in Table 1 and in order to improve the FAR of the system under the assumption of systematic theft of the key, it is important to have a high THlocal. We aware that when comparing our results with the previous works of Dong et al. [9] there is some differences. Throughout this paper, we focused on attacking templates represented as a set of integer vectors and not only based on a single binary vector, in fact more complex. Dealing with such templates, GA attack is not enough effective if the size of the template is well tuned.

4.2 Reversibility estimation

Considering privacy preservation assessment, we verify whether the biometric template estimated during GA attack would be useful for the attacker.

Table 2: GA attack success rate

	THlocal							
	100 bits	105 bits	110 bits	115 bits	120 bits	125 bits	130 bits	145 bits
Normal FAR@ET	6.11%	1.17%	1.1%	%0.9%	0.5%	0	0	0
GA attack FAR@ET	100%	100%	50%	50%	50%	20%	0	0

Table 3: Average local similarity comparison

Average distance between two minutiae of the original and estimated model	Average distance between two minutiae instances of the same fingerprint
1500	650

1. First we test the irreversibility criterion : In GA attack, we estimated a biometric model $\grave{X}$ from a model protected $H(X, K)$ and the key K. We want to study the relationship between X and the estimated model $\grave{X}$. We study distances by comparing the original model with the one estimated. Average distances obtained are reported in tables 3. We notice that the estimated model cannot be considered as the inverse of the original model because of the large distance between the biometric vectors.
2. We now use this generated model to attack another system with a different key $\grave{K}$: we study the similarity between $H(X, \grave{K})$ and $H(\grave{X}, \grave{K})$ for different THlocal values. We obtain the same previous remark, the more we increase the constraints on the local similarities, the more the overall score decreases and therefore the attack becomes ineffective. This avoids tracking individuals which increases privacy preservation.

4.3 Comparison with other attacks

1. In **the brute force attack**, we randomly generate biometric templates $\grave{X}$ and we estimate the similarity score between $H(X, K)$ and $H(\grave{X}, K)$. In Table 4 , we represent a comparison between the results of this attacks with GA attack. We notice that the attack by the GA has higher success rate.
2. In **dictionnary attack**, we generate a synthetic fingerprint using the SFinge tool[25]. We then submit this fingerprint to our system to generate a template biometric $\grave{X}$. We then make the comparisons between $H(X, K)$ and $H(\grave{X}, K)$. We set THlocal(145/180) where we saw that the genetic algorithm always obtained scores at 0%. The average similarity score obtained over the database is 50%. We conclude by saying that the use of a synthetic image brings an attack success more relevant than GA or brute force attack.

5 Conclusion

In this work, we implemented a GA to inverse cancelable templates expressed as a set of unordered strings of integer for privacy assessment. Contrary to binary

Table 4: Comparison between the convergence scores of GA attack and brute force attack.

	THlocal							
	100 bits	105 bits	110 bits	115 bits	120 bits	125 bits	130 bits	145 bits
Average score by GA	92%	71%	43%	19%	9%	3%	0	0
Average score by brute force	66%	36%	14%	6%	0	0	0	0

vector format, we conclude that this representation is more difficult to inverse with GA and we point the relevant problem of synthetic generation of raw biometrics which exhibits more correlation with the original data. In the future work, other metheuristics can be checked for a better optimisation of the fitness value and we would explore the attack when multiple protected templates of the same user are compromised simultaneously.

References

1. https://www.researchandmarkets.com/reports, (2020)
2. Liu, S., Shao, W., Tan, Li., Xu, W., Song, L.: Recent advances in biometrics-based user authentication for wearable devices: A contemporary survey. Digital Signal Processing, (2021)
3. Sudhakar, T., Gavrilova, M.: Cancelable biometrics using deep learning as a cloud service. IEEE Access journal **8**, 112932–112943 (2020)
4. Campisi, P.: Security and privacy in biometrics. Springer, (2013)
5. Jain, A. K., Nandakumar, K., Nagar, A.: Biometric Template Security. EURASIP Journal on Advances in Signal Processing **35**(12), 1–178 (2008)
6. Gomez-Barrero, M., Galbally, J.: Reversing the irreversible : A survey on inverse biometrics. Computers & Security **90**, 1–178 (2020)
7. ISO/IEC JTC1 SC27: ISO/IEC 24745:2022. Information Technology – Security Techniques – Biometric Information Protection.(2022)
8. Simoens, K., Yang, B., Zhou, X., Beato, F., Busch, C., Newton, E.M., Preneel, B.: Criteria Towards Metrics for Benchmarking Template Protection Algorithms. In 5th IAPR International conference on biometrics, pp. 498—505. IEEE, (2012)
9. Dong, X., Jin, Z., Jin, A.T.B.:A genetic algorithm enabled similarity-based attack on cancellable biometrics. In : 10th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), (2019)
10. Jafari, O., Maurya, P., Nagarkar, P., Islam, K. M., Crushev, C.:A survey on locality sensitive hashing algorithms and their applications, (2021)
11. Teoh, A. B. J., Goh, A., Ngo, A. B. J.: Random multispace quantisation as an analytic mechanism for BioHashing of biometric and random identity inputs. IEEE Trans. Pattern Anal. Mach. Intell, **28** (2006)
12. Kaski, S. : Dimensionality reduction by random mapping. In Int. Joint Conf. on Neural Networks. pp. 1—8. (1998).
13. Manisha, R.,Nitin, K.: Cancelable Biometrics: a comprehensive survey. Artificial intelligence Review,**53**(5), 3403–3446 (2020)
14. Wang, S., Deng, G., Hu, J.: A partial Hadamard transform approach to the design of cancelable fingerprint templates containing binary biometric representations. Pattern Recognition, **61**, 447–458 (2017)

15. Jin, Z., Hwang, J.Y., Lai, Y.L, Kim, S., Teoh, A.B.J. :Ranking-Based Locality Sensitive Hashing-Enabled Cancelable Biometrics: Index-of-Max Hashing. IEEE Transactions on Information Forensics and Security. **13**(2), (2018)
16. Rathgeb, C., Breitinger, F., Busch, C. : Alignment free cancelable iris biometric templates based on adaptive bloom filters. In 2013 international conference on biometrics (ICB). pp. 1-–8. IEEE (2013).
17. Wang, S., Hu, J.: Design of alignment-free cancelable fingerprint templates via curtailed circular convolution. Pattern Recognition, **47**(3), 1321–1329 (2014)
18. Cappelli, R., Ferrara, M., Maltoni, D.: Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition. IEEE transactions Pattern Analysis and Machine Intelligence, **32**(12), 2128–2141 (2010)
19. Ferrara, M., Maltoni, D., Cappelli, R.: A two-factor protection scheme for MCC fingerprint templates. In : 2014 International Conference of the Biometrics Special Interest Group (BIOSIG), pp. 1-8. IEEE,(2014)
20. Bedari, A., Wang, S., Yang, W.: Design of cancelable MCC-based fingerprint templates using Dyno-key model. Pattern Recognition, **119**, (2021)
21. Feng, Y. C., Lim, M.-H, Yuen, P.C. :Masquerade attack on transform-based binary-template protection based on perceptron learning. Pattern Recognition, **47**(9), 3019–3033 (2018)
22. Bergel, A.: Agile Artificial Intelligence in Pharo - Implementing Neural Networks, Genetic Algorithms, and Neuroevolution. Apress Media, (2021)
23. Belguechi, R., Cherrier, E., Rosenberger, C., Ait-Aoudia, S.: An integrated framework combining Bio-Hashed minutiae template and PKCS15 compliant card for a better secure management of fingerprint cancelable templates. Elsevier Journal on Computers & Security, **39**(3), 325–399 (2013)
24. Fingerprint verification competition, http://bias.csr.unibo.it/fvc2002/. 2022
25. Biometric system laboratory, SfinGE, http://biolab.csr.unibo.it

A review of Lightweight block ciphers for Embedded based devices

Amina Souyah and Mohammad Hammoudeh

[1] LabRi Laboratory, Ecole Superieure en Informatique, Sidi Bel Abbes, Algeria
[2] School of Computing, Mathematics and Digital Technology
Manchester Metropolitan University, United Kingdom

Abstract. Security in communication over the Internet has turned out to be more complex, due to the advent of advanced technology and the frequent exchange of confidential data. In particular, when dealing with resource limited environments such like embedded systems, IoT based systems, and wireless sensors. The demand of security on such environments has motivated cryptographers to design lightweight cryptographic primitives that can make a good trade-off between security, cost and performance. This paper provides a review of popular contemporary lightweight block ciphers that have been designed and tailored for resource-limited environments, introduces a comparative analysis between these lightweight proposals in terms of some specific metrics, and concludes with discussion about some research issues to consider when dealing with the design of new efficient block ciphers.

Keywords: Cryptography, lightweight block cipher, Embedded devices, IoT systems, building block.

1 Introduction

Lightweight cryptographic primitives are designed and optimized to be efficiently workable on resource-constrained devices, such like embedded devices, IoT based devices, RFID tags that are restricted with respect to their processing power, energy consumption, memory, and battery power. In such environments, it is of an important concern to design cryptographic primitives that are as lightweight as possible, along with ensuring satisfactory level of security to withstand possible attacks [1,2]. To this end, a number of lightweight cryptographic primitives have been developed with a good compromise between security, cost and performance in mind. This paper provides a review of popular contemporary lightweight block ciphers that have been designed to meet resource-limited environments requirements, presents a comparative analysis between these lightweight proposals in terms of some specific metrics, and then with discusses some research issues to consider when dealing with the design of new efficient block ciphers.

The rest of the paper is organized as follows: Embedded systems constraints are discussed in section 2. Existing lightweight cryptographic primitives are presented in section 3, through which a total focus is given to lightweight block

ciphers, by introducing different employed building block techniques. A description of the fundamental features of lightweight cryptographic primitives, a comparison of different lightweight block ciphers, and the a discussion about some research issues to consider when dealing with lightweight block cipher design are covered in section 4. Section 5 concludes this paper.

2 Embedded systems constraints

Embedded systems are characterized by their inherent restrictions regarding the reduced computing power, limited memory (registers, RAM, ROM), low battery power (or no battery) small physical area and limited energy consumption [1, 3–5]. Internet of things (IoT, for short) is defined as a network of connected embedded physical devices (things) that are uniquely identified within the network, and capable to gather and exchange data over the internet with or without human interaction [4]. Security over embedded systems such like IoT remains a challenging issue due to the limited resources embedded in IoT devices from one hand, and the issue of real-time applications from the other hand. Under these considerations, substantial efforts are devoted to design new efficient cryptographic primitives that can deal well with the new requirements of these resource-constrained devices.

3 Lightweight cryptographic primitives

Lightweight cryptography (LWC, for short) refers to the incorporation of lightweight cryptographic primitives that represent the basic building block in the design and realization of lightweight cryptographic means tailored for resource-constrained devices. Lightweight cryptography is primarily subdivided into two categories: symmetric lightweight cryptography and asymmetric lightweight cryptography (see Figure 1). In symmetric lightweight cryptography, four types of lightweight cryptographic primitives are approved by NIST and CRYPTREC [6], which are: lightweight block ciphers, lightweight stream ciphers, lightweight hash functions and lightweight message authentication code and authenticated encryption. When employing these lightweight cryptographic primitives as basis to design lightweight symmetric cryptographic means, it is intended to provide performance advantages over conventional cryptographic algorithms, by achieving a better balance between security, time efficiency and resource requirements for specific resource-constrained environments like IoT systems [6]. In what follows, the main focus will be given on symmetric lightweight block ciphers.

3.1 Lightweight block ciphers

Conventional block ciphers fall under symmetric cryptographic algorithms. They are the most commonly employed cryptographic primitives on rich resources applications. A well-designed block cipher should mandatory guarantee the confusion and diffusion properties introduced by Shannon in his paper [7]. Basically,

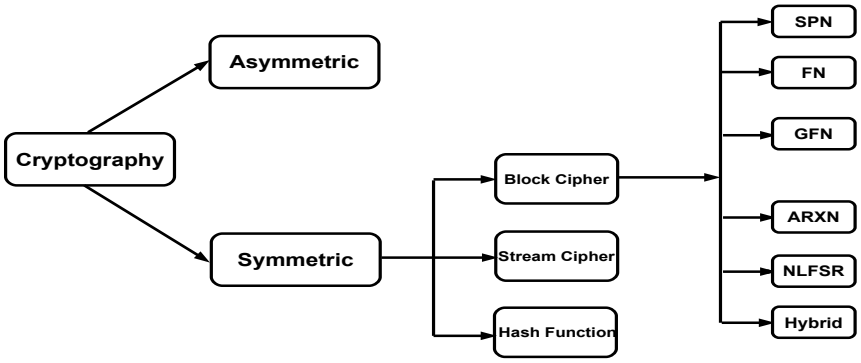


Fig. 1. Classification of lightweight cryptographic primitives.

a block cipher takes as input a block of a fixed size of data (plaintext and key), and produces a block of the same fixed size as output (ciphertext). On the other hand, lightweight block ciphers have been designed to be suited for resource-constrained devices such like IoT systems, WSN, RFID tags, embedded systems, etc. lightweight block cipher is featured by smaller block size, smaller key size, simpler key schedule routine, simpler rounds, and with a minimal implementation compared to conventional block cipher, aiming to attain a better balance between security, time efficiency and resource requirements. On the basis of their internal structure, lightweight block ciphers can be classified as follows:

Substitution Permutation Networks (SPNs)

In the substitution permutation-based block cipher, the plain blocks undergo a series of sequential substitution and permutation boxes to produce cipher blocks as output (see Figure 2). Such two important mechanisms are employed with the aim of achieving the confusion and diffusion properties of a well-designed cipher. The substitution module that is a non-linear primitive renders the relationship between the block cipher's input and its matching output as complex as possible (confusion property), such non-linear mechanism is mainly ensured by employing either one s-box like AES conventional block cipher, or multiple s-boxes like Blowfish conventional block cipher. The permutation module that is a linear primitive works to make any bit-alteration on the plaintext or the key spreads over many bits on the ciphertext (diffusion property), such linear mechanism is usually ensured by using permutation P-boxes like PRESENT lightweight block cipher or matrix operation like AES conventional block cipher. It is worthy to mention that Advanced En-

cryption Standard (AES) is the best popular conventional block cipher that follows the SPN structure. For the sake of responding to the challenging issue of ensuring data confidentiality over recent applications like the ones running of resource-constrained devices and real-time applications, a lot of lightweight SPN-based block ciphers have been figured in the scientific literature among which: PRIDE,NOEKEON,ICBERG,PRESENT,PUFFIN-2,mCrypton,PRINTcipher,

EPCBC,LED,KLEIN,PRINCE,PICARO, and SKINNY [8–21].

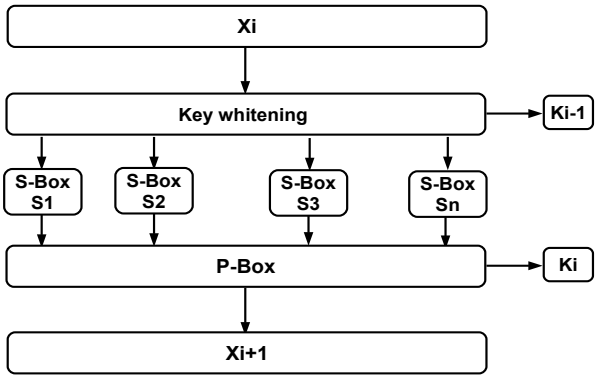


Fig. 2. Substitution Permutation Network (SPN) structure for block ciphers.

mCrypton: is a lightweight block cipher proposed by C.Lim et al. in their contribution [13]. The proposal is of 64-bit block length, and supports three key size options, namely 64-bits, 96-bits and 128 bits for different level of security (minimal, moderate and standard security, respectively). mCrypton block cipher is a SPN-based, and its core components adopt the structure of Crypton [22] with devoted simplification of each transformation function to ensure much compact implementation in both hardware and software, and hence adapt the cipher for use on resource-constrained environments. mCrypton block cipher is mainly performed under 13 rounds, each of which comprises four stages: nonlinear substitution, bit permutation, column to row transposition and key addition for each encryption round. The mechanism of key scheduling consists of the employment of nonlinear s-box to generate the sub-keys of each round and rotation operation at both word and bit level for key variables updating.

PRESENT: is a lightweight block cipher proposed by A.Bogdanov et al. in their contribution [11]. The proposal is of 64-bit block length, and supports two key size options, namely 80-bits and 128 bits for different level of security (moderate and standard security, respectively). PRESENT block cipher is an

SPN-based, which is designed with a careful way aiming to meet high hardware implementation efficiency and to be well suited for use on resource-constrained environments, it is also considered as one of the first ultra-lightweight block ciphers, that is standardized in ISO/IEC [3]. PRESENT block cipher is mainly performed under 31 rounds, each of which comprises three stages: XOR operation with the round sub-key, a linear bitwise permutation and a nonlinear substitution whereby only a single S-Box is employed rather than multiple S-Boxes, and at the end a XOR operation with the last sub-key is conducted for post-whitening. The mechanism of key scheduling consists of: at first storing the user key in key register K, then the 64 leftmost bits of the register key K represents the 64-bits of the round sub-key, after that the content of the key register must be updating by applying 61-bits left rotation, then PRESENT S-Box is performed to the leftmost four-bits, and the round counter value denoted as i is exclusive-ored with a five selected bits of the key with the least significant bit of round counter on the right.

Feistel Networks (FNs)

Feistel network is another building block structure in the design of block ciphers which was proposed by the cryptography Horst Feistel in his design of Lucifer [23]. In the Feistel network-based block cipher, the plain block is divided in two halves (see Figure 3), a round function F that is the fundamental building block of any Feistel network, provides the confusion property overall the cryptographic scheme, and the more nonlinear this function is, the difficult cryptanalysis will be. This nonlinear round function F is applied to one-half of the plain block within the encryption round, and its output is then exclusive-ored with the other half for reaching the diffusion property. The final cipher blocks are obtained after passing through a number of identical encryption rounds besides to any initial and/or final extra procedure if exists. Moreover, the employed sub-keys within each encryption round are produced from the original key by following the key scheduling mechanism, and the decryption can be performed by applying the same encryption algorithm with reverse-ordered round sub-keys. It is worth noting that Data Encryption Standard (DES) is the best popular conventional block cipher that follows the FN structure. Block ciphers which are Feistel network based are featured by their little cost for both encryption and decryption functionalities, however a lot of lightweight Feistel network based block ciphers suffer from serious security problems, opposed to SPN-based block ciphers which are characterized by their high security level. Furthermore, the decryption functionality within many tag-based applications is uncommonly needed, accordingly SPN-based block ciphers can be considered as a strong competitor and the best suited in use for lightweight cryptography field [3]. Aiming to ensure data confidentiality over recent applications like the ones running of resource-constrained devices and real-time applications, a lot of lightweight Feistel network-based block ciphers have been figured in the scientific literature

among which: SEA, CLEFIA, Piccolo, LBlock, TWINE, SIMON, MIBS, GOST, FeW, Robin, HISEC and RoadRunneR [24–35].

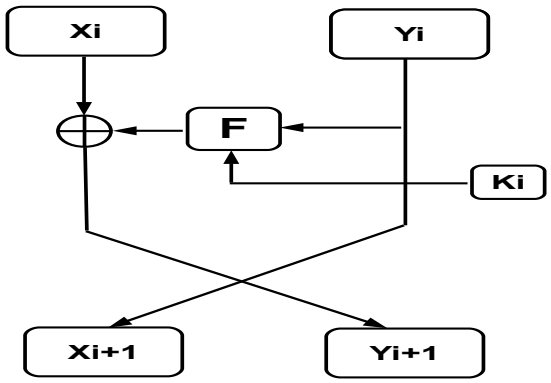


Fig. 3. Feistel Network (FN) structure for block ciphers.

CLEFIA: is a highly efficient block cipher proposed by T. Shirai et al. in their contribution [25]. The proposal is of 128-bit block length, and supports three key size options, namely 128-bits, 192-bits and 256 bits such like AES block cipher. CLEFIA block cipher is Feistel network-based, which is designed with the scope to attain a good balance in both security level, time-efficiency and cost implementation to be well suited for use on resource-constrained environments, it is designed by SONY and standardized in ISO/IEC 29129. CLEFIA block cipher is mainly performed under 18, 22 and 26 rounds, and it employs a 4-branch generalized Feistel network (GFN for short) that is perceived as an extension of the classical Feistel network, this GFN structure utilizes two F-functions per round for the sake of reducing the overall number of rounds, these F-functions use the Diffusion Switching Mechanism (DSM for short), in which two diffusion matrices and two 8-bit S-boxes are conducted aiming to withstand known attacks and to meet efficient implementation especially in hardware. The mechanism of key scheduling consists of the employment of the generalized Feistel network (GFN) and the DoubleSwap function to produce the needed sub-keys for each encryption round. The most compact implementation is performed under 2488 GE for the encryption procedure only, an extra 116 GE is needed to achieve the decryption procedure. Moreover, the most lightweight implementation for both encryption and decryption can be realized with 2604 GE, which is 23 % smaller than AES with 128-bits key [3].

SIMON: is a lightweight block cipher proposed by NSA [29]. The proposal is of 32-bits, 48-bits, 64-bits, 96-bits and 128-bits block lengths, supports seven key

size options, namely 64-bits, 72-bits, 96-bits, 128-bits, 144-bits, 192 bits and 256 bits, and performed under different round numbers 32, 36, 42, 44, 52, 68, 69 and 72. SIMON block cipher is Feistel network-based, which is designed by employing the simplest components possible for attaining the desired compromise between linear diffusion and nonlinear confusion properties, and the searched flexibility to be well suited for use on resource-constrained environments, its performance evaluation is introduced in [29]. The fundamental structure of SIMON block cipher consists of employing the 2-branch traditional Feistel network, in which a nonlinear function is performed to the left half sub-block, the output of this later is then exclusive-ored with the other half along with the round sub-key, and after the two halves sub-blocks are swapped. The mechanism of key scheduling consists of the employment of two rotations to the right and XOR the obtained results together with a fixed constant and five constant sequences that are version dependent to produce the needed sub-keys for each encryption round.

Add-Rotate-XOR Networks (ARXNs)

ARXNs based block cipher employs modular addition, rotation and XOR operation as fundamental operations in the design and realization of the cipher. These three operations diminish RAM memory consumption but at the cost of weakening cryptographically the block cipher, leading to the use of large number of rounds for the sake of achieving satisfactory level of security. The security property of ARXNs block ciphers is not studied extensively as compared to SPN and Feistel network based ciphers. It should be pointed out that IDEA is the best popular conventional block cipher that follows the ARXNs structure. As examples of lightweight block ciphers that follow ARXN structure, we state: SPECK [29], HIGHT, LEA and BEST-1, and SPARX [36–39].

HIGHT: is a lightweight block cipher proposed by D.Hong et al. in their contribution [36]. The proposal is of 64-bit block length, and supports a 128-bit key size. HIGHT block cipher is an ARXN-based, which is designed to be well suited to the hardware implementation on the resource-constrained environments such like RFID tag, and as each operation within HIGHT is 8-bit processor oriented, HIGHT can provide efficient performance within the 8-bit oriented software implementation that can outperform the performance of AES block cipher with 128-bit key size. HIGHT block cipher employs simple operations (e.g., XOR, modular addition and bitwise rotation) jointly with 8-branch generalized Feistel network (GFN for short), and it undergoes 34 rounds in which: the first round consists of an initial transformation that transforms the plain blocks to the input of the round function by means of the produced four whitening keys, then the next 32 rounds consist of employing a round function, which is governed by GFN inner structure combined with the simple aforementioned operations and ruled by means of the generated sub-keys, after the last round comprises a final transformation that transforms the resulted blocks from the previous step to a final cipher blocks using four produced whitening keys. The mechanism of key scheduling is based mainly on LFSR stream cipher and modular addition,

to produce the needed eight whitening keys and sub-keys for each encryption round.

SPECK: is a lightweight block cipher proposed by NSA [29]. Similarly to SIMON, SPECK also performs well in both software and hardware, however the former is more optimized and oriented for hardware, and the latter for software implementations. The proposal employs the same block and key lengths as SIMON under 22, 23, 26, 27, 28, 29, 32, 33 and 34 rounds. SPECK block cipher is ARXN-based, the round function consists of using simple operations (e.g., XOR, modular addition and right/left rotations) and the adoption of 2-branch Feistel network structure, at each round: firstly the left plain sub block is circularly shifted by a fixed number of bits to the left, modular addition is applied to left and right sub blocks, after XOR operation is handled between the round sub-key and the left sub block, next the right sub block is circularly shifted by a fixed number of bits to the right, finally XOR operation is handled between the left and right sub blocks. The mechanism of key scheduling is based on the SPECK round function.

Non-Linear Feedback Shift Register based (NLFSR)

NLFSR-based block cipher employs the building block of a stream cipher, and it is commonly used for hardware implementations. The security of a block cipher that adopts NLFSR structure is relied on stream cipher analysis. It should be emphasised that KeeLoq is the best popular conventional block cipher that follows the NLFSR structure. As examples of lightweight block ciphers that follow NLFSR structure, we report: KeeLoq, KATAN and KTANTAN family [40, 41].

KATAN and **KTANTAN:** are highly efficient oriented hardware block ciphers proposed by C.D. Canniere et al. in their contribution [41]. The proposals adopt 32-bit, 48-bit and 64-bit block lengths, and support 80-bit key size under 254 rounds. KATAN and KTANTAN ciphers are designed to cope with hardware implementation requirements for resource-constrained environments such like RFID tag, it should be noted that KTANTAN ciphers family are well suited on devices in which the key is first initialized and doesn't undergo any changes. The ciphers adopt mainly the design of KeeLoq cipher under less number of rounds (254 rounds), employ LFSR rather than NLFSR structure, and performed under the following steps: at first the plain block is loaded into two registers, then at each round a number of bits is taken from the registers and becomes the input of two nonlinear boolean functions, after the output of these nonlinear boolean functions is shifted and then loaded to the least significant bits of the registers, such mechanism is repeated through 254 rounds.

Hybrid

Hybrid ciphers combine both SPN, FN, and ARXN for the sake of benefiting from the advantage performance of each building block structure, such like throughput. The best known hybrid block ciphers are those of Hummingbird

family [42,43], that is a particular example of a hybrid structure including stream and block cipher.

Hummingbird (HB): is an ultra-lightweight cryptographic algorithm proposed by D.Engels et al. in their contribution [42]. The proposal is of 16-bit block length, and supports 256-bit key size and 80-bit internal state. Hummingbird cipher is both block cipher-based and stream cipher-based, which is designed to be well suited to both software and hardware implementations for the resource-constrained environments such like RFID tag and wireless sensor nodes. Hummingbird cipher is mainly performed under 20 rounds, it consists of four 16-bit block ciphers, four 16-bit internal state registers, 16-bit stage LFSR, and four 64-bit sub-keys which are derived from the secret key of 256-bit length, and it is performed as follows: at first a modular addition is applied to the 16-bit plain block and the content of the first 16-bit internal state register, then the resultant 16-bit are enciphered by means of the first block cipher, these two steps are repeated for three additional times to produce the first 16-bit cipher block. The content of four internal state registers is updated by means of their current states, the output of first three block ciphers, and the state of LFSR.

Hummingbird-2 (HB-2): is a lightweight authenticating encryption primitive proposed by D.Engels et al. in their contribution [43]. The proposal is of 16-bit block length, and supports 128-bit key size, 128-bit internal state, and 64-bit initial vector (IV). Hummingbird-2 authenticated encryption algorithm is both block cipher-based and stream cipher-based, which is designed particularly for resource-constrained environments such like RFID tag and wireless sensor nodes. Hummingbird-2 cipher is mainly based on XOR operation, modular addition and nonlinear mixing function performed all on 16-bit words, the nonlinear mixing function comprises 4-bit S-box permutation lookups on every nibble of the 16-bit word, and linear transformation which is left-rotation based. Hummingbird-2 cipher design fulfills the requirements of IS0 18000-6C protocol. Hummingbird-2 cipher has as an extra convenience among its predecessor Hummingbird and other existing lightweight encryption primitives that it ensures both confidentiality and integrity protection for messages, by producing a message authentication code.

3.2 The fundamental features of lightweight cryptographic primitives

The three fundamental features of lightweight cryptographic primitives and their offerings are illustrated in Table 1 [4, 44].

As it is above-mentioned in Table 1, physical cost, performance and security are the leading concerns that should be carefully studied, in the design of lightweight cryptographic primitives for resource-constrained devices. The physical cost can be regarded through memory demand, energy consumption and physical space. The performance cost can be estimated via the processing power (latency and throughput metrics). To meet the aforementioned two features, smaller block lengths, smaller key sizes, simpler key scheduling, simpler round functions and less number of rounds. The security can be checked and analyzed

through block/key lengths (to assess possible key attacks) and other possible attacks such like side-channel and fault-injection attacks, and it is guaranteed by the employment an adequate internal structure(s) and robust round function(s).

Table 1. Fundamental features of lightweight cryptographic primitives.

Fundamental features		Achieved by
Physical cost	Physical area (Gates Area, logic blocks)	Smaller block lengths Smaller key sizes Simpler key scheduling simpler round function(s)
	Memory (registers, RAM, ROM)	
	Battery supply	
performance	Processing power (latency and throughput)	less number of rounds
Security	Minimum securith strengths (bits)	Robust building block mechanism
	Key-related attacks	
	Side-channel and Fault injection attacks	

3.3 Comparison of different lightweight block ciphers

A number of lightweight block ciphers have been proposed in the scientific literature, to deal with the challenging issue of confidentiality preservation over resource-constrained devices, especially under real-time applications scenarios. These proposals designed by employing different building block mechanisms, aiming to achieve a good compromise between physical cost, performance and security. Table 2 gives a comparative analysis between a list of lightweight block ciphers in terms of some specific metrics.

3.4 Discussion and open research challenges

A block cipher is a widely investigated cryptographic mean that is used to guarantee various cryptographic security goals among which : confidentiality (data encryption), integrity (building hash function), message authentication (building hash function for digital signature mechanism),... soon. A well designed cryptographic mean should maintain a proper trade-off between security, cost and performance, however it is reachable to fulfill any two of these searched metrics whilst achieving them all in the same designed cryptographic mean is challenging [4, 34].

The existing lightweight block ciphers differ in their inner construction as depicted in table 2, for example in what concerns the building block mechanism, some block ciphers employ SPN structure aiming to achieve high security degrees, others use FN structure to benefit from the fact that encryption mechanism is similar to decryption (i.e., no cryptographic primitive inverse to compute) aiming to reduce ROM memory, others are ARXN structure related to minimize memory use, others favorite efficiency in hardware implementation as the case of

Table 2. An overview of a list of lightweight block ciphers.

LWC Block cipher	Key size	Block length	No. of rounds	Structure	Security goal	Target
BEST-1	128	64	12	ARXN-based	Confidentiality	Software+Hardware
CLEFIA	128/192/256	128	18/22/26	FN-based	Confidentiality	Software+Hardware
EPCBC	96	48/96	32	SPN-based	Confidentiality	Hardware
FeW	80/128	64	32	FN-based	Confidentiality	Software
GOST	256	64	32	FN-based	Confidentiality	Software+Hardware
HIGHT	128	64	32	ARXN-based	Confidentiality	Hardware
HISEC	80	64	15	FN-based	Confidentiality	Software+Hardware
Hummingbird (HB)	256	16	20	Hybrid	Confidentiality	Software+Hardware
Hummingbird-2 (HB-2)	128	16	4	Hybrid	Confidentiality+ Message authentication	Software+Hardware
ICBERG	128	64	16	SPN-based	Confidentialy	Hardware
KATAN	80	32/48/64	254	LFSR-based	Confidentiality	Hardware
KTANTAN	80	32/48/64	254	LFSR-based	Confidentiality	Hardware
KeeLoq	64	32	528	NLFSR-based	Confidentiality	Hardware
KLEIN	64/80/96	64	12/16/20	SPN-based	Confidentiality	Software+Hardware
LBlock	80	64	32	FN-based	Confidentiality	Software+Hardware
LEA	128/196/256	128	24/28/32	FN-based	Confidentiality	Software+Hardware
LED	64/128	64	32/48	SPN-based	Confidentiality	Software+Hardware
mCrypton	64/96/128	64	13	SPN-based	Confidentiality	Software+Hardware
MIBS	64/80	64	32	FN-based	Confidentiality	Hardware
NOEKEON	128	128	16	SPN-based	Confidentiality	Software+Hardware
PICARO	128	128	12	SPN-based	Confidentiality	Hardware
Piccolo	80/128	64	25/31	FN-based	Confidentiality	Hardware
PRESENT	80/128	64	31	SPN-based	Confidentiality	Hardware
PRIDE	128	64	20	SPN-based	Confidentiality	Software
PRINTcipher	80/160	48/96	48/96	SPN-based	Confidentiality	Software+Hardware
PUFFIN-2	80/128	64	34	SPN-based	Confidentiality	Software+Hardware
PRINCE	128	64	12	SPN-based	Confidentiality	Hardware
SEA	48/96/144	48/96/144	51/93/135	FN-based	Confidentiality	Software+Hardware
SPECK	64/72/96/128/144/ 192/256	32/48/64/96/ 128	22/23/26/27/ 28/29/32/33/34	ARXN-based	Confidentiality	Software+Hardware (more Software oriented)
SIMON	64/72/96/128/144/ 192/256	32/48/64/96/ 128	32/36/42/44/ 52/68/69/72	FN-based	Confidentiality	Software+Hardware (more Hardware oriented)
TWINE	80/128	64	36	FN-based	Confidentiality	Software+Hardware
RoadRunneR	80/128	64	10/12	FN-based (using S-Box)	Confidentiality	Software+Hardware
SKINNY	64-384	64/128	32-56	SPN-based	Confidentiality	Software+Hardware
SPARX	128/256	64/128	24-40	ARXN-based (using S-Box)	Confidentiality	Software

NLFSR structure, besides to the hybrid construction that aims to benefit from the advantage of each building block structure. In addition to the difference in block size (16,32,64,... etc), key size (64,80,96,... etc), and the number of rounds (4,12,15,... etc). Moreover, some block ciphers are more dedicated for efficient software implementation such as SIMON, SPECK, PRIDE and LEA,... etc, others are more hardware implementation oriented such as Piccolo, GOAST, SEA, KATAN/KTANTAN,... etc.

Overall, none of existing lightweight block ciphers satisfies all the efficiency metrics of software and hardware requirements, and hence considerable effort

still be to devote in such direction for better lightweight cryptographic primitives in IoT security. We have pointed out the following research issues to take into consideration when designing new efficient lightweight block cipher :

1. Design of simple, fast and robust substitution-box (S-box) and permutation-box (P-box) that can attain desired confusion and diffusion degreed and make a good compromise between security, cost, and performance. Rather than using multiple S-boxes as the case in AES, reducing the number of the employed S-boxes is desired but without affecting the required security degrees. Furthermore, searching for efficient alternative confusion techniques that can successfully replace S-box primitive is still an open research problem
2. Design of a key schedule mechanism with less memory overhead and processing cost.
3. With the aim of preserving a good compromise between security, cost and performance, the designer of lightweight cryptographic primitives should employ smaller block length, less number of rounds and lighter round function

4 Conclusion

Embedded devices that are extensively employed in different platforms are featured by their own needs (e.g., reduced computing power, limited memory,..etc.), so that preserving data content by applying conventional cryptographic primitives is not the appropriate solution to face these limited constraints. To this end, lightweight cryptography has been emerged recently to deal with embedded devices' features. In this paper, a review of popular contemporary lightweight block ciphers has been covered, a comparative analysis in terms of some specific metrics has been discussed, and then a number of research issues have been pointed to take into consideration when dealing with the design of new efficient lightweight block ciphers.

References

1. Saci Medileh, Abdelkader Laouid, Reinhardt Euler, Ahcène Bounceur, Mohammad Hammoudeh, Muath AlShaikh, Amna Eleyan, Osama Ahmed Khashan, et al. A flexible encryption technique for the internet of things environment. *Ad Hoc Networks*, 106:102240, 2020.
2. Hassan N Noura, Ali Chehab, and Raphael Couturier. Efficient & secure cipher scheme with dynamic key-dependent mode of operation. *Signal processing: Image communication*, 78:448–464, 2019.
3. George Hatzivasilis, Konstantinos Fysarakis, Ioannis Papaefstathiou, and Charalampos Manifavas. A review of lightweight block ciphers. *Journal of cryptographic Engineering*, 8(2):141–184, 2018.
4. Vishal A Thakor, Mohammad Abdur Razzaque, and Muhammad RA Khandaker. Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and research opportunities. *IEEE Access*, 9:28177–28193, 2021.
5. Marcus Walshe, Gregory Epiphaniou, Haider Al-Khateeb, Mohammad Hammoudeh, Vasilios Katos, and Ali Dehghantanha. Non-interactive zero knowledge proofs for the authentication of iot devices in reduced connectivity environments. *Ad Hoc Networks*, 95:101988, 2019.
6. Kerry McKay, Lawrence Bassham, Meltem Sönmez Turan, and Nicky Mouha. Report on lightweight cryptography. Technical report, National Institute of Standards and Technology, 2016.
7. Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.
8. Martin R Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar, and Tolga Yalçın. Block ciphers–focus on the linear layer (feat. pride). In *Annual Cryptology Conference*, pages 57–76. Springer, 2014.
9. Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie proposal: Noekeon. In *First open NESSIE workshop*, pages 213–230, 2000.
10. François-Xavier Standaert, Gilles Piret, Gaël Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat. Iceberg: An involutional cipher efficient for block encryption in reconfigurable hardware. In *International Workshop on Fast Software Encryption*, pages 279–298. Springer, 2004.
11. Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *International workshop on cryptographic hardware and embedded systems*, pages 450–466. Springer, 2007.
12. Cheng Wang and Howard M Heys. An ultra compact block cipher for serialized architecture implementations. In *2009 Canadian Conference on Electrical and Computer Engineering*, pages 1085–1090. IEEE, 2009.
13. Chae Hoon Lim and Tymur Korkishko. mcrypton–a lightweight block cipher for security of low-cost rfid tags and sensors. In *International workshop on information security applications*, pages 243–258. Springer, 2005.
14. Lars Knudsen, Gregor Leander, Axel Poschmann, and Matthew JB Robshaw. Printcipher: a block cipher for ic-printing. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 16–32. Springer, 2010.
15. Huihui Yap, Khoongming Khoo, Axel Poschmann, and Matt Henricksen. Epcbc-a block cipher suitable for electronic product code encryption. In *International Conference on Cryptology and Network Security*, pages 76–97. Springer, 2011.

16. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The led block cipher. In *International workshop on cryptographic hardware and embedded systems*, pages 326–341. Springer, 2011.
17. Zheng Gong, Svetla Nikova, and Yee Wei Law. Klein: a new family of lightweight block ciphers. In *International workshop on radio frequency identification: security and privacy issues*, pages 1–18. Springer, 2011.
18. Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences*, 58(12):1–15, 2015.
19. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, et al. Prince–a low-latency block cipher for pervasive computing applications. In *International conference on the theory and application of cryptology and information security*, pages 208–225. Springer, 2012.
20. Gilles Piret, Thomas Roche, and Claude Carlet. Picaro–a block cipher allowing efficient higher-order side-channel resistance. In *International Conference on Applied Cryptography and Network Security*, pages 311–328. Springer, 2012.
21. Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The skinny family of block ciphers and its low-latency variant mantis. In *Annual International Cryptology Conference*, pages 123–153. Springer, 2016.
22. Chae Hoon Lim. A revised version of crypton: Crypton v1. 0. In *International Workshop on Fast Software Encryption*, pages 31–45. Springer, 1999.
23. Horst FEISTEL. Cryptography and computer privacy. In *Scientific american*, volume 228, pages 15–23, 1973.
24. François-Xavier Standaert, Gilles Piret, Neil Gershenfeld, and Jean-Jacques Quisquater. Sea: A scalable encryption algorithm for small embedded applications. In *International Conference on Smart Card Research and Advanced Applications*, pages 222–236. Springer, 2006.
25. Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-bit blockcipher clefia. In *International workshop on fast software encryption*, pages 181–195. Springer, 2007.
26. Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: an ultra-lightweight blockcipher. In *International workshop on cryptographic hardware and embedded systems*, pages 342–357. Springer, 2011.
27. Wenling Wu and Lei Zhang. Lblock: a lightweight block cipher. In *International conference on applied cryptography and network security*, pages 327–344. Springer, 2011.
28. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. Twine: A lightweight, versatile block cipher. In *ECRYPT workshop on lightweight cryptography*, volume 2011, 2011.
29. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck lightweight block ciphers. In *Proceedings of the 52nd Annual Design Automation Conference*, pages 1–6, 2015.
30. Maryam Izadi, Babak Sadeghiyan, Seyed Saeed Sadeghian, and Hossein Arabnezhad Khanooki. Mibs: a new lightweight block cipher. In *International Conference on Cryptology and Network Security*, pages 334–348. Springer, 2009.
31. Axel Poschmann, San Ling, and Huaxiong Wang. 256 bit standardized crypto for 650 ge–gost revisited. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 219–233. Springer, 2010.

32. Manoj Kumar, PAL Sk, and Anupama Panigrahi. Few: a lightweight block cipher. *Turkish Journal of Mathematics and Computer Science*, 11(2):58–73, 2014.
33. Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varıcı. Ls-designs: Bitslice encryption for efficient masked software implementations. In *International Workshop on fast software encryption*, pages 18–37. Springer, 2014.
34. Sufyan Salim Mahmood AlDabbagh, Imad Fakhri Taha Al Shaikhli, and Mohammad A Alahmad. Hisec: A new lightweight block cipher algorithm. In *Proceedings of the 7th International Conference on Security of Information and Networks*, pages 151–156, 2014.
35. Adnan Baysal and Sühap Şahin. Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors. In *Lightweight Cryptography for Security and Privacy*, pages 58–76. Springer, 2015.
36. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, et al. Hight: A new block cipher suitable for low-resource device. In *International workshop on cryptographic hardware and embedded systems*, pages 46–59. Springer, 2006.
37. Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Dong-Geon Lee. Lea: A 128-bit block cipher for fast encryption on common processors. In *international workshop on information security applications*, pages 3–27. Springer, 2013.
38. John Jacob. Best-1: a light weight block cipher. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 16(2):91–95, 2014.
39. Daniel Dinu, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Johann Großschädl, and Alex Biryukov. Design strategies for arx with provable bounds: Sparx and lax. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 484–513. Springer, 2016.
40. Sebastiaan Indesteege, Nathan Keller, Orr Dunkelman, Eli Biham, and Bart Preneel. A practical attack on keeloq. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–18. Springer, 2008.
41. Christophe De Cannière, Orr Dunkelman, and Miroslav Knežević. Katan and ktantan—a family of small and efficient hardware-oriented block ciphers. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 272–288. Springer, 2009.
42. Daniel Engels, Xinxin Fan, Guang Gong, Honggang Hu, and Eric M Smith. Hummingbird: ultra-lightweight cryptography for resource-constrained devices. In *International conference on financial cryptography and data security*, pages 3–18. Springer, 2010.
43. Daniel Engels, Markku-Juhani O Saarinen, Peter Schweitzer, and Eric M Smith. The hummingbird-2 lightweight authenticated encryption algorithm. In *International workshop on radio frequency identification: Security and privacy issues*, pages 19–31. Springer, 2011.
44. Bassam J Mohd, Thaier Hayajneh, and Athanasios V Vasilakos. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. *Journal of Network and Computer Applications*, 58:73–93, 2015.

High accuracy fall detection method based on image analysis deep learning Xception Model and accelerometer data

Brahim Achour [1], Idir Filali [1], Malika Belkadi [1] and Mourad Laghrouche[2]

[1] LARI Laboratory, University of Tizi Ouzou, Algeria
[2] LAMPA Laboratory, University of Tizi Ouzou, Algeria
Brahim Achour
brahimachour@live.fr
Idir Filali
idir.filali@ummto.dz
Malika Belkadi
belkadi_dz@yahoo.fr
Mourad Laghrouche
larouche_67@yahoo.fr

Abstract. Automatic recognition of a human fall allows for prompt medical assistance and injury prevention. However, current detection systems are limited by their high acquisition frequency (50 Hz and more) and low detection rates. To overcome these limitations, this study proposes a new fall detection algorithm based on 3D accelerometer data. Indeed, a novel data selection approach is proposed. This approach is designed using the "sleep-down/wake-up" method. In addition, we propose an advanced method for human fall detection. This method consists in using deep learning image classification techniques. The Xception model determines whether or not the fall was present in the acquired data. As a result, the proposed method reduces transmitted data by 87% and acquired data by 79%. The classification method achieves a 98.2% accuracy rate. The proposed algorithm performed better than those reported in other research studies while using a lower acquisition frequency (14 Hz). This result shows the efficiency of our approach.

Keywords: Artificial intelligence, Deep learning, Fall detection, Activity recognition, Behavior classification, Accelerometer.

1 Introduction

New information and communication technologies are increasingly deployed to improve the lives of individuals. Recently, we have observed an expanding use of smart systems in homes [1] [2], cities [3] [4] and farms [5] [6]. The recognition of human physical activity is essential in many fields. Indeed, medicine, well-being, sports, energy consumption, construction, behavioral analysis and many other fields are increasingly interested in the state of activity of an individual. In this context,

detecting human activities helps to better understand their behavior and requirements [7]. Therefore, several works have been conducted from this perspective [8] [9] [10]. The fall of humans is considered abnormal behavior. It is frequently appearing in the elderly. Instantaneous detection of the fall allows for rapid medical assistance and prevents injury degradation [11].

Fall detection systems have been developed in several studies [12] [13]. However, these studies are limited by the following: (1) in most cases, the methods used in the studies are trained using inertial data acquired at high frequencies (between 50 Hz and 200 Hz). These frequencies significantly reduce the deployment time of fall detection systems. (2) Most works use a data mining procedure that transforms the data and causes an alteration of potentially valuable data. (3) The learning algorithms used (SVM, RNN, etc.) focus on the value of each inertial axis (or fusion of axes) to predict the classes (fall or no fall). However, no comparison of the values of the various axes of the accelerometer is performed. (4) Most of the works require a protocol for installing data acquisition sensors on individuals (fixed location).

The aim of this paper is to develop a novel method for detecting falls in different participants using 3D accelerometer data. No protocol for the installation of the data collection sensor is specified. The individual can place the accelerometer in any pocket and in any direction. Moreover, a new data selection method is proposed to decrease the data collection frequency while maintaining the important accelerometer data. The "sleep down/wake up" approach is used to select the data. Moreover, a classifier system is provided. Indeed, the accelerometer data is plotted in graphical form. Then, the existence or absence of a fall for each plot is determined by an image processing algorithm. To perform this step, the deep learning model Xception [14] is used. This model exploits the spacing that exists among the different axes of the accelerometer during prediction.

The main contributions of this study are the following:

- A human fall detection technique is introduced using the Xception deep learning model.
- The approach is designed using the data from a 3D accelerometer. The Xception model uses the spacing among the accelerometer axes.
- This spacing cannot be exploited in other algorithms that use numerical series like RNN, LSTM, etc.
- A new data selection approach based on sleep-down/wake-up is used to reduce the acquisition frequency while keeping the accelerometer data significant, which significantly decreases the acquisition frequency of the system.
- High accuracy rates are obtained. These rates exceed those achieved in numerous studies [12] [13] [15].

The rest of the document is structured as follows: Section 2 describes the different phases of data processing and classification. Section 3 gives the results. Finally, Section 4 provides the conclusion and perspectives of this research.

2 Materials and methods

2.1 Data selection

The primary aim of this research is to develop a new algorithm that reduces the frequency of data acquisition while retaining useful data for classification. For this purpose, we propose an algorithm that allows variation in the acquisition frequency of an instant T depending on the accelerometer value acquired during a previous time (T-1).

After the sensor is turned on, a frequency of 5 Hz is defined. This frequency corresponds to collecting the accelerometer data every 200 milliseconds. The data is collected at two different times. One vector called last_acc_vect saves the data acquired the first time (from the accelerometer's three axes). A second vector called acc_vect saves the data acquired the second time. Then, the difference between the data of these two vectors is calculated. If this difference is significant (above a threshold called T1), the frequency of 5 Hz will be changed to 50 Hz. Otherwise, the algorithm keeps the 5 Hz frequency and continues its operation. When the frequency of 50 Hz is defined, the standard deviation of the data is calculated. If this deviation is significant (above a threshold called T2), the vector is saved, and the 50 Hz frequency is maintained. Otherwise, the vector is deleted, and a frequency of 5 Hz is defined. Fig. 1 shows the flowchart of the algorithm.

The values of the T1 and T2 thresholds used in this study are obtained using a Gaussian mixture model [16]. The T1 threshold is determined by a mixture of two normal laws using data from all accelerometer axes. Each axis allows for the definition of a threshold (T1_x, T1_y and T1_z). The sum of these thresholds allows defining the T1 threshold (T1_x + T1_y + T1_z). The T2 threshold was established using a mixture of two normal distributions on the standard deviation variable of the accelerometer axes (T2_std_x, T2_std_y and T2_std_z). The sum of the three thresholds allow for the establishment of the value of the T2 threshold.

2.2 Data collection

The data selection algorithm has been implemented in an ESP 32 microcontroller. The ESP is linked to an accelerometer module (MPU-9250 module). The device was placed on 30 people (in their pockets) who had to perform a series of behaviors: walking, going up and down stairs, running, doing sports movements, standing still, etc. The individuals were free to do any activity they wanted at any time. The only constraint is to fall forwards, backwards, or either the left or right side. Fig. 2-(1) shows the individual (ID 27) simulating a fall event. Fig. 2-(2) shows the individual (ID 15) playing sports. Fig. 2-(3) shows the data collection sensor.

The data obtained from the 30 individuals were divided into training and test data. Indeed, the data obtained from the first 20 individuals (ID0, ID1, ... , ID19) are used as training data. The data of 10 individuals (ID 20, ID21, ... , ID29) were used during the validation phase.

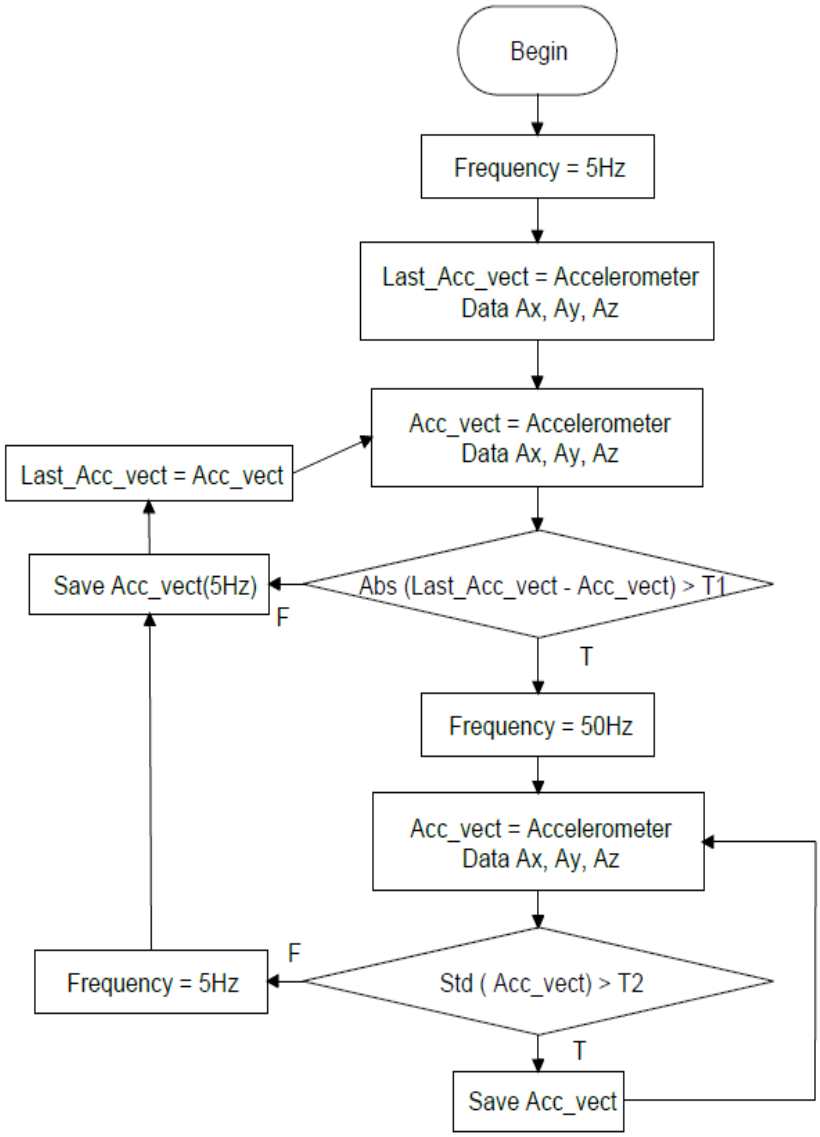


Fig. 1. The flowchart of the data collection method.

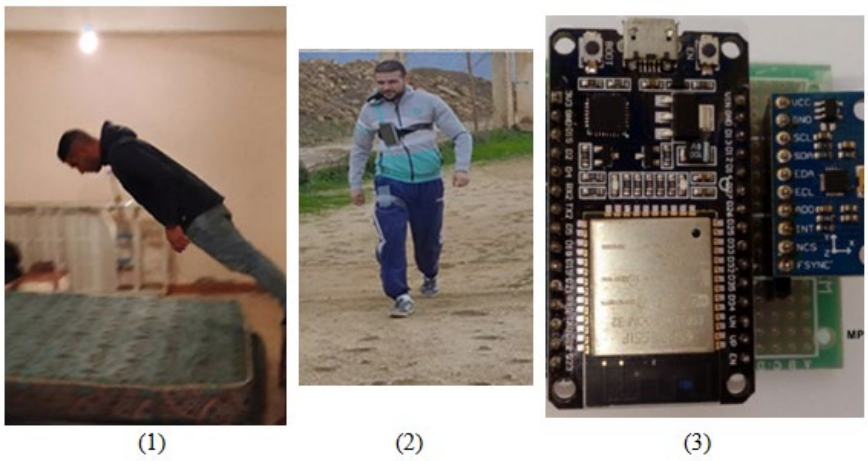


Fig. 2. Deployment of the data collection sensor

The acquisition step was mainly done at the multipurpose stadium of Ait Aggouacha, Tizi ouzou, Algeria and on the premises of the computer science department of UMMTO (Mouloud Mammeri University of Tizi-Ouzou) with the participation of 30 people aged between 18 and 55 years old belonging to different categories. Table 1 shows information about some participants. During data acquisition no bugs were encountered and the data was not affected by any noise source.

Table 1. Information about some participants

First name	Id	Age	Gender	Height (cm)	Weight (kg)
Omar	01	42	male	178	75
Chabane	03	26	Male	180	78
Rabah	05	27	Male	185	85
Rachid	06	54	Male	175	81
Nassim	08	20	Male	170	66
Karima	12	30	Female	165	62
Nabila	15	18	Female	172	71
Brahim	17	38	Male	191	102
Mélissa	22	32	Female	158	60
Sofiane	26	21	Male	188	87

2.3 Data classification

The collected data were displayed as image graphs with a segmentation window of 10 seconds. Each image was annotated into one of two classes: fall-image and other-activities-image. Fig. 3 presents a graph that includes the fall class. The colors of the accelerometer axes can be affected by the orientation of the sensor in my pocket. In this figure the blue represents the z axis, the green represents the x axis and the red represents the y axis of the accelerometer. During image partitioning, the images obtained from 20 individuals were used to train the classification model, and the data obtained from 10 other individuals were used as a validation data set.

In order to automatically classify the images and predict the class of each image, an image processing algorithm was used. This algorithm is based on convolution neural networks. This type of network has shown great efficiency in classifying and segmenting images. The variant of CNN used in this study is the Xception model [14]. This model comprises 36 convolution layers followed by a logistic regression layer. Fig. 4 shows the architecture of this algorithm. The Xception model has been chosen because it has shown very good results in several image classification studies. This algorithm is implemented in a centralized terminal. The role of the sensor is to acquire the data with the minimum of energy. Then, the Xception model classifies these images in a centralized terminal where we can use a power consuming model.

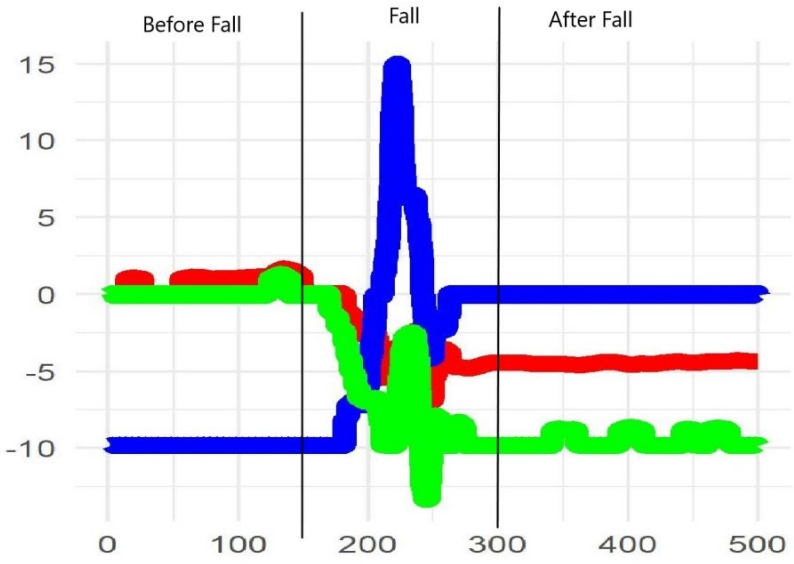


Fig. 3. The plot of the fall class.

The system's accuracy is calculated using equation 1 to validate the classification algorithm.

$$\text{Acc} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{true negative} + \text{false positive} + \text{false negative}} \quad (1)$$

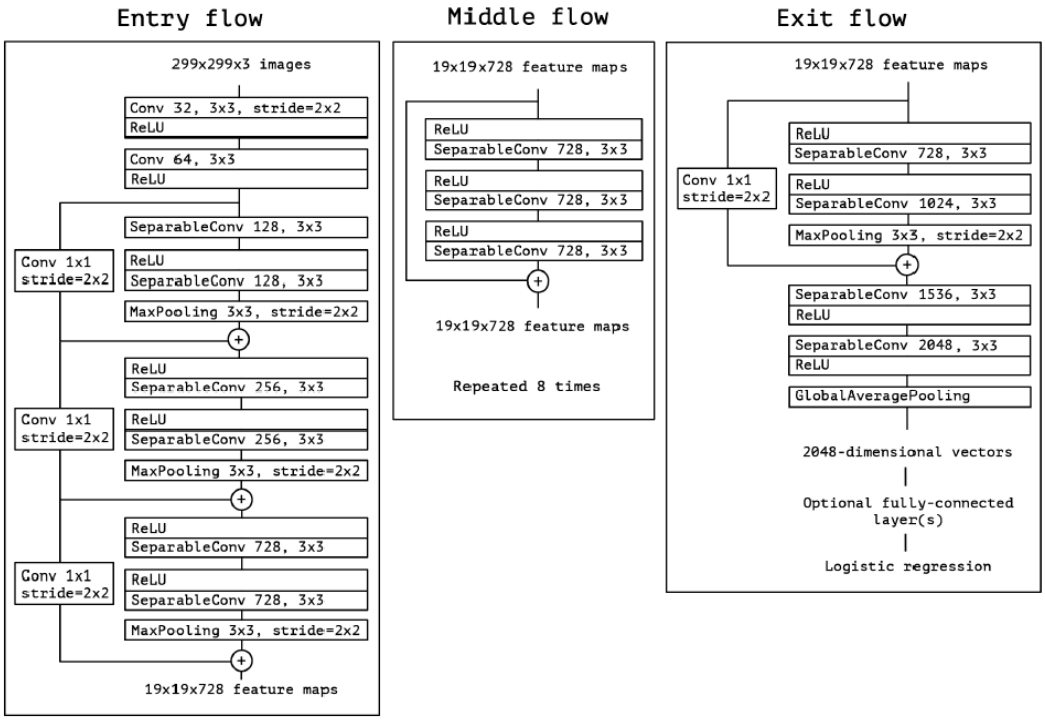


Fig. 4. The Xception architecture [14].

3 Results and discussion

The data collection strategy used in this research (Fig. 1) allowed for a 79% reduction in data collection and a significant reduction in data transmission (by 87%). The acquisition sensor reduced 79% of the data. This is because the acquisition frequency of 50 HZ was activated for 20% of the experiment's total duration. The frequency of 5 Hz was activated in the remaining time. On average the acquisition frequency was 14 Hz. The variance analysis of the acquired data allowed reducing the amount of transmitted data by 87%. This was done because wireless communication is very expensive in terms of energy. On average the acquisition frequency was 10 Hz. The value of the T1 and T2 thresholds greatly impacts these frequencies. Indeed, we chose the

lowest value for these two thresholds to ensure that the selection algorithm retains the integrity of the important accelerometer data. The change in the number of normal laws used when calculating the T1 and T2 thresholds caused the loss of some fall event in the dataset.

A total of 1245 images were constructed from the data of 30 individuals. During data acquisition, each person made 5 fall events. The total number of falls was 150. The falls of 20 people were used to train the model (100 fall events) and the falls of 10 people were used for testing (50 images). For the others activities, the test data consisted of 357 images. The training data consisted of 738 images. In all the dataset of the other activities consisted of 1095 images.

The classification accuracy of the approach is 98.2%. This accuracy shows the great interest in using the image classification approach to classify accelerometer data. The confusion matrix of the system is illustrated in Fig. 5. The training and testing accuracy was almost similar, this demonstrates that the proposed classification model does not exhibit overfitting.

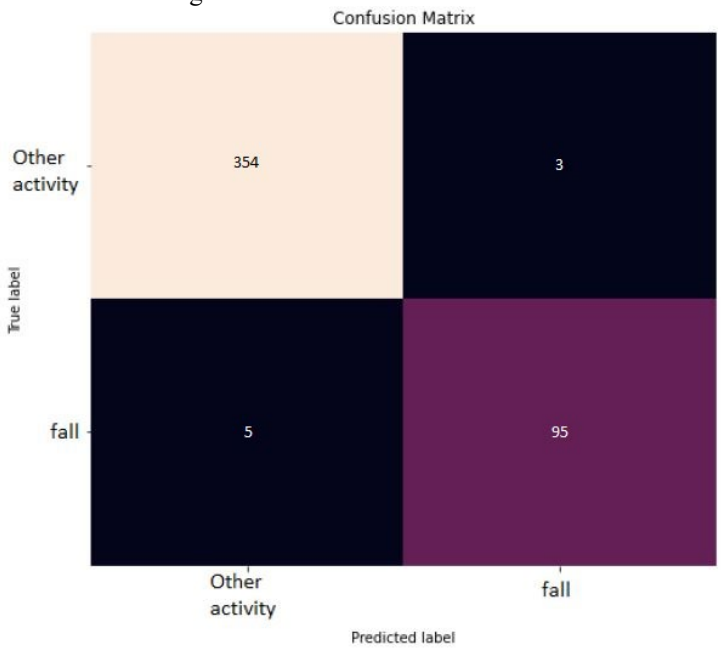


Fig. 5. The confusion matrix of the Xception model.

The results obtained in this study are significantly better than those obtained in several other studies. The accuracy of our approach is 3% better than [15], 13% better than [17], 21% better than [12], 1% better than [13]. Moreover, in this study, a very low frequency has been used. Fig. 6 allows comparing the results obtained and the frequency used.

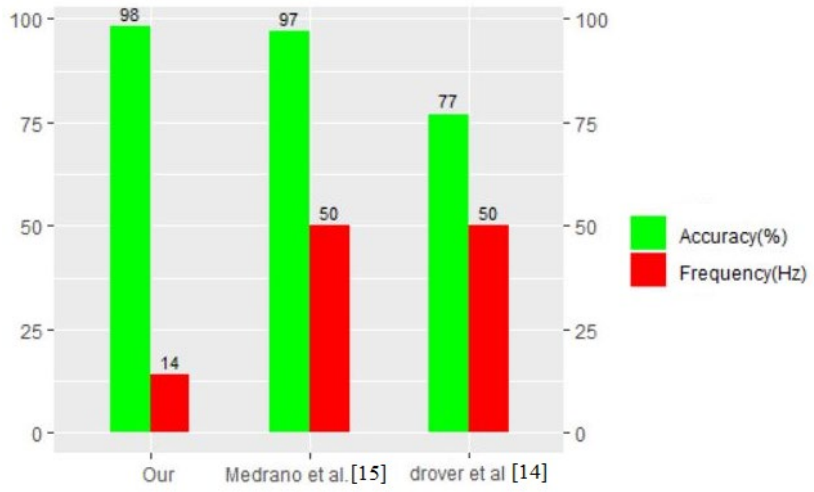


Fig. 6. Comparison between the current study and other research

The data used in this study are different from those used in other studies. This difference can be explained because the objective of the current study is to reduce the acquisition frequency of accelerometer data. For this purpose, we used a variable acquisition frequency much lower than the frequency applied in other researches. The use of high frequencies (50Hz and above) impacts the life of the system which must be constantly recharged. For example, it is impractical to insert a fast-discharging system directly into clothing.

4 Conclusion

Detecting a human fall can prevent injury and provide immediate medical attention. This study presents a completely novel method for detecting falls in 30 people. Indeed, a new technique for data selection allowing reducing the acquisition frequency from 50 Hz to an average of 10 Hz was designed and used. In addition, an Xception classifier was used to automatically recognize human falls. This classifier is a variant of the convolutional neural network that has proven to be very effective for image classification in several other domains. As a result, the approach achieved high accuracy (98.2%), demonstrating the proposed approach's high interest. In our future work, we plan to extend the capability of the system to detect other activities such as walking, running, etc.

Acknowledgement

We thank all the participants in the different experiments conducted during the data collection and the LARI laboratory members (Algeria) for their assistance.

References

1. Marikyan, D., Papagiannidis, S. and Alamanos, E., 2019. A systematic review of the smart home literature: A user perspective. Technological Forecasting and Social Change, 138, pp.139-154.
2. Boultache, T., Achour, B. and Laghrouche, M. (2022) 'Human activity detection from inertial data using RNN and LSTM network', Int. J. Sensor Networks, Vol. 39, No. 3, pp.156–161.
3. Ismagilova, E., Hughes, L., Dwivedi, Y. and Raman, K., 2019. Smart cities: Advances in research—An information systems perspective. International Journal of Information Management, 47, pp.88-100.
4. C. I. Ngabo and O. El Beqqali, "3D tilt sensing by using accelerometer-based wireless sensor networks: Real case study: Application in the smart cities," 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), 2018, pp. 1-8, doi: 10.1109/ISACV.2018.8354013.
5. Aoughlis, S., Saddaoui, R., Achour, B. and Laghrouche, M. (2021) 'Dairy cows' localisation and feeding behaviour monitoring using a combination of IMU and RFID network', Int. J. Sensor Networks, Vol. 37, No. 1, pp.23–35.
6. F. Tian, J. Wang, B. Xiong, L. Jiang, Z. Song and F. Li, "Real-Time Behavioral Recognition in Dairy Cows Based on Geomagnetism and Acceleration Information," in IEEE Access, vol. 9, pp. 109497-109509, 2021, doi: 10.1109/ACCESS.2021.3099212.
7. Biagetti, G., Crippa, P., Falaschetti, L., Luzzi, S., Turchetti, C. (2019). Recognition of Daily Human Activities Using Accelerometer and sEMG Signals. In: Czarnowski, I., Howlett, R., Jain, L. (eds) Intelligent Decision Technologies 2019., Smart Innovation, Systems and Technologies, vol 143. Springer, Singapore. https://doi.org/10.1007/978-981-13-8303-8_4.
8. Jeffin Gracewell, J., Pavalarajan, S. Fall detection based on posture classification for smart home environment. J Ambient Intell Human Comput 12, 3581–3588 (2021).
9. Kerdjidj, O., Ramzan, N., Ghanem, K. et al. Fall detection and human activity classification using wearable sensors and compressed sensing. J Ambient Intell Human Comput 11, 349–361 (2020).
10. Bet, P., Castro, P. and Ponti, M., 2019. Fall detection and fall risk assessment in older person using wearable sensors: A systematic review. International Journal of Medical Informatics, 130, p.103946.
11. G. Bergen, Falls and fall injuries among adults aged ≥ 65 years – United States, MMWR. Morbidity and Mortality Weekly Report 65, (2014).
12. Drover, D., Howcroft, J., Kofman, J. and Lemaire, E., 2017. Faller Classification in Older Adults Using Wearable Sensors Based on Turn and Straight-Walking Accelerometer-Based Features. Sensors, 17(6), p.1321.
13. Medrano, C., Igual, R., Plaza, I. and Castro, M., 2014. Detecting Falls as Novelties in Acceleration Patterns Acquired with Smartphones. PLoS ONE, 9(4), p.e94811.
14. F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.
15. Theodoridis, T.; Solachidis, V.; Vretos, N.; Daras, P. Human fall detection from acceleration measurements using a Recurrent Neural Network. In Precision Medicine Powered by pHealth and Connected Health; Springer: Berlin, Germany, 2018; pp. 145–149
16. McLachlan, G., Peel, D., 2000. Finite Mixture Models. John Wiley & Sons Publishers, NewYork.

17. Santos GL, Endo PT, Monteiro KHdC, Rocha EdS, Silva I, Lynn T. Accelerometer-Based Human Fall Detection Using Convolutional Neural Networks. Sensors. 2019; 19(7):1644. https://doi.org/10.3390/s19071644

Artificial Immune Systems for Software Change Prone Prediction

Kamilia MENGHOUR[0000-0002-8522-8199], Amina BOUDJEDIR[0000-0003-1917-6674],
Boubaker NASRI, Labiba SOUICI-MESLATI[0000-0003-1022-9995]

LISCO Laboratory, Computer Science Department, Badji Mokhtar-Annaba University, Annaba, Algeria

K-menghour@yahoo.fr, a.boudjedir@hotmail.fr, souici_labiba@yahoo.fr

Abstract. Software projects change frequently during the whole cycle life. In fact, the bugs, the defaults, the change of user requirements and the improvement of product quality are the main reasons of software changing. Software change prone prediction aims to predict the parts or classes of source code which are most susceptible to change in the future. This prediction allows developers to focus on reducing maintenance cost, time and managers to allocate resources more efficiently. Several machine learning techniques have been proposed to predict the software change prone using quality metrics as dependents variables. In this area, bio-inspired methods for classification have attracted the attention of researchers. In this work, we propose a prediction model based on artificial immune systems to predict the classes which are susceptible to change in object-oriented software. We have implemented five of the most popular variants of the artificial immune systems which are: AIRS1, AIRS2, CLONALG, CSCA and IMMUNOS99. In order to evaluate our models, we have constructed a dataset using two versions of the « Apache Commons Pool » library. We have also compared the performances achieved by our models with classifiers based on standard machine learning methods like naïve bayes, Decision tree and multilayer perceptron. The experimental results proved that the artificial immune systems predict effectively change prone classes in object-oriented software. Furthermore, the results showed that the prediction performances of the immunological models are statistically better than those based on standard machine learning techniques.

Keywords: Software Change Prone, Artificial Immune System, Machine Learning, Prediction, Empirical Software Engineering.

1 Introduction

Continuous improvement is one of the basic principles of development, production and offer of services throughout the world. It is even one of the basic requirements of the standards ISO 9001 series. The field of software development also follows this approach throughout its life cycle under certain attributes such as maintainability,

default detection, etc. These attributes can be predicted with several software quality metrics which subsequently offers possibilities of improvement by meeting user requirements and achieving customer satisfaction.

The early identification of the main change prone classes of software helps developers to limit the risks (appearance of new defects, decrease in performance, etc.) and leads to a lower development cost for the final product of quality. Software change prone prediction is the procedure that aims to develop a prediction model which is able to automatically predict the entities of the software those are susceptible to be changed in the future versions of software.

The new directions in the field of scientific research in software engineering tend to develop good quality indicators and prediction models using machine learning techniques [1-3]. Machine learning is a subfield of artificial intelligence, it consists of automatically solving complex problems such as pattern recognition, classification and optimization. The research trend in this field has been oriented in recent years towards the use of bio-inspired methods such as: artificial neural networks, genetic algorithms, swarm intelligence and artificial immune systems. These methods are widely used in various fields and they have proved their effectiveness in solving problems and improving model's performances [4].

In this paper, we propose a prediction model based on the artificial immune systems (AIS) to predict the change prone classes in software. AIS are computational systems inspired by theoretical immunology and observed immune functions. Many AIS algorithms are proposed and applied in pattern recognition and classification domains. In the literature, there is a significant paucity of works that have used artificial immune systems for prediction of changes in software. In reality, the only work found in this area is that of Malhotra and Khanna [5].

The remainder of this paper is organized as follows. Section 2 introduces some related works. Section 3 describes research background. In section 4, we present the adopted research methodology including data collection, prediction model construction and performances evaluation. We discuss, in section 5, the obtained results for our immunological models and compare their performances with the classical machine learning models. Finally, section 6 concludes this paper.

2 Related works

Many studies have proposed prediction models based on machine learning techniques. In [1], the authors use decision tree algorithm as classifier in a hybrid model which combines features such as execution time, trace events and behavioral dependency, generated from source code to predict the change-proneness of classes. In [2], the authors propose a prediction model based on ensemble voting classifier. They have used different fitness functions to build a set of five particle swarm optimization classifiers. The results of these classifiers are combined by voting to predict the change prone classes in each one of the five popular Android application packages used in their experiments.

In [3] the authors proposed an ensemble learning approach based on several machine learning algorithms and deep learning classifiers to predict the future change software entities. This approach has been tested on various applications of android framework.
Other works paid attention to study the relationship between object oriented (OO) metrics and the change proneness of classes in software. In this context, Kumer et al [6] used 62 metrics belonging to four metrics categories in order to develop their prediction model and they have applied five feature selection methods to select the most significant ones. The authors proved that the performances of the developed models using coupling metrics are better compared with other metrics categories like cohesion, size and inheritance metrics. In [7], authors use 11 feature selection techniques to select the most suitable metrics over 21 object oriented metrics for software change prone prediction. Martins et al [8] perform an empirical study of 8 machine learning methods using 8 OO metrics and 21 types of code smells to build class change proneness prediction models. In [9] the authors predict the change prone classes in software using a combination of 11 feature reduction methods with 3 classifiers. The employed techniques are testes on six java datasets contained 60 OO and 26 graph-based metrics.

3 Background

In this section, we present the different metrics chosen in this work. In addition, we present the artificial immune systems used to predict the change prone classes in software.

3.1 Dependent and independent Variables

The need to identify the characteristics of software which affect complexity has encouraged research and development of software metrics. The most dominant metrics for construction of prediction models are object-oriented metrics because these metrics are the most effectively during the whole process of software maintainability [6-9]. In this section, we present the depended and the independent variables used in this work.

Independent variables. In the literature, many OO metrics are developed to ensure the quality of software such as Chidamber and Kemerer (CK), Lorenz and Kidd, Li and Henry, Hitz & Montazeri, MARTIN, Kim and Ching, Tang et al, MOOD, QMOOD and more others metrics[10]. These metrics are used to characterize various software proprieties like: Complexity, inheritance, coupling, cohesion and others. Most of these metrics have been confirmed empirically to be useful for software change prone prediction. Table 1 resumes OO metrics used in this work which are grouped by categories. They represent the input variables of the prediction models.

In additional to these metrics of table 1, the McCabe's Cyclomatic Complexity (CC) metric is used in this work [11].This metric is equal to number of different paths in a method (function) plus one. The cyclomatic complexity is counts as: $CC = E – N + P$. where: E is the number of edges, N is the number of nodes (block of code in the method) and P is the number of connected components in the method control graph.

Table 1. Summary of the object oriented metrics used in this work.

Categories	Metrics	Source	Definition
Complexity metric	WMC :Weighted Methods per Class	C&K[12]	It counts the sum of static complexity of all local methods in a class.
	AMC :Average Method Complexity	TANG [13]	It is calculated as the average of methods size for each class.
inheritance Metrics	DIT :Depth of Inheritance Tree	C&K[12]	It calculates the length of the longest path, from the class to the root of the inheritance tree. DIT is a measure of how many ancestor classes can potentially affect a class.
	NOC (Number of Children)	C&K[12]	It counts number of immediate subclasses subordinated to a class in the class hierarchy. i.e. the number of children in the inheritance hierarchy.
Coupling Metrics	CBO (Coupling Between Object)	C&K[12]	Number of classes to which the considered class is coupled. Two classes are coupled, if the methods of one use methods or instance variables of the other.
	RFC (Response For a Class)	C&K[12]	It counts the number of all methods of the class and the number of methods called by the methods of the class.
	Ca (Afferent Couplings)	Martin [14]	The number of classes outside this category that depend upon classes within this category. A category is a set of classes that belong together in the sense that they achieve some common goal.
	Ce (Efferent Couplings)	Martin [14]	The number of classes inside this category that depend upon classes outside this category.
	IC (Inheritance Coupling)	TANG [13]	It is counted as the number of parent classes to which a given class is coupled.
	CBM (Coupling Between Methods)	TANG [13]	It counts the total number of new/redefined methods to which all the inherited methods are coupled.
Cohesion Metrics	LCOM(Lack of Cohesion of Methods)	C&K[12]	It is calculated by counting the number of disjoint sets formed by the intersection of the sets of instance variables used by methods of the class.
	LCOM3 (Lack of Cohesion Among Methods of a Class)	Hitz & Montazeri [15]	Consider an undirected graph G, where the vertices are the methods of a class, and there is an edge between two vertices if the corresponding methods use at least an attribute in common. LCOM3 is then defined as the number of connected components of G.
	CAM (Cohesion Among Methods of Class)	QMOOD [16]	This metric computes the relationship among methods of a class. It counts the sum of intersection of the method parameters and the max independent set of all the types of the method parameters.
Size	NPM (Number of Public Methods)	QMOOD [16]	It is calculated by counting the number of public methods in a given class.
Others	DAM (Data Access Metric)	QMOOD [16]	It is the ratio of the number of private (or protected) attributes to the total number of class attributes of the class.
	MOA (Measure Of Aggression)	QMOOD [16]	It counts the number of user-defined data declared insaid the class.
	MFA (Method of Functional Abstraction)	QMOOD [16]	It is counted as the ratio of inherited methods in the class, which are accessible by to the total number of their methods members.

Dependent variables. In this work, the change proneness of the classes is used as a dependent variable. In the case that there is a change in the source code of the class, it belongs to "changed" class, otherwise it belongs to "unchanged" class.

3.2 Artificial immune systems

The biological immune system is a very complex system that uses many mechanisms for defending the human body organisms from foreign pathogens. It is able to organize all cells (or molecules) within the body to classify them as self-cells or non-self cells.

Artificial immune system (AIS) has emerged as a computational intelligence (CI) technique. AIS inspired some of the aspects of a Nature Inspired System (NIS) in order to solve problems. It has been defined by De Castro and Timmis in 2002 [17] as:"adaptive systems, inspired by theoretical immunology and observed immune functions, principle and models, which are applied to problem solving". AIS have been mainly applied in different areas such as clustering/ classification, anomaly detection, computer security, numeric function optimization, learning and more others areas[18;19].

In this section, we briefly present the most population AIS algorithms which are proposed and widely used for pattern recognition and classification systems.

AIRS1 Algorithm. AIRS is one of the first artificial immune system techniques for classification, it is proposed by Watkins and Boggess in 2002[20; 21]. This algorithm based on the paradigm of sources limited artificial immune system.

AIRS algorithm tried to build pool of recognition (memory) cells which are excellent representative of the data training for unseen data classification. The training process starts with an initialization step which occurs one time. This step is a combination of pre-preprocessing (normalization) and parameters discover (affinity threshold definition) stages. The three others steps involved in the AIRS algorithm represent a loop which is performed for each antigen in the training set. The first is the memory cells identification and the ARB generation (Artificial Recognition Ball). ARBs represent a set of identical B cells, where the number of cells B is the number of the ARB limited resources. The next step is competition for limited resources and the development of a candidate memory cell by selecting the ARB with the greatest stimulation level. The final step in this process is the introduction of the candidate memory cell in the memory cell pool. When the training process is completed, the memory cell pool is available to be used in the classification where the K-nearest neighbor approach is performed at this step.

AIRS2 Algorithm. The AIRS2 algorithm is a revisited and affined version of the AIRS1 algorithm [22]. It was developed to be smaller and simpler but more effective. AIRS2 is generally very similar to the first version AIRS1 despite significant differences that have been introduced in this version [23]. The main differences are that AIRS1 uses the ARB pool as a permanent source while it is used as temporary in the AIRS2. AIRS1 enabled the mutation class process, so classes of clones may be changed whereas this process is not allowed to change in AIRS2. Another important difference is that AIRS2 uses somatic hyper mutation concepts while the AIRS1 uses

a user-definition mutate rate parameter to determinate the produced clone mutation degree.

IMMUNOS99 Algorithm. IMMUNOS81 is the first classification algorithm that uses immune system metaphor, it is proposed by Carter in 2000 [24]. It was created using abstractions of T cells, B cells, antibodies, and their interactions [23]. In 2005, Brownlee [25] improved this classifier by integrating elements from other AIS algorithm inspired by the clonal selection theory. The obtained version was named IMMUNOS99. The original IMMUNOS99 algorithm starts by devising data into antigen groups according to classification labels. For each group, the algorithm creates an initial B-cell population which will be exposed to the antigens from all groups in order to calculate the fitness scoring and population purning. Afterwards, some antigens of the same groups will be selected randomly and inserted. After several generations, the final pruning for each B-cell population will be performed and the final B-cell population will be returned as the classifier.

CLONALG Algorithm. CLONALG (CLONal selection ALGorithm) [26] (called CSA) is an AIS inspired by clonal selection theory of immune systems. The algorithm offers two search mechanisms (local and global search) in order to produce a population of antibodies and antigens. The antibody represents a single solution of the problem whereas the antigen represents an evaluation to the problem space [27]. In a local search, the better matched antibodies are selected to product clones using the affinity maturation process. While in the global search, randomly generated antibodies are inserted into the population to augment the variety and offer a manner to avoid the local optima in the future.

CSCA Algorithm. CSCA is a variant of CLONALG proposed by Brownlee in 2005 [27]. The object of this variant is to maximize the accuracy of the correct classified instances and minimize the number of incorrect classified instances. The algorithm performed for many generations and during each generation, the full set of antibodies is presented to all antigens in order to maximize prediction accuracy.

4 Research methodology

The goal of this work is to propose an approach based on artificial immune systems for predicting the susceptibility of classes to changes in software. The proposed approach consists of three critical steps:

- Build a database using the collected data from two versions of given software;
- Propose a prediction model based on artificial immune systems;
- Evaluate the results obtained by AIS variants and compare them with those obtained using standards machine learning techniques.

4.1 Data collection

In our experiments, we have used the java open-source software library “Apache Commons Pool”. All the releases are available from the apache website [28].

For database construction (fig.1), we have collected the data from the two versions of the chosen software: "2.4.3" and "2.5.0". The constructed dataset contains 70 classes (samples) with 19 "changed" classes.

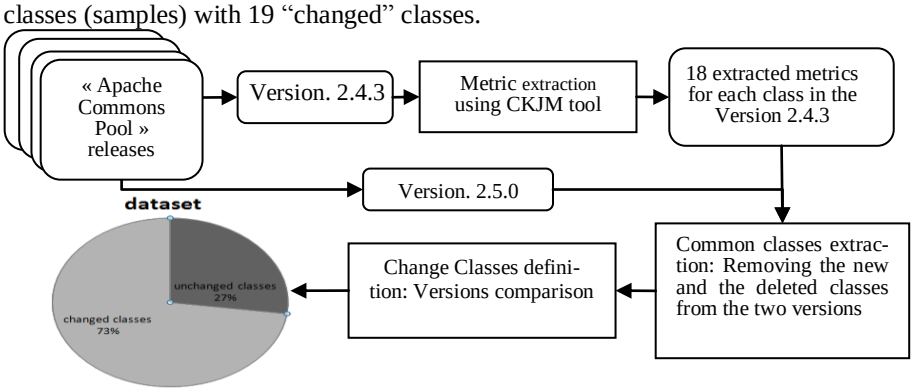


Fig.1. Data extraction approach.

The construction procedure starts by removing the new and the deleted classes from the latest version. So we use only the common classes between both versions of the software. Afterwards, a set of object oriented quality metrics, which serve as input variables to the models, will be extracted.

The extended version of CKJM (Tool for Calculating Chidamber and Kemerer Java Metrics) [29] is used for the extraction of metrics. CKJM_extended is able to calculate the following 18 OO metrics for each class of the given software: WMC, DIT, NOC, CBO, RFC, LCOM, Ca, Ce, NPM, LCOM3, DAM, MOA, MFA, CAM, IC, CBM, AMC and CC. These metrics belong to one of the following four categories: coupling, cohesion, inheritance, and complexity.

After extracting the metrics, the last step in the data building procedure is to define the labels of the classes (over all the selected classes which one are changed or unchanged). We have manually defined the sample's labels by comparing the source code files to determine the change between the classes of the both versions. In the case of the modification in the source code of the class, it belongs to class "changed", otherwise it belongs to class "unchanged".

4.2 Construction of the prediction model

Our motivation for this study comes from the fact that there is a significant lack of work that has explored the use of the immune system for the prediction of changes in software. In fact, the only work found is that of Malhotra and Khanna [5], in which the authors use three AIS variants to predict the change prone of classes in software.

In this paper, we have used the five AIS algorithms: AIRS1, AIRS2, IMMUNOS99, CLONALG and CSCA for change prone prediction. Those algorithms are implemented in WEKA Tool [30]. The default parameters of each algorithm in the WEKA tool are presented in table 2.

Table 2: Default parameters setting of the five AIS algorithms in WEKA.

Algorithms	Parameters
AIRS1	AffinityThresholdScalar ATS:0.2, ArbinitialPoolSize :1, ClonalRate:10, HypermutationRate: 2,NumInstancesAffinityThreshold:-1, Knn:3, MutationRate:0.1, Seed:1, StimulationValue: 0.9, TotalResources:150, MemInitialPoolSize: 1
AIRS2	AffinityThresholdScalar: 0.2, MemInitialPoolSize:1, ClonalRate: 10, NumInstancesAffinityThreshold: -1, Seed: 1, HypermutationRate: 2, StimulationValue: 0.9, Knn: 3, TotalResources: 150
IMMUNOS99	MinimumFitnessThreshold: -1,SeedPopulationPercentage: 0.2, , Seed: 1, TotalGenerations: 1
CLONALG	AnticorpPool_Size: 30, ClonalFactor: 0.1, RemainderPoolRatio: 0.1, Seed: 1, SelectionPoolSize: 20, NumGenerations: 10, TotalReplacement:0
CSCA	KNN: 1, MinimumFitnessThreshold: 1, ClonalScaleFactor: 1, NumPartitions: 1, Seed: 1, InitialPopulationSize:50, TotalGenerations: 5

4.3 Performance evaluation criteria

In general, four parameters are used to evaluate performances of the classification methods. True Positive (TP) is the cases having positive class label which were classified as positive. False Positive (FP) is the cases having negative class label which were classified as positive. True Negative (TN) is the cases having negative class label which were classified as negative. And False Negative (FN) is the cases having negative class label which were classified as positive.

Evidently, the best method is that offers maximum of TP and TN and minimum of FP and FN. The most used metrics for evaluating prediction models are based on these parameters, like accuracy, recall, precision and F_measure [31]. These metrics are based on the number of the right and false classifications that the algorithm makes. In this work, we have used those four metrics for evaluating the proposed change prone prediction models. The used metrics are presented in table 3.

Table 3. Evaluation criteria

Evaluation criteria	Definition	Formula
Accuracy	Refers to the percent of the correct classifications in the evaluation set .	$\frac{TP+TN}{N} * 100$
Precision	Refers to the portion of actual positives that was correctly classified as positive.	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}}$
Recall	Refers to the portion samples classified as positive which are true positives.	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{TN}}$
F-measure	Defined as the harmonic mean of precision and recall.	$\frac{2(\mathrm{precision})(\mathrm{recall})}{\mathrm{precision}+\mathrm{recall}}$

5 Results and discussion

In this work, we carried out a series of experiments by varying the AIS algorithms and the values of the most significant parameters which have a great influence on their performances. Five immunological algorithms are used (AIRS1, AIRS2,

CLONAG, CSCA and IMMUNOS99). Figures 2-6 present variation of the prediction accuracy in term of one of the most significant parameters and which have a great influence on the performances of the algorithm.

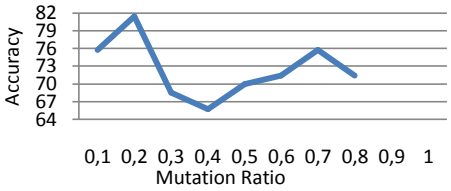


Fig. 2. Learning accuracy vs mutation ratio for AIRS1 algorithm

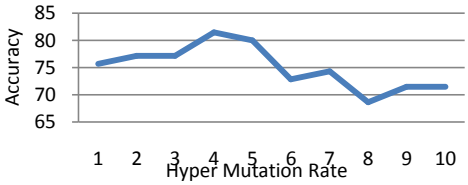


Fig. 3. Learning accuracy vs Hyper mutation rate for AIRS2 algorithm

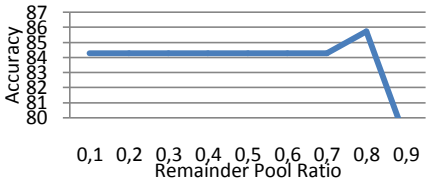


Fig. 4. Learning accuracy vs remainder pool ration for CLONALG algorithm

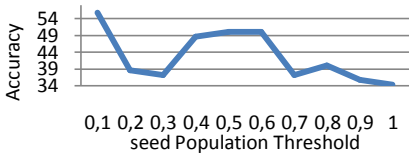


Fig. 5. Learning accuracy vs seed population threshold for IMMUNOS99

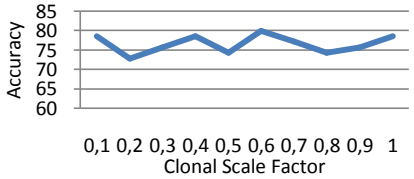


Fig. 6. Learning accuracy vs clonal scale factor for CSCA algorithm

Table 4 presents the best parameters setting obtained for the five AIS based algorithms used in this work. These parameters are experimentally adjusted.

Table 4: Setting parameters for the five AIS algorithms**.**

Algorithms	Parameters
AIRS1	Mutation_Rate:0.2, Clonal_Rate:5, hyper_mutation-rate: 0.4
AIRS2	Clonal_Rate: 9, Hyper_mutation_Rate: 4,
IMMUNOS99	Seed_Population_Percentage: 0.1, Seed: 10
CLONALG	Anticorp_Pool_Size: 500, remainder_Pool_Ratio: 0.8
CSCA	Clonal_Scale_Factor: 0.6, initialPopulationSize: 30

Table 5 presents a summary of our experimentation undergone in the AIRS1, AIS2, CLONAG, CSCA and IMMUNOS99 algorithms. In this table, we have explained the results obtained using default parameters of WEKA (presented in table 2) and the optimized parameters which are presented in table 3. Correct classification rate (accuracy), precision, recall and f-measure metrics are calculated to assess learning performance. All the proposed models are evaluated using K- cross-validations method. We have chosen K= 10 (the default value defined in weka tool).

Table 5. Performances achieved by different AIS algorithms.

Algorithms	Parameters	Accuracy	Precision	Recall	f-measure
AIRS1	Default configuration	68.571	0.696	0.686	0.69
	Best configuration	81.428	0.807	0.814	0.799
AIRS2	Default configuration	70	0.676	0.7	0.684
	Best configuration	81.428	0.805	0.814	0.805
CLONAG	Default configuration	68.571	0.686	0.686	0.686
	Best configuration	85.714	0.857	0.857	0.857
CSCA	Default configuration	77.143	0.761	0.771	0.731
	Best configuration	80	0.811	0.8	0.764
IMMUNOS99	Default configuration	40	0.634	0.4	0.392
	Best configuration	55.714	0.654	0.557	0.582

From the table 5, we can see that among the five developed models based on the AIS, four models have achieved good results. It can be observed that the CLONAG algorithm, using setting parameters, has shown the best performances in term of accuracy, precision, recall and f-measure with values of 85.714%, 85.7%, 85%7 and 85%7 respectively. The IMMUNOS99 model achieved the worst performance with 55% in term of accuracy.

In order to better evaluate AIRS based models, we carried out others experiments using well known machine learning algorithms which prove their efficacy in the field of classification and pattern recognition. Table 6 presents the results obtained by five software change prone prediction models developed using: Naïve bayes; Support vector machine (SVM), multilayer perceptron (MLP), decision tree (we have used to methods: J48 and Random forest) and K-nearest-neighbored (KNN). These algorithms are implemented in Weka Tool using the default parameters setting.

Table 6. Performances achieved by the classical machine learning algorithms.

Algorithms	Accuracy	Precision	Recall	f-measure
Naïve bayes	75.714	0.736	0.757	0.738
SVM	77.142	0.778	0.771	0.719
MLP	75.714	0.741	0.757	0.744
J48	77.143	0.759	0.771	0.762
Random Forest	78.571	0.773	0.786	0.761
KNN (with k=1)	74.285	0.735	0.743	0.738

From table 6 and comparing performance of classical machine learning methods, with those of our models, we can see that four based AIS models have accuracy value greater than 80%, followed by Random Forest classifier with accuracy value equal to78%. The others classical classifiers achieved an average of 76% in term of accuracy.

6 Conclusion

In this work, we have proposed prediction model based on artificial immune systems to predict the change proneness of classes in software. After the collecting of OO metrics and change data from two versions of the Apache Commons Pool software,

we have used five well known immunological classifiers to build our prediction models. These models were evaluated using four evaluation criteria. Finally, in order to prove the effectiveness of our methods, we have compared the obtained results with the performances of six classical machine learning methods.

The results obtained in this experimental study are promising and confirm the interest of using bio-inspired immunological approaches for the prediction of classes change prone in software. Immunological classifiers outperform classical machine learning models.

In future, a study must be conducted on the influence of other variants of the artificial immune systems and other bio-inspired classification methods on the change prediction in software. We would like to experiment these models with many datasets.

References

1. Godara, D., Singh, R.K.: A New Hybrid Model for Predicting Change Prone Class in Object Oriented Software. International Journal of Computer Science and Telecommunications 5(7).1-6. (2014).
2. Malhotra, R., Khanna, M. : Software Change Prediction using Voting Particle Swarm Optimization based Ensemble Classifier. In: Proceedings of The Genetic and Evolutionary Computation conference, Berlin, (GECCO'17), pp. 311-312, (2017).
3. Khanna, N., & Agarwal, O. Software Change Prediction using Ensemble Learning on Object Oriented Metrics. In 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS) pp. 1369-1373. IEEE. (2022, May).
4. Fan, X., Sayers, W., Zhang, S., Han, Z., Ren, L., & Chizari, H.: Review and Classification of Bio-inspired Algorithms and Their Applications. Journal of Bionic Engineering, 17(3), 611–631. (2020).
5. Malhotra, R., Khanna, M.: Analyzing software change in open source projects using Artificial Immune System algorithms. In 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). (2014).
6. Kumar, L., Rath, S. K., & Sureka, A.: Empirical Analysis on Effectiveness of Source Code Metrics for Predicting Change-Proneness. In: Proceedings of the 10th Innovations in Software Engineering Conference on - ISEC '17. pp. 4-14. (2017).
7. Kumar, L., Lal, S., Goyal, A., Murthy. N. L. B.: Change-Proneness of Object-Oriented Software Using Combination of Feature Selection Techniques and Ensemble Learning Techniques. In :12th Innovations in Software Engineering Conference (formerly known as India Software Engineering Conference) (ISEC'19). pp. 1-11. (2019).
8. Martins, A., Melo, C., Monteiro, J. Machado, J: Empirical Study about Class Change Proneness Prediction using Software Metrics and Code Smells. In: Proceedings of the 22nd Int. Con. on Enterprise Information Systems (ICEIS 2020) - V 1, pp 140-147 (2020).
9. Malhotra, R., Kapoor, R., Aggarwal, D., & Garg, P. Comparative study of feature reduction techniques in software change prediction. In 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR) (pp. 18-28). IEEE. (2021, May).
10. Goel, B. M., Bhatia P. K.: An Overview of Various Object Oriented Metrics. International Journal of Information Technology & Systems. 2(1). 18-27. (2013).
11. McCABE T. J.: A complexity measure. IEEE Trans. on Software Engineering, 2(4), 308-320. (1976).

12. Chidamber, S. R., Kemerer, C. F.: Towards a metrics suite for object oriented design. In : Proc. Of the 6th ACM Conference on Object-Oriented Programming, Systems Languages, and Applications, OOPSLA '91, (1991).
13. TANG M-H., KAO M-H.,CHEN M-H.: An Empirical Study on Object-Oriented Metrics, in: Proc. of The Software Metrics Symposium, 242-249. (1999).
14. MARTIN R.: OO Design Quality Metrics - An Analysis of Dependencies, In: Proc. of Workshop Pragmatic and Theoretical Directions in Object-Oriented Software Metrics, OOPSLA'94, (1994).
15. Hitz, M., Montazeri, B.: Measuring Coupling and Cohesion in Object-Oriented Systems, in: Proc. Int. Symposium on Applied Corporate Computing, Monterrey, Mexico, (1995).
16. Bansiya,J., Davis, C. G.: A hierarchical model for object-oriented design quality assessment, IEEE Trans. Software Engineering, 28(1),4-17, (2002).
17. De Castro, LN., Timmis J.: Artificial immune systems: a new computational intelligence approach. Springer, New York, (2002).
18. Hart, E., Timmis, J.: Application Areas of AIS: The Past, The Present and The Future, Springer, ICARIS 2005, LNCS 3627, pp.483–497, (2005).
19. Hart, E., Timmis, J.: Application areas of AIS: The past, the present and the future, Applied Soft Computing 8,191–201, (2008).
20. Watkins, A. B., Boggess, L. C.: A New Classifier Based on Resource Limited Artificial Immune Systems, In: Proce. of Congress on Evolutionary Computation, Part of the 2002 IEEE World Congress on Computational Intelligence held in Honolulu, HI, USA, IEEE, University of Michigan, pp. 1546-1551, (2002).
21. Watkins, A. B., Boggess, L. C.: A Resource Limited Artificial Immune Classifier. In: Proceedings of Congress on Evolutionary Computation, Part of the 2002 IEEE World Congress on Computational Intelligence held in Honolulu, HI, USA, pp. 926-931.(2002)
22. Watkins,A. B., Timmis, J.: Artificial Immune Recognition System (AIRS): Revisions and Refinements, in: 1st International Conference on Artificial Immune Systems (ICARIS2002), University of Kent at Canterbury, pp. 173-181, (2002).
23. Brownlee, J.: Artificial immune recognition system (AIRS) a review and anal-ysis. Technical report No. 1-02. Centre for Intelligent Systems and Complex Processes (CISCP), (ICT), Swinburne University of Technology, (2005)
24. Carter, J., H.: The immune system as a model for classification and pattern recognition, Journal of the American Informatics Association, vol. 7, (2000).
25. Brownlee, J.: Immunos-81: The Misunderstood Artificial Immune System, Technical Report, No. 3-01, Centre for Intelligent Systems and Complex Processes (CISCP), (ICT), Swinburne University of Technology, (2005).
26. De Castro, L.N., Zuben, J.V.: Learning and optimization using clonal selection principle, IEEE Trans. Evol. Comput. Spec. Issue Artif. Immune Syst. 6 (3), 239–251, (2002).
27. Brownlee, J.: Clonal Selection Theory & Clonalg, The Clonal Selection Classification Algorithm (CSCA), Technical Report, No. 2-02 Centre for Intelligent Systems and Complex Processes (CISCP), (ICT), Swinburne University of Technology, (2005).
28. The apache software foundation, https://apache.org/, last accessed 2022/09/01.
29. CKJM extended. http://gromit.iiar.pwr.wroc.pl/p_inf/ckjm/ last accessed 2022/09/01.
30. WEKA Tool, https://sourceforge.net/projects/wekaclassalgos/, last accessed 2018/12/09.
31. AlKhiaty, M., Abdel-Aal, R., Elish, M.: Abductive Network Ensembles for Improved Prediction of Future Change-Prone Classes in Object-Oriented Software, The International Arab Journal of Information Technology, 14(6), 803-811, (2017).

Source Reliability Estimation for the Verification of the Authenticity of Information: an Evidential Approach

Hamza Tarik Sadouk, Faouzi Sebbak, Walid Cherifi, and Mohamed Amine Chouarfia

Ecole Militaire Polytechnique, PO Box 17, 16111 Bordj El Bahri, Algiers, Algeria
{sadouk.hamza.tarik,faouzi.sebbak,wa.cherifi,
,amine2chouarfia}@gmail.com

Abstract. The rapid evolution of technology has led to the increasing use of social networks. These have allowed various users to post whatever they want without any guarantee of the exactitude of the content posted, which has led to the rapid spread of false information, with severe consequences for the community. Despite extensive field study, dealing with the trustworthiness of sources remains an unanswered question. The main problem is that there is no efficient way to discern the difference between a genuine and a false assertion. To address this challenge, we have developed a hybrid approach to verify the authenticity of the information on Twitter based, on the one hand, on the stylistic analysis of the content of tweets, and the other hand, on the exploitation of the opinion of Twitter users by taking into account their reliability using the theory of evidence. Experimental assessments using publicly available datasets demonstrate very optimistic and better results when compared to other techniques.

Keywords: Fake news detection, Social media, Machine learning, Evidence theory, Source reliability.

1 Introduction

The emergence of social media (like Facebook and Twitter) has been propelled by the human urge to communicate and developments in digital technology, which have opened the door for the massive distribution of information. Social media users nowadays are producing and disseminating more information than ever before. Moreover, according to the latest recent statistics, half of the world's population (58.7%), or around 4.65 billion people, currently utilizes social media. In addition, 196 million new users entered the Internet the year before. On average, these people utilize social media for 2 hours and 29 minutes per day [1]. The dissemination of false news is a significant issue brought on by the growth of social networks. The latter is viewed as a form of propaganda meant to sway people's perceptions of the truth [10]. By inventing information and disseminating it via traditional print, the radio, and social media, this is content that

is portrayed as news content but is promotional material that the promoter is aware is false. Fake news has affected all facets of global civilization, including the economy and politics [17]. Finding such material online has become crucial due to the massive proliferation of false news and its detrimental effects. Different methods can be used to spot this incorrect information. These tactics often rely on fact-checking websites whose verification is based on lists of websites and items suspected of being false that experts provide. The issue with this method is that it needs human experience, which takes time and money. Additionally, web-based fact-checking services only include articles from particular fields, frequently politics, making it challenging to spot false information in other contexts [8].

Fortunately, things were made simpler by using machine learning algorithms based on the linguistic content of these posts and other features[6]. These methods can replace an expert while speeding up and saving on verification. With the help of these techniques, we were able to create a system for determining the veracity of the information on "Twitter." Our solution uses style content verification on the one hand and stance identification of comments towards the target tweet on the other. It might be challenging to determine a tweet's credibility based only on the opinions of other users. The stance detection technique may reveal user opinions, but appropriate mathematical methods are needed to estimate the validity of sources and reach an agreement in a noisy environment.

Dempster-Shafer Theory (DST), also identified as evidence theory [15], is a flexible mathematical framework that extends Bayesian theory by allowing each data source to include information at varying degrees of detail for addressing uncertainty [7]. It offers a powerful mechanism for consensus decision-making and has been extensively applied in many fields, especially in machine learning [18]. In this work, we look into how evidence theory is applied to classifying fake content. To the best of our knowledge, it is the first piece of work to address source reliability in the context of false content categorization using an evidence-based technique. The strategy put forward in this study emphasizes the potential of diverse information sources to discern between normal and aberrant behaviour. The suggested contextual discounting mechanism weakens source indicators differently according to their reliability. Consequently, we provide a single approach that uses content features and social environment attributes to identify fake information automatically.

The remaining of the paper is structured as follows. The theoretical background is reviewed in the second section. Section 3 discusses the Evidential System for the VeRification of the Authenticity of Information. Section 4 discusses the experimental results by comparing them with other techniques. Section 5 concludes the paper by summarizing our work and suggesting potential future work.

2 Theoretical Background

2.1 The theory of evidence

This section briefly recalls some basic notions of the Dempster-Shafer theory. For this purpose, let $\Omega = \{\omega_1, ..., \omega_k\}$, and let $\wp(\Omega) = \{A_1, ..., A_q\}$ be its power set, with $q = 2^k$. A function m defined from $\wp(\Omega)$ to $[0, 1]$ is called a "basic belief assignment (bba)" or "masse function" if $\sum_{A \in \wp(\Omega)} m(A) = 1$.

A *bba* m defines then a "plausibility" function Pl from $\wp(\Omega)$ to $[0, 1]$ by $Pl(A) = \sum_{A \cap B \neq \emptyset} m(B)$, and a "credibility" function Cr from $\wp(\Omega)$ to $[0, 1]$ by $Cr(A) = \sum_{B \subset A} m(B)$. Also, both aforementioned functions are linked by $Pl(A) + Cr(A^c) = 1$. Furthermore, a probability function p can be considered as a particular case for which $Pl = Cr = p$.

It is possible to implement a combination rule to provide a combined set of masses from the sets of masses obtained from each of the information sources. Several combination rules have been developed within the framework of belief theory, for example:

Dempster's Rule (DR) This rule was initially proposed by *Dempster* [3]. *Dempster*'s combination rule is the normalized conjunctive operation that aims to aggregate evidence from multiple independent sources. This rule is given for any $A \neq \emptyset$ of $\wp(\Omega)$ and for $N \geq 2$ information sources by:

$$m_{DR}(A) = \frac{1}{1 - K} m_{Conj}(A) \tag{1}$$

Where K and $m_{Conj}(A)$ denote the conflict measure and the unnormalized conjunctive rule respectively:

$$K = \sum_{\substack{B_i \in \wp(\Omega) \\ \cap_{i=1}^{l} B_i = \emptyset}} \prod_{j=1}^{N} m_j(B_i) \tag{2}$$

$$m_{Conj}(A) = \sum_{\substack{B_i \in \wp(\Omega) \\ \cap_{i=1}^{l} B_i = A}} \prod_{j=1}^{N} m_j(B_i) \tag{3}$$

However, scholars suggest that when the conflict between sources is severe, this combination rule behaves abnormally [14]. An alternative is:

Majority Consensus Rule (MCR) Proposed by *Sebbak et al.* [13], the rationale behind this combination rule is that it redistributes global conflict into already involved focal element sets, resulting in a majority and consensus. This rule is given for any $A \neq \emptyset$ of $\wp(\Omega)$ and for $N \geq 2$ information sources by:

$$m_{MCR}(A) = \frac{1}{N + 1 - K} \left(m_{Maj}(A) + m_{Conj}(A) \right) \tag{4}$$

Where $m_{Maj}(A)$ denotes the majority rule, it is given by:

$$m_{Maj}(A) = \sum_{j=1}^{N} m_j(A) \tag{5}$$

Discounting methods can estimate the weakening coefficients assigned to a source to correct its decision. These adjustments differ depending on whether it is a classic or contextual weakening.

Classical discounting The weakening of mass functions makes it possible to model sources' reliability by introducing a coefficient α_s where for each source s ,we have:

$$\begin{cases} m'_s(A) = \alpha_s.m_s(A) & \forall A \in 2^{\Omega}, A \neq \Omega \\ m'_s(\Omega) = (1 - \alpha_s) + \alpha_s.m_s(\Omega) \end{cases} \tag{6}$$

Domain discounting The idea behind domain weakening is that the reliability of a source can vary depending on the object to be recognized. The method we propose belongs to this category and is described below.

$$\begin{cases} m'_s(A) = \alpha_s^d.m_s(A) & \forall A \in 2^{\Omega}, A \neq \Omega \\ m'_s(\Omega) = (1 - \alpha_s^d) + \alpha_s^d.m_s(\Omega) \end{cases} \tag{7}$$

Where α_s^d is the weakening coefficient of the s^{th} source in the domain d.

2.2 Fake news detection

"Fake news" is untrue material that looks like news media content but has a different organizational structure and intended audience. *Misinformation* (false or misleading information) and *disinformation* (false information that is purposely spread to deceive people) are two information pathologies that share characteristics with fake news [10]. In our work, we use the term "fake news" to address both "disinformation" and "misinformation," These two occurrences have one thing in common: they both include incorrect information that may be identified by the features described in figure 1.

In contrast to social media, where extra information from the social environment can be utilized as additional data to detect fake news, conventional media primarily rely on news content. *K. Shu et al.* [16] defines a taxonomy of these characteristics to help detect false information.

2.3 Content-based approaches

Content-based strategies use various news sources' material, including article content, headlines, images, and videos. Specific categories for current methods include *Knowledge-based* and *Style-based* [6].

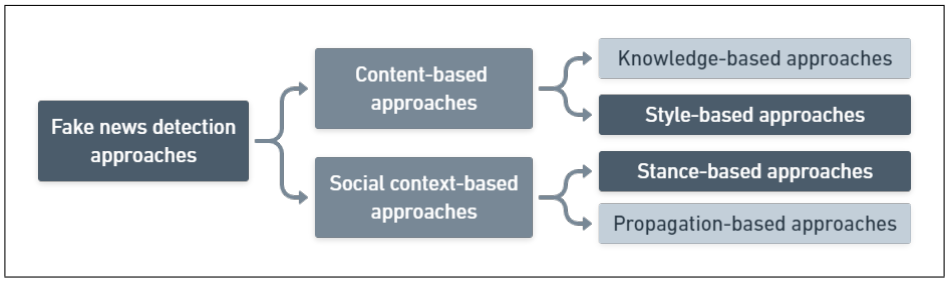


Fig. 1. Fake news detection taxonomy.

- *Knowledge-based:* Fake news attempts to promote false claims in news content. The most straightforward technique to identify fake news is to evaluate the veracity of the claims made in a news item to assess the news' credibility. In this process, suggested material is fact-checked using outside sources.
- *Style-based:* In order to appeal and persuade a broad spectrum of customers, fake news writers usually have a malicious goal of spreading erroneous information. This necessitates using specialized writing styles that are not readily apparent in legitimate news articles.

2.4 Social context-based approaches

Researchers have greater tools to amplify and enhance *content-based* models due to the nature of social media. Social context modeling methods now in use may be split into two categories: *stance-based* and *propagation-based*.

- *Stance-based:* Users' viewpoints from pertinent post contents are used to infer the veracity of original news items. Automatically determining from a post whether a person is in favour of, neutral toward, or opposed to a particular thing, event, or concept is referred to as *stance detection*.
- *Propagation-based:* The interrelationships of relevant social media posts are taken into account by fake news detection techniques based on propagation to anticipate news plausibility. The fundamental assumption is that a news event's credibility and the veracity of relevant social media posts are strongly connected.

3 The Evidential System for the VeRification of the Authenticity of Information *ES-VRAI*

The system automatically pulls the tweet's text, including the comments. Then, specific text prepossessing procedures are carried out on this data at the application server level. The inference server and the database receive the output of these operations. The two models are housed on the inference server, and the findings are kept in the database for future use by other components of our system. Figure 2 represents the system modules and their interactions.

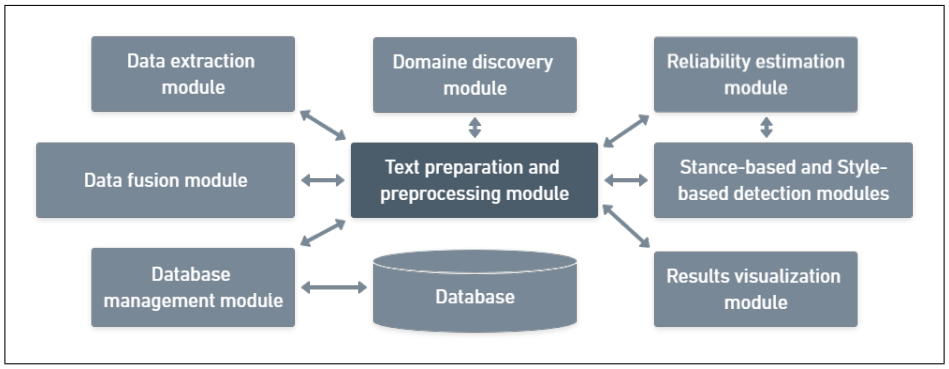


Fig. 2. ES-VRAI modules and interaction.

3.1 Data extraction module

The module retrieves the tweet's content and comments. It has an "Extractor" class responsible for extracting data from the social network Twitter. The class succeeded in extracting data from Twitter thanks to the "tweepy" library, which is used to interact with the Twitter database.

3.2 Text preparation and preprocessing module

Preprocessing of the text containing the information is necessary in order to verify the accuracy of the information. This module applies natural language processing preparation phases using the "nltk" library. The major procedures are lowercase text conversion, punctuation removal, stop word removal, word stemming, and lemmatization of words. The preprocessing results are saved in the same file after receiving the file containing the texts that the extraction module has obtained.

3.3 Style-based and stance-based detection modules

The models' architecture uses Long Short-Term Memory (LSTM) networks [9]. The "Embedding" layer uses Word2Vec [11] to calculate the feature vector of each word. The style-based model was trained on the "Fake and real news dataset", where it reached a validation accuracy of 97.75%. The stance-based model was trained on the "Fake News Challenge" stance detection dataset, where it reached a validation accuracy of 96.06%.

3.4 Domain discovery module

Each tweet extracted by the system has a particular domain. It can relate to one of the following classes: {Politics, Business, Entertainment, Sport and Technology}. The reliability estimation module will use it to evaluate the trustworthiness

of each data source, i.e. the Twitter users. We used a deep learning model based on LSTM networks in this module. The experimental results section will discuss the result and comparison with other machine learning models.

3.5 Reliability estimation module

In order not to bias the results of the stance detection model towards tweets because of malicious accounts, we exploited the style verification model to assign reliability to commentators. This is done by analyzing a commenter's (s) available tweets in the domain (d). For each commentator, we get:

$$\alpha_s^d = \frac{\sum_k p_k^d}{n_s^d} \tag{8}$$

Where p_k^d represents the probability that the commentator (s) k^{th} tweet in the domain d is true according to the style based model, and n_s^d represents the number of available tweet of the commentator s in the domain d.

3.6 Data fusion module

The system must generate two probability values $P_{t,C}(True)$ and $P_{t,C}(False)$. First, the fusion module obtains the values $P_t(True)$ and $P_t(False)$, which respectively represent the probability that the tweet is true and the probability that it is false according to the Style-based module. Then, it retrieves the values of the position of each comment relative to the content of the same tweet. So for each comment $c_i \in C$, we will have: $P_{c_i}(Agree), P_{c_i}(Disagree), P_{c_i}(Discuss)$ and $P_{c_i}(Unrelated)$. After, the module retrieves the reliability of each commentator relative to the domain of the tweet, so for each commentator (s), we will have α_s^d.

We must convert the probability functions into mass functions. For the first model the conversion is straightforward, $m_t(True) = P_t(True)$, $m_t(False) = P_t(False)$ and $m_t(\{True, False\}) = 0$. For the second model, we elaborated a new "bba" strategy. It proceeds in two steps:

- *Hypotheses restriction:* in this step we exclude all the comment c_j that are unrelated to the tweet t, i.e., $argmax(P_{c_j}) = Unrelated$.
- *Hypotheses projection:* each person commenting on the tweet implicitly indicates their opinion concerning this tweet. That is, if the user's opinion agrees with the tweet, so it shows that the tweet is true: $m_{c_i}(True) = P_{c_i}(Agree)$. If it disagrees with the tweet so it shows that the tweet is fake then: $m_{c_i}(False) = P_{c_i}(Disagree)$. The rest of the belief will be considered as ignorance: $m_{c_i}(\{True, False\}) = P_{c_i}(Discuss) + P_{c_i}(Unrelated)$.

Before the combination, we use the weakening coefficients assigned to each source to correct its decision. Domain discounting will be used in our case where for each comment c_i, we will weaken its mass function using equation 7. Once all the "bbas" are constructed, we aggregate them in two steps:

- *Stance bba's fusion:* for the mass functions issued from the *Stance-based* model, we will use the "MCR" fusion rule. The final mass function denoted $m_{C'}$ will describe the majority consensus opinion of the crowd.
- *Style and Stance fusion:* once the $m_{C'}$ calculated we will use "DR" to aggregate it with m_t. It will only produce a mass of certainty for the singletons denoted m. This is a probabilistic case and the desired probability will be: $P_{t,C}(True) = m(True)$ and $P_{t,C}(False) = m(False)$.

3.7 Results visualization module

Our system is intended for use in social networks (Twitter in the first place), which are websites. That is why we use an extension to offer our services because it is the most suitable solution in our case. Figure 3 represents the *ES-VRAI* home page.

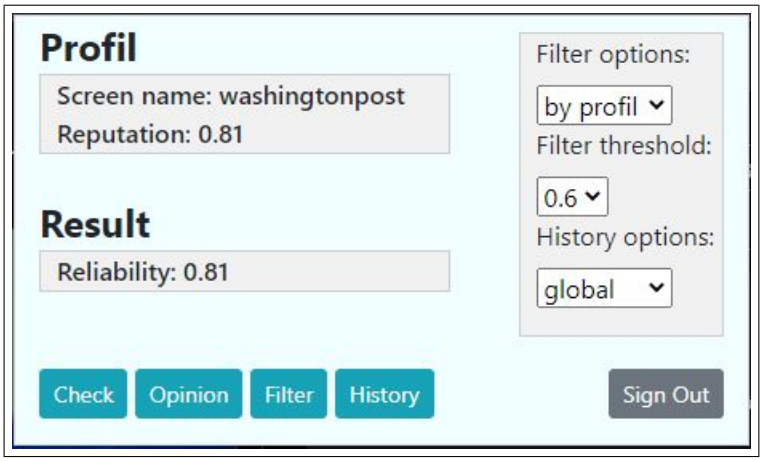


Fig. 3. *ES-VRAI* home page.

3.8 Database management module

Our system needs a database to store the different data indicated during the information verification phase to be able to satisfy the other functionalities: consultation of the history of tweets already verified, filtering of the news feed by profile or by tweet and the expression of opinion, which allows the users to give their opinion on the information they receive on Twitter.

4 EXPERIMENTAL RESULTS

In this section, we assess the proposed domain discovery model's efficiency compared with some classical machine learning and deep learning models. The tests

are conducted on the "BBC News Archive" [5]. It is a series of news stories from BBC News that are made available for use as comparisons in machine learning research. For the convenience of use, the original data is transformed into a single file while preserving the news title, the name of the relevant text file, the news content, and its category. It comprises 2225 documents from the BBC news website that relate to articles covering five different topics: {Politics, Business, Entertainment, Sport and Technology}. Figure 4 represents the distribution of the classes in the dataset.

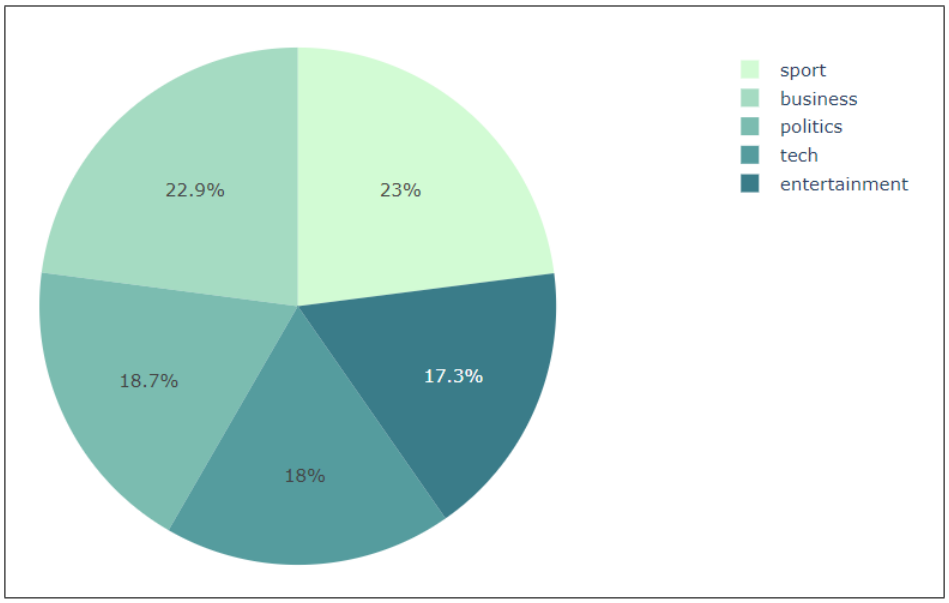


Fig. 4. Class distribution in the "BBC News Archive" dataset.

In this experiment, we used the Text Frequency inverse Document Frequency (TFiDF) of uni-grams for feature representation. The number of tokens was limited to 1000, and we used TensorFlow [2] for deep learning models, and scikit-learn [12] for machine learning models on the Colab environment [4]. We tested four machine learning models:

- Logistic Regression: with default parameters;
- Linear SVC: with default parameters;
- Random Forest: with the function to measure the quality of a split using the Shannon information gain (entropy);
- Multinomial Naive Bayes: with defaults parameters;

The findings are reported in table 1.

For the deep learning models, we used tree layers fully connected network with 16 hiding units with 'relu' activation function in one case and 16 'LSTM'

Table 1. Test metrics for machine learning models.

Model	Test Accuracy	Precision	Recall	F1
Random Forest	49.29	0.49	0.49	0.49
Logistic Regression	56.67	0.57	0.57	0.57
Multinomial Naive Bayes	57.86	0.58	0.58	0.58
Linear SVC	51.46	0.51	0.51	0.51

units with 0.7 of 'recurrent dropout' in the second. We used the 'rmsprop' optimize, the 'categorical cross entropy' as the loss function and accuracy as the performance measure. The training was done in 30 epochs with a batch size of 64.

Figure 5 shows the first model's training and validation accuracy and loss variation.

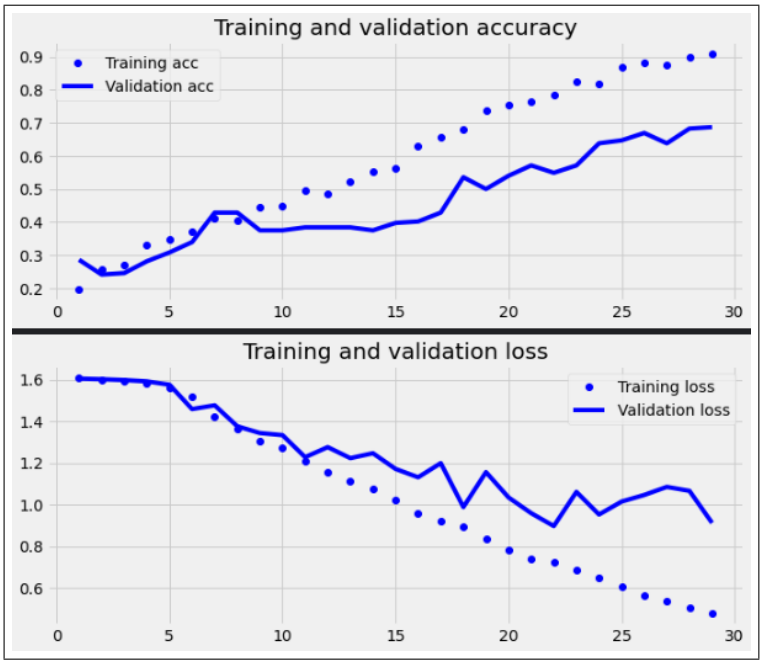


Fig. 5. Dense model accuracy and loss variation.

Figure 6 shows the second model's training and validation accuracy and loss variation.

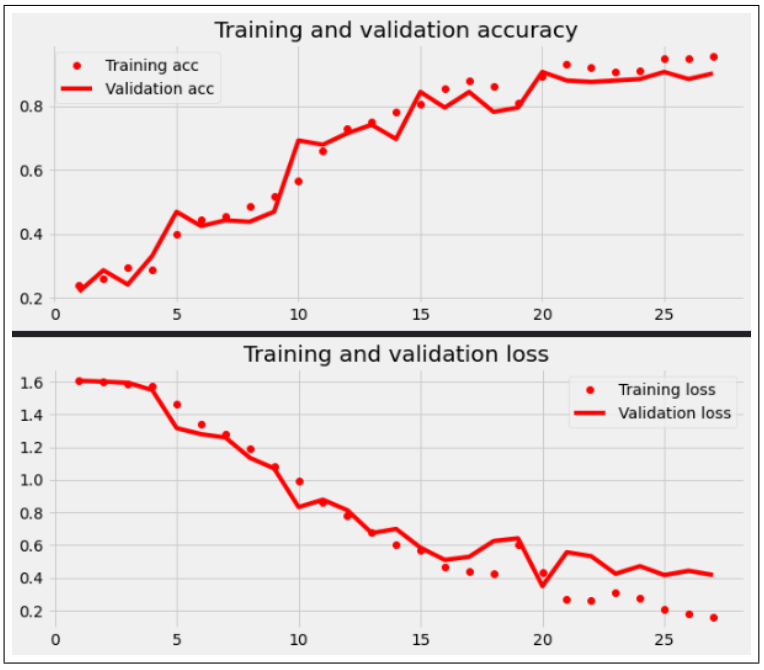


Fig. 6. LSTM model accuracy and loss variation.

5 Conclusion

We developed an evidence-based method to lower data uncertainty in misleading content classification. In this study, we investigated the use of the evidence theory to categorize false content. This study created a technique that integrates many information sources to distinguish between normal and aberrant behaviour. As a result, we offered a unified strategy that automatically detects false information by using both content aspects and social context factors. Experiments were done to assess the efficiency of the proposed LSTM architecture as part of The Evidential System for the VeRification of the Authenticity of Information, where it could successfully discern the domain of the targeted tweet.

References

1. Digital 2022: April Global Statshot Report mdash; DataReportal – Global Digital Insights (apr 21 2022), [Online; accessed 2022-06-08]
2. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng,

X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
3. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. The annals of mathematical statistics pp. 325–339 (1967)
4. Ekaba, B.: Google colaboratory. Building Machine Learning and Deep Learning Models on Google Cloud Platform; A Comprehensive Guide for Beginners (2019)
5. Greene, D., Cunningham, P.: Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proceedings of the 23rd international conference on Machine learning. pp. 377–384 (2006)
6. Gupta, A., Anjum, A., Gupta, S., Katarya, R.: Recent trends of fake news detection: A review. Machine Learning, Advances in Computing, Renewable Energy and Communication pp. 483–492 (2022)
7. Hamache, A., Boudaren, M.E.Y., Boukersoul, H., Debicha, I., Sadouk, H.T., Zibani, R., Habbouchi, A., Merouani, O.: Uncertainty-aware parzen-rosenblatt classifier for multiattribute data. In: International Conference on Belief Functions. pp. 103–111. Springer (2018)
8. Hassan, N., Adair, B., Hamilton, J.T., Li, C., Tremayne, M., Yang, J., Yu, C.: The quest to automate fact-checking. In: Proceedings of the 2015 computation+ journalism symposium (2015)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Lazer, D.M., Baum, M.A., Benkler, Y., Berinsky, A.J., Greenhill, K.M., Menczer, F., Metzger, M.J., Nyhan, B., Pennycook, G., Rothschild, D., et al.: The science of fake news. Science **359**(6380), 1094–1096 (2018)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26** (2013)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
13. Sebbak, F., Benhammadi, F.: Majority-consensus fusion approach for elderly iot-based healthcare applications. Annals of Telecommunications **72**(3), 157–171 (2017)
14. Sebbak, F., Benhammadi, F., Mataoui, M., Bouznad, S., Amirat, Y.: An alternative combination rule for evidential reasoning. In: 17th International Conference on Information Fusion (FUSION). pp. 1–8. IEEE (2014)
15. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
16. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter **19**(1), 22–36 (2017)
17. Thijssen, Y.: Breaking the news: the effects of fake news on political attitudes. Master's thesis, University of Twente (2017)
18. Xu, P., Davoine, F., Zha, H., Denoeux, T.: Evidential calibration of binary svm classifiers. International Journal of Approximate Reasoning **72**, 55–70 (2016)

Enhanced merging order: A novel architecture for merging sub-triangulations

Z. Tchantchane[1, 2], and O. Azouaoui[1], N.Goualmi[2], M. Bey[1]

[1] Centre de Développement des Technologies Avancées (CDTA)
Division Productique et Robotique (DPR)
BP 17, Baba Hassen, Algiers 16303, Algeria
[2] Université Badji Mokhtar, Faculté des Sciences de l'Ingénieur
Département d'Informatique, Laboratoire Réseaux et Systèmes
BP 12, Annaba 23000, Algeria.
tarabet.zahida@gmail.com, ztchantchane@cdta.dz

Abstract. With the technological evolution in different industries such as automotive, aeronautics, industrial and medical design, the need to produce complex objects from a point cloud using the Reverse Engineering process is still present. The acquired point cloud is very dense and unstructured, which requires a high processing time and an increased cost for the reconstruction step. Yet, surface reconstruction techniques are limited or insufficient to achieve the desired shapes. The 3D Delaunay triangulation is one of the oldest and most fundamental surface reconstruction techniques. Its objective is to create triangles from points using different algorithms. The fastest algorithm is based on the divide-and-conquer, which is generally designed to be used for parallelism. This algorithm is carried out in two steps: (i) the first step recursively partitions the points set into sub-regions; each is assigned to a processor. Independently, these regions are further triangulated simultaneously. (ii) The second step merges the sub-regions into the final mesh, which is applied in the reverse order of points set partitioning. Therefore, merge step is complex and hard to be implemented especially in 3D. This work deals with the generation of a 3D triangulation from any point cloud, which is partitioned to several sub-points using Octree. Independently, the sub-points are further triangulated simultaneously by parallelizing the calculations on several processors. Then, a novel 3D merging architecture, Enhanced Merging Order (EMO), is developed to merge the Octree nodes without knowing the partitioning direction. Finally, this merge architecture is tested and validated through many unstructured point clouds.

Keywords: Unstructured point cloud, Divide-and-conquer algorithm, Merging Order, Octree, Delaunay triangulation.

1 Introduction

With technological advancement in different industries such as automotive, robotics, aeronautics, industrial and medical design. The need to produce more complex objects

is always present with more details and a higher degree of precision. In industrial practice, surface models are obtained either (i) by using CAD software if shapes are more or less simple, or (ii) by using the reverse engineering process if shapes are very complex or if CAD models are not available. The reverse engineering process has now become a very common practice in the industrial world. The reconstruction of the CAD model is a very delicate and time-consuming step, which necessitates the development of methods allowing the improvement of computation time to generate the desired model. The existing methods are classified into two categories, implicit methods and explicit methods. The latter is the most commonly used, since they are mainly local geometrical approaches based on Delaunay triangulation or Dual Voronoï diagram. After a study on main advantages and drawbacks of several reconstruction algorithms, the Delaunay triangulation method is the most suitable with three kinds of algorithms: divide-and-conquer, sweep line and incremental insertion. Generally, divide-and-conquer is designed for parallelism and needs multi-core processor computers. It is realized in two steps, the first step recursively divides the points set into sub-regions; each is assigned to a processor. Independently, these regions are further triangulated, simultaneously. The second step merges the sub-regions into the final mesh; the merge is applied in the reverse order of partitioning the points set. This step still represents a connection problem, especially for Delaunay triangulation of non-structured point clouds. Merge details are complex and hard to be implemented especially in 3D. This work deals with the generation of a 3D triangulation from any points set. This latter is partitioned on several subset points using Octree; each is assigned to a different processor. Independently, these subset points are further triangulated at the same time by parallelizing the calculations on several processors to reduce the processing times. The main contribution is to propose a novel 3D merge architecture, Enhanced Merging Order (EMO), that specifies merge order and agreement assigned for each Octree node without being aware of the partitioning direction. The aim is being to reduce the processing time by parallelizing the calculations on several processors. The remainder of this paper is organized into five sections. Section 1 consisted of a general introduction. Section 2 reviews the related research works on Delaunay triangulation. Section 3 describes the proposed parallel Delaunay triangulation algorithm. Section 4 analyzes and compares the obtained results. Section 5 provides a brief conclusion and future works.

2 Related work

The need for reconstructing a 3D model from a point cloud is still present. The choice of the reconstruction method depends on the point cloud type, which may be uniform or non-uniform [1]. Since the object scanning gives a non-uniform point cloud, the surface reconstruction became a difficult task, especially in 3D spaces or more [2]. In general, surface reconstruction methods are classified into four classes [3]: (i) explicit form, (ii) implicit form, (iii) computer vision and (iv) soft computing. Explicit form methods are able to represent faithfully the surface compared to implicit form [4]. As indicated in [4], two different types of explicit surfaces exist: (a) parametric surfaces are topologically limited by the initial model; this means that complex surfaces are not easily

represented. (b) triangulated surfaces are the most intuitive version of surface representation; the surface is described by triangles connected from the input points. This justifies the development of Dual Voronoï diagram and Delaunay triangulation algorithms. Many research works studied the main advantages and drawbacks of the different reconstruction algorithms, depending on several criteria [5] such as algorithm complexity, point cloud, and surface topology. Delaunay triangulation method seems to be the most appropriate. Three types of algorithms are commonly used to build Delaunay triangulations: (i) sweep line algorithms, (ii) incremental insertion algorithms and (iii) divide-and-conquer algorithms. The sweep line algorithms [6] and the incremental insertion algorithms [7,8] are sequential algorithms. The fastest algorithm is based on divide-and-conquer and is generally designed to be used for parallelism. The divide-and-conquer algorithm is presented for the first time by Shamos and Hoey [9]. With the appearance of multi-core machines, several parallel algorithms are thus developed using more than one processor such as in [10]. Here, the point cloud is recursively divided into sub-regions; each is assigned to a processor. Independently, these regions are further triangulated and merged into one domain, simultaneously. Cignoni et al. [11] proposed DeWall which is a slightly different approach; instead of merging partial triangulations, a more complex division phase is applied, which partitions the points set and builds the merging triangulation first. The result is used subsequently to build the triangulate of the two subsets points. Authors in [12] defined a parallel algorithm based on partitioning the point cloud along a single dimension. Each processor performed then the sub-triangulation; after that, the sub-triangulation is merged in the reverse order of the points set partitioning. Since it is difficult to merge sub-triangulations, Bin Chen [13] introduced the affected zone approach, which determines the zone that can be modified when merging the sub-triangulations. Later, the generated triangulation interface (global triangulation) is given by introducing the first tetrahedron and generating the generic tetrahedron. However, these methods contain complicated merging steps. With the emergence of multi-core machines, several parallel algorithms for Delaunay triangulation are proposed to improve performances and overcome the limitations of existing techniques. Additionally, the 3D parallel Delaunay triangulation for a non-uniform distribution of points remains an interesting question, which could be investigated. To face this issue, this work proposes a methodology, which divides the point cloud on several sub-clouds using Octree. After that, these sub-cloud points are further triangulated independently; this is done by parallelizing the calculations on several processors to reduce the processing time, simultaneously. The contribution of this work is manifold. First, it proposes a novel 3D merge architecture that specifies the merge order and agreement assigned for each Octree node without being aware of the partitioning direction. Second, it reduces the processing times by parallelizing the calculations on several processors. Finally, it validates the developed merge architecture on many unstructured point clouds.

3 Proposed methodology

The proposed methodology, Enhanced Merging Order (EMO), is based on divide-and-conquer and incremental insertion algorithm to generate the 3D triangulation of an unstructured point cloud. This solution aims at twofold: (i) proposition of a strategy to partition the point cloud on several sub-points using Octree and (ii) development of a novel merge architecture that specifies merge order and agreement assigned for each Octree node without being aware of the partitioning direction. Figure 1 shows the general structure of the proposed solution.

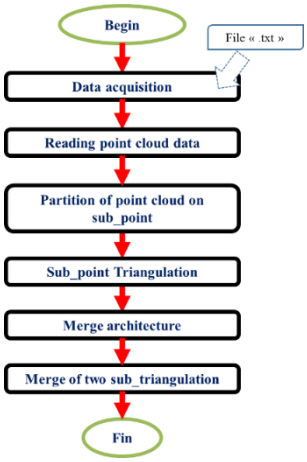


Fig 1: General structure of the proposed EMO solution

3.1 Reading point cloud data

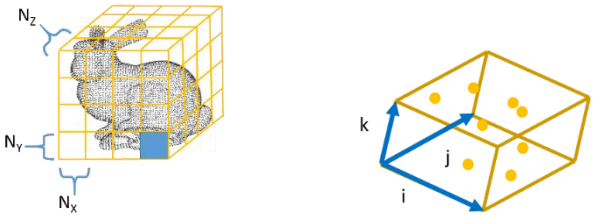
After acquiring the file containing the point cloud representing the object, the file is verified syntactically and semantically. In case this file is correct, all the points will be stored in an array of point structure. This operation allows defining the object envelope defined by its extremum points (X_max,X_min,Y_max,Y_min,Z_max,Z_min); accordingly, its dimensions (length, height, width) are calculated

3.2 Partition of the point cloud

The main objective of this approach is to parallelize the Delaunay triangulation of the unstructured point cloud. For this purpose, the Octree is used to have more or less equal number between partitions; this drives to a balanced workload at different processors. The following subsections describe the creation of the Octree

Creation of boxes: To facilitate the points manipulation when creating the Octree, the raw is subdivided into small cells of the same size called boxes (Figure 2a). Thereafter,

each point is assigned to its box using its indices (Figure 2b). Algorithm 1 shows the creation of boxes.



a. boxes b. Indices of boxes

Fig 2: Creation of boxes

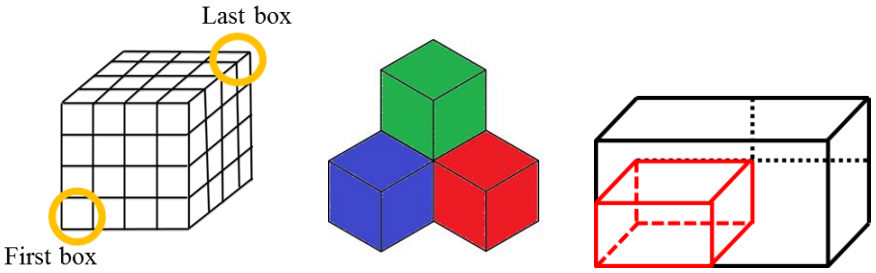
Algorithm 1: Boxes creation{
Input
Set of points: X;
Boxes number: n_x, n_y, n_z;
Output
Boxes $(box_1, box_2, box_3 \dots box_k)$;
Calculate the limit of points set: X_{min}, X_{max}, Y_{min}, $Y_{max}, Z_{min}, Z_{max}$;
Calculate the dimensions of the raw material: $length$, $width$ and $height$;
Calculate the number of boxes: $Nbr\_x = 2^{nx}, Nbr\_y = 2^{ny}, Nbr\_z = 2^{nz}$;
Calculate the dimensions of the box: $pasX = \frac{X_{max} - X_{min}}{Nbr\_x}, pasY = \frac{Y_{max} - Y_{min}}{Nbr\_y}, pasZ = \frac{Z_{max} - Z_{min}}{Nbr\_z}$
Create 3D arrays: boxes along X-axis, Y-axis and Z-axis;
for each point{
Calculate I, J, K indices as follows:

$$I = \frac{x - X_{min}}{pasX}, J = \frac{y - Y_{min}}{pasY}, K = \frac{z - Z_{min}}{pasZ}$$

Assign the point $P(x, y, z)$ to the box (I, J, K);
}
}

Algorithm 1: Boxes creation algorithm

Creation of the Octree: In this step, the point cloud is transformed into an Octree (Figure 3). To minimize the storage space, the father and children nodes are represented directly by the indices of boxes previously defined. The creation of the Octree is realized as shown in *Algorithm 2*.



a. Octree limits b. Neighbors of one node c. Children nodes

Fig 3: Creation of the Octree

Algorithm 2: Octree creation{
Input
Boxes: $box_1, box_2, box_3 \dots box_k$;
Maximum number of points in each node of Octree: N;
Octree Root (father): composed of all boxes;
Limits of father defined by: indices of first and last boxes;
The first node: Octree Root;
Output
Octree($Root, child_1, child_2 \dots child_8, \dots$);
for each node of *Octree*{
if the points number in this node $\geq N$
Divide this node on eight nodes;
}
Create an array from the obtained nodes of the Octree;
Define each node of the Octree by its father, children, three neighbors, position, number of points and limits;
Define the limits based on boxes indices;
}

Algorithm 2: Creation of the Octree

3.3 Triangulation of the sub-points

To reduce the processing time of 3D Delaunay triangulation, the calculations of sub-points triangulations are parallelized on several processors. Each processor runs the *Destruction-construction* procedure detailed in what follows [15]:

Creation of the super-tetrahedron: To ensure that each inserted point belongs to a tetrahedron, a virtual tetrahedron, super-tetrahedron, is created.

Generation of the 3D triangulation: The generation of the 3D Delaunay triangulation and the *Destruction-construction* procedure follows the steps described bellow.

Insertion of points: the generation of 3D triangulation is done by inserting one point at a time in *sequential* or *random*.

Determination of the Delaunay tetrahedra kinds: once a point is inserted, all tetrahedra whose circumscribed spheres contain this inserted point are identified (Figure 4a). The kind depends on the distance between the inserted point and the center of the tetrahedron circumscribed sphere. Two cases are possible: If the distance is less than the radius, the tetrahedron is non-Delaunay. Otherwise, the tetrahedron is Delaunay.

Modification of the 3D triangulation: for each non-Delaunay tetrahedron, its faces are stored in a list without repetition. Next, these tetrahedra are deleted. Based on the list of stored faces and inserted point, new valid tetrahedra are created (Figure 4b).

a. Circumscribed sphere verification b. 3D triangulation update

Fig 4: 3D triangulation modification

3.4 Merge architecture

Once the sub-triangulations are calculated, their merging is necessary to generate the global triangulation without being aware of the partitioning direction. To face this, a novel merge architecture is proposed to specify merge order and determine the agreement assigned for each Octree node. The architecture is carried out in three directions: X-axis, Y-axis and Z-axis; it is described in the diagram of Figure 5 and *Algorithm 3* summarizes this approach.

.

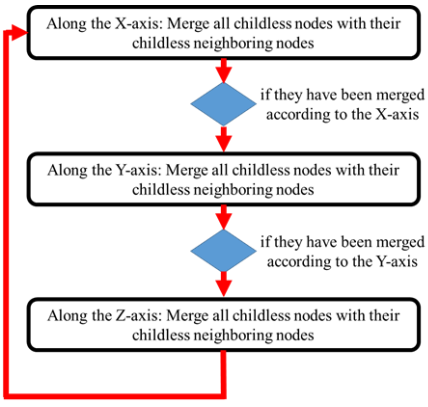


Fig 5: Merge architecture steps

Algorithm 3: Merge architecture{
Introduce the nodes of the Octree;
repeat{
Find all nodes without children along X-axis;
for each node{
Determine its neighbor nodes without children along X-axis;
if exist, merge all nodes two-by-two along the X-axis in parallel (new parallelism level);
}
Find all nodes without children along Y-axis;
for each node{
Determine its neighbor nodes without children along Y-axis;
if exist, merge all nodes two-by-two along the Y-axis in parallel (new parallelism level);
}
Find all nodes without children along the Z-axis;
for each node{
Determine its neighbor nodes without children along the Z-axis;
if exist, merge all nodes two-by-two along the Z-axis in parallel (new parallelism level);
}
}**until** (the parent node is achieved);
}

Algorithm 3: Merge architecture

3.5 Merging approach of two sub-triangulations

To accelerate the merge of two sub-triangulations, the affected zones of adjacent nodes are defined and used thereafter. It determines all the tetrahedra that may be modified during merge step. Once the affected zones are defined, *Algorithm 4* is implemented.

Algorithm 4: Merge of two sub-triangulations{
Introduce the sub-triangulations of two nodes T_1 and T_2;
Introduce the defined affected zones of two nodes AZ_1 *list* and AZ_2 *list*;
Create a small subset *B* that contains without repetition:

- All points defining tetrahedra of AZ_1 *list*;
- All points defining tetrahedra of AZ_2 *list*;

Triangulate the small subset *B* into *T(B)*;
Find the final tetrahedron *Final_T* from T_1, T_2 and *T(B)*;
for each tetrahedron of T_1{
if it does not belong to AZ_1 *list*, add it to *Final_T*;
else continue to next tetrahedron;
}
for each tetrahedron of T_2{
if it does not belong to AZ_2 *list*, add it to *Final_T*;
else continue to next tetrahedron;

```
        }
        for each tetrahedron of T(B){
                if points are from both partitions, add them to Final_T;
                if points are within one partition (T1 and T2) and this tetrahedron replaces a previously found tetrahedron of AZ1 list or AZ2 list, add them to Final_T;
        }
}
```

Algorithm 4. Merge of two sub-triangulations

4 Results and discussions

The proposed EMO methodology is implemented in C++Builder and OpenGL graphics library running on Windows 7. Validation tests are performed on an Intel Core i3 PC with 6GB RAM.

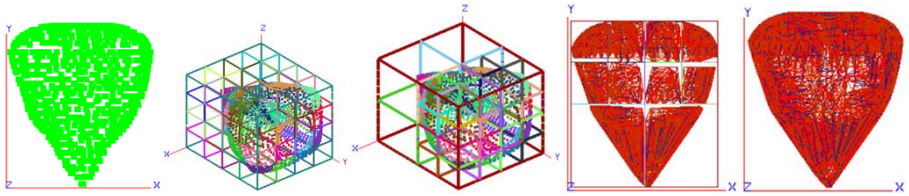
4.1 Obtained results

The effectiveness and performances of EMO methodology are demonstrated on Tooth model (Figure 6a) with a resolution of 1366×768 pixels. For the validation, Tooth model contains 4300 points; boxes number along three directions X-axis, Y-axis and Z-axis is equal to 2^2 in each direction (Figure 6.b). The maximum number of points in each node of the Octree is equal to 500 points. Thus, an Octree is created which contains 40 nodes; amongst them 36 nodes without children (Figure 6c). The nodes without children are triangulated in parallel using Destruction-construction procedure. Therefore, only 36 are triangulated in parallel (Figure 6d). Since the triangulation is performed for the whole points set, the sub-triangulations are merged two-by-two until finding the final triangulation. To merge these nodes along the three directions, Algorithm 3 gives five parallelism levels (Table 1):

Levels	DIRECTIONS	Exanimated nodes
Level 1	X-axis	18 pairs
Level 2	Y-axis	8 pairs
Level 3	Z-axis	4 pairs
Level 4	X-axis	2 pairs
Level 5	Y-axis	2 pairs

Table 1 illustrates the five parallelism levels, the direction of merge and the concerned nodes.

At this moment, these sub-triangulations are merged. The obtained triangulation is very close to the theoretical model (Figure 6e).



a. Point cloud b. Created boxes c Created Octree tree d. sub-triangulations e. global triangulation

Fig6: Obtained triangulation of the Tooth model

4.2 Comparative study

To assess the performances of the proposed methodology, a comparative study is carried between our EMO approach and the merge of nodes in the reverse order of their partitions, Reverse Order of Merge (ROM) approach [12]. Tables 1 and 2 illustrate the levels of the examined nodes and merge directions for both architectures, respectively.

Table 1 indicates that the EMO approach gives five parallelism levels. Whereas, Table 2 shows that the ROM approach gives six-parallelism levels.

Levels	DIRECTIONS	Exanimated nodes
Level 1	X-axis	16 pairs
Level 2	Y-axis	8 pairs
Level 3	Z-axis	4 pairs
Level 4	X-axis	4 pairs
Level 5	Y-axis	2 pairs
Level 6	Z-axis	1 pairs

Table 2: Parallelism levels for the ROM architecture

It must be mentioned that each direction represents a parallelism level. From both tables, the number of parallelism levels of EMO architecture is less than that of ROM architecture. Thus, one more direction along the Z-axis is needed in the ROM architecture to obtain the final triangulation. Additionally, considering the number of pairs by level, the total number of the novel EMO architecture is less than that of the ROM architecture.

The EMO approach reduced the number of directions; as a result, it decreased the number of parallelism levels. Further, the EMO architecture minimized the number of pairs; thus, it diminished merge steps of pairs. Both consequences have a positive effect on the calculation time. Therefore, it can be deduced that the novel EMO architecture is able to return better solutions. This observation is more visible especially with a deep tree structure and availability of multicore computers for parallel computing.

A second comparison is made regarding the computation times of the 3D Delaunay triangulation. Results returned by the EMO approach are compared with two sequential approaches of the literature:

- Local Transformations Method (LTM) [16]

- Bowyer Algorithm (BA) [15].

For this purpose, the two previous approaches, LTM and BA, have been implemented and compared with EMO using another model, Convex part (Figure 2a), with different samples and densities. Table 3 illustrates the computation times for generating 3D Delaunay triangulation for all algorithms. The table clearly shows that the total computation times of EMO approach (using parallel computing) is better than that of both sequential approaches (LTM and BA).

Density / Methodology	**EMO** (s)	**LTM (s)**	**BA (s)**
300	0.8	46	35
400	1.2	90	70
700	3.7	217	169
900	5.3	367	302
1500	17	947	821
3200	102	4440	3861

Table 3: Computation times between EMO, LTM and BA

The examination reveal that the generated triangulations for all the approaches (BA, LTM and EMO) are very close to the theoretical model with a denser point cloud. However, BA and LTM methods required and important processing time of the Delaunay triangulation. On the other hand, the proposed EMO method considerably reduced the computation times with denser points.

In summary, the comparative study demonstrated that the sequential Delaunay triangulations gave good shapes with higher computation times. Moreover, their Delaunay triangulations are close to the theoretical model when the point clouds are dense (where parallel computing is recommended). Additionally, comparison of obtained results with the ROM architecture proved the superiority of the proposed EMO architecture in terms of levels number and pair nodes number, especially with a deep tree structure and the utilization of multicore computers.

5 Conclusions and future work

This paper presented the parallel generation of 3D Delaunay triangulation for unstructured point clouds. Its main objective consists of proposing a novel merge architecture of the sub-triangulations. The Enhanced Merging Order (EMO) method used Octree to partition the points set into subsets (nodes) with nearly equal density. To merge the obtained sub-triangulations, a novel merge architecture is applied along X-axis, Y-axis and Z-axis directions. The developed approach has been tested on different samples of two models, Tooth model and Convex part. Comparisons of obtained results with the Reverse order partition merging architecture, Local Transformations Method and Bowyer Algorithm demonstrated the superiority of the proposed methodology in terms of levels number, pair nodes number and mainly computation time, especially

when dealing with non-structured point clouds, using a deep tree structure and the utilization of multicore computers. Future work will utilize the parallelism of the GPU and examine other partitioning modes such as 3D multigrid, clustering methods, etc.

References

1. M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," ACM Trans. Graph., vol. 32, no. 3, 2013.
2. R. L. M. E. Do Rêgo and A. F. R. Araújo, "A surface reconstruction method based on self-organizing maps and intrinsic Delaunay triangulation," Proc. Int. Jt. Conf. Neural Networks, 2010.
3. S. P. Lim and H. Haron, "Surface reconstruction techniques: A review," Artif. Intell. Rev., vol. 42, no. 1, pp. 59–78, 2014.
4. A. Khatamian and H. R. Arabnia, "Survey on 3D surface reconstruction," J. Inf. Process. Syst., vol. 12, no. 3, pp. 338–357, 2016.
5. K. Khanna and N. Rajpal, "Survey of curve and surface reconstruction algorithms from a set of unorganized points," Adv. Intell. Syst. Comput., vol. 258, pp. 451–458, 2014.
6. B. Žalik, "An efficient sweep-line Delaunay triangulation algorithm," CAD Comput. Aided Des., vol. 37, no. 10, pp. 1027–1038, 2005.
7. B. Joe, "Construction of three-dimensional Delaunay triangulations using local transformations," Comput. Aided Geom. Des., vol. 8, no. 2, pp. 123–142, 1991.
8. P. Maur and I. Kolingerova, "Post-optimization of Delaunay tetrahedrization," in Proceedings Spring Conference on Computer Graphics, 2001, pp. 31–38.
9. M. I. Shamos and D. Hoey, "Closest-point problems," in Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS, 1975, vol. 1975-Octob, pp. 151–162.
10. H. Wu, X. Guan, and J. Gong, "ParaStream: A parallel streaming Delaunay triangulation algorithm for LiDAR points on multicore architectures," Comput. Geosci., vol. 37, no. 9, pp. 1355–1363, 2011.
11. P. Cignoni, C. Montani, and R. Scopigno, "DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed," CAD Comput. Aided Des., vol. 30, no. 5, pp. 333–341, 1998.
12. M. Bin Chen, "A parallel 3D Delaunay triangulation method," Proc. - 9th IEEE Int. Symp. Parallel Distrib. Process. with Appl. ISPA 2011, pp. 52–56, 2011.
13. M. Bin Chen, "The merge phase of parallel divide-and-conquer scheme for 3D delaunay triangulation," Proc. - Int. Symp. Parallel Distrib. Process. with Appl. ISPA 2010, pp. 224–230, 2010.
14. W. Wu, Y. Rui, F. Su, L. Cheng, and J. Wang, "Novel parallel algorithm for constructing Delaunay triangulation based on a twofold-divide-and-conquer scheme," GIScience Remote Sens., vol. 51, no. 5, pp. 537–554, 2014.
15. Z. Tchantchane, M. Bey, K. Azouaoui, and H. Bendifallah, "STL Model of Sculptured Surfaces from 3D Unstructured Cloud of Points," 11th EMP Mechanical Days (JM'11–EMP), pp. 3–7, 2018.
16. S. H. Lo, "3D Delaunay triangulation of non-uniform point distributions," *Finite Elem. Anal. Des.*, vol. 90, pp. 113–130, 2014.

Adding Robustness against Geometric Attacks to a DFT-DCT Based Watermarking Approach

Abdelhamid Saighi, Salim Chitroub

USTHB, Algiers, ALGERIA

abd_saighi@yahoo.fr, s_chitroub@hotmail.com

Abstract. This paper investigates a robust digital image watermarking scheme that uses discrete Fourier transform (DFT), discrete cosine transform (DCT), and scale-invariant feature transform (SIFT). In fact, it is of paramount concern to design watermarking techniques based on image features that withstand standard signal processing transformations as well as geometric distortions, while keeping a low induced distortion. To this end, a robust watermark is inserted within the DFT-DCT space to enhance the robustness against operations of image processing. SIFT, on the other hand, allows to improve the watermark robustness against geometrical distortions. Firstly, the watermark is embedded in the middle band of the DCT coefficients of the DFT magnitude of the host image, using a secret key. Then, the SIFT feature points, are stored for a subsequent use, when extracting the watermark, in order to determine and invert the geometric distortions, such as rotation, scaling, as well as translation (RST). Several experiments were carried out on grayscale images to evaluate the method. Results show that, compared to existing methods, the approach presented herein, is robust against common image transformations while it maintains good imperceptibility.

Keywords: DFT, DCT, SIFT, RST, copyright protection, robustness, imperceptibility, oblivious.

1 *Introduction.*

In our modern society, we rely more and more on digitalized information that can be easily accessed, duplicated and then transmitted through open networks. Therefore, the need to design new systems that can protect digitized contents from abuse has increased substantially over the past thirty years. Watermarking, a set of techniques, used to add extra information, called watermark, into the original image, can be a potential solution for enforcing image copyright protection [1]. In case of dispute over image ownership-right, the issue can be solved by extracting the embedded hidden information.

As a general rule, digital watermarking scheme should satisfy three main requirements which are imperceptibility, capacity and robustness [2]. Imperceptibility is a measure of the similarity of the watermarked image and the cover image. Capacity refers to the maximum number of bits which ca be hidden in a given cover image without inducing noticeable artifacts. Robustness describes how well watermarks survive

malicious and non-malicious attacks. Ensuring the best tradeoff between these three conflicting properties is of paramount importance to the watermarking method to be effective. Increasing the capacity results in a decrease of both robustness and imperceptibility, and vice-versa. The ultimate goal of this work is copyright protection and as such, capacity is not an important issue. In fact, the two main properties we should consider when targeting image copyright protection are robustness and imperceptibility.

Digital watermarking techniques are classified into different categories depending on different parameters, such as the embedding domain, the resistance to attacks, and the extraction scheme. Watermark embedding techniques can be performed using one of two domains: spatial domain or transform domain. In techniques that operates in spatial domain [3], the operation of embedding is carried out by altering the pixels intensity of the image while in transform domain techniques, embedding is done after applying a transformation, like DCT [4], DWT [5], and DFT [6]. In terms of attack resistance, the watermark embedding process can be qualified as robust [7][8], semi-fragile [9], or fragile [10]. The extraction scheme can be non-oblivious, semi-oblivious, or oblivious technique. In non-oblivious methods [11], the original image is required when performing extraction. In semi-oblivious methods, the watermark is essential during extraction. In oblivious methods [12], watermark extraction is carried out using only the secret key.

In [13], an oblivious robust watermarking algorithm that combines DCT and DWT is proposed. First, one-level DWT is performed on the cover image giving four sub-bands: one coarse sub-band (LL_1), a horizontal (HL_1), vertical (LH_1) and diagonal (HH_1) details sub-bands. Next, the HL_1 sub-band is shuffled to increase security. Afterwards, the watermark, which is resized to 32×32 pixels before it is encrypted, is embedded in the 8×8 block size DCT coefficients of HL_1 sub-band coefficients. The authors claim that their method is robust while maintaining good imperceptibility.

In [14], a robust watermarking method for images that uses both DWT and DCT, is presented. Before embedding the watermark, the cover image is first split using a 2-level DWT. Then, HL_2 sub-image is divided into 4×4 blocks. Next, the 2D-DCT is carried out on each block. Watermark embedding is done using two pseudo-random (PRN) sequences, implanted in the mid-band DCT coefficients, in conformity with the watermark bit, 0 or 1. During extraction, the watermarked image is decomposed using 2-level DWT and subsequently, 2D-DCT is applied to each block. Then, correlations between middle band of DCT coefficients and the two PRN sequences are computed and the watermark bit, 0 or 1, is extracted, according to correlation values.

In Hamidi et al. [15], authors have proposed a SIFT-based resilient image watermarking scheme that operates in DWT-DCT space. The technique, which uses the frequency domain, inserts the watermark bits, in a chosen set of DCT coefficients of the HL_1 sub-band, after applying 2D-DWT to the host image. Next, inverse DCT (2D-IDCT) operation is performed on each modified block of HL_1 to generate the modified HL1 sub-band followed by inverse DWT (2D-IDWT) operation to produce the watermarked image. In the end, SIFT features are retrieved from the watermarked image and saved. The extraction mechanism consists of two subsequent stages: geometric attacks correction followed by watermark recovery. In the first stage, SIFT features extraction

from the attacked image is first performed, and then, a matching operation between these features and the features saved during embedding is carried out. Next, the corrected image against RST attacks is generated. In the second stage, the watermarked image is first decomposed using 1-level 2D-DWT into four sub-bands, followed by a 2D-DCT operation on HL_1 sub-band. Next, two PRN sequences are generated using the same key as the one used during the embedding process. Finally, the correlations between middle band of DCT coefficients and the two PRN sequences are computed to determine the watermark bit value, 0 or 1 that was inserted.

In [7], a SIFT-based image watermarking algorithm which combines the two transforms, namely, DWT and SVD, is presented. In the first place, a 3-level DWT (2D-DWT) is performed on the host image. Second, the low-frequency LL3 sub-band is decomposed using the singular value decomposition (SVD), and the watermark is embedded using an additive procedure. RST distortions are rectified by computing the match between the keypoints of the watermarked and the attacked images.

A literature review shows that most digital watermarking techniques are either not tested for geometric attacks [12][13] or susceptible to this type of attacks [16].

This work describes a resilient hybrid SIFT-based approach to image watermarking that uses DFT and DCT. The proposed data hiding technique inserts the watermark in the middle band of DCT coefficients of the DFT magnitude of the cover image. Embedding the watermark in the DFT magnitude ensures a good imperceptibility. Unfortunately, using the DFT alone does not yield a robust system. Therefore, we decided to perform DCT on the DFT magnitude, for its many good properties, mainly its resistance to standard image processing operations. Besides DCT and DFT techniques, SIFT transform is also used in this work, to improve robustness against geometric attacks. This produces a robust method that can resist to both commonly used image processing manipulations like filtering, adding noise, lossy and lossless compression, etc., and geometric manipulations while reducing the effect on imperceptibility.

The remaining of this paper is structured as follows. In section 2, the principle of the SIFT transform [17] is explained. Section 3 gives a detailed description of the proposed algorithm. The performance of our method is assessed and discussed in section 4. Finally, the conclusions are drawn in section 5.

2 SIFT algorithm

The SIFT algorithm first introduced by [17], is an image descriptor that extracts low-level feature points, which proved to be invariant to geometric distortions, i.e., rotation, scaling, and translation. The SIFT transform extracts the feature points from the scale-space of the image, denoted as $L(x, y, \alpha)$ which is given by (1).

$$L(x, y, \sigma) = g(x, y, \sigma) * P(x, y), \;\; g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{(-x^2+y^2)/2\sigma^2} \tag{1}$$

where $P(x, y)$ denotes the digital image, $g(x, y, \sigma)$ is the variable-scale Gaussian function, (x, y) is the pixel coordinates, $*$ represents the convolution operator, and σ is the scale-space parameter, that characterizes the image's smoothness.

To detect SIFT feature points in an effective way, the difference-of-Gaussians (DoG) is calculated as follows.

$$D(x,y,\sigma) = \big(g(x,y,k\sigma) - g(x,y,\sigma)\big) * P(x,y) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (2)$$

where k is a constant, multiplicative factor, between two neighboring scale-spaces and $L(x,y,k\sigma)$ is a scale-space function, which can be used to determine smoothed images at different scales.
In order to determine candidate keypoints of the difference-of-Gaussian function, $D(x,y,\sigma)$, each point is first compared with its immediate neighbors in the current image. It is also compared to the nine neighbors in each of the adjacent scales to determine whether it is a maximum or a minimum. The point is selected only when its value exceeds or is lower than values of all its neighbors. Also, candidate points which are the result of low local contrast (low edge strength) or are poorly localized along an edge are rejected.

Next, one or more directions are assigned to each point based on image gradient directions. Let the gradient magnitude be $GM_{SIFT}(x,y)$, and the orientation of the feature point at coordinates (x,y) be $\theta_{SIFT}(x,y)$ which are computed using, respectively, equations (3) and (4).

$$GM_{SIFT}(\mathrm{x},\mathrm{y}) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2} \quad (3)$$

$$\theta_{SIFT}(x,y) = \tan^{-1}\{(L(x+1,y) - L(x-1,y)) \,/\, (L(x,y+1) - L(x,y-1))\} \quad (4)$$

where L is the Gaussian smoothed image. The peak of the histogram of the orientations in the region of a keypoint is then selected to be the local direction of the feature. The reader can refer to [17] which studies the algorithm's basis in much greater details and outlines factors essential to its performance. In this work, the SIFT algorithm is used to correct geometric attacks to improve robustness of the watermark.

3 Proposed Framework

This proposed method describes an oblivious watermarking approach for images robust to intentional and unintentional attacks. The method that operates in the transform domain, inserts the watermark bits in the twenty-two middle-frequency DCT coefficients of the DFT magnitude of the original image. Choosing DFT magnitude for embedding is motivated by a gain in imperceptibility. Performing DCT on the DFT magnitude of the host image yields a robust scheme since DCT transform is known for its robustness against standard image processing techniques. First, the DFT is performed on the original image to produce magnitude and phase components. Afterwards, the DFT magnitude is decomposed into non-overlapping 8×8 blocks and then 2D-DCT is carried out on each block. Next, a secret key is used to produce two PRN sequences. The first one (PRN_0) is used to encode watermark bit value 0 whereas the second one (PRN_1) is used to encode watermark bit value 1. The embedding of the watermark bits is

performed as in (5), in the case the watermark bit value is 0, and as in (6), in the case the watermark bit value is 1.

$$DCT_{F_M}^{M} = DCT_{F_M} + \delta \times PRN_0 \tag{5}$$

$$DCT_{F_M}^{M} = DCT_{F_M} + \delta \times PRN_1, \tag{6}$$

where DCT_{F_M} denotes the initial middle band DCT coefficients of the DFT magnitude, and $DCT_{F_M}^{M}$ is the marked middle band DCT coefficients of the DFT magnitude. The subscript F_M is used to denote the middle band DCT coefficients of the DFT magnitude. F_M is chosen as the embedding region to obtain better robustness against lossy compression operations, while avoiding noticeable modification of the cover image.

3.1 Embedding Scheme

Fig. 1 presents the watermark embedding mechanism and the details of watermark embedding is described below:

1. Perform DFT on the cover image to generate magnitude and phase matrices.
2. Divide the magnitude into blocks of 8×8, then perform 2D-DCT on each block.
3. Compute two highly uncorrelated PRN sequences PRN_0 and PRN_1 with the help of a secret key. To insert a 0, we use PRN_0 and to insert a 1, we use PRN_1.
4. For each block, embed the two PRN sequences into the DCT middle-frequency coefficients of the magnitude taking the watermark bit value into consideration. To insert a 0, use equation (5), to insert a 1, use equation (6).
5. Apply the inverse DCT (2D-IDCT) to each modified magnitude block.
6. Restore the watermarked image with the altered magnitude and the unaltered phase.
7. Retrieve SIFT features from the watermarked image and store them.

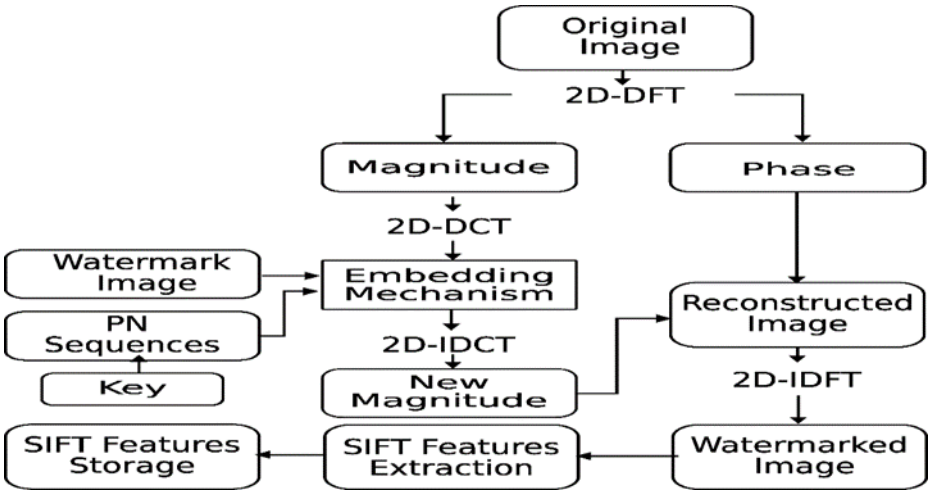


Fig. 1. Block diagram of the watermark insertion mechanism.

3.2 Extracting Scheme

The extracting mechanism is carried out in two stages: geometric attacks correction and watermark recovery. Before proceeding with watermark extraction, we need to rectify the geometrical distortions the attacked image has experienced. For this reason, first, SIFT features are retrieved from the attacked image, then a matching process is operated between them and the SIFT features stored during the embedding stage. This matching operation provides pairs of matching points which can then be used to perform the correction.

To perform rotation attack correction, the attacked image is rotated by a certain angle φ_c. From basic mathematics, the mathematical formula of the correction angle is defined by.

$$\varphi_c = \frac{1}{N}\sum_{j=1}^{\mathrm{N}} \gamma_j \ , \quad \gamma_j = \arccos\left(\frac{\overrightarrow{m_w n_w}\cdot\overrightarrow{m_r n_r}}{|\overrightarrow{m_w n_w}|\,|\overrightarrow{m_r n_r}|}\right), \tag{7}$$

where N refers to the number of true matching points pairs, $\overrightarrow{m_w n_w}$ and $\overrightarrow{m_r n_r}$ are two vectors formed by two keypoints picked respectively, from the watermarked image and the rotated image. Thus, the attacked image can then be rotated by the angle φ_c to achieve rotation correction.

In the same way, to perform scale attack correction, the attacked image is scaled by a scaling factor χ.

$$\chi = \frac{1}{M}\sum_{l}^{M} \frac{S_{w_l}}{S_{s_l}} \tag{8}$$

where S_{w_l} and S_{s_l} denote the k$^{\text{th}}$ matching points scale values in, respectively, the watermarked image and scaled image. M is the number of pairs of matching points. The attacked image can then be scaled with the factor χ to perform scaling correction.

Coordinates of matching points are used to correct translation. Let (x_w, y_w) be the watermarked image's coordinates, (x_t, y_t) the translated image's coordinates, and $N \times N$ the dimension of the image. Then, to correct translation attack, new coordinates are computed using equation (9).

$$x_c = \begin{cases} x_t - x_w + N, & x_t < x_w \\ x_t - x_w, & else \end{cases}, \quad y_c = \begin{cases} y_t - y_w + N, & y_t < y_w \\ y_t - y_w, & else \end{cases} \tag{9}$$

Following this, the attacked image can then be corrected by using x_c, and y_c on the horizontal and vertical axes in a plane coordinate system.

The watermark extraction mechanism is depicted in Figure 2.

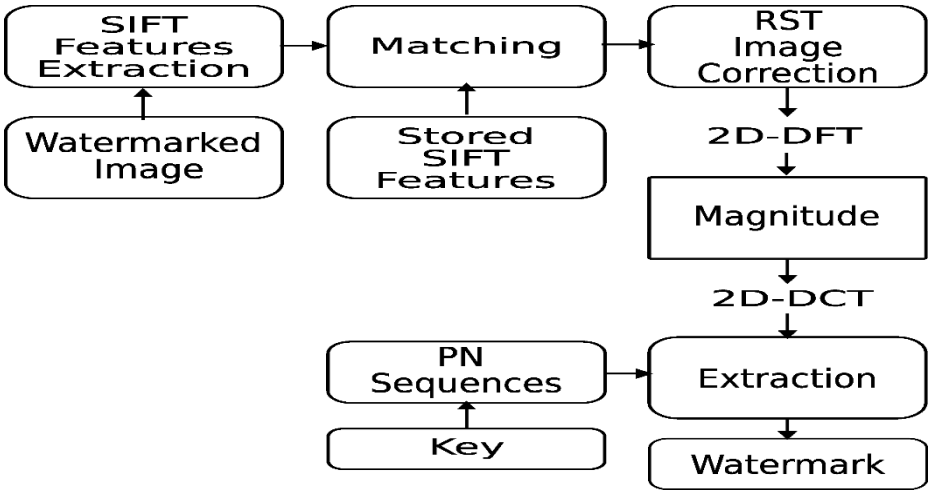


Fig. 2. The proposed extraction mechanism.

The detailed description is given below:

1. Retrieve SIFT features.
2. Perform feature matching.
3. Perform RST corrections.
4. Perform DFT on the watermarked image to generate the magnitude.
5. Produce two PRN sequences (PRN_0 and PRN_1) with the help of the same key as the one used in the embedding process.
6. Perform DCT on the DFT magnitude, then retrieve middle band DCT coefficients.
7. Compute correlations $Corr0$ and $Corr1$ between the middle band DCT coefficients and the two PRN sequences.
8. Recover watermark bit using (10).

$$Watermark\ bit = \begin{cases} 0, & Corr0 > Corr1 \\ 1, & Corr1 > Corr0 \end{cases} \tag{10}$$

4 Experimental Results and Discussion

In this section, performance evaluation in terms of imperceptibility and robustness of our method is presented and discussed. To do so, several experiments were conducted using various grayscale (512×512) images. The images used in these experiments are Baboon, Boat, Cameraman, Lena, and Peppers, as illustrated in figure 3. The watermark used for embedding is a binary logo of size (19×52) as illustrated in figure 4.

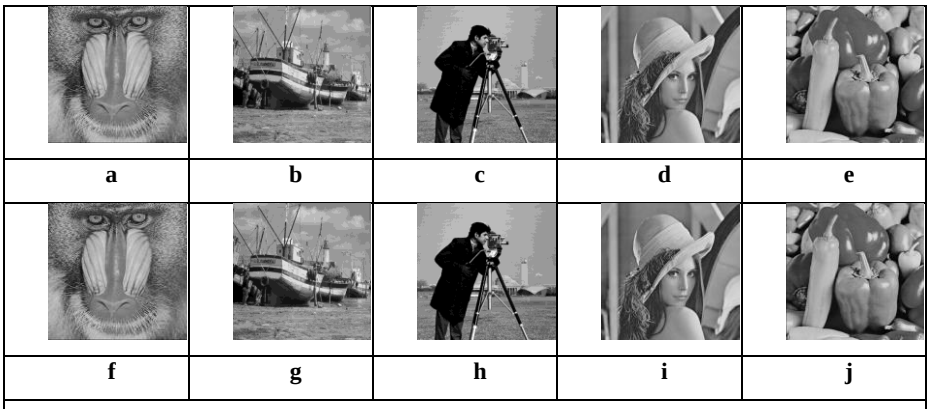


Fig. 3. Original images: **a** Baboon, **b** Boat, **c** Cameraman, **d** Lena, **e** Peppers. Watermarked images: **f** Watermarked Baboon **g** Watermarked Boat, **h** Watermarked Cameraman, **i** Watermarked Lena, **j** Watermarked Peppers.

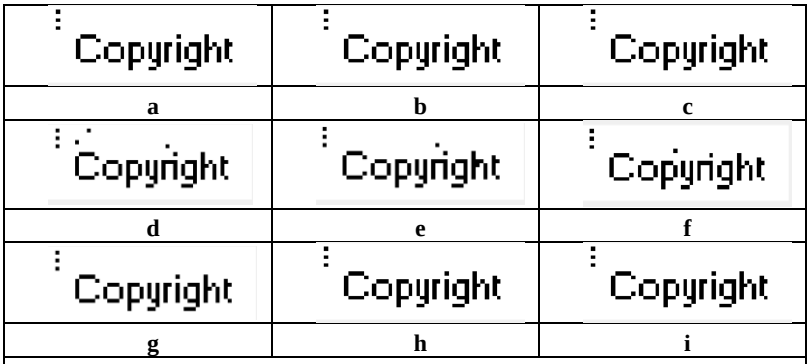


Fig. 4. Extracted watermarks after attacks: **a** Original watermark, **b** Histogram equalization, **c** Salt and pepper noise, **d** JPEG (60), **e** JPEG (65), **f** JPEG (70) **g** JPEG (80), **h** Gaussian smoothing, **i** Sharpening.

To perform the performance evaluation of the proposed method, a comparative study with existing methods is necessary. We compared the robustness and the imperceptibility with four robust watermarking methods [7], [13], [14], and [15]. To perform a fair comparison with other alternative works, another binary logo having size 64×64 is also used as watermark.

4.1 Imperceptibility Assessment

Peak signal-to-noise ratio (PSNR) is the visual quality evaluation metric used in most watermarking processes to assess the imperceptibility performance of the method. For an image P of size M x N, it is computed as in (11).

$$PSNR = 10log10\left(\frac{P_{max}^2}{MSE}\right) \quad (11)$$

where P_{max} denotes the maximum pixel value in the original image P, and MSE is the mean square error between the host image P and the watermarked image $\tilde{P}$. MSE is calculated as in (12).

$$MSE = \frac{1}{M \times N}\sum_{m=0}^{M}\sum_{n=0}^{N}[P(m,n) - \tilde{P}(m,n)]^2 \quad (12)$$

where $P(m,n)$ and $\tilde{P}(m,n)$ denotes the pixel intensity values on the coordinates of row m and column n in the cover image and the watermarked image, respectively.
Table 1 contains the results obtained for all five watermarked images in PSNR (dB) with no attack performed on the images, using a binary (19×52) image as watermark. From the results provided by Table 1 and the images shown in figure 3, it can be verified that our watermarking scheme preserves a good imperceptibly.

Table 2 shows the imperceptibility comparison measured by the well-known PSNR metric between our scheme and the scheme described in reference [13]. It can be concluded from the obtained PSNR values in this table that our proposed method presents good perceptual quality. We presume that these good results are due to embedding the watermark in the middle band DCT coefficients of the DFT magnitude that guarantees high perceptual visual quality. Moreover, results contained in Table 2 shows that our scheme outperforms scheme in [13] for Mandrill, Boat, Cameraman, Lena, and Peppers images.

Table 3 reports the comparison in terms of imperceptibility between our scheme and the scheme described in [13], after several attacks have been applied to the watermarked Lena image. From Table 3, it is clear that our scheme maintains a good imperceptibility against all experienced attacks and gives better results than the approach described in [13].

Table 1. PSNR (dB) for various images using a watermark of size (19×52).

Images	Our scheme
Mandrill	47.4220
Boat	47.9627
Cameraman	48.0531
Lena	48.9985
Peppers	49.9565

Table 2. PSNR (dB) for various images of the proposed method and scheme [13].

Images	Our scheme	[13]
Mandrill	47.4220	47.2718
Boat	47.9627	—
Cameraman	48.0531	47.2724
Lena	48.9985	47.2717
Peppers	49.9565	47.2717

Table 3. Imperceptibility comparison against various attacks for Lena image using PSNR (dB).

Images	Attack Parameter	Scheme [13]	Our scheme
		DWT+DCT	DFT-DCT
JPEG	(QF=50)	45.9736	47.0345
	(QF=80)	44.7711	48.0125
	(QF=90)	47.3282	49.0236
Gaussian filter	(3x3)	47.4099	48.7245
	(5x5)	47.4163	48.8675
Median filter	(3x3)	49.4379	49.9632
Gaussian noise (μ=0.01)	(σ = 0.01)	20.0581	43.0124
	(σ = 0.02)	17.2366	42.8793
Salt and pepper noise	0.01	25.2894	42.1328
	0.02	22.2896	42.0145
	0.04	19.3324	41.9654
	0.05	18.3447	41.7859
Speckle noise	0.01	25.8118	42.8823
	0.02	22.8595	42.6532
	0.04	19.9513	42.3215
	0.06	18.2868	41.9965

4.2 Robustness Assessment

Robustness is a measure of the capacity of a watermark to withstand removal due to malicious and non-malicious attacks. The normalized correlation (NC) is generally used to measure the similarity between two images, and therefore it is adopted in this paper to assess the robustness of our watermarking scheme. It is calculated as follows.

$$NC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} W(i,j) \times W'(i,j)}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}[W^2(i,j)]} \times \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} W'^2(i,j)}} \quad (13)$$

where W and W' denote the original watermark and the extracted watermark respectively. $m \times n$ represents the size of the watermark.

Table 4 gives the robustness comparison between our scheme and the scheme in [13], before attacks. From Table 4, NC values are all equal to 1.

To test the robustness of the method, various attacks were performed on the watermarked image. These attacks are noising attacks (Gaussian noise, Salt and pepper noise), JPEG compression (with different quality factors), image processing attacks (Sharpening, Gaussian filter, median filter, average filter, histogram equalization), and geometric attacks (scaling to different sizes, cropping a small area from the original, rotation with different degrees. The retrieved watermarks from the watermarked and attacked image are shown in figure 4. It can visually be observed from the figure, that despite the attacks, the watermarks are easily recognized.

Table 5 contains the results of robustness compared to schemes in [7, 14, 15]. The comparison has been made for Lena. The results in this Table 5 reveal the good robustness of the proposed method. Also, it can be observed that our method compares favorably with scheme [14] and has similar performance with scheme [7]. On the other hand, one can see from Table 5 that the method in [15] is slightly better than ours.

Table 4. NC values for various images before attacks.

Images	Proposed scheme	Scheme [13]
Mandrill	1.0	1.0
Boat	1.0	—
Cameraman	1.0	1.0
Lena	1.0	1.0
Peppers	1.0	1.0

Table 5. Robustness comparison with schemes in [7], [14] and [15] for Lena.

Attacks	Scheme [7]	Scheme [15]	Scheme [14]	Our scheme
Median filter (3x3)	0.9913	0.9802	0.9032	0.9887
Median filter (4x4)	—	—	0.6387	0.9654
JPEG (10)	—	0.9995	0.3675	0.7321
JPEG (20)	—	0.9994	0.8095	0.7566
JPEG (30)	—	0.9987	0.8943	0.7693
JPEG (50)	—	0.9832	0.9342	0.9785
JPEG (70)	—	1.0	0.9721	0.9896
JPEG (90)	—	1.0	0.9783	1.0
JPEG (100)	0.9966	1.0	—	1.0
Sharpening	—	—	0.9520	1.0
Gaussian filter (3x3)	—	—	0.9117	0.9878
Average filter (3x3)	—	—	0.8432	0.9835
Gaussian noise (0.001)	0.9788	1.0	0.8895	0.9776
Salt & pepper (0.001)	0.9758	0.9907	0.8990	0.9756
Rotation (0.25°)	—	—	0.8235	—
Rotation (0.75°)	—	—	0.7995	—
Rotation (1°)	—	—	0.8390	—
Rotation (2°)	0.9741	0.9998	—	0.9969
Rotation (5°)	0.9813	0.9998	—	0.9927
Rotation (10°)	0.9861	0.9995	0.7817	0.9897
Rotation (30°)	0.9861	0.9987	0.7580	0.9796
Rotation (-0.25°)	—	—	0.8242	—
Rotation (-0.75°)	—	—	0.7986	—
Rotation (-1°)	—	—	0.8394	—
Cropping (25%)	0.9179	1.0	0.9364	0.9213
Cropping (40%)	—	—	0.8731	—
Scaling (25%)	0.9744	0.9831	0.9364	0.9763
Scaling (50%)	0.9919	0.9987	0.8667	0.9896

5 Conclusion

One of the most challenging problems in digital image watermarking is watermark recovery in the presence of geometric attacks like rotation, scaling, translation, and cropping. In this case, in order to re-establish synchronization, an approach based on SIFT in the DFT-DCT space, has been developed in this study. The method combines the advantages offered by DFT and DCT transforms to gain robustness against standard signal processing, such as would occur in a content creation and distribution process, while minimizing the effect on visual quality. At the same time, SIFT descriptor properties allow to achieve robustness against RST attacks. The experiments results and comparisons showed the high robustness of our scheme for image processing operations as well as geometric distortions while keeping a low induced visual distortion.

References

1. Hamidi, M., Chetouani, A., Haziti, M.E., Hassouni, M.E., Cherifi, H.: A robust blind 3-d mesh watermarking technique based on scs quantization and mesh saliency for copyright protection. In: International Conference on Mobile, Secure, and Programmable Networking. pp. 211–228. Springer (2019).
2. Shih, F.Y.: Digital watermarking and steganography: fundamentals and techniques. CRC press (2017).
3. Su, Q., Chen, B.: Robust color image watermarking technique in the spatial domain. Soft Computing 22(1), 91–106 (2018).
4. Das, C., Panigrahi, S., Sharma, V.K., Mahapatra, K.: A novel blind robust image watermarking in dct domain using inter-block coefficient correlation. AEU-International Journal of Electronics and Communications 68(3), 244–253 (2014).
5. Keshavarzian, R., Aghagolzadeh, A.: Roi based robust and secure image watermarking using dwt and arnold map. AEU-International Journal of Electronics and Communications 70(3), 278–288 (2016).
6. Poljicak, A., Mandic, L., Agic, D.: Discrete fourier transform-based watermarking method with an optimal implementation radius. Journal of Electronic Imaging 20(3), 033008 (2011)
7. Zhang, Y., Wang, C., Zhou, X.: Rst resilient watermarking scheme based on dwt-svd and scale-invariant feature transform. Algorithms 10(2), 41 (2017).
8. Hamidi, M., Chetouani, A., Haziti, M.E., Hassouni, M.E., Cherifi, H.: Blind robust 3-d mesh watermarking based on mesh saliency and qim quantization for copyright protection. In: Iberian Conference on Pattern Recognition and Image Analysis. pp. 170–181. Springer (2019).
9. Qi, X., Xin, X.: A singular-value-based semi-fragile watermarking scheme for image content authentication with tamper localization. Journal of Visual Communication and Image Representation 30, 312–327 (2015).
10. Ansari, I.A., Pant, M., Ahn, C.W.: Svd based fragile watermarking scheme for tamper localization and self-recovery. International Journal of Machine Learning and Cybernetics 7(6), 1225–1239 (2016).
11. Saha, B.J., Kabi, K.K., Pradhan, C., et al.: Non blind watermarking technique using enhanced one-time pad in dwt domain. In: Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT). pp. 1–6. IEEE (2014).
12. Fazlali, H.R., Samavi, S., Karimi, N., Shirani, S.: Adaptive blind image watermarking using edge pixel concentration. Multimedia Tools and Applications 76(2), 3105–3120 (2017).
13. Veni, M., Meyyappan, T.: Dwt dct based new image watermarking algorithm to improve the imperceptibility and robustness. In: 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE). pp. 1–6. IEEE (2017).
14. Mingzhi, C., Yan, L., Yajian, Z., Min, L.: A combined dwt and dct watermarking scheme optimized using genetic algorithm. Journal of multimedia 8(3), 299–305 (2013)
15. Hamidi, M., El Haziti, M., Cherifi, H., El Hassouni, M.: A hybrid robust image watermarking method based on dwt-dct and sift for copyright protection. Journal of Imaging 7(10), 218 (2021).
16. Ahmed, M.R., Rahman, M.M., Ahammed, M.S.: A semi blind watermarking technique for copyright protection of image based on dct and svd domain. Global Journals of Research in Engineering 16(F7), 9–15 (2016).
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91–110 (2004).

A Comparative Study of Supervised Machine Learning Models for Cyberattacks using Edge-IIoTset Dataset

Rafika Saadouni[1], Amina Khacha[1], Yasmine Harbi[2], and Zibouda Aliouat[2]

Faculty of Sciences, Ferhat Abbas University, Setif, Algeria
{rafikasaadouni124, khachaamina0}@gmail.com
LRSD Laboratory, Ferhat Abbas University, Setif, Algeria
{yasmine.harbi, zaliouat}@univ-setif.dz

Abstract. Nowadays, millions of users can access any information and communicate everywhere and anytime through the Internet. The volume of network traffic continues to grow, and as technology develops, so do the number of cyberattacks. Intrusion detection is one major research problem in network security, whose aim is to identify unusual access or attacks to secure network traffic. Machine learning (ML) based intrusion detection systems have become widespread in recent years due to their ability to provide embedded intelligence in electronic devices and networks to cope with different security problems. In this study, we investigate and analyze six widely-known supervised ML techniques, including, support vector machine, Naïve Bayes, decision tree, random forest, k-nearest neighbors, and logistic regression. We evaluate the performance of these algorithms using the novel Edge-IIoTset dataset in terms of ten metrics, namely, accuracy, precision, recall, specificity, F1-score, detection time, false detection rate, false positive rate, false negative rate, and false omission rate. All ML models achieve 100% of accuracy and 0% of negative measures in binary classification. The decision tree model has the highest accuracy in multiclass classification.

Keywords: Supervised learning · Intrusion detection · Cyber security · Edge-IIoTset.

1 Introduction

The usage of the Internet has a great commercial and social impact on our daily lives. Internet-connected systems allow users to share resources, services, and information through wireless channels. The tremendous growth of Internet traffic makes such systems luring targets for cyberattacks ranging from simple hacks to well-coordinated security breaches.

Cyberattacks have become more sophisticated with the development of new technologies. Attackers are not only launching flooding and probing attacks but also spreading malware and virus to exploit the software vulnerabilities and

disclose sensitive information. According to Cisco report [2], Trojan was classified as one of the top five malware used to gain unauthorized access.

Various preventive measures such as firewalls, antivirus software, and intrusion detection systems (IDS) are deployed to mitigate malicious activities [6]. Signature-based IDS is the most popular tool for scanning network traffic and has gained commercial success. However, it requires a regular update to add the attack signature and maintenance of the signature database. In addition, it is not suitable for real-time network intrusion detection [21]. Therefore, the IDS has to be effective and new mechanisms are needed to fulfill the requirement of real-time intrusion detection.

Machine learning (ML) is a promising technology that has attracted significant attention in recent years, specifically in intrusion detection [13]. ML-based IDSes have shown considerable performance in the classification of normal and malicious traffic. They require a particular set of features and provide a prediction for the upcoming traffic based on the training data [16]. Several datasets [1, 3–5] were proposed by considering different categories of attacks for intrusion detection. Motivated by this fact, a new real traffic dataset named Edge-IIoTset [10] is analyzed and used to study the effectiveness of various supervised ML techniques.

The following are the primary contributions provided by this work:

- Review of conventional and ML-based intrusion detection systems.
- Description of 6 common supervised ML techniques.
- Analysis of the recently proposed Edge-IIoTset dataset.
- Performance evaluation of ML techniques in terms of 10 well-known metrics in binary and multiclass classifications.

The remainder of the paper is organized as follows. Section 2 provides an overview of IDS. Section 3 introduces the investigated ML techniques. A description of the used dataset and evaluation metrics is provided in Section 4 and Section 5, respectively. In Section 6, we study the performance of each Ml model using the Edge-IIoTset dataset. Section 7 concludes our work.

2 From Conventional to ML-based IDS

Intrusion detection systems have emerged to handle malicious network activities and network policy violations. They are categorized into two main types: host-based and network-based, as presented in Figure 1. A host-based IDS is implemented on individual hosts or devices to monitor the computer system and detect malicious activities (e.g., modification of the system file). A network-based IDS is placed on network points (e.g., routers) to examine the network traffic and identify unauthorized access, anomalous behavior, and attacks [12].

The detection approaches of IDSes can be classified into signature-based, rule-based, anomaly-based, and ML-based, as depicted in Figure 1. A signature-based IDS maintains a database of known attack signatures to identify attacks. A rule-based IDS uses rules to detect known attacks based on expert systems.

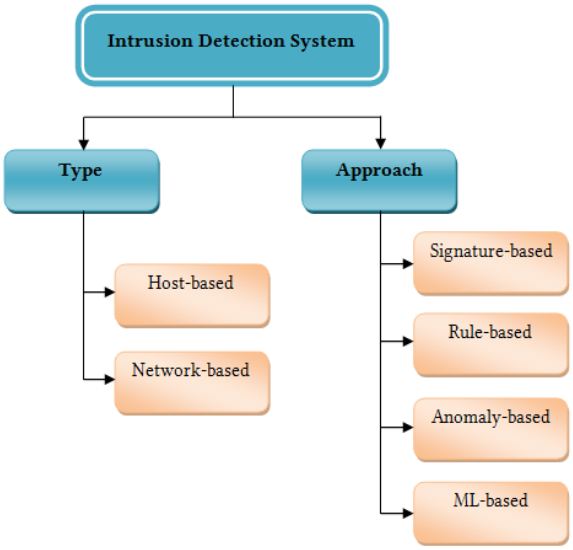


Fig. 1. Intrusion Detection System

An anomaly-based IDS models the normal behavior of the system and requires a regular update of the system to identify unknown attacks. ML-based IDS learns normal and malicious behavior of the traffic flow to classify attacks. There are several advantages of employing ML-based IDS over conventional one [15]:

- Low computational cost.
- Detection of new attacks.
- High detection accuracy.
- Database updates are not required.
- Real-time intrusion detection.

There are three main types of ML algorithms: supervised, unsupervised, and reinforcement learning [8]. The supervised learning models deal with labeled data and predict the class of input data based on the training sample. The unsupervised learning models deduce the description of hidden structures from unlabeled data to classify the input data. The reinforcement learning models continuously learn from previous knowledge to achieve the right decision-making.

3 ML Models

In this section, we describe six common supervised ML techniques, namely support vector machine, Naïve Bayes, decision tree, random forest, k-nearest neighbors, and logistic regression. These algorithms were chosen due to their efficient performance and application in the field of network security.

3.1 Support Vector Machine (SVM)

The SVM is one of the standard ML algorithms used in classification and regression analysis. It works by creating a hyperplane that separates the labeled data into classes, as demonstrated in Figure 2. The goal of the hyperplane is to distinguish each class with a minimum error at a maximum margin using complex data transformation [8, 18].

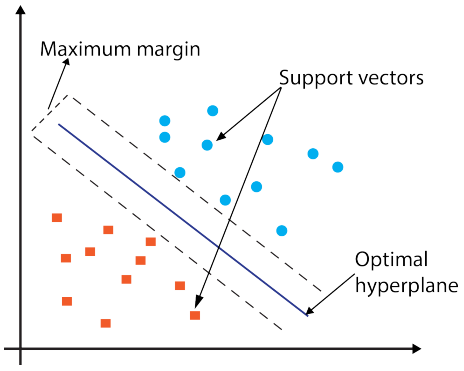


Fig. 2. Support Vector Machine Model

3.2 Naive Bayes (NB)

The NB belongs to the group of probabilistic classifiers since it implements the Bayes theorem for classification problems. The Bayes theorem is a mathematical formula that finds the probability of an event occurring given the probability of another event that has already occurred. The NB is usually used in IoT for anomaly and intrusion detection [7,17]. It requires less data for classification and collects simple per-class statistics from each feature, as shown in Figure 3.

3.3 Decision Tree (DT)

The DT is mainly used to solve both classification and regression problems. It has different branches (i.e., edges) and leaves (i.e., nodes). The algorithm starts from the root node and compares the values of the root attribute with the record attribute (i.e., real dataset). Then, it follows the branch and jumps to the next node to compare the attribute value with other sub-nodes and continues the process until it reaches the leaf node of the tree [14, 20]. Figure 4 illustrates the flowchart of the DT algorithm.

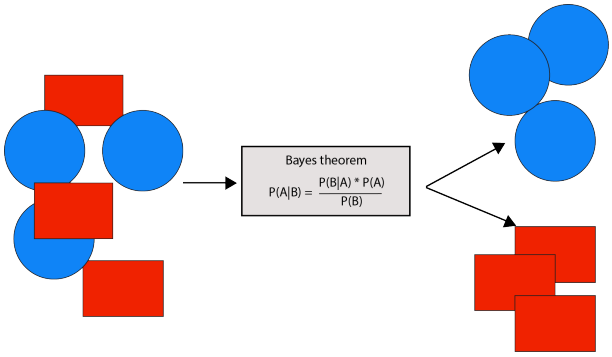


Fig. 3. Naive Bayes Model

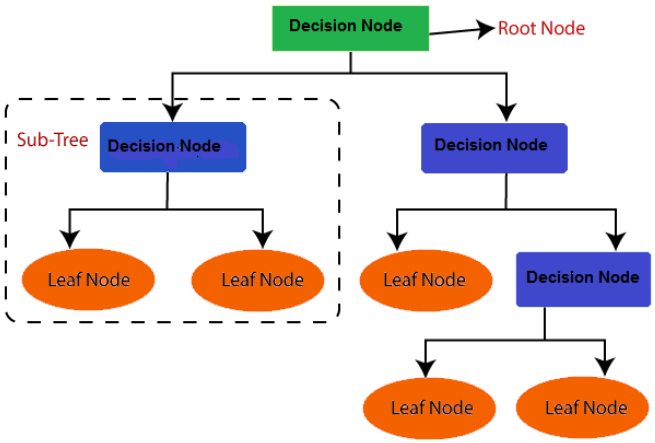


Fig. 4. Decision Tree Model

3.4 Random Forest (RF)

The RF is used to solve regression and classification problems. It is composed of many DTs working in parallel to provide the solution. Each DT works on a random subset of features to calculate the output that will be combined to extract the highest vote and the final result, as shown in Figure 5.

3.5 K-Nearest Neighbors (K-NN)

The k-NN is one of the simplest ML algorithms used for solving regression and classification problems. It assumes the similarity between the new case/data and available cases by calculating the distance between training data and testing data, then taking the k nearest neighbors in each category. Among these k neighbors, it counts the number of the data points in each category and classifies the new case into the category for which the number of the neighbors is

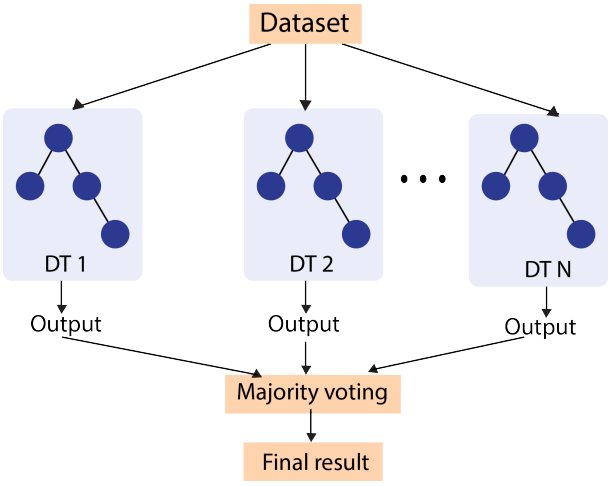


Fig. 5. Naive Bayes Model

maximum [11, 20]. Figure 6 shows the steps of the classification process using the k-NN algorithm.

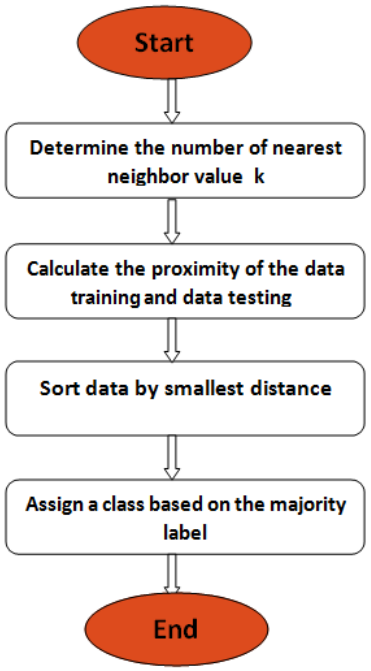


Fig. 6. K-Nearest Neighbors Model [9]

3.6 Logistic Regression (LR)

The LR is one of the most popular ML algorithms that predicts a dependent data variable by analyzing the relationship between one or more existing independent variables [19].

4 Dataset Description

Edge-IIoTset [10] contains real traffic data generated by a large number of IoT devices such as temperature sensors, humidity sensors, heart rate sensors, etc. It consists of 20,952,648 records of five threat classes, including, DoS/DDoS attacks, Information gathering, Man in the middle attacks (MITM), Injection attacks, and Malware attacks. It contains 61 features, as presented in Table 2.

Initially, we filtered the dataset in order to remove redundant lines and unnecessary characteristics. Then, an analysis was carried out to detect NaN or NUL values since they negatively affect the performance of ML models. Table 1 shows the classes of Edge-IIoTset after the filtering process.

Attack category	**Attack name**	**Number of instances**
DDOS	DDoS_UDP	14498
	DDoS_ICMP	13096
	DDoS_HTTP	10495
	DDoS TCP	10247
Injection	SQL_injection	10282
	XSS	9543
	Uploading	10214
MITM	MITM	358
Malware	Ransomware	9689
	Backdoor	9865
	Password	9972
Scanning	Vulnerability_scanner	10062
	Finger_printing	853
	Port_Scanning	8921
Normal		24101

Table 1. Classes of Edge-IIoTset

5 Evaluation Metrics

As mentioned previously, six supervised ML models are employed for intrusion detection based on labeled traffic data. For each model, we evaluate its perfor-

No.	**Feature Name**	No.	**Feature Name**	No.	**Feature Name**
1	frame.time	21	tcp.ack	41	dns.qry.qu
2	ip.src_host	22	tcp.ack_raw	42	dns.qry.type
3	ip.dst_host	23	tcp.checksum	43	dns.retransmission
4	arp.dst.proto_ipv4	24	tcp.connection.fin	44	dns.retransmit_request
5	arp.opcode	25	tcp.connection.rst	45	dns.retransmit_request_in
6	arp.hw.size	26	tcp.connection.syn	46	mqtt.conack.flags
7	arp.src.proto_ipv4	27	tcp.connection.synack	47	mqtt.conflag.cleansess
8	icmp.checksum	28	tcp.dstport	48	mqtt.conflags
9	icmp.seq_le	29	tcp.flags	49	mqtt.hdrflags
10	icmp.transmit_timestamp	30	tcp.flags.ack	50	mqtt.len
11	icmp.unused	31	tcp.len	51	mqtt.msg_decoded_as
12	http.file_data	32	tcp.options	52	mqtt.msg
13	http.content_length	33	tcp.payload	53	mqtt.msgtype
14	http.request.uri.query	34	tcp.seq	54	mqtt.proto_len
15	http.request.method	35	tcp.srcport	55	mqtt.protoname
16	http.referer	36	udp.port	56	mqtt.topic
17	http.request.full_uri	37	udp.stream	57	mqtt.topic_len
18	http.request.version	38	udp.time_delta	58	mqtt.ver
19	http.response	39	dns.qry.name	59	mbtcp.len
20	http.tls_port	40	dns.qry.name.len	60	mbtcp.trans_id
				61	mbtcp.unit_id

Table 2. Features in Edge-IIoTset

mance by visualizing the confusion matrix demonstrated in Table 3. The assessment is provided in terms of accuracy, precision, recall, specificity, F1-score, detection time, false detection rate, false positive rate, false negative rate, and false omission rate, as presented in Table 4.

	Predicted as attack	**Predicted as normal**
Classified as attack	True Positive (TP)	False Negative (FN)
Classified as normal	False Positive (FP)	True Negative (TN)

Table 3. Metrics of confusion matrix

Evaluation metric	Description	Formula
Accuracy (Acc)	Percentage of correct classifications to the total number of records	$Acc = \frac{TP+TN}{TP+TN+FP+FN}$
Precision (Pr)	Percentage of correct attacks classifications to data predicted as attacks	$Pr = \frac{TP}{TP+FP}$
Sensitivity/Recall (Rc)	Percentage of identified attacks to the total number of attacks	$Rc = \frac{TP}{TP+FN}$
Specificity (Sp)	Percentage of correct normal classifications to data predicted as normal	$Sp = \frac{TN}{TN+FP}$
F1-score (F1)	Computed as the weighted average of precision and recall	$F1 = 2 \times \frac{(Pr \times Rc)}{(Pr+Rc)}$
False detection rate (FDR)	Probability of incorrectly rejecting the null hypothesis	$FDR = \frac{FP}{TP+FP}$
False positive rate (FPR)	Probability of falsely rejecting the null hypothesis for a particular test	$FPR = \frac{FP}{FP+TN}$
False negative rate (FNR)	Proportion of positives which yield negative test outcomes with the test	$FNR = \frac{FN}{FN+TP}$
False omission rate (FOR)	Proportion of false negatives which are incorrectly rejected	$FOR = \frac{FN}{TN+FN}$
Detection cost	Total test time divided by the size of the dataset used in the test phase	$cost = \frac{Time_{test}}{Size_{testdataset}}$

Table 4. Evaluation metrics

6 Results and Discussions

In this section, we evaluate and discuss the effectiveness of ML models that are trained and tested using the Edge-IIoTset dataset. Both training and testing processes were performed on Kaggle [1].

6.1 Binary classification

Table 5 depicts the performance of supervised ML models in binary classification. It is obviously observed that all models provide 100% of accuracy, precision, recall, specificity, and F1-score and achieve 0% of negative measures (i.e., FDR, FNR, FPR, and FOR). This is due to the large amount of data collected from different sources in the Edge-IIoTset dataset. Moreover, the dataset pre-processing is performed to remove redundant and unnecessary information and improve the performance of ML models. The LR model is more efficient in terms of detection time compared to other models.

ML model	Acc	Pr	Rc	Sp	F1	FDR	FNR	FPR	FOR	Time
SVM	100	100	100	100	100	0	0	0	0	8.55×10^{-7}s
NB	100	100	100	100	100	0	0	0	0	9.49×10^{-8}s
RF	100	100	100	100	100	0	0	0	0	1.05×10^{-7}s
KNN	100	100	100	100	100	0	0	0	0	1.84×10^{-3}s
LR	100	100	100	100	100	0	0	0	0	1.24×10^{-8}s
DT	100	100	100	100	100	0	0	0	0	1.85×10^{-8}s

Table 5. Binary classification results

[1] https://kaggle.com

6.2 Multiclass classification

The performance results of supervised ML models related to different types of attacks and normal traffic are demonstrated in Table 6. It is noted that all models provide 100% of accuracy, precision, recall, specificity, and F1-score and achieve 0% of negative measures (i.e., FDR, FNR, FPR, and FOR) for MITM and normal classes. The DT model has the highest accuracy with 94.47, 95.94, and 94.45 for DDoS, malware, and injection classes, respectively.

The RF model gives the best accuracy with 95.94 for the scanning class compared to other models. For both DDoS and injection classes, the DT has the lowest FDR with 8.79 and 13.97, respectively. The LR outperforms other models in terms of FDR with 4.04 for the malware class. The NB model gets a reduced FDR with 6.25 for the scanning class. The LR model is the most efficient in terms of detection time.

7 Conclusion

The huge amount of Internet traffic may lead to many complex attacks over the network and raises the need for powerful security mechanisms. An IDS is a fundamental security tool that examines the network traffic and detects suspicious activities. The goal of supervised ML-based IDS is to generate a general representation of known attacks. Moreover, supervised learning algorithms provide high accuracy for detecting well-known attacks. This study investigates the use of a new real traffic dataset to evaluate the performance of six major supervised ML models in terms of several well-known metrics. The obtained results showed that all models achieve the highest performance in binary classification than in multiclass classification. These results can be used for further research works that focus on supervised ML applications for network security.

Our future work will investigate the supervised ML models for attack detection using other relevant datasets. Moreover, it will cover the analysis of unsupervised ML models that deal with unknown traffic.

ML Model	Metric	DDos	MITM	Malware	Normal	Scanning	Injection	Average
SVM	Acc	90.62	100	92.79	100	91.47	88.86	93.95
	Pr	87.92	100	95.49	100	67.48	65.51	86.06
	Rc	81.76	100	65.74	100	67.33	92.77	84.60
	Sp	94.75	100	99.25	100	95.11	87.89	96.17
	F1	84.73	100	77.87	100	67.41	76.79	84.46
	FDR	12.07	0.00	4.50	0.00	32.51	34.48	13.93
	FOR	8.24	0.00	7.61	0.00	4.92	1.99	3.79
	FNR	18.23	0.00	34.25	0.00	32.66	7.22	15.39
	FPR	5.24	0.00	0.74	0.00	4.88	12.10	3.82
	Time	5.81×10^{-3}s						
NB	Acc	86.73	100	87.80	100	86.86	80.80	90.36
	Pr	84.86	100	68.53	100	93.75	50.79	82.99
	Rc	70.28	100	69.01	100	0.74	97.95	73.00
	Sp	94.26	100	92.34	100	99.99	76.57	93.86
	F1	76.88	100	68.77	100	1.47	66.90	69.00
	FDR	15.13	0.00	31.46	0.00	6.25	49.20	17.00
	FOR	12.60	0.00	7.50	0.00	13.14	0.65	5.65
	FNR	29.71	0.00	30.98	0.00	99.25	2.04	26.99
	FPR	5.73	0.00	7.65	0.00	0.01	23.42	6.13
	Time	1.32×10^{-7}s						
RF	Acc	94.41	100	95.65	100	95.94	93.62	96.60
	Pr	90.53	100	94.84	100	84.29	80.34	91.67
	Rc	92.21	100	82.03	100	83.68	89.07	91.16
	Sp	95.45	100	98.92	100	97.73	94.72	97.80
	F1	91.36	100	87.97	100	83.99	84.48	91.30
	FDR	9.46	0.00	5.15	0.00	15.70	19.65	8.32
	FOR	3.70	0.00	4.18	0.00	2.37	2.71	2.16
	FNR	7.78	0.00	17.96	0.00	16.31	10.92	8.83
	FPR	4.54	0.00	1.07	0.00	2.26	5.27	2.19
	Time	1.92×10^{-7}s						
KNN	Acc	89.46	100	92.07	100	91.08	87.65	93.37
	Pr	82.14	100	81.80	100	67.18	67.97	83.18
	Rc	84.27	100	75.90	100	64.29	71.94	82.73
	Sp	91.78	100	95.95	100	95.18	91.56	95.74
	F1	83.19	100	78.74	100	65.70	69.90	82.92
	FDR	17.85	0.00	17.19	0.00	32.81	32.02	16.81
	FOR	7.13	0.00	5.67	0.00	5.43	7.08	4.21
	FNR	15.72	0.00	24.09	0.00	35.70	28.05	17.26
	FPR	8.21	0.00	4.04	0.00	4.81	8.43	4.24
	Time	9.20×10^{-4}s						
LR	Acc	90.93	100	92.62	100	91.51	88.79	93.97
	Pr	88.54	100	95.95	100	65.59	65.08	85.86
	Rc	82.50	100	65.73	100	67.45	92.06	84.62
	Sp	94.93	100	99.31	100	94.94	87.99	96.19
	F1	85.41	100	78.01	100	66.51	76.25	84.36
	FDR	11.45	0.00	4.04	0.00	34.40	34.91	14.13
	FOR	8.04	0.00	7.89	0.00	4.66	2.14	3.78
	FNR	17.49	0.00	34.26	0.00	32.54	7.93	15.37
	FPR	5.06	0.00	0.68	0.00	5.05	12.00	3.79
	Time	1.22×10^{-8}s						
DT	Acc	94.47	100	95.91	100	95.90	94.45	96.79
	Pr	91.20	100	89.26	100	84.29	86.02	91.79
	Rc	91.58	100	89.38	100	83.95	85.54	91.74
	Sp	95.83	100	97.46	100	97.67	96.61	97.93
	F1	91.39	100	89.32	100	84.12	85.78	91.77
	FDR	8.79	0.00	10.73	0.00	15.70	13.97	8.20
	FOR	3.97	0.00	2.50	0.00	2.37	3.51	2.06
	FNR	8.41	0.00	10.61	0.00	16.04	14.45	8.25
	FPR	4.16	0.00	2.53	0.00	2.32	3.38	2.06
	Time	1.35×10^{-8}s						

Table 6. Multiclass classification results

References

1. Bot-iot dataset https://research.unsw.edu.au/projects/bot-iot-dataset
2. Cisco annual cyber security report. Tech. Rep.
3. Cse-cic-ids2018 dataset https://www.unb.ca/cic/datasets/ids-2018.html
4. Nsl-kdd dataset https://www.unb.ca/cic/datasets/nsl.html
5. Unsw-nb15 dataset https://research.unsw.edu.au/projects/unsw-nb15-dataset
6. Aburomman, A.A., Reaz, M.B.I.: Survey of learning methods in intrusion detection systems. In: 2016 international conference on advances in electrical, electronic and systems engineering (ICAEES). pp. 362–365. IEEE (2016)
7. Almseidin, M., Alzubi, M., Kovacs, S., Alkasassbeh, M.: Evaluation of machine learning algorithms for intrusion detection system. In: 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY). pp. 000277–000282. IEEE (2017)
8. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications surveys & tutorials **18**(2), 1153–1176 (2015)
9. Dewia, A.M.S.I., Dwidasmaraa, I.B.G.: Implementation of the k-nearest neighbor (knn) algorithm for classification of obesity levels. Jurnal Elektronik Ilmu Komputer Udayana p-ISSN **2301**, 5373 (2020)
10. Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L., Janicke, H.: Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. IEEE Access **10**, 40281–40306 (2022)
11. Gohar, F., Butt, W.H., Qamar, U., et al.: Terrorist group prediction using data classification. Work. MultiRelational Data Min. MRDM2003 **10**, 199–208 (2014)
12. Hande, Y., Muddana, A.: A survey on intrusion detection system for software defined networks (sdn). In: Research Anthology on Artificial Intelligence Applications in Security, pp. 467–489. IGI Global (2021)
13. Harbi, Y., Aliouat, Z., Refoufi, A., Harous, S.: Recent security trends in internet of things: A comprehensive survey. IEEE Access (2021)
14. Kingsford, C., Salzberg, S.L.: What are decision trees? Nature biotechnology **26**(9), 1011–1013 (2008)
15. Mishra, P., Varadharajan, V., Tupakula, U., Pilli, E.S.: A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE communications surveys & tutorials **21**(1), 686–728 (2018)
16. Sahoo, K.S., Tripathy, B.K., Naik, K., Ramasubbareddy, S., Balusamy, B., Khari, M., Burgos, D.: An evolutionary svm model for ddos attack detection in software defined networks. IEEE Access **8**, 132502–132513 (2020)
17. Tahsien, S.M., Karimipour, H., Spachos, P.: Machine learning based solutions for security of internet of things (iot): A survey. Journal of Network and Computer Applications **161**, 102630 (2020)
18. Vapnik, V.: The nature of statistical learning theory (1999)
19. Vinayakumar, R., Soman, K., Poornachandran, P., Akarsh, S.: Application of deep learning architectures for cyber security. In: Cybersecurity and Secure Information Systems, pp. 125–160. Springer (2019)
20. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Wang, C.: Machine learning and deep learning methods for cybersecurity. Ieee access **6**, 35365–35381 (2018)
21. Zaman, M., Lung, C.H.: Evaluation of machine learning techniques for network intrusion detection. In: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. pp. 1–5. IEEE (2018)

Combining Resnet 34 with U-net for the Segmentation of retinal blood vessels

Mohamed Elssaleh Bachiri[1,*] Adel Rahmoune [2] Faycal Rahmoune[3]

[1,2,3] LIMOSE Laboratory M'Hamed Bougara University Boumerdes, Algeria
[1] m.bachiri@univ-boumerdes.dz
[2] adel.rahmoune@gmail.com
[3] faycal_rahmoune@yahoo.com

Abstract. Blood vessels eye in the human body give us important information about a person's health, it allows us to identify diseases. Blood vessel segmentation facilitates these operations. In deep learning, especially when convolutional layers are used more frequently within the training model, there is deterioration and divergence from the best results. In this paper, we have treated and solved the gradient problem by proposing a deep learning architecture for the segmentation of the vascular networks of blood vessels in fundus images. This architecture combines residual learning and U-Net. As we know U-net, it consists of two parts, coder and decoder paths, we combined Resnet 34 with U-net as follows. In downsampling where we are extracted features of the image by convolution with a filter in every layer that we define. Here and at this stage, we used Resnet 34 for extracted our features. After extraction of features in the step of downsampling, we go to a stage upsampling with the same number of layers that we used in downsampling, and of course, we apply a concatenation between the patches. Finally, we got the value of recall equal to 0.9794, Accuracy 0.9692, sensitivity equal to 0.7859, specificity equal to 0.9870 and 0.9832 F1-Score for DRIVE (Digital Retinal Images for Vessels Extraction) database, and with STARE (Structuring Analysis of the Retina) database, we got recall equal 0.9961, Accuracy 0.9363, sensitivity equal 0.9335, plus specificity equal 0.9246 and 0.9649 F1-Score. We can apply our model for the segmentation of vessels similar to different elements in the medical field. This work outperforms many previous contributions.

Keywords: Blood vessels segmentation, Convolution Neuron Network, Downsampling, U-Net, Deep Residual.

1 INTRODUCTION

Segmentation of blood vessels helps specialists to recognize diseases such as arteriosclerosis and diabetes, hypertension. The retinal checkup is based on careful observation by the expert. The general structure of the retina is complex because it's content various shapes, branching patterns, and angles. However, the visual inspection of the

vessels is hard and challenging too. Many techniques of retinal blood vessel segmentation are proposed. This paper used a branch from deep learning , called the Convolutional Neuron Network (CNN) for semantic segmentation. Convolutional neural network CNN, case U-net O. Ronneberger [3] gave excellent results in medical imaging segmentation. The method gave us excellent results compared to the different methods before, because it depends on learning to solve a specific problem that the difficulty such as vanishing gradients, this net allow us to concatenate feature map from different levels. We will be inspired by our work through deep residual learning and U-net, In this paper, we use datasets from DRIVE and STARE that contain a group of images of blood vessels in the retina. The model U-net will be hybridized with Resnet 34 architecture as its encoder (downsampling).
This paper contains the introduction, which presents the importance of retinal vascular segmentation in the medical and diagnostic field. Secondly, related works show various contributions to vascular segmentation. In the third part, we will explain our method for segmentation, fourthly we will give the results we obtained and compare them with the previous contributions. In conclusion, we summarize our contribution.

2 RELATED WORK

There are many algorithms and techniques for the segmentation of blood vessels in the retina, these methods are divided into supervised methods and unsupervised methods. In an unsupervised method, there is no training, the algorithm or technique that is used in the processing stage, gave us six major categories: (i) kernel-based techniques;(ii) vessel-tracking Y. A. Tolias [4]; (iii) mathematical morphology based T. Walter [5]; (iv) multiscale approaches; (v) model based;(vi) adaptive local thresholding Joao VB [6]. Supervised learning; the model learns from the examples and data we give it, using manual classification images, then we predict the images for testing. Moreover, Al-Rawi and H. Karajeh, [7] used a genetic algorithm (GA) to find the best parameters for filter (MF) response. Kundu, A, and Chatterjee [8] used a technique of morphological angular scale-space. Jiang, Z, and Yepez, J [9] implemented a rule of global thresholding based on morphological operations. Kavya et al. [10] combined both methods, the ABC mono-objective optimization algorithm with the Fuzzy C-Means (FCM) clustering method it has given significant and good results for the segmentation of retinal vessels. Additionally, Asad et al. [11] improved a method to reduce the time and complexity by relying on the ant colony. Emary et al. [12] it's using a cuckoo search to find the optimal segmentation. We always stay with Emary et al. [13] Suggest a technique that depends on (FPSA) flower pollination search algorithm and (PS) pattern search, The first algorithm (FPSA) finds the entire blood vessel, then the algorithm (PS) follows it to find the thin vessels which the first algorithm did not recognize.
In this paper, we will use a deep residual U-Net that has not been used before for blood vessel segmentation, we will apply this method to the database DRIVE and STARE, and we will discuss the results obtained and compare them with the methods used before.

3 OUR RESEARCH METHOD

We propose the deep ResUnet, by integrating Resnet 34 with U-net. As we know U-net is recommended by [3], for cell segmentation, which has given excellent results and success. O. Ronneberger and his group, decided to make this network because the previous model that was done by Ciresan et al. [14] which was allowing us to predict the class of each pixel by training the network, they obtained excellent results in the EM segmentation challenge at ISBI 2012.
Obviously, Ciresan's model had two major flaws. First, there are a lot of overlapping corrections and slow execution. Secondly, requires more max-pooling layers that reduce the accuracy. U-net works with fewer images and gives us accurate segmentations. In Figure 1, we can see the visualization of U-Net architecture, The U-shape of architecture is the reason behind its name, and the building blocks of the method and each operation, convolution, max pool, downsampling, upsampling, feature space, etc, are given below.

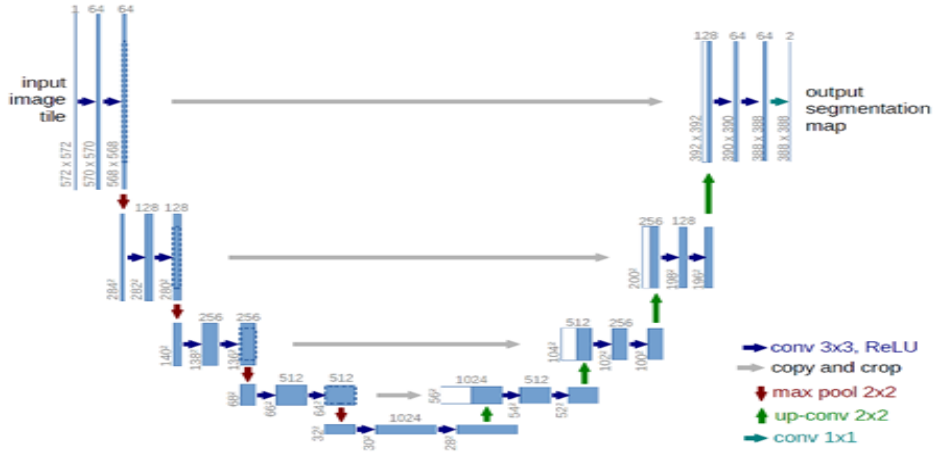


Fig. 1. The U-Net architecture.

The architecture contains two paths, encoder, and decoder, the path on the left, is called the contraction path (encoder path) or downsampling, and on the right, it is called the decoder path or upsampling. Downsampling is the operation, where the output represents information from the input for other steps. Upsampling has the same structure as downsampling but in the opposite direction. It is the network, which takes the output features from downsampling as input and provides the same match of the actual input. The operation of the encoder allows for reducing the size input matrix and increasing the feature maps, and on the contrary, the decoder path is returning the matrix to its original size by unpooling and reducing the feature maps. The advantage of U-net increases images quality and resolution by replacing pooling with upsampling operators because we combined high-resolution features from the downsampling with upsampling output. The U-net allows the user to do the segmentation by an overlap-tile strategy (pixel induction by mirroring the input image). The network gives us good and important results if we have a large number of images, so we have augmented our data

DRIVE and STARE database by application deformations such rotate images, flipping, and adding noise with different parameters. These operations allow us to give multiple examples and are easy to learn. This difference in tissues is widely seen in medical images.

If we add the number of layers and go deeper, the effectiveness will increase of a multi-layer neural network, but not all systems are easy to optimize. However could hinder the training, and might be a degradation problem. To find a solution to this problem, He et al. [19] where give us a model with no higher training error and solve the degradation problem.

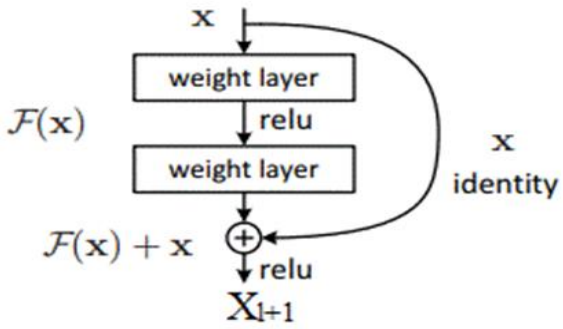


Fig. 2. Residual learning blocks.

The residual network is composed of a series of blocks, as it generally appears in the following equations:

$$Pl = t(xl) + F(xl, Wl) \qquad (1)$$

$$Xl + 1 = f(Pl)$$

Where F() is the residual function, f(Pl) is function activation. Xl+1 the input and output of the l-th residual unit, t(xl) is a identity mapping function t(xl) = xl.

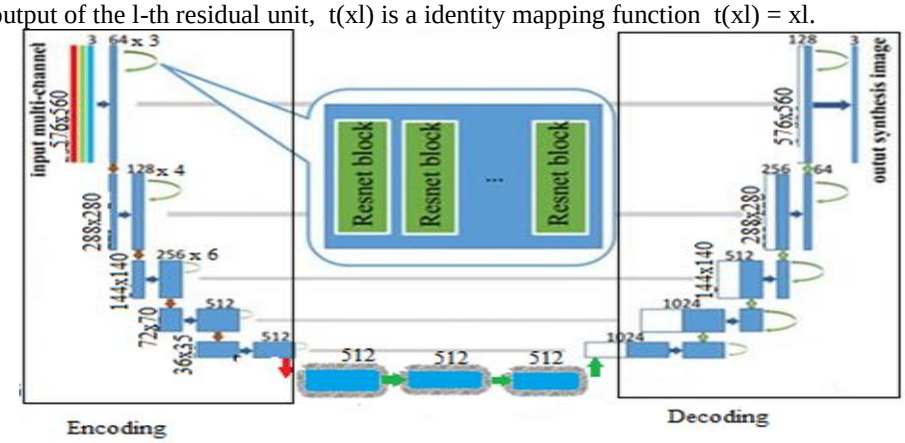


Fig. 3. The architecture of Resnet +Unet.

We have proposed this model Resnet+Unet for semantic segmentation, which consists of strengths of both residual neural network and U-net. The main features of this combination are: Facilitate work of the network by the residual unit; residual connections with low and high levels will advance diffusion of information without degradation, which gives us minimal parameters, which gives us better performance.

In this paper, we used Resnet 34 + U-net for blood vessel extraction, as shown in fig. 3. The network has two parts: encoding and decoding. First, we encoded our image input into a consolidated representation. In the second stage, we retrieved the representations to classify pixels, in other words, semantic segmentation. Encoding and decoding are built with residual units and identity mapping plus convolution blocks, which include a BN layer, and a ReLU activation. First off, in the encoding path, we used a pooling operator with a stride of two to decrease the feature map by half. Moreover, on the other side for decoding, we use upsampling of features and concatenation with the feature from the encoding path. Finally, we used a layer of convolution and a sigmoid activation to get the required segmentation. We used 32 convolutional layers in the encoding path, as the Resnet 34 contains, without 2 convolutional layers (fully connected and a 1 × 1 convolution and a sigmoid activation layer) that is used during we classification by Resnet 34. In this paper, we will focus on loss and accuracy, Sensitivity, and Specificity to measure model performance. The loss function it quantifies the error between the output of the algorithm and the given target value. The goal is to make this value as minimal as possible. We use Loss Dice as the loss function:

$$DL\ (Y,\hat{Y}) = 1-(2\ Y,\ \hat{Y}+1)/\ (\ Y+\hat{Y})\ +1) \qquad (2)$$

We use the Adam optimization algorithm, which will reduce the value of the error function. Moreover, accuracy it's the value of convergence between predicted and actual value. To get accuracy, we use the confusion matrix, which is detailed in Table 1.

Table 1. Confusion matrix

Actual Value	**Predicted Value**	
	True	False
True	(True Positive)	FN (False Negative)
False	FP (False Positive)	TN (True Negative)

We used two databases to train our model. DRIVE and STARE. DRIVE means Digital Retinal Images for Vessel Extraction. DRIVE consists of 40 images, 20 for training and 20 for tasting, type of images RGB with the size of 565 × 584 pixels. STARE means Structuring Analysis of the Retina it contains 20 images, 10 of which are images of a normal retina, a type of images PPM with the size of 700 × 605 pixels.

4 SEGMENTATION RESULTS

4.1 IMPLEMENTATION DETAILS

We have used environment Tensorflow 2.0.0, Keras 2.2.4, and NumPy libraries to operate the network. All the images are taken as 576x560 size in the DRIVE database, and we take 688x592 with STARE without the operation resizing to not lose important details in the images. Our model is trained with a batch size equal to 4, on an NVIDIA GeForce GTX 1080 Ti GPU, and i7-7700K CPU @ 4.20GHz (8 CPUs).

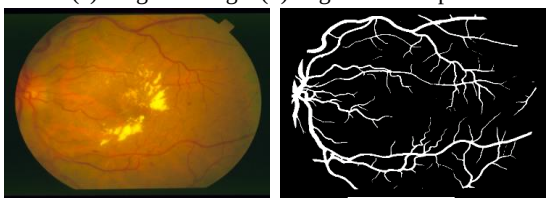
The visualization of segmentation results it's described in figures 4, 5, 6, and 7.

(a) (b)

Fig.4. The first test results on DRIVE
(a) Original image. (b) Segmented output

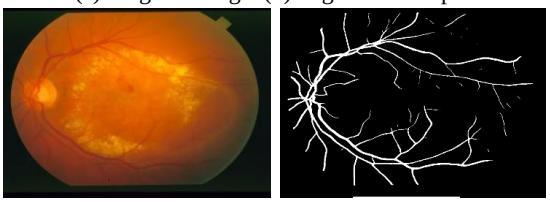
(a) (b)

Fig.5. The second test results on DRIVE
(a) Original image. (b) Segmented output



(a) (b)

Fig.6. The first test results on STARE
(a) Original image. (b) Segmented output



(a) (b)

Fig.7. The second test results on STARE
(a) Original image. (b) Segmented output

The predictions that we obtained after applying our model to the databases show us, that in Figures 4 and 5 the segmentation result was excellent as we can observe the intensity of convergence in DRIVE. Figure 6 the main vessels are clearly visible, but in Figure 7 we notice cuts in the thin vessels and some of them are not clear.

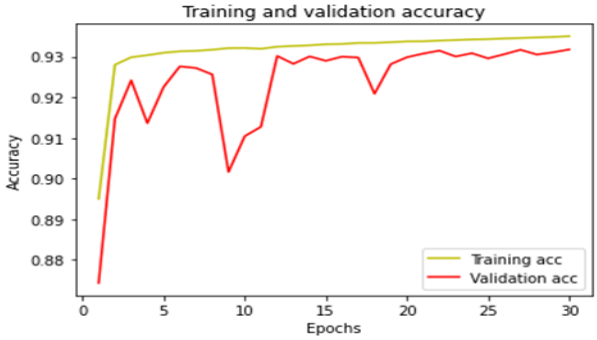


Fig.8. Curve of accuracy from Resnet34+Unet applied in DRIVE database.

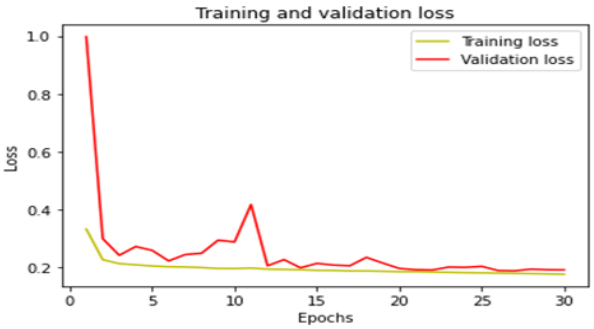


Fig.9. Curve of loss from Resnet34+Unet applied in DRIVE database.

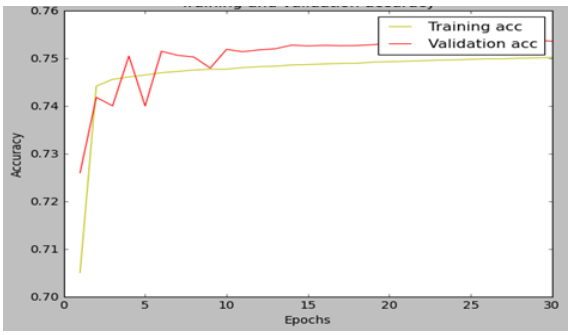


Fig.10. Curve of accuracy from Resnet34+Unet applied in STARE database.

Fig.11. Curve of loss from Resnet34+Unet applied in STARE database.

Figures 8, 9,10, and 11 demonstrate the accuracy and loss for DRIVE and STARE in order, we can see the training process with 30 epochs, it gives us the model a minimum loss value and maximum accuracy, and then loss, the accuracy of the proposed model are faster convergent and stable with smoother movements.

Table 3. Comparison of the proposed with other methods

Data	**DRIVE**			**STARE**		
Method	sens	spec	acc	sens	spec	acc
Budai [15]	0.759	0.968	0.949	0.580	0.982	0.938
Marin [16]	0.706	0.980	0.945	0.694	0.981	0.952
Joes. S [17]	0.719	0.977	0.944	0.694	0.981	0.952
Soares [6]	0.733	0.978	0.946	0.720	0.975	0.948
Mendonc [18]	0.734	0.976	0.945	0.699	0.973	0.944
Resnet+Unet	**0.987**	**0.785**	**0.969**	**0.924**	**0.933**	**0.936**

Table 3. Cont.

Data	**DRIVE**		**STARE**	
	Recall	F1-Score	Recall	F1-Score
Proposed	**0.9794**	**0.9832**	**0.9961**	**0.9649**

We can conclude from the values of recall and F1-Score in the previous table 3 we have successfully performed a vascular network segmentation. Also, the values we obtained for accuracy, sensitivity, and specificity it was high compared to other contributions.

5 CONCLUSION

We can conclude that this model has contributed to solving one of the most common problems that occur in convolutional layers, especially when we increase the depth, as we did not encounter the problem of deterioration because of the algorithm we followed. We have transformed the working principle Resnet 34 from classification work to segmentation with problem degradation solving. The results obtained were among the best and we can rely on this model for retinal vascular segmentation. For future research, we can use more architecture such as AlexNet, and LeNet as encoder-decoder to get more alternative models.

REFERENCE

1. V. Jaiswal, V. Sharma, S. Varma. (2019) 'An Implementation of Novel Genetic-Based Clustering Algorithm for Color Image Segmentation', TELKOMNIKA Telecommunication Computing Electronics and Control, vol 17, no 3, pp 1461- 1467, June 2019.
2. P. B. Prakoso and Y. Sari. (2019) 'Vehicle Detection using Background Subtraction and Clustering Algorithms', TELKOMNIKA Telecommunication Computing Electronics and Control, vol 17, no 3, pp 1393-1398, June 2019

3. O. Ronneberger, P. Fischer, and T. Brox. (2015) 'U-Net: Convolutional networks for biomedical image segmentation', in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2015, pp. 234–241
4. Y. A. Tolias and S. M. Panas. (1998) 'A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering', IEEE TMI, vol. 17, no. 2, pp. 263–273.
5. T. Walter and J. Klein. (2001) 'Segmentation of color fundus images of the human retina: Detection of the optic disc and the vascular tree using morphological techniques', Paper Presented at the International Symposium on Medical Data Analysis. Springer, 2000 pp. 282–287.
6. Joao VB Soares, Jorge JG Leandro, Roberto M Cesar, Herbert F Jelinek, and Michael J Cree. (2006) 'Retinal vessel segmentation using the 2-d Gabor wavelet and supervised classification', IEEE TMI, vol. 25, no. 9, pp. 1214–1222, 2006
7. M. Al-Rawi, H. Karajeh, Genetic algorithm matched filter optimization for automated detection of blood vessels from digital retinal images, Computer methods and programs in biomedicine 87 (3) (2007) 248-253.
8. Kundu, A and Chatterjee, R.K. (2012) 'Retinal vessel segmentation using Morphological Angular ScaleSpace'. In Proceedings of the 2012 Third International Conference on Emerging Applications of Information Technology, Kolkata, India, 30 November–1 December 2012; pp. 316– 319.
9. Jiang, Z.; Yepez, J.; AN, S.; KO, S. (2017) 'Fast, accurate and robust retinal vessel segmentation system'. Biocybern. Biomed. Eng. 37, 412–421.
10. Kavya k, Dechamma m.g, Santhosh kumar b.j, extraction of retinal blood vessel using artificial bee-colony optimization, Journal of theoretical and applied information technology,88 (3) (2016).
11. A. Asad, A. T. Azar, N. El-Bendary, A. E. Hassaanien, et al., Ant colony based feature selection heuristics for retinal vessel segmentation, arXiv preprint arXiv:1403.1735.
12. E. Emary, H. M. Zawbaa, A. E. Hassanien, G. Schaefer, A. T. Azar, Retinal vessel segmentation based on possibilistic fuzzy c-means clustering optimised with cuckoo search, in: Neural Networks (IJCNN), 2014 International Joint Conference on, IEEE, 2014, pp. 1792–1796.
13. E. Emary, H. M. Zawbaa, A. E. Hassanien, B. Parv, Multi-objective retinal vessel localization using flower pollination search algorithm with pattern search, Advances in data analysis and classification (2016) 1–17.
14. Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS. pp. 2852{2860 (2012).
15. Budai, A.; Michelson, G.; Hornegger, J. (2010). 'Multiscale Blood Vessel Segmentation in Retinal Fundus Images'. In Proceedings of the Bildverarbeitung für die Medizin, Aachen, Germany, 14–16 March 2010; pp. 261–265
16. Marin, A. Aquino, M. E. Gegundez-Arias, J. M. Bravo. (2011) 'A new supervised method for blood vessel segmentation in retinal images by using gray level and moment invariants-based features'. Medical Imaging, IEEE Transactions on 30 (1) pp.146–158.
17. Joes Staal, Michael D Abramoff, Meindert Niemeijer, Max A Viergever, and Bram van Ginneken. (2004) 'Ridgebased vessel segmentation in color images of the retina', IEEE TMI, vol. 23, no. 4, pp. 501–509, 2004.
18. M. Mendonca and A. Campilho. (2006) 'Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction'. Medical Imaging, IEEE Transactions on 25 (9) pp.1200-1213.
19. K. He, X. Zhang, and S. Ren, ''Deep residual learning for image recognition,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–6.

Enhanced Flip in 3D Delaunay Triangulation

Z. Tchantchane[1,2], and O. Azouaoui[1], N.Goualmi[2], M. Bey[1]

[1] *Centre de Développement des Technologies Avancées (CDTA), Division Productique et Robotique (DPR), BP 17, Baba Hassen, Algiers 16303, Algeria*

[2] *Université Badji Mokhtar, Faculté des Sciences de l'Ingénieur*, Département d'Informatique, Laboratoire Réseaux et Systèmes, *BP 12, Annaba 23000, Algeria*
ztchantchane@cdta.dz, tarabet.zahida@gmail.com

Abstract. The reconstruction of Computer Aided Design (CAD) model of objects based on a points cloud has become a very common practice in the industrial world. This process is arduous and time consuming since it depends on CAD software functions and requires high interactivity between software and designer. To shorten this reconstruction cycle, the points cloud is approximated by tetrahedron. For this purpose, several methods are proposed in the literature. Delaunay triangulation method is chosen for its property of uniqueness, even it is delicate and time consuming, especially if it deals with non-structured points clouds. This paper proposes a methodology to generate a 3D triangulation approximating objects from non-structured points cloud. This methodology is based on Delaunay triangulation method in 3D by local modification (Split and Flip). It consists of determining the starting tetrahedron used to find the shortest path to the tetrahedra containing the inserted point and thus minimizing the search area. Subsequently, the flip operation is executed by locating invalid tetrahedron using neighboring features, which has the effect of decreasing the triangulation update area. At the end, a comparative study is established to assess the performances of the proposed approach.

Keywords: Non-structured points cloud, Split and Flip, Delaunay triangulation.

1 Introduction

The direct process of constructing an object starts with CAD (Computer-Aided Design) where the object is designed and then transmitted to CAM (Computer-Aided Manufacturing) for manufacturing to obtain the desired part. In case of duplication of an existing part (part reconstruction), another path is followed. Indeed, the process starts by the digitization phase, followed by the reconstruction phase and then moving to CAD and CAM phase to obtain the duplicated part.
The universal process of model reconstruction begins by digitizing the object to obtain several points cloud that will be subsequently adjusted. The second phase, decimation,

consists of simplifying the points cloud. After that, the next phase, segmentation, consists of dividing the object into regions. At the end, the definition of the continuous model is generated.
The reconstruction of models based on a points cloud acquired by scanning real objects on 3D Measuring Machines "CMM" became a very common practice in industrial world. It is worth to mention that the segmentation and definition of the continuous model are very delicate and time-consuming steps. For this reason, the process of models reconstruction may take another way that starts from a points cloud representing the object and then connecting the model points by simple geometric shapes (triangles) to generate the triangular facets that tile the object surface (STL model). Since STL model is a data exchange format between CAD and CAM, it will be transferred to CAD and CAM to obtain the object that closely represents the target objet. In the literature, several reconstruction methods were carried out and in a general way, they are classified into two categories, implicit methods and explicit methods. The latter is the most commonly used, since they are mainly local geometrical approaches based on Delaunay triangulation or Dual Voronoï diagram. After a study on main advantages and drawbacks of several reconstruction algorithms, the Delaunay triangulation method is the most suitable with three kinds of algorithms: divide-and-conquer, sweep line and incremental insertion. Generally, the incremental insertion algorithm is much simpler to implement and has the advantage of generalizing to higher dimensionality. The algorithm adds points into the existing triangulation one-by-one and updates the triangulation in a reasonable time. The implementation is relatively simple, but runs slowly. This work deals with the generation of a 3D triangulation from any points set. This latter use the incremental insertion algorithm. The main contributions of this paper is twofold. First, a fast and efficient approach for localizing the point to add is proposed in case of non-structured cloud points. Second, a strategy is developed to update the tetrahedron meshes in better conditions in order to improve the efficiency of the incremental Delaunay triangulation for non-uniformly distributed points cloud data.
The remainder of this article is organized as follows. Section 2 reviews the related research works on Delaunay triangulation. Section 3 describes the design and implementation of the proposed methodology. Section 4 presents and discusses the experimental results and then compares results with those obtained in [19]. Finally, Section 5 presents limitations and future works.

2 Related work

Many research works studied the main pros and cons of the several reconstruction algorithms such as in [1], In a general way, surface reconstruction methods are classified into four classes: (i) explicit form, (ii) implicit form, (iii) computer vision and (iv) soft computing. Explicit form is able to represent faithfully the surface compared to implicit form [3]. As indicated in [2], two different types of explicit surfaces exist. First, parametric surfaces are topologically limited by the initial model; this means that complex surfaces are not easily represented. Second, triangulated surfaces are the most intuitive version of surface representation; here, the surface is described by triangles connected

from the input points. This justified the development of Dual Voronoï diagram and Delaunay triangulation algorithms. Delaunay triangulation method seems to be the most suitable. Three types of algorithms are commonly used to build Delaunay triangulations: (i) divide-and-conquer based algorithms, (ii) sweep line algorithms and (iii) incremental insertion algorithms.

As indicated in [4], the fastest algorithm is based on divide-and-conquer. Since this algorithm is generally designed to be used for parallelism, several approaches are thus developed such as in [5]. Here, the points cloud is recursively divided into sub-regions; each is assigned to a processor. Independently, these regions are further triangulated simultaneously and merged into one domain. Merging the sub-regions into the final mesh still represents a connection problem, especially for Delaunay triangulation of non-structured points cloud. The merge details are complex and hard to be implemented especially in 3D.

To overcome the merging stage, other approaches are developed, where points are assigned to each processor. In [8], authors presented the 3D Parallel Optimistic Delaunay Meshing (PODM) generation method for polyhedral domains. In [6], Lo proposed a parallel insertion scheme by zonal subdivision, in which points are partitioned into cells which are then grouped into zones for parallel insertion. When all processors complete the Delaunay triangulation of assigned points, an overall Delaunay mesh is obtained without the need for complex merging of sub-meshes constructed by the different processors. Although the performance of parallel strategies is better than that of serial ones, the realization depends on a many-core CPU or GPU. Multi-core CPU have complex hardware requirements, and ordinary computers usually have an insufficient number of cores (about 8 to 16); therefore, the improvement in construction speed is limited in most cases.

The second type is sweep line algorithms; they are also used to construct a Voronoï diagram, i.e., dual Delaunay triangulation for points cloud in the plane. Its idea is that points in the plane are sorted horizontally (or vertically); then, the next point is incrementally extracted from the points set and new triangles associated with this point are constructed [9]. The implementation of this method has gradually been simplified over the years, but still remains of significant programming complexity. For this reason, a number of researchers continue to implement simpler methods.

The third type and the simplest is incremental insertion algorithm. It is effective, simple and does not require complex post-processing steps (such as merge step). Additionally, it is much simpler to implement and has the advantage of generalizing to higher dimensionality [10]. This algorithm adds points into the existing triangulation one by one and updates the triangulation in a reasonable time. This implementation is relatively simple, but it runs slowly and its complexity is $O(n^2)$ in the worst case. Guibas et al. [11] improved this algorithm by using a random insertion method, which reduced the complexity to O(nlogn).

Incremental insertion algorithm provides two approaches. The first is given by Rebay [12]; it works by adding points, one at a time, to a valid Delaunay triangulation. After every insertion, any triangle whose circumcircles contain the new point are deleted, thus leaving a star-shaped polygonal hole, which is then, re-triangulated using the new

point. This approach can be extended to 3D while there are particular degenerate cases where the hole is not star-shaped.

Authors in [11] made available another approach for incremental construction "flip". It works by adding points, one at a time to a valid Delaunay triangulation. After every insertion, the approach searches for the triangle that contains the point (localization point); later, this triangle is subdivided into three new triangles. Afterwards, a sequence of flips are used to update Delaunay triangulation. Authors in [13] provided an efficient implementation, the algorithm spent most of the time in localizing point (split step) and updating triangulation (flip step). The key factor influencing the incremental algorithms performances includes finding the tetrahedra where the inserted point is located. Subsequently, tetrahedron meshes are updated by finding the invalid tetrahedron that does not satisfy the Delaunay criteria, and then modified with flips sequences. For point location step, "walking method" is described in [14].

Authors in [15] improved the point location efficiency by dividing grids to record the neighboring tetrahedra. Nevertheless, this approach consumes substantial amount of time building and updating the grids; hence, the overall efficiency is not high. Amenta et al. [16] presented a Biased Randomized Insertion Order (BRIO), which can decrease the frequency of circumcircle judgement. However, it wastes large time on point location, namely, searching the triangle containing the inserted point.

Sloan [17] presented a method of adding points using uniform grids. The time complexity of this approach reached approximately O(n), but the execution time was sensitive to the order of inserted points.

In [18], authors proposed a method based on uniform grid division that traverses divided grids by different space-filling curves (e.g. Hilbert curve) and sorts points in this traversing order. This approach improved upon the performances of the uniformly divided grids method. Nonetheless, the uniform grid method is effective for uniformly distributed points set but not for non-uniformly distributed ones.

Authors in [18] presented k-d tree (short form for k dimensional tree) grid division scheme. Compared with the regular grid insertion scheme, the k-d tree scheme is more efficient for non-uniformly distributed points set. However, since the result of high complication of the k-d tree, constructing k-d tree grids would cost much more time and lead to low efficiency ultimately. On that basis, Lo [7] proposed a multi-grid insertion scheme, that is not only more simple than k-d tree scheme but also can extremely improve the efficiency of Delaunay triangulation for non-uniformly distributed points. Nevertheless, because of the line-by-line grid traversing manner, the multi-grid insertion will generate many conflicting elongated triangles, which costs extra time on constructing and deleting, and thus decreases its efficiency.

The study of the abovementioned works shows that the incremental algorithms are adopted and favored for two reasons. First, parallel algorithms for Delaunay triangulation require the presence of multi-processor computers. Second, merging of sub-regions always represents a stitching problem in the merge step, especially when dealing with non-structured points cloud.

Based on the analysis of the factors influencing the efficiency of the incremental insertion algorithm, the main contributions of this paper is twofold. First, a fast and efficient approach for localizing the point to add is proposed in case of non-structured cloud

points. Second, a strategy is developed to update the tetrahedron meshes in better conditions in order to improve the efficiency of the incremental Delaunay triangulation for non-uniformly distributed points cloud data.

3 Methodology

The proposed solution, "Enhanced bistellar flips algorithms", aims at twofold: (i) a fast and efficient approach to locate the point to be added and (ii) a strategy to update the tetrahedron meshes under better conditions, all in the context of unstructured cloud points data. Figure 1 shows the general structure of the proposed solution.

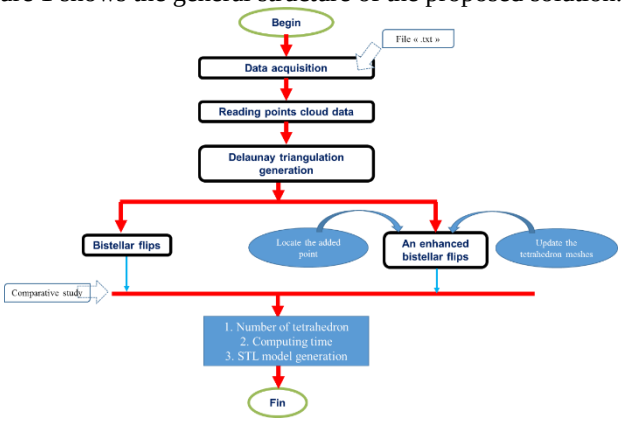


Fig 1: General structure of the proposed solution

3.1 Reading points cloud data

After the acquisition of the file containing the points cloud representing the object, the file is verified syntactically and semantically. In case this file is correct, all the points will be stored in an array of point structure. This operation allows defining the envelope of object defined by its extremum points ($X_{max}, X_{min}, Y_{max}, Y_{min}, Z_{max}, Z_{min}$); consequently, its dimensions (length, height and width) are calculated.

3.2 Implementation of the Delaunay triangulation

This section describes in more detail the implementation of the proposed Delaunay triangulation approach

Data structure

One of the keys to enhancing the performance of a Delaunay triangulation algorithm is to improve the data structure used to store the triangulation. Here, the goal is being to have a structure that is as light and fast as possible for dealing exclusively with 3D triangulations. Implementation is coded in simple C++ language, with arrays of doubles, integers, a point and tetrahedron structure to store the topology and geometry of

the meshes. Point structure is defined by three doubles representing (X,Y,Z) coordinates. An integer index to identify the number of the point in the array, and a Boolean as an indicator of addition to the triangulation. Likewise, tetrahedron structure specified by four integers representing the four points of the tetrahedron and four integers defining its four neighbors. Other structures are established for easy manipulation of data in the implementation phase such as sphere center, plane and cloud

Improved In-Sphere predicate computation (Delaunay criterion)

All algorithms for constructing a Delaunay triangulation are based on quick operations to determine the position of a point relative to a circle circumscribed by a triangle. In 3D, it consists of determining if a given point e is inside, outside or on the circumscribed sphere of the tetrahedron defined by four points (a, b, c, d). This evaluation is done using *In-Sphere predicate* which computes the sign of the following determinant (Equation (1)):

$$\text{Insphere(a,b,c,d,e)} = \begin{vmatrix} a_x & a_y & a_z & \|a\|^2 & 1 \\ b_x & b_y & b_z & \|b\|^2 & 1 \\ c_x & c_y & c_z & \|c\|^2 & 1 \\ d_x & d_y & d_z & \|d\|^2 & 1 \\ e_x & e_y & e_z & \|e\|^2 & 1 \end{vmatrix} \quad (1)$$

It must be mentioned that this operation is time consuming. In order to make it more efficient, we propose to calculate and stock both parameters defining the tetrahedron during its creation (i) center P_{center} and (ii) radius R_{sphere} of the circumscribed sphere. Next, when checking the Delaunay triangulation criterion, the distance is calculated to verify the membership of the point $\boldsymbol{e}$ to the circumscribed sphere (Equation (2)). Certainly, the number of operations to determine the position of a point with respect to the circumscribed sphere becomes negligible compared to the use of *In-Sphere predicate*

$$Distance = \sqrt{(X_e - X_{center})^2 + (Y_e - Y_{center})^2 + (Z_e - Z_{center})^2} \quad (2)$$

Delaunay triangulation process

The general process of Delaunay triangulation by flip is executed in two stages. First, an initial Delaunay triangulation is generated; it consists of one big tetrahedron that encloses the whole of points cloud. Second, a recursive function is applied; points are inserted one at time to the initial Delaunay triangulation. Then, tetrahedra containing the point are determined and subdivided into new four tetrahedra (subsection 3.5). After that, the new tetrahedron must check the Delaunay criterion (subsection 3.6). These two steps are presented in the following paragraphs.

3.3 Initial Delaunay triangulation (big tetrahedron)

In this step, an initial triangulation is created. It is composed of one big tetrahedron that encloses the whole points cloud and satisfies the Delaunay criterion. It is done in five steps:

- Calculate the raw center of the object C_P.
- Calculate the radius of the big sphere R: it is defined by the maximum distance between the center C_P and one of the extremum points.
- Generate the four points on the sphere that allow defining the tangent planes using the spherical coordinates.
- Generate the four tangent planes on the sphere passing through the previously defined points.
- Obtain the four vertices of the enclosing tetrahedron by the intersection between the different planes using the Gaussian pivot method.

Each tetrahedron is characterized by four (04) vertices, four (04) normals, four (04) faces, four (04) neighbors and its circumscribed sphere (center C_P and radius R) which will be used in next steps.

3.4 Point insertion step

This step consists of adding points (one by one) to the Delaunay triangulation in a sequential or a random way. Note that the random mode is the fastest

3.5 Split step

The proposed enhanced Delaunay triangulation starts with localizing the tetrahedron containing the inserted point (target tetrahedron). According to the authors' best knowledge, the localization step depends on the first tetrahedron that will be used (i.e., starting tetrahedron). For this purpose, the starting tetrahedron is selected based on the following three parameters: *width*, *height* and *length* of the raw part. Two steps are used in this stage as follows

Finding the starting tetrahedron

To improve the search of the tetrahedron that contains the added point, a starting tetrahedron is selected. The algorithm starts with the definition of the sphere enclosing the whole points. For this, the zone radius R_{zone}, which represents the maximum value between the dimensions of the raw part, is introduced. Subsequently, the envelope of the sphere is calculated. At this moment, the starting tetrahedron can be identified by testing the overlap of the mesh tetrahedra one by one with the envelope of the sphere. The overlapping tetrahedron will be considered as the starting tetrahedron.

Finding the target tetrahedron

The algorithm1 is proposed to identify the tetrahedron containing the added point based on the starting tetrahedron. The algorithm is defined as follows

Algorithm 1:

- Introduce the coordinates of the inserted point;
- Introduce the starting tetrahedron;
- Build a vector from the center of starting tetrahedron to the inserted point $\vec{V}$;
- Build four vectors $\overrightarrow{S_1}$, $\overrightarrow{S_2}$, $\overrightarrow{S_3}$, $\overrightarrow{S_4}$; each vector is from the center of starting tetrahedron to one normal of its face;
- Calculate the scalar product between $\vec{V}.\overrightarrow{S_1}$, $\vec{V}.\overrightarrow{S_2}$, $\vec{V}.\overrightarrow{S_3}$ and $\vec{V}.\overrightarrow{S_4}$;
- Test the sign of the four scaler products;
- If all scaler products have the same sign, this tetrahedron is the target one;
- If one of the scaler products has a different sign, go to the next tetrahedron in the list;

Algorithm 1 . Target tetrahedron algorithm.

Subdivision of the target tetrahedron

Once the target tetrahedron is known, it is replaced by four new tetrahedron. For each new tetrahedron, normals, faces, neighbors and its circumscribed sphere (Center C_E and Radius R) are identified

3.6 Updating Flip

Several modifications are necessary after inserting the new point. the tetrahedron that does not check the test is considered as invalid, and therefore a Flip sequence will be applied. The details of these steps are given in the following

New tetrahedron test

The new tetrahedron will be tested with its neighbors using the sphere radius already calculated during the tetrahedron creation. Three cases, resulting after calculating the distance between inserted point and the center of the circumscribed sphere of neighboring tetrahedron, are identified:

- The distance is greater than R_{sphere}: this means that the inserted point is outside the sphere and does not need a modification; this tetrahedron is valid.
- The distance is equal to R_{sphere}: the inserted point is on the sphere and requires a coordinates modification; this tetrahedron is invalid.
- The distance is less than R_{sphere}: inserted point is inside the sphere; thus, this tetrahedron is invalid

Bistellar flips

In this stage, invalid tetrahedra are considered and need to be modified. For this aim, each invalid tetrahedron is flipped with its neighbor. Three cases are possible:

- FLIP 2-3 is performed; it consists of replacing these two tetrahedron by three new ones.
- FLIP 3-2 is realized; three tetrahedron are replaced by two new ones.
- FLIP 4-4 is performed; four new tetrahedron are created from the four existing ones

Updating tetrahedron neighbors

When bistellar flips are applied, some tetrahedra are deleted and new ones are created and added to obtain a valid mesh; consequently, the neighbor tetrahedra changed. Therefore, it is necessary to update the neighbors of each tetrahedron. Since this is a very time-consuming task, it is improved by locating the modification zone of the neighbors which is done in two levels (Figure 10):

- First level: creation of a neighbor array that contains the neighbors of each created tetrahedron.
- Second level: creation of a zone array to show the neighbors of each neighbor

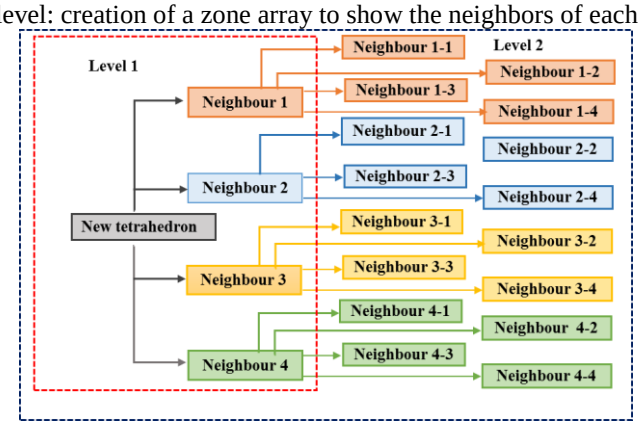


Fig 2: Updating tetrahedron neighbors.

4 Results and discussions

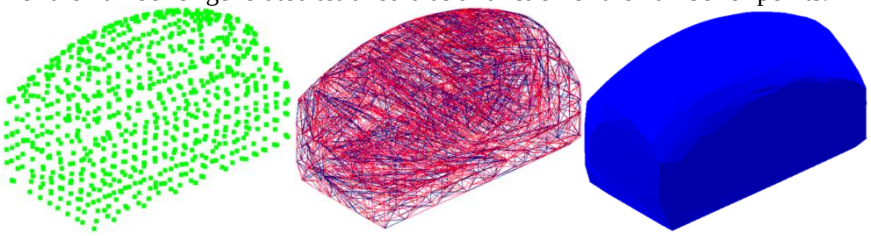
The proposed methodology has been implemented using C++Builder and OpenGL graphics library running on Windows 7. Validation tests are performed on an Intel Core i3 PC with 6 GB of RAM and a resolution of 1366 × 768 pixels. The effectiveness and performances of such a methodology are demonstrated on several samples of *"Convex part model"* (Figure 3.a).

4.1 Obtained results

Figure 3.b shows the triangulation of the "*Convex part model"* with 900 points that generate about 10550 tetrahedra in 16s. At this moment, this triangulation is used to generate the STL models *"* (Figure 3.c).and saved as an *".stl"* file.

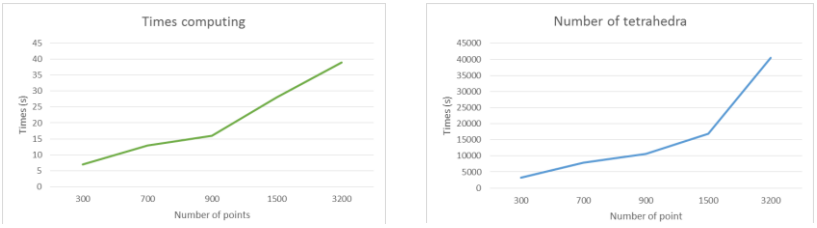
The proposed methodology has been evaluated in terms of two constraints: (*i*) the global computation time of the Delaunay triangulation and (*ii*) the number of generated tetrahedra.

Figure 4.a Figure 4.b show respectively, the evolution of the computation time of Delaunay triangulation as a function of the number of points and illustrates the evolution of the number of generated tetrahedra as a function of the number of points.



a. Points Cloud b. Delaunay triangulation c. STL model

Fig 3: Obtained result of Delaunay triangulation of the "*Convex part model*



a. Computation times b. Number of tetrahedra

Fig 4: Computation times and Number of tetrahedra for "Convex part"

4.2 Comparative study

To assess the performances of the proposed approach, a comparative study is carried between this methodology (Method_1) and that developed in [19] (Method_2).

Figure 6 illustrates the computation times for generating 3D Delaunay triangulation for both algorithms (Method_1 and Method_2).

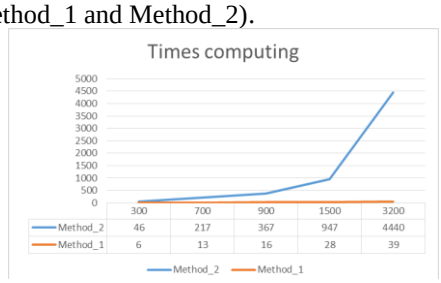


Fig 5: Computation times for Delaunay triangulation using Method_1 and Method_2

Results reveal that computation time is proportional to the number of points. Further, it is low and stable with Method_1 even though points number increases compared to method Method_2.

Another comparison is made about the localization of the inserted point. Table 1 and Table 2 illustrate the iterations number executed to find the tetrahedron containing such a point for both methods (Method_1 and Method_2), respectively. It can be clearly seen that the iterations number for Method_1 is less than that of Method_2. For example, to insert the 7th point using Method_1, the number of iterations required to find the tetrahedron containing this point is equal to four (04); this gives the tetrahedron number 12 (amongst 120 tetrahedra), on the one hand. On the other hand, Method_2 performs 22 iterations to find the targeted tetrahedron (index 21) amongst the other mesh tetrahedra. Accordingly, Method_1 is better than Method_2 since this latter performs the search in all the mesh tetrahedra (one by one). Indeed, the reduced iterations number and the update of neighbors in a restricted area allows minimizing the time spent to compute the Delaunay triangulation despite the fact that points cloud is unstructured and unsorted

Index point / Method	Number of iteration (Method_1)	Index tetrahedron (Method_1)	Number of iteration (Method_2)	Index tetrahedron (Method_2)
1	2	2	3	2
2	3	3	3	2
3	4	15	6	5
7	4	34	22	21
9	5	56	16	15
10	75	66	41	40

Table 1: Target tetrahedron using *Method_1* and *Method_2*

5 Conclusions and future work

This paper presented an enhanced Delaunay triangulation methodology for unstructured point clouds. Its main objective is to propose a fast and efficient approach for localizing the new point to insert; further updating tetrahedra meshes in a restricted area. The proposed methodology has been tested on different samples of Convex part model, and assessed in terms of two criteria (i) global computation time of the Delaunay triangulation and (ii) number of generated tetrahedra. Its effectiveness and performances for localizing the new point to add may be observed thanks to the reduced number of iterations executed to find the target tetrahedron (containing the point). Further, invalid tetrahedra that do not satisfy the Delaunay criterion are located by registering the neighboring tetrahedron to update the mesh. The minimized iterations number and the update of neighbors in a restricted area allow enhancing the triangulation time, especially when dealing with non-structured points cloud. Comparison of obtained results with another method described in [19] demonstrated the superiority of the proposed methodology in terms of both criteria (global computation time and number of generated tetrahedra). This work can be extended to other Delaunay triangulation methods such as divide and conquer.

References

1. S. P. Lim and H. Haron, “Surface reconstruction techniques: A review,” *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 59–78, 2014.
2. A. Khatamian and H. R. Arabnia, “Survey on 3D surface reconstruction,” J. Inf. Process. Syst., vol. 12, no. 3, pp. 338–357, 2016.
3. H. K. Zhao, S. Osher, and R. Fedkiw, “Fast surface reconstruction using the level set method,” *Proc. - IEEE Work. Var. Lev. Set Methods Comput. Vision, VLSM 2001*, pp. 194–199, 2001.
4. M. I. Shamos and D. Hoey, “Closest-point problems,” in *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 1975, vol. 1975-Octob, pp. 151–162.
5. H. Wu, X. Guan, and J. Gong, “ParaStream: A parallel streaming Delaunay triangulation algorithm for LiDAR points on multicore architectures,” *Comput. Geosci.*, vol. 37, no. 9, pp. 1355–1363, 2011.
6. S. H. Lo, “Parallel Delaunay triangulation in three dimensions,” *Comput. Methods Appl. Mech. Eng.*, vol. 237–240, pp. 88–106, 2012.
7. S. H. Lo, “3D Delaunay triangulation of non-uniform point distributions,” *Finite Elem. Anal. Des.*, vol. 90, pp. 113–130, 2014.
8. D. Nave, N. Chrisochoides, and L. P. Chew, “Guaranteed-quality parallel Delaunay refinement for restricted polyhedral domains,” *Comput. Geom. Theory Appl.*, vol. 28, no. 2-3 SPEC. ISS., pp. 191–215, 2004.
9. S. Fortune, Handbook of Discrete and Computational Geometry, Third Edition. Chapman and Hall/CRC, 2017.
10. C. L. Lawson, “Transforming triangulations,” *Discrete Math.*, vol. 3, no. 4, pp. 365–372, 1972.
11. L. J. Guibas, D. E. Knuth, and M. Sharir, “Randomized incremental construction of delaunay and voronoi diagrams,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 443 LNCS, pp. 414–431, 1990.
12. S. Rebay, “Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm,” *J. Comput. Phys.*, vol. 106, no. 1, pp. 125–138, May 1993.
13. B. Joe, “Construction of three-dimensional Delaunay triangulations using local transformations,” *Comput. Aided Geom. Des.*, vol. 8, no. 2, pp. 123–142, 1991.
14. P. J. Green and R. Sibson, “Computing dirichlet tessellations in the plane,” *Comput. J.*, vol. 21, no. 2, pp. 168–173, 1978.
15. H. Borouchaki and S. H. Lo, “Fast Delaunay triangulation in three dimensions,” *Comput. Methods Appl. Mech. Eng.*, vol. 128, no. 1–2, pp. 153–167, Dec. 1995.
16. N. Amenta, S. Choi, and G. Rote, “Incremental constructions con BRIO,” *Proc. Annu. Symp. Comput. Geom.*, pp. 211–219, 2003.
17. S. W. Sloan, “A fast algorithm for constructing Delaunay triangulations in the plane,” *Adv. Eng. Softw.*, vol. 9, no. 1, pp. 34–55, 1987.
18. Y. Liu and J. Snoeyink, “A Comparison of Five Implementations of 3D Delaunay Tessellation In Combinatorial and Computational Geometry,” *Comb. Comput. Geom.*, vol. 52, pp. 439–458, 2005.
19. Z. Tchantchane, M. Bey, K. Azouaoui, and H. Bendifallah, “Triangulation d ‘ un Nuage de Points 3D Non-Structuré par la Méthode de FLIP Triangulation d ‘ un N uage de Points 3D Non-Structuré par la Méthode de FLIP,” no. November, 2018.

An Automatic Deep Learning Mask Detection System for Small Devices

Rachid Djerbi[1], Mohamed T. Bennai[1], Rabah Imache[1], Abdelkrim Halimi[1], Mahdi Touhent[1], and Neil Benahmed[1]

LIMOSE Laboratory, Faculty of Sciences, University of M'hamed Bougara of Boumerdes, Avenue de l'indépendance, 35000, Boumerdes, Algeria
r.djerbi@univ-boumerdes.dz

Abstract. Deep Learning (DL) is a recent branch of artificial intelligence enabling significant progress in many fields, including computer vision. With DL, computer vision solutions are making major advancements such as facial recognition, person and object detection, tracking, and medical imaging. Mask detection solutions are among these DL applications. Since the appearance of Covid19, the prevention of pandemics has become a crucial public health issue. In this context, verifying mask-wearing is critical in response to control virus propagation. In this paper, we present an automatic mask detection tool based on DL. After reviewing some Convolutional Neural Networks (CNN) used for computer vision, we selected the MobileNetV2 model with some adjustments to improve its efficiency. Thus, the obtained model was trained and tested with an open-source dataset we improved. Our model was designed to use few resources, making it able to be deployed on equipment with limited performance. The experimental results demonstrated the proposed model's effectiveness for correctly detecting surgical masks on images.

Keywords: Deep Learning, CNN, Surgical mask, MobilNetV2, ReLU6, LeakyRelu, ELU.

1 Introduction

Following Deep-Leaning (DL) democratization, there has been an increase in interest in artificial intelligence in general and computer vision in particular. Substantial improvements were made in the automated persons and object recognition thanks to the availability of DL methods, the improvements made in hardware, and the easy access to resources. We find mask detection solutions among the solutions using DL for computer vision.

Since the appearance of Covid19, the prevention of epidemics and pandemics has become a crucial public health issue. In this context, the verification of mask-wearing became a key element in response to the control of virus propagation. Automating mask detection solutions would therefore be useful for this purpose. During the last few years, several DL solutions were proposed in the literature related to mask detection issue [23]. These methods can be divided according

to their complexity from the least complex systems (single-stage methods) to more complex ones (Two and multiple stages methods). Therefore, the single-stage methods offer many advantages for automatic mask detection, including their simplicity and limited resource requirements. Thus, Convolutional Neural Networks (CNNs) are one of the most popular techniques in this category [7]. Accordingly, many CNN-based approaches were proposed for mask detection; however, there is still room for improvement.

In this context, we propose a new DL system for mask detection based on the MobileNetV2 model [19]. Our approach aims to improve MobileNetV2 performance by changing its activation function and adding some output layers. Moreover, the resulting neural network was trained on an open-source dataset after being improved using data augmentation. Our experimentations demonstrate the detection efficiency achieved by our approach in comparison with the MobileNetV2 model.

In this paper, we present our work according to the following organization. The background of this research and the related works are presented in Section 2. Then, Section 3 describes in detail the improvements proposed in our approach. In Section 4, we provide information concerning the implementation of our solution and the conducted experiments. Finally, a conclusion and some perspectives are proposed in Section 5.

2 Background and related work

Computer vision has received increasing interest in recent years, especially since the democratization of deep learning. The multiplication of deep learning tools and the availability of resources have allowed significant advances in the automatic recognition of people and objects. The most used methods in this field are based on convolutional neural networks.

2.1 Use of the CNN for image recognition

Convolutional neural networks (also called CNNs) are multilayered neural networks whose connection architecture is inspired by the mammalian visual cortex[7]. Their design follows the discovery of visual mechanisms in living organisms. These CNNs can categorize the information from the simplest to the most complex. They consist of a multilayer stack of neurons and mathematical functions (also known as activation functions) with several adjustable parameters, which preprocess small amounts of information as described in Fig. 1. Convolutional networks are characterized by their first convolutional layers (usually one to three neurons). As its name suggests, a convolutional layer is based on the mathematical principle of convolution. It seeks to identify the presence of a pattern (in a signal or an image) [26].

An Example of image CNN-based image recognition is the use of three different layers. The first convolutional layer detects the contours of objects, whereas

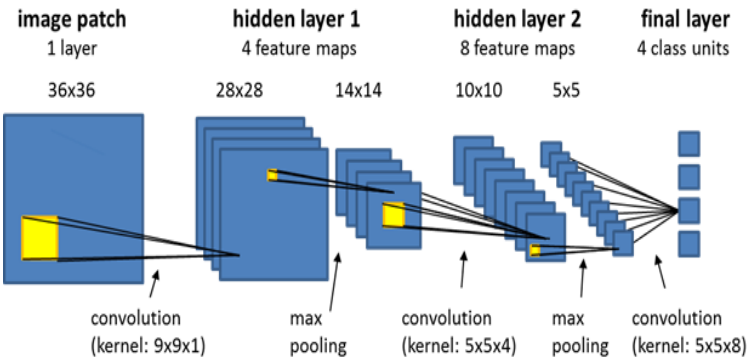


Fig. 1: Convolutional neural network with two hidden layers [21]

the second one combines the contours into objects. Finally, the subsequent layers (not necessarily convolutional) use the previously obtained information to distinguish a car from a motorbike. Therefore, a learning phase is needed, and is based on model's parameters tuning using ground truth image examples. Accordingly, a significant challenge of CNN models is to find methods that adjust these parameters as quickly and efficiently as possible. Hence, convolutional neural networks have many applications in image and video recognition and natural language processing. The elaboration of CNN models for image recognition raises many challenges. One of the critical characteristics of a CNN model is the chosen activation function. The activation function primarily aims to modify data in a non-linear way. This non-linearity enables the modification of data spatial representation. In CNNs, selecting a practical activation function affects the results of the proposed model [2]. In what follows, we will define some DL terms used in the rest of this paper.

Epochs The number of epochs is a hyperparameter specifying how many rounds the learning algorithm will cover the training dataset. One epoch indicates that the inner model parameters had a chance to be adjusted for each sample in the training dataset. Traditionally, the number of epochs is high, enabling the learning procedure to operate until the model error is appropriately reduced [3].

Training loss It indicates how well a deep learning model fits the training data. Thus, it evaluates the model's error on the training set. The total errors number for each sample in the training set is used to determine the training loss computationally [3].

Validation loss It is an indicator of how well a deep learning model performed on the validation set. The validation loss is determined from the total mistakes for each sample in the validation set and is comparable to the training loss [3].

2.2 Related works

Masked face detection is a recent trend in artificial vision, starting from 2021 and the Covid pandemic. Accordingly, most related works were published during the last two years. Those works can therefore be divided into Neural Network-based methods, and hand-crafted feature-based methods (conventional methods) [23]. The latter methods have the major disadvantage of lacking adaptation when dealing with complex scenes and illumination changes. Thus, neural network-based methods are much more suitable for mask detection issues since they can widely adapt when fed with enough balanced training images. Therefore, these methods can be divided as follows:

Single-stage methods: Based on deep learning, these methods use a model to classify images into two classes (masked person and unmasked person), or three (correctly masked, incorrectly masked, and unmasked). Many different DL models were used in this category like Yolo (from V1 to V5) [25,13,18,5,27], InceptionV3 [11], MobileNet [6], ...

Two-stages methods: These methods are divided into two different tasks: Face detection and mask detection task [23]. In these systems, DL can be used in one task [16,1] or both [9]. They, generally, offer better accuracy detection, but at the cost of performance degradation.

Multi-stages methods: They consist of a sequence of different operations used to enhance the effectiveness of the mask-detecting process [12]. One of the most added operations in these systems is human posture detection which can significantly improve the detection results [12]. Increasing the number of operations involved in the mask detection process helps to improve accuracy. However, it also induces an expansion in resources consumption

According to the recent existing works on mask detection, deep learning-based systems offer better results than conventional methods. Therefore, the literature review suggests that multiplying detection stages improves quality detection. However, stages' multiplication increases the need for computational resources. Thus, an effective mask detection system must be as light as possible to be installed on affordable devices and widely used. Consequently, we focus our research on developing a single-stage deep learning mask detection system using MobilNetV2 model. The latter is described in the following section.

3 Our Approach

3.1 Using MobilNetV2 for classification

MobileNetV2 is a CNN for low-powered machines, such as smartphones. This is mainly due to its architecture, where there are two types of blocks[19].

- The first one is a residual block with a stripe of 1 pixel (px)
- The second is a block with a stripe of 2 px, used for size reduction.

These latter contain three layers:

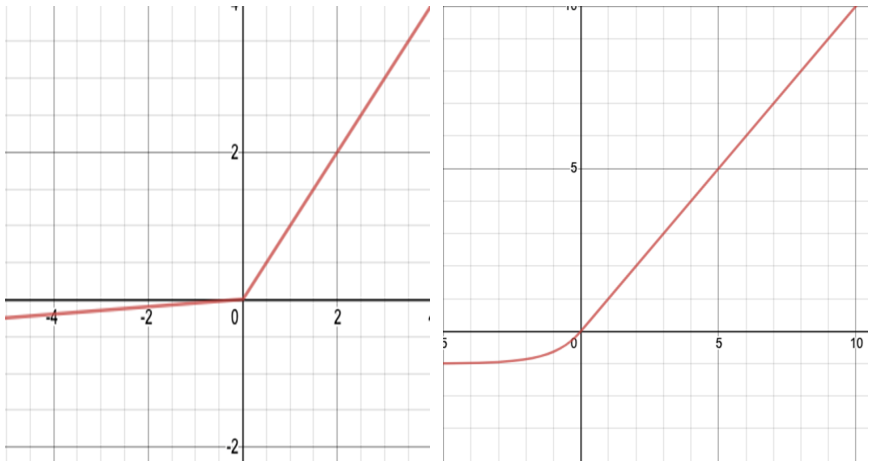
- The first layer, unlike MobileNetV1, is a 1×1 convolution followed by a Batch normalization [10] and a ReLU6 activation function.
- The second layer is Depthwise convolution [17] followed by a Batch normalisation and a ReLU6 activation.
- The third layer is another 1×1 convolution followed by a Batch normalization but without any non-linearity, which according to the author of [19], will impact the performance.

After performing several tests by changing our model's hyperparameters, we conclude that the best parameters to use are the following: a Batch-Size of 32 images, a Test-Size of 20%, a 10^{-4} Learning-Rate, and a 20 value EPOCHS. Furthermore, we noticed that the model's training was very fast, fifty-four seconds per epoch, i.e., 20 minutes, with 99% accuracy. Finally, MobileNetV2 takes up the least amount of memory with only 14MB, and is very interesting to use when adapting AI solutions to smaller machines (smartphones, cameras, ...).

Table 1: Architecture of MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

New activation function to solve the problem of dying neurons It is noted that the presence of ReLU6 activation function in the architecture of MobileNetV2 can cause information loss.
Indeed, knowing that the ReLU6 function returns 0 for any value < 0, the same value for values between 0 and 6 and finally 6 for all values $>= 6$, it is more probable that with the Batch Normalization which normalizes the data between -1 and 1, that we end up with a great loss of information in the data. This issue is called the dying neurons problem [14]. To improve MobileNetV2 results, and to overcome the latter issue, we made several improvements described in the following section.



(a) Leaky Relu with $\alpha = 0.05$ [20] (b) Leaky Relu with $\alpha = 0.05$ [20]

Fig. 2: Comparison between Leaky Relu (2a) and Elu activation functions (2b)

3.2 The Used Dataset

Our research was conducted using Deb's Face Mask Detection as a starting point for our dataset [4]. It comprises 4095 images divided into two classes: with mask images (2165 images) and without mask images (1930 images). These images were collected from different sources (Bing Search API, Kaggle datasets [24], RMFD dataset [15]). Therefore, we choose to extend Deb's Face Mask Detection using data augmentation to ensure the effectiveness of our model despite the real-world images' variability. Augmentation is a process that randomly increases the changeability of the images using different ImageDataGenerator." The resulting dataset was divided into two sets of images (80% for training and 20% for testing).operators (rotation, image resizing, shifting, ...) [22]. The augmentation process used is provided by Tensorflow [8] under the name "

3.3 The proposed model

To improve the performance of the MobilNetV2, we suggested to add the following layers to the model output:

- A network of 128 neurons with Relu activation,
- a 50% dropout,
- , and a network of 2 neurons with an output Softmax activation function to perform the final prediction.

Moreover, changing the activation functions may impact the results of the deep learning architectures. The fact is that Leaky-ReLU and ELU can improve the model's performance because they do not neglect negative values in the

outputs. Therefore, we first test the MobileNetV2 with both Leaky-ReLU and ELU activation functions. Then, the best resulting model will be improved with the previously described final layers.

Replacement of ReLU6 by Leaky-ReLU in MobileNetV2 Using the Leaky-ReLU, we obtained the results illustrated in Fig.3. Thus, we notice that the "Validation Loss" curve has slightly decreased (189%). Therefore, the needed training time was 196 seconds per epoch, much slower than with ReLU6. Moreover, it should be noted that the over-fitting is still very visible, and the accuracy level is 96.7% in the training and 95.7% in the validation.

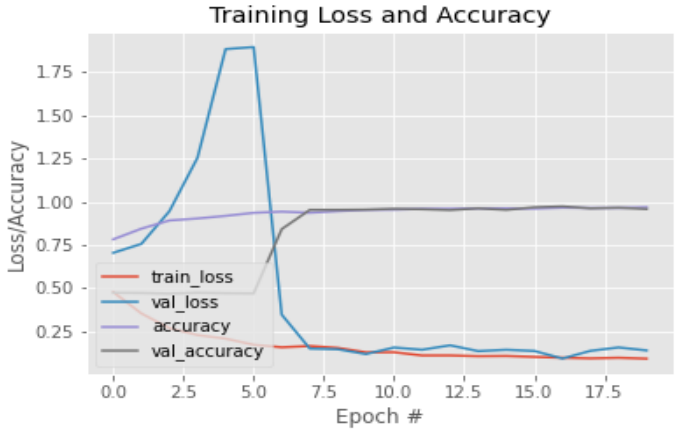


Fig. 3: Training by replacing the ReLU6 activation function with Leaky ReLU

Replacement of ReLU6 by ELU in MobileNetV2 According to the results illustrated in Fig.4, we notice performance improvements due to ELU function. This improvement is outlined in Fig.4 by the over-fitting decrease (the superposition of $train\_loss$ and val_loss). Additionally, we notice that using ELU helps reduce the peak of the validation loss during the first epochs and, thus, improves the convergence (the accuracy increases towards the 5th-6th epochs, unlike the others activation functions). Concerning the training time, ELU requires about 185 seconds per epoch.

According to the results observed during these tests, we choose to implement the model with the ELU activation function and with additional output layers to improve the MobilNetV2 mask detection capabilities. The final results are described in the following section.

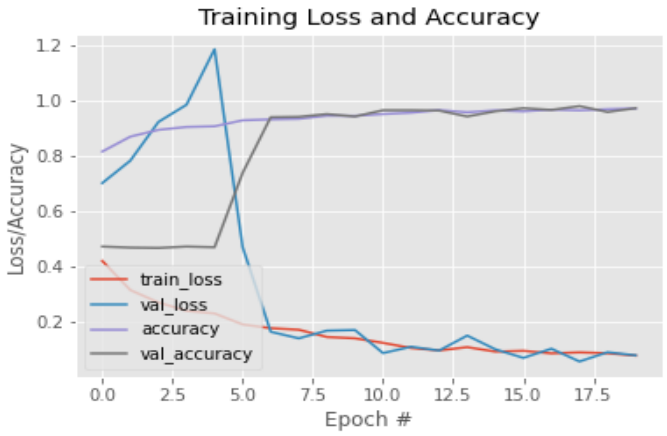


Fig. 4: Training by replacing the ReLU6 activation function with ELU

4 Implementation

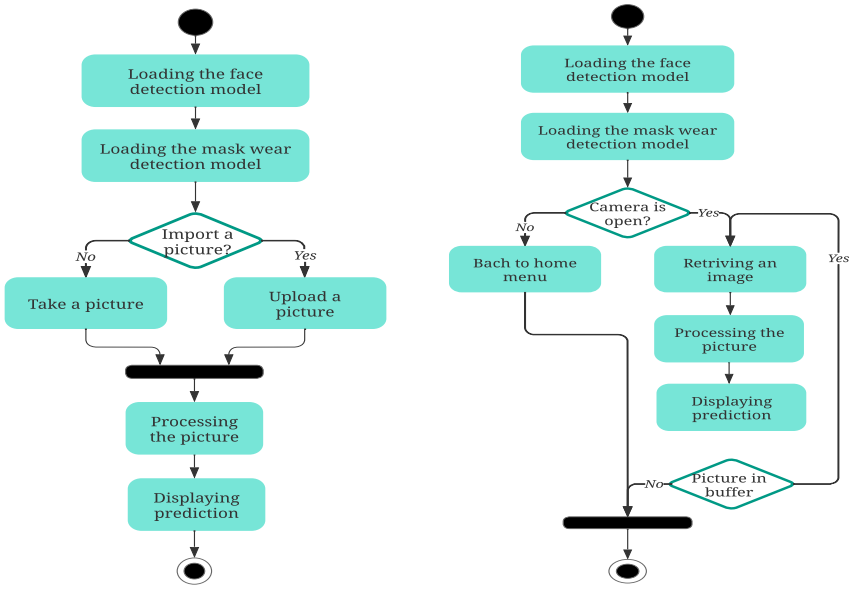
The mobile implementation is done with React Native accompanied by the framework Expo. The ELU model, which has a size of 31.6 MB instead of 30.0 MB for the original model, is used in our application. Since the model was written in Python, we had to convert it into JavaScript with TensorflowJS (after conversion, the model has a size of 9.14 MB and 8.52 MB for the basic one), and we imported it into our application. Then we imported a pre-trained face detection model called BlazeFace, on which we will apply our mask detection model. Finally, to make it friendly, we have set up an interface to choose the detection mode to be used image-based or real-time detection

4.1 The mobile application workflow

Image-based detection: In this interaction, the application takes as input an image provided by the user through the phone's camera or from its gallery. The operation is described below in Fig.5a.

Real-time detection: In this case, the application takes as input a real-time image stream (video) using the user's smartphone camera. Then, for each received image, a prediction is made. The operation is explained below in Fig.5b.

Pre-processing and predictions As shown in the flowcharts in Fig.5a and Fig.5b, after image retrieval in both modes (image-based and video detection), a pre-processing of the image is carried out. The prediction is applied to the resulting image. The process is described in Fig.6.



(a) Image-based detection (b) Real-time detection

Fig. 5: System's execution in image-based (Fig.5a) and real-time (Fig.5b)

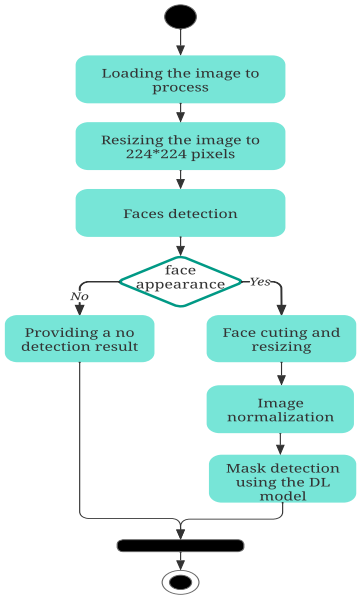


Fig. 6: Diagram explaining how image processing works with the application

4.2 Experimental results

Our experiments were performed using the model with the ELU activation function and the additional output layers. The detection results are therefore described in Table 2 and Fig.7.

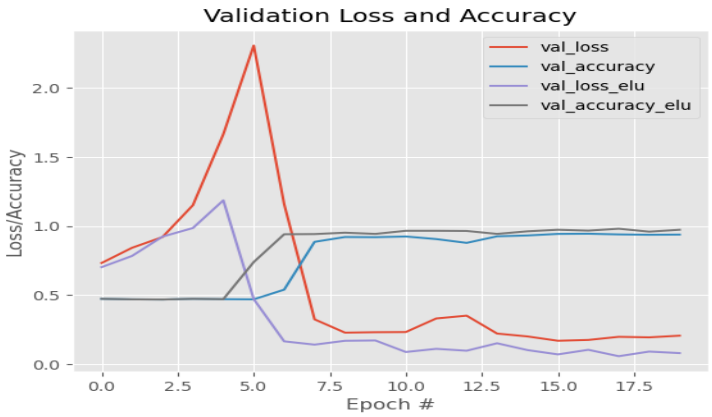


Fig. 7: Results' comparison between the MobilNetV2 and our approach

Table 2: MobileNetV2 VS Our model

	MobilNetV2	Our Approach
Training Accuracy	94.970%	97.270%
Validation Accuracy	93.870%	97.370%
Training Loss	00.137%	00.079%
Validation loss	00.206%	00.080%

According to Table 2, the several upgrades introduced in this paper improved the detection accuracy by 2.42% during the training and by 3.72% during the validation. Also, both the training loss and the validation loss were reduced (from 00.137% to 00.079% during the training and from 00.206% to 00.080% during the validation). Moreover, we can observe in Fig.7 an apparent reduction in over-fitting and an early convergence compared with the initial model. These significant results motivate us to further investigate improving the model's performance.

5 Conclusion

The Covid19 pandemic raised the issue of automating the mask-wearing verification process. This problem raised the need for robust computing systems to

distinguish the persons wearing masks from the others. Hence, these systems need to be as light as possible to be installed on cheap hardware. This paper describes our work on developing a deep-learning system for automatic mask detection. The proposed system is based on the MobileNetV2 model with several improvements.

To improve the results of MobileNetV2, we first switch the activation function from the ReLU6 to the ELU function. Additionally, we included a set of output layers composed of a 128 neurons network with an output Softmax. The resulting DL model was trained on a real-world image dataset [4] and improved with some data augmentation. Therefore, we implemented our model as a multi-platform mobile application that can be used on most common mobile devices.

We tested our system on a subset of the used dataset to ensure its performance. The obtained results illustrate an improvement in detection accuracy and a significant decrease in validation loss. These results demonstrate an improvement in mask detection rates in both image and real-time image detection. In future works, we plan to enhance our model by exploring the possibility of pre-trained weights integration and increasing the size of the training dataset.

References

1. Adhinata, F.D., Rakhmadani, D.P., Wibowo, M., Jayadi, A.: A deep learning using densenet201 to detect masked or non-masked face. JUITA: Jurnal Informatika **9**(1), 115–121 (2021)
2. Apicella, A., Donnarumma, F., Isgrò, F., Prevete, R.: A survey on modern trainable activation functions. Neural Networks **138**, 14–32 (2021)
3. Baeldung: Training and validation loss in deep learning (2022). URL https://www.baeldung.com/cs/training-validation-loss-deep-learning
4. Deb, C.: Face Mask Detection (2022). URL https://github.com/chandrikadeb7/Face-Mask-Detection
5. Degadwala, S., Vyas, D., Chakraborty, U., Dider, A.R., Biswas, H.: Yolo-v4 deep learning model for medical face mask detection. In: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), pp. 209–213. IEEE (2021)
6. Dey, S.K., Howlader, A., Deb, C.: Mobilenet mask: a multi-phase face mask detection model to prevent person-to-person transmission of sars-cov-2. In: Proceedings of International Conference on Trends in Computational and Cognitive Engineering, pp. 603–613. Springer (2021)
7. Fu, H., Niu, Z., Zhang, C., Ma, J., Chen, J.: Visual cortex inspired cnn model for feature construction in text analysis. Frontiers in computational neuroscience **10**, 64 (2016)
8. Goldsborough, P.: A tour of tensorflow. arXiv preprint arXiv:1610.01178 (2016)
9. Hussain, S., Yu, Y., Ayoub, M., Khan, A., Rehman, R., Wahid, J.A., Hou, W.: Iot and deep learning based approach for rapid screening and face mask detection for infection spread control of covid-19. Applied Sciences **11**(8), 3495 (2021)
10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp. 448–456. PMLR (2015)

11. Jignesh Chowdary, G., Punn, N.S., Sonbhadra, S.K., Agarwal, S.: Face mask detection using transfer learning of inceptionv3. In: International Conference on Big Data Analytics, pp. 81–90. Springer (2020)
12. Lin, H., Tse, R., Tang, S.K., Chen, Y., Ke, W., Pau, G.: Near-realtime face mask wearing recognition based on deep learning. In: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), pp. 1–7. IEEE (2021)
13. Loey, M., Manogaran, G., Taha, M.H.N., Khalifa, N.E.M.: Fighting against covid-19: A novel deep learning model based on yolo-v2 with resnet-50 for medical face mask detection. Sustainable cities and society **65**, 102,600 (2021)
14. Lu, L., Shin, Y., Su, Y., Karniadakis, G.E.: Dying relu and initialization: Theory and numerical examples. arXiv preprint arXiv:1903.06733 (2019)
15. MakeML: Mask dataset. URL https://makeml.app/datasets/mask
16. Meivel, S., Indira Devi, K., Muthamil Selvam, T., Uma Maheswari, S.: Real time analysis of unmask face detection in human skin using tensor flow package and iot algorithm. Materials Today: Proceedings (2021)
17. Patel, H., Prajapati, K., Sarvaiya, A., Upla, K., Raja, K., Ramachandra, R., Busch, C.: Depthwise convolution for compact object detector in nighttime images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 379–389 (2022)
18. Ren, X., Liu, X.: Mask wearing detection based on yolov3. In: Journal of Physics: Conference Series, vol. 1678, p. 012089. IOP Publishing (2020)
19. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520 (2018)
20. Singh, S.: Leaky relu as an activation function in neural networks (2021). URL https://deeplearninguniversity.com/leaky-relu-as-an-activation-function-in-neural-networks/
21. Timilsina, S., Sharma, S., Aryal, J.: Mapping urban trees within cadastral parcels using an object-based convolutional neural network. ISPRS annals of the photogrammetry, remote sensing and spatial information sciences **4**, 111–117 (2019)
22. Van Dyk, D.A., Meng, X.L.: The art of data augmentation. Journal of Computational and Graphical Statistics **10**(1), 1–50 (2001)
23. Wang, B., Zheng, J., Chen, C.P.: A survey on masked facial detection methods and datasets for fighting against covid-19. IEEE Transactions on Artificial Intelligence **3**(3), 323–343 (2021)
24. Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., Yi, P., Jiang, K., Wang, N., Pei, Y., et al.: Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093 (2020)
25. Wang, Z., Wang, P., Louis, P.C., Wheless, L.E., Huo, Y.: Wearmask: Fast in-browser face mask detection with serverless edge computing for covid-19. arXiv preprint arXiv:2101.00784 (2021)
26. Wu, J.: Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China **5**(23), 495 (2017)
27. Yang, G., Feng, W., Jin, J., Lei, Q., Li, X., Gui, G., Wang, W.: Face mask recognition system with yolov5 based on image recognition. In: 2020 IEEE 6th International Conference on Computer and Communications (ICCC), pp. 1398–1404. IEEE (2020)

CONCEPTION OF NEW ARTIFICIAL NEURAL NETWORKS FOR MICROWAVE CHARACTERIZATION ENHANCEMENT

F. DJERFAF[1], S. ROBERT[2] ,T. ALI OUAR [1], N. MELLAK [1]

[1] Laboratoire Matériaux, Systèmes Énergétiques, Energies Renouvelables et Gestion de l'Énergie. University of Laghouat, Algeria.
[2] Télécom Saint-Étienne, Laboratoire Hubert Curien, Lyon, France
Corresponding author: ✉ *f.djerfaf@gmail.com*
T.Aliouar@gmail.com; s.robert@univ-st-etienne.f
N.Mellak@gmail.com

Abstract. Artificial neural networks are designed to enhance the microwave characterizations using reduced parameters. In this case, big size data are used for the training step. This data size is directly affecting the relationship's complexity that decreases the network's performances. To overcome this difficulty, novel neural networks are designed for searching the dependence in scattering parameters and permeability parameters.

The originality of this work is the detection of the dependence between parameters using artificial neural networks (ANNs). The elimination of this dependence defines new parameters (reduced parameters). Moreover, the microwave characterization performances are improved using these reduced parameters. The results obtained are validated by simulation and measurement samples. Consequently, ANNs found a new effectiveness in the data size reduction at that time their enhancement in the microwave characterization.

Keywords: Artificial neural networks, Characterization, Microwave, Materials.

1 INTRODUCTION

The properties of saturated thin ferrite films are commonly described by the permeability tensor and the permittivity [1-4]. In general, the relationship between these properties and the scattering parameters is a difficult function.

Artificial neural networks (ANNs) [7-9] have experienced a very rapid development in recent years. Their ability to learn then generalization the problems is proven very useful in many cases [5-6].

Recently, the ANN are used to determine the magnetic [11], the dielectric and the thickness using the scattering parameters.

The purpose is to use the minimum information that reduces the data size used in the training step (search of the parameters' number reduction).

In this work, new neural networks establish their efficiency in the data size reduction and their improvement of the Microwave characterization.

2. METHOD

The originality of this work is the detection of the dependencies in the scattering parameters as well as in the permeability tensor elements. The elimination of these dependencies leads to the reduction of the variables number.

Subsequently, these new reduced parameters are used as inputs and outputs in the designed neural network for magnetic thin layer's characterization.

This method is intended at extracting significant data, in order to prioritize them and eliminate marginal effects. The result of this method is unique, which ensures rigor, flexibility and adaptability.

As a result, contributes to improve the Microwave characterization performances [12-14]. Furthermore, numerous thin ferrite films at microwave frequencies will be characterized without being limited by the big data's size.

2.1 Microwave characterization using conventional ANNs

A good choice of the neural network structure is taken to enhance the Microwave characterization (Fig.1). In this case, a specific structure of multilayer perceptron is used (Fix the number of layers and neurons then select the activation function and the learning algorithm type).

Also, activations functions are *SIGMOID* in the hidden layers *LINEAR* for the outputs layer. The training algorithm is *LEVENBERG MARQUARDT* [15].

The training is done though a particular input presented to the network matches a specific output.

Moreover, the adjustment of the weights is done by comparison between the response of the network and the desired outputs [16].

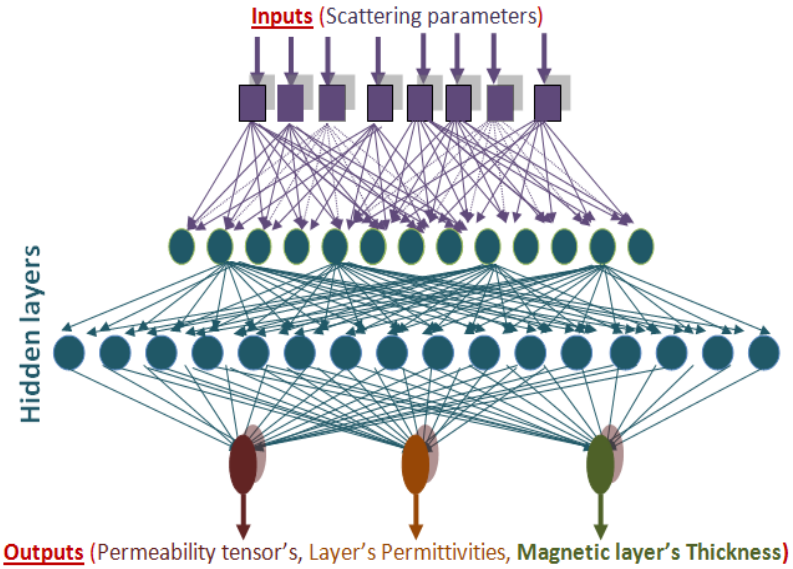


Fig.1. Design of conventional ANNs (without reduction: using conventional INPUTS/OUTPUTS).

These conventional ANNs determine the characteristics of thin ferrite films described in the following Table.1.

Table. 1. Characteristics of the thin ferrite samples (YIG).

Length of the sample	20 mm
Ferrite layers' permittivity	$15 \leq \varepsilon'_{rf} \leq 15.3$
(real and imaginary parts)	$0.08 \leq \varepsilon''_{rf} \leq 0.1$
Dielectric layer's thickness (mm)	0.635
Substrate's relative permittivity	$9.5 \leq \varepsilon'_{rs} \leq 10$
(real and imaginary parts)	$0.008 \leq \varepsilon''_{rs} \leq 0.01$
Thickness of the conductor layer (nm)	600
Slit widths (mm)	0.3
Magnetic applied field (KA/m)	$Ho \leq 220$
Saturation magnetization (KA/m)	$300 \leq Ms \leq 400$
Ferrite layer's thickness (μm)	$10 \leq e \leq 13$
Damping factor	$0.05 \leq \alpha \leq 0.1$
Frequency band (GHz)	$4 \leq f \leq 9$

The ANNs' performances are measured from the mean squared error (MSE) calculated between the estimated outputs (ANN response) and those calculated:

$$MSE_1 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mu_c(i) - \mu(i))^2} \qquad (1)$$

$$MSE_2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\varepsilon_c(i) - \varepsilon(i))^2} \qquad (2)$$

$$MSE_3 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_c(i) - e(i))^2} \qquad (3)$$

Where:
N : Number of samples.
μ_C: Calculated permeability tensor;
μ : Desired permeability tensor
ε_C : Calculated permittivity ;
ε : Desired permittivity
e_C: Calculated thickness;
e: Desired thickness

The data's used in the training step of ANNs consist of complexes matrix (INPUTS-OUTPUTS). These variables are complexes (real and imaginary parts).

The first matrix (INPUTS) consists of scattering parameters *(S^*_{11}, S^*_{12}, S^*_{21}, S^*_{22})*.

The second matrix (OUTPUTS) consists of the elements of the permeability tensor (μ^*, κ^*), permittivities of the substrate and the thin ferrite layer (ε^*_s, ε^*_f) as well as thickness (e) of the thin ferrite layer.

2.2 Reduction of the scattering parameters number using ANNs

New ANNs (Fig.2) are designed for detecting dependence in scattering parameters and in permeability parameters.

This extraction of dependence consists a new ANNs' feature to search then reduce the parameters' number.

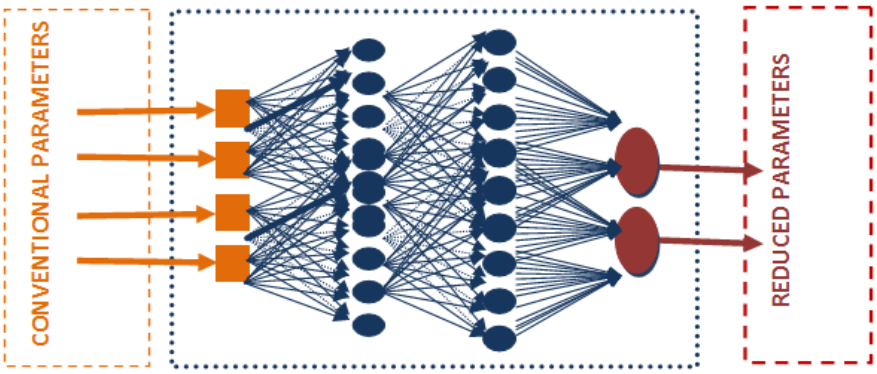


Fig .2. Neural networks for detecting dependence in parameters.

In the calculations, we need to reduce the matrix' sizes (parameter's reduction). While keeping the maximum of useful information for the characterization is our aim.

The dependencies in parameters are searched using optimized ANNs.

2.3 Microwave characterization using improved ANN

Another ANNs are trained by the reduced parameters. These ANNs will be able to characterize the samples having characteristics mentioned in Table.1.

In this case; ANNs predict permeability, permittivity and thin thickness.

Using the proposed structure (Fig.3), performances will be are increased

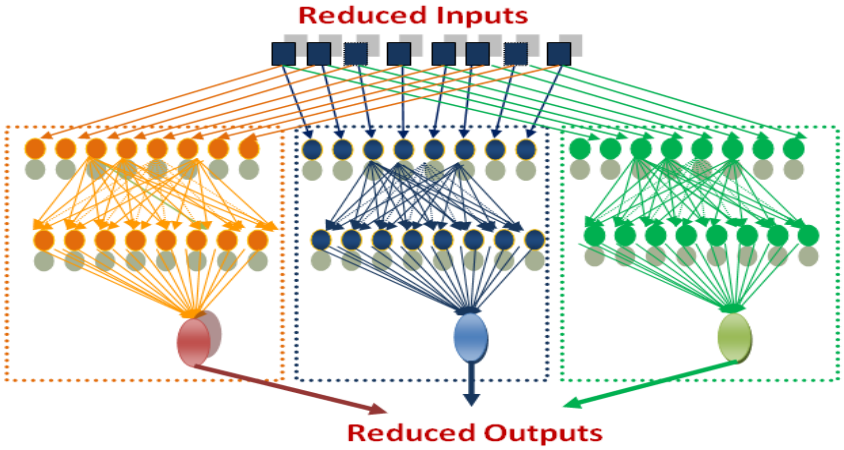


Fig.3. Design of improved ANNs with reduced INPUTS/OUTPUTS.

3. RESULTS AND DISCUSSIONS

As shown in the following Fig.4, ANNs detect the dependence in the scattering parameters [13]. As a result, the network calculates the complexes parameters (S^*_{22} , S^*_{21}) from (S^*_{11} , S^*_{12}.).

This network proves dependence between S parameters. The elimination of these dependencies leads to reduction of 50 % in the original parameters size.

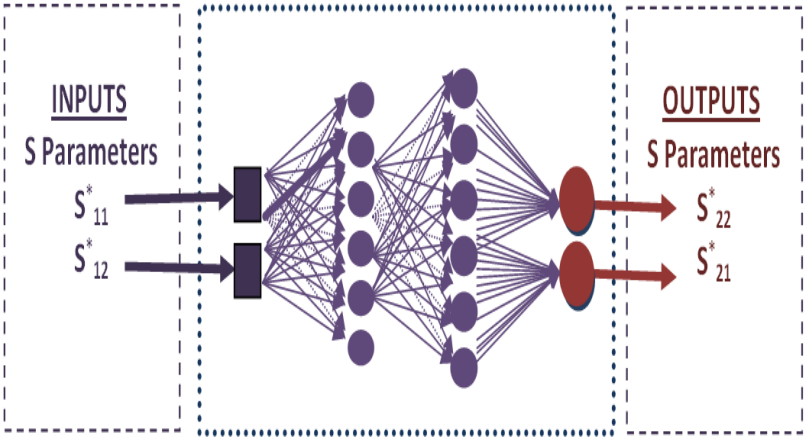


Fig .4. Neural networks for detecting dependence in scattering parameters (50% reduction rate in the original parameters).

The dependences in the scatting parameters are demonstrated using these ANNs that allow the passageway between S parameters (Fig.5 and Fig.6).

Fig.5.a and Fig.5.b confirm that Network N°1 predicts the real part and the imaginary of S^*_{21} from the measured parameters S^*_{11}, S^*_{12} .

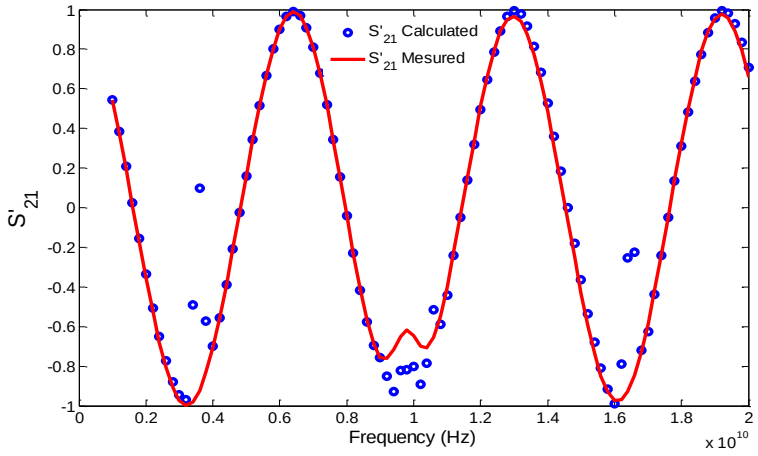


Fig.5.a Prediction of real part of S^*_{21} from measured S^*_{11}, S^*_{12} parameters using Network N°1.

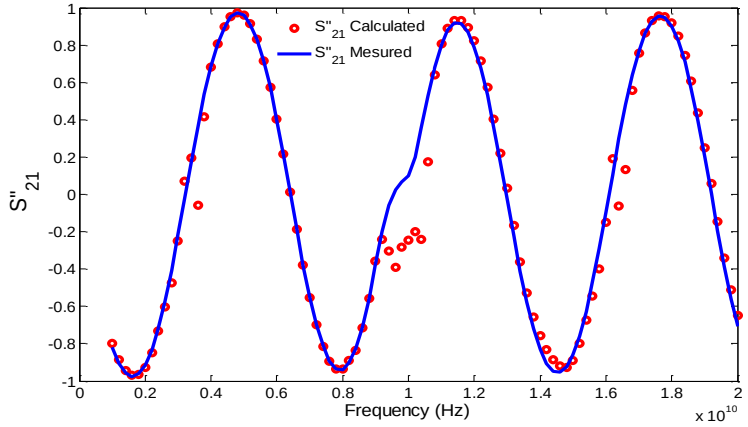


Fig.5.b Prediction of Imaginary part of S^*_{21} from measured S^*_{11}, S^*_{12} parameters using Network N°1.

Moreover, Fig.6.a and Fig.6.b approve that Network N°1 calculates the real part and the imaginary of S^*_{22} from the measured parameters S^*_{11}, S^*_{12} . For this network, the Mean Square Error between calculated and measured is about $9.10\ 10^{-9}$ and Reduction rate is 50 % (Table.2)

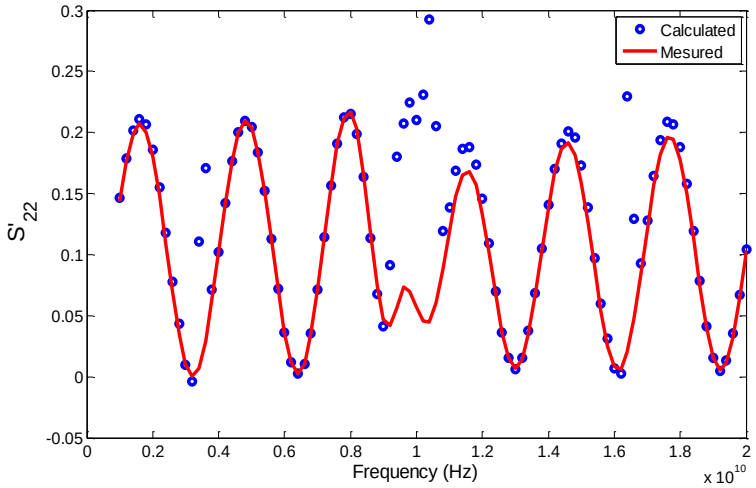


Fig.6. a Prediction of real part of S^*_{22} from measured S^*_{11}, S^*_{12} parameters using Network N°1.

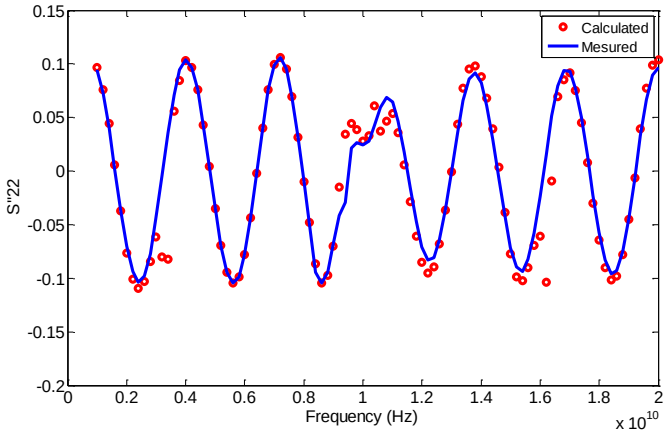


Fig.6. b Prediction of imaginary part of S^*_{22} from measured S^*_{11}, S^*_{12} parameters using Network N°1.

As well, the following Fig.7.a shows dependence in permeability tensor parameters [10] detected by ANN. This network calculates the κ'' parameter from μ', μ'' and κ' parameters (': Real part; '': Imaginary part.).

The elimination of these dependencies reduces of 25% in the original parameters size. This network proves dependence in the permeability tensor parameters.

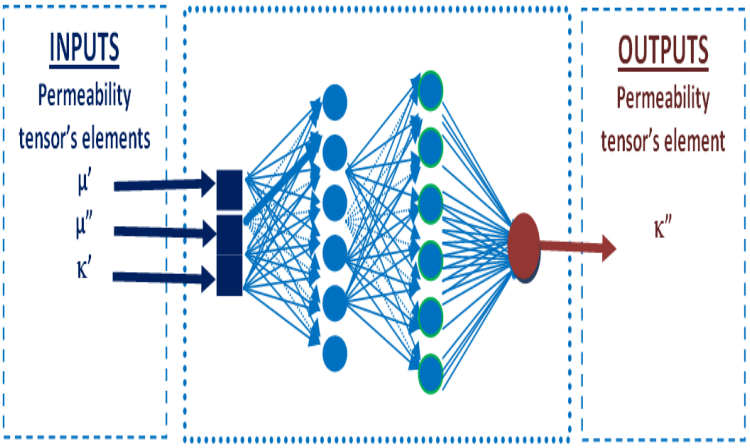


Fig .7.a Neural networks for detecting dependence in permeability tensor parameters (Reduction of 25%)

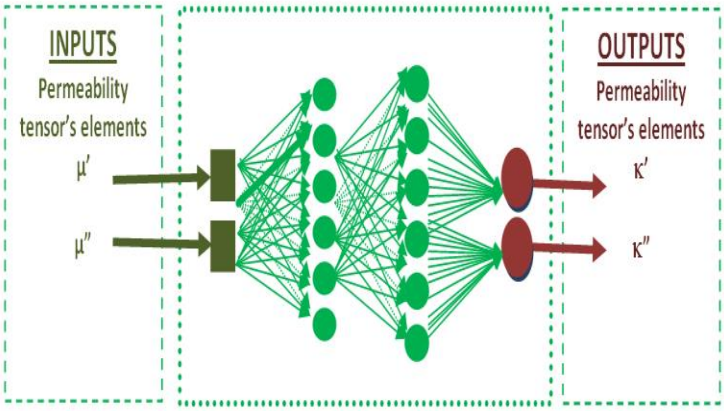


Fig .7.b Neural networks for detecting dependence in permeability tensor parameters (Reduction of 50%) .

Moreover, Fig.7.b shows that the new designed ANN computes (κ' , κ") from parameters (μ' and μ").

This network permits the reduction of 50 % in the original parameters size.

Three new designed networks (Table. 2) confirm the dependence in parameters.

Table.2. Dependence' extraction in parameters using ANNs .

The scattering parameters (INPUTS)				
Designed Networks	Inputs	Outputs	MSE	Reduction rate
Network N°1	S^*_{11}, S^*_{12}	S^*_{21},S^*_{22}	$9.10\ 10^{-9}$	50%
The permeability tensor parameters (OUTPUTS)				
Designed Networks	Inputs	Outputs	MSE	Reduction
Network N°2	μ', μ", κ'	κ"	$4.10\ 10^{-9}$	25%
Network N°3	μ',μ"	κ' ,κ "	$4.69\ 10^{-4}$	50%

Results (in Table 2) illustrate dependence in parameters. The original matrix size of scattering parameters (network N°1: Fig.4) is [8 x 5500]. After reduction, the new size is [4 x 5500] . In this case, the reduction is of 50% in the original parameters size.

In network N°2 (Fig.7.a), the original matrix size of permeability tensor is [4 x 5500], the new size kept is [3 x 5500]. In this case, the reduction rate reached is about 25%.

Moreover, for Network N°3 (Fig.7.b), only two components μ ', μ" are reserved [2 x 5500] in the new size. This network allows the reduction of 50 % in the original parameters size.

Fig.8 confirms the dependence in the permeability tensor elements. Parameters κ', κ" are predicted from μ', μ" (Network N°3).

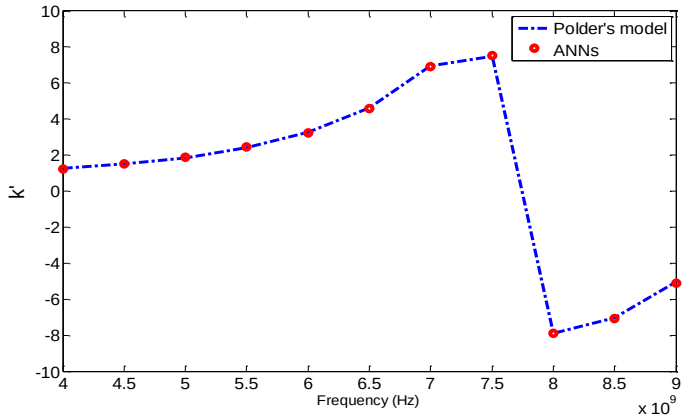


Fig.8.a Prediction of k' parameters from μ' and μ" parameters using network N°3.

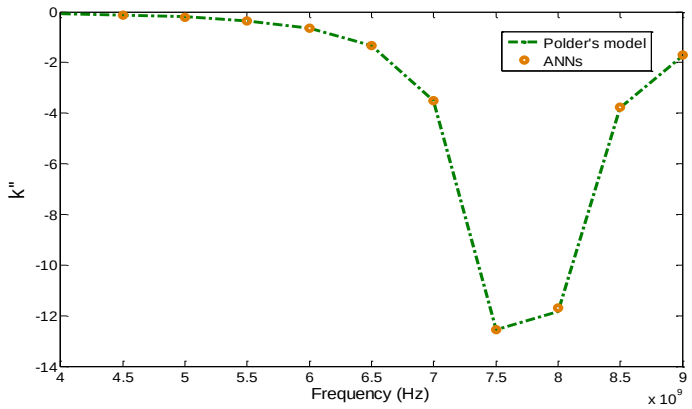


Fig.8.b Prediction of k" parameters from μ' and μ" parameters using network N°3.

Table.3 summarizes the results of conventional ANNs and the improved ANNs.

This table compares performances of improved networks (ANNs trained with reduced matrix) and conventional networks (ANNs trained without reduction).

From these results (Table.3), the error "MSE" decreases using the improved ANNs based on the reduction data (Inputs/outputs).

The improved networks give better results than the conventional networks.

Figure.9. shows that the improved ANNs calculate the permeability elements of thin ferrite layer.

These results show a good agreement with the Polder's model.

Moreover, the designed ANNs (Table.4) calculate the averages values of ferrite layer thickness's and permittivities.

These results show good agreement with samples described in Table 1.

Table.3. Comparison between conventional and improved ANNs' performances**.**

Designed networks	Inputs number	Outputs number	MSE
Conventional Network N°4	4 inputs without reduction S^*_{11}, S^*_{12} ; S^*_{22}, S^*_{21}	2 outputs without reduction ε^*_s, ε^*_f	$4.37\ 10^{-8}$
Improved Network N°5	2 inputs (with 50%reduction rate) S^*_{11}, S^*_{12}	2 outputs without reduction ε^*_s, ε^*_f	**$3.24\ 10^{-8}$**
Conventional Network N°6	4 inputs without reduction S^*_{11}, S^*_{12} ; S^*_{22}, S^*_{21}	2 outputs without reduction μ^*,κ^*	$3.15\ 10^{-5}$
Improved Network N°7	4 inputs without reduction S^*_{11}, S^*_{12} ; S^*_{22}, S^*_{21}	1 outputs (with reduction 50%) μ^*	**2.3110^{-5}**

Table.4. Calculation of the permittivities of thin ferrite layer and the substrate.

	Thin ferrite layer's permittivity		substrate's permittivity		Thin ferrite thick-ness's
	ε'_{rf}	ε''_{rf}	ε'_{rs}	ε''_{rs}	*e (µm)*
ANN calculation	*15.2*	*0.1*	*10*	*0.009*	*12.83*

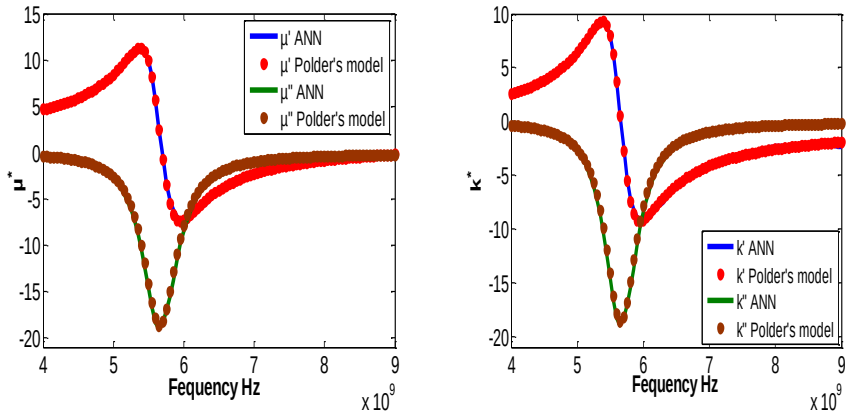


Fig.9. Calculation of the permeability elements (Ms=300KA/m, α=0.05, Ho = 161 KA / m).

4. CONCLUSION

In this work, various improved neural networks (ANNs) are designed for enhancing the microwave characterization. We have demonstrated that artificial neural networks can reduce the parameters' number used.

As a result, the performances of the ANNs designed are enhanced taking into account the dependence in the parameters used for the training phase.

This method has been validated by simulation and by measurement. The removal of these dependencies in parameters led to the reduction of the data set sizes. Also, the proposed method is intended for extracting significant data, in order to prioritize them and eliminate marginal effects. The result of this method is unique, which ensures rigor, flexibility and adaptability.

This work can be extended to a different range of composite materials and others category.

Furthermore, it will be necessary to initiate researches with neural networks for data size reduction.

References

1. J. M. D. Coey, Stuart S.P. Parkin,Handbook of Magnetism and Magnetic Materials, Sringer 2021
2. N. Velhal and al., Ba-M+Y (Ba4−XCo2+XFe36O60) complex hexaferrite: structural, electrical, magnetic, and enhanced microwave absorbing properties over 8–18 GHz frequency, Journal of Materials Science: Materials in Electronics volume 32, pages 10240–10254 ,2021
3. M. Jiabin and al., High-entropy spinel ferrites MFe2O4 (M = Mg, Mn, Fe, Co, Ni, Cu, Zn) with tunable electromagnetic properties and strong microwave absorption, Journal of Advanced Ceramics ,2022
4. G.Vincent Harris and al., Recent advances in processing and applications of microwave ferrites, Journal of magnetism and magnetic materials, pp2035-2047, 2009
5. D. Vincent and al., "New broad-band method for magnetic thin-film characterization in Microwave range," IEEE. Trans. Microw. Theory. Tech. Vol.53. NO. 4. pp 1174-1180. April 2005.
6. L. Weiwei and al.,Metamaterial Absorbers: From Tunable Surface to Structural Transformation Wiley , Journal Advanced materials, 23 May 2022
7. P. Borne, M. Benrejeb, J. Haggège „" Les réseaux de neurones: Présentation et applications"", Edition Technip, Paris, 2007.
8. Y. Djeriri,"Les Réseaux de Neurones Artificiels", University of Sidi-Bel-Abbes 2017.
9. S. Kiat, T. S. Guan, Q. Yinan, L. Shanchun , „"Output partitioning of neural networks„". In Elsevier Science, editor, Neurocomputing, volume 68, pp 38-53, October 2005.
10. D. Polder, " On the theory of ferromagnetic resonance". Philos. Mag., Vol. 40, pp. 99–115, 1949
11. F. Djerfaf., D. Vincent, S. Robert A. Merzouki, "Caractérisation des couches minces magnétiques en hyperfréquences par les réseaux de neurones", 11ème JCMM, Brest, 2010.
12. Siblini, O. Jalled, C. Nader, "Méthode de micro-inductance pour la mesure de la perméabilité de couches minces magnétiques en basse fréquence", J. Phys. IV France, vol. 124, N° , pp. 171-176, 2005

13. T. Ithoh and R. Mittra, Spectral-Domain Approach for Calculating the Dispersion Characteristics of Microstrip IEEE trans. Microwave theory Tech. , 21.. pp 496-499. July 1973.
14. F. Djerfaf, D. Vincent, S. Robert and A. Merzouki, Determination of thickness and permeability tensor using the combination (models-neural networks) , EDP Science 2015
15. S. Abouzaid and al., ,FMCW Radar-Based Material Characterization Using Convolutional Neural Network and K-Means Clustering 24th International Microwave and Radar Conference (MIKON) IEEE Xplore 2022
16. A.Bonginkosi,On the Application of Artificial Neural Network for Classification of Incipient Faults in Dissolved Gas Analysis of Power Transformers MPDI, Mach. Learn. Knowl. Extr., 4,839–851. 2022

Using Machine Learning for Scientific Journals Classification

Razika Lounas, Zahra Haddar, Hocine Mokrani, Achwak Salmi, and Dhai Eddine Salhi

LIMOSE Laboratory, Faculty of Sciences
University of Boumerdes, Independency Avenue, 35000 Algeria
{razika.lounas}@univ-boumerdes.dz

Abstract. This paper presents research about the classification of scientific journals into predatory and correct journals. Prediction of predatory behavior of journals is a serious issue for scientists since the publication in predatory journals harms the scientist and his scientific career and reputation. We used data analytics to detect predatory journals. We started with a preprocessing phase where we selected the most relevant features then, we applied three data techniques (Decision Tree, Support Vector Machine, and K-Nearest Neighbour) on a dataset from the Algerian Ministry official site. This study uses several machine learning algorithms and provides satisfying results for evaluation metrics.

Keywords: Machine Learning, Predator Detection, Decision Tree, Support Vector Machine , K-Nearest Neighbour.

1 Introduction

Publication is a central activity in the scientific community. It allows science dissemination and progress. Scientific publishing is important for researchers, institutions, and countries [1]. Indeed, the progress of a researcher's career is closely tied to the publication of results in recognized events or journals [2]. The dissemination of results allows the scientific to get cited, ranked, and recognized. Scientific publishing is a major criterion in ranking institutions and the health of scientific research in countries around the world [3]. Besides, the scientific career of a researcher or a PhD student is evaluated using publications, especially in scientific journals [4].

Scientific journals represent one of the means of scientific publishing. Through peer review processing, a rigorous evaluation is performed to guarantee that the submitted work is worth publishing and is then accepted for publication [5,6]. Scientific journals adopt several economical models such as the open access model, restricted access model, and hybrid model. The open access model refers to the immediate availability of the work for readers, for free, after publications. This model implies generally article processing charges for the authors [7,8]. In the restricted model, the reader, person or institution, is required to pay the

access to the articles. This model is generally free of charge to the authors. In the hybrid model, the journal gives the choice to the authors to make their articles restricted or open access.

The implication of the financial aspect in the process of publishing leads to the apparition of predatory journals. These journals aim to publish papers, with publication fees, without the effort of reviewing. The papers are generally poorly written and the results contain errors [9]. Detecting such journals is not always an easy task, since they use some techniques to attract young researchers. These techniques include invitation emails, promises of rapid publishing, and the imitation of the names of serious journals [10,11].

Several works endeavored to present a list of criteria to help researchers in selecting journals [12]. The most known reference is the Beall list that provides a list of journals that researchers must avoid. In Algeria, the Ministry of Higher Education and Scientific Research publishes every year a list at the intention of scientists and PhD students. Every researcher or PhD student is required to consult this list before publishing his work. Publications in predatory journals and editors lead the rejection of defense and career ascension. However, the use of these criteria is manual and error-prone. Besides, the list of both predatory and not predatory journals is periodically subject to update since new journals appear continually. In order to help both researchers and institutions, this works aims to propose an automatic tool based on machine learning techniques to classify journals into predatory or reliable journals in order to prevent scientists from the harm caused by predatory journals.

This paper is organized as follows: In Section 2 the related work is presented. The proposed approach is presented in Section 3. In Section 4, the focus is given to the implementation and the results. We conclude in Section 5.

2 Related works

The detection of predatory journals is an important issue. For this reason, several recent works studied this topic. Some of these works are base on journals' attributes, called heuristics, and others are based on textual analysis of journals' contents. In this section, we present an overview of these studies with regard to the used techniques, considered criteria and obtained results.

In [13], the authors presented a methodology for automatic detection of predators journals. The classification is based on heuristics and textual features of journals websites. The authors used heuristics to identify predatory journals (publication period, key words. . .). The used machine learning techniques are K-Nearest Neighbour (KNN), Naïve Bayes, and Support Vector Machine (SVM). The results showed that classifiers work better with heuristics and that SVM gave better results in both features categories. In [14], the authors proposed to classify scientific journals on the exploration of the text of the journals' websites. The technique aims to construct a list of keywords characterizing predatory and non-predatory journals. The authors used the following algorithms: Naïve Bayes, KNN, SVM, and Random Forest. The study is based on features such as

the number of journals volumes, and publication frequency. The results showed that KNN obtained the best score.

In [15], the authors presented a system to classify scientific journals based on journal articles. The system uses a vectorization of 100-dimensions for articles and uses supervised learning to predict the belonging of an article to a predatory journal. The approach used several machine learning algorithms: Naive Bayes, Random Fores, Decision Trees. The results established that Naïve Bayes provides the best results followed by Random forest and Decision Tree. In [16], the authors present a method to detect predatory journals that imitate well-known journals. To detect these journals, the authors analyzed the websites of non-predatory journals to extract common features of these journals. They selected features to recognize predatory behaviors and proposed a numerical measure based on the features of the non-predatory journal sites. The authors used several features such as the age of the domain, the country of the website, the number of dead links, the number of broken links, the number of articles per year, and the scope of the journals. They used Decision Tree, J48, Random Tree, and Random Forest. The best results are provided by the Decision Tree algorithm.

The existing works showed that the approaches based on journals' features gave better results than textual analysis of the content of the journal (articles or journal websites). The main drawback of the existing approaches, except for [13] is the small size of datasets (104 items for [16]). Besides, the indexation in recognized databases, which is an important feature for journals, is not considered in these works. In this paper, we propose an approach based on journal features to detect predatory journals. The approach is based on academically selected attributes and a larger dataset provided by an official site.

3 The Proposed Approach

Machine Learning is a subfield of Artificial Intelligence that focuses on developing operational models regarding relevant parameters. The models are trained on a large amount of data to get appropriate results. Machine learning projects are based on several steps [17]. Figure 1 illustrates the principal steps of this proposal: The first step, data collection, is defined as the procedure of collecting, measuring, and analyzing accurate insights for the research. The second phase, called data preprocessing, is performed to ensure the quality and to study the usefulness of data. In this step, the developer fixes missed data and errors in the dataset. The dataset is split into a training set and a test set. In the third step, we apply Machine Learning algorithms, build and evaluate the models..

3.1 Dataset Collection and Preprocessing

Manual exploration is the initial step in data analysis, where users explore large unstructured datasets to uncover initial patterns, characteristics, and points of interest. In our case, we use original primary data collected for the research

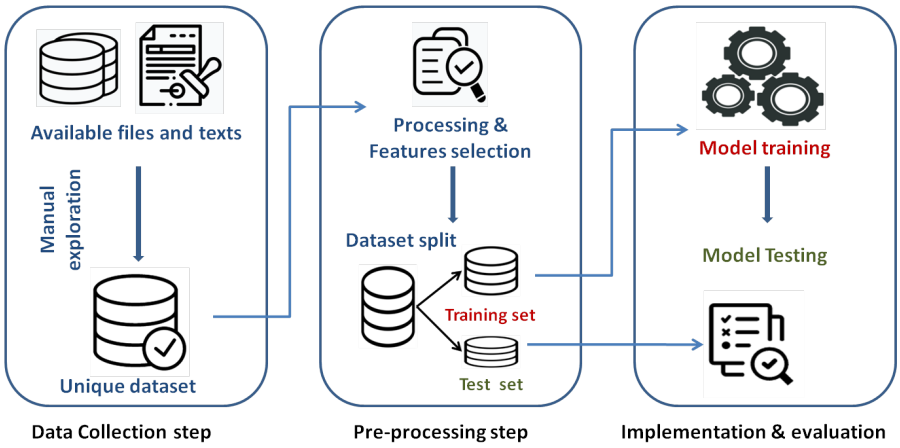


Fig. 1: The steps of the approach

problem from an official list published on the Algerian Ministry of Higher Education and Scientific Research website [18]. The dataset is represented as a matrix, where rows represent the journals and columns represent the attributes. This process does not involve revealing every bit of information a dataset holds, but rather helps create a broad picture of trends and major points to study in more detail. Moreover, some critical features are not always directly available. This phase is critical in the process of Machine Learning algorithms application. Dataset is manually analyzed to label each journal as predatory or not. This annotated data will be used by the different algorithms during the training step.

Features Selection. In a dataset containing a large set of features, the choice of the most relevant ones is based on several methods. In this study, the features are chosen from official texts. We use correlation matrix to get insight about features selected from official texts.

Splitting the dataset. To train the Machine Learning algorithm, we divide the dataset into two sub-datasets, namely, training-set (80% of the dataset) to build the models and test-set (20% of the dataset) to evaluate the models.

3.2 Modeling and Evaluation

Different machine learning classifiers are built for the classification of scientific journals into predatory and non-predatory. The selected algorithms are the most successful with regard to existing works. The models are evaluated to identify the best classifier. In order to evaluate the performance of machine learning techniques, several metrics are used, namely the precision, the recall,

the F1-score:

$$Precision = TP/(TP + FP)$$
$$Recall = TP/(TP + FN)$$
$$F1 - score = (2 * (Precision * Recall)/(Precision + Recall))$$

Where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. Precision is the ability of the algorithm not to label as positive a sample that is negative. The recall is the ability of the classifier to find all the positive samples. The F1-score can be explained as a weighted average of the precision and recall, where an F1-score reaches its best value at one and worst score at zero.

4 Implementation and Results

In this section, we provide details of the implementation for the different phases of our approach.

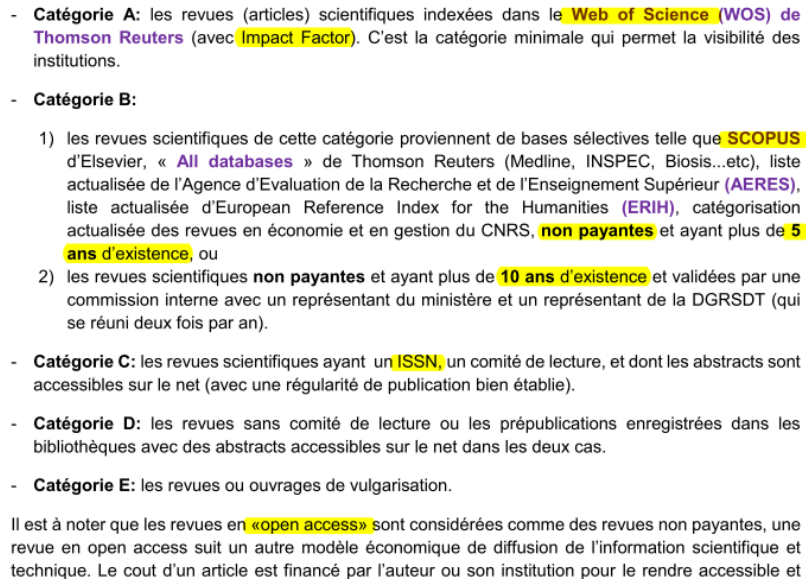
4.1 Data Collection, Labeling, and Preprocessing

This research collected data from an official Ministry website. Indeed, the Ministry website provides lists of legitimate journals from different categories. It also provides a list of predatory journals and a list of predatory editors. Due to the growth of scientific journal numbers, in both legitimate and predatory sides, this list is updated yearly. Researchers face continually the question of whether a journal is predatory or not. The Ministry's legitimate list provides names of journals, identifiers (ISSN and EISSN), and publishers. The list of predatory journals provides the names and URLs of the journals, whereas the list of predatory publishers provides the names of the publishers and their websites. Firstly, this step uses manual exploration through the list of legitimate journals. The size of the list of predatory journals raised the issue of dataset balance. This list contains only 1464 titles against more than 24268 titles for legitimate journals. To solve this issue, the list of predatory editors is used to search for more predatory journals.

The step of features' selection is based on criteria recommended by scientific institutions and legal texts [19]. Figure 2 shows an extract from an official document where keywords related to some recommended criteria, related to indexations of the journals, journals ages, impact factor, publication fees, and open access are highlighted.

The construction of the dataset started with the official lists and texts, however; some information were not available such as the impact factor, the age of journals, and indexations in recognized databases. To obtain this information, the journals' websites and specialized websites about scientific journals [1] have been browsed. Table 1 illustrates the description of the following criteria:

[1] https://academic-accelerator.com/
https://www.scijournal.org/



- **Catégorie A:** les revues (articles) scientifiques indexées dans le **Web of Science (WOS) de Thomson Reuters** (avec Impact Factor). C'est la catégorie minimale qui permet la visibilité des institutions.
- **Catégorie B:**
  1) les revues scientifiques de cette catégorie proviennent de bases sélectives telle que **SCOPUS** d'Elsevier, « **All databases** » de Thomson Reuters (Medline, INSPEC, Biosis...etc), liste actualisée de l'Agence d'Evaluation de la Recherche et de l'Enseignement Supérieur **(AERES)**, liste actualisée d'European Reference Index for the Humanities **(ERIH)**, catégorisation actualisée des revues en économie et en gestion du CNRS, **non payantes** et ayant plus de **5 ans** d'existence, ou
  2) les revues scientifiques **non payantes** et ayant plus de **10 ans** d'existence et validées par une commission interne avec un représentant du ministère et un représentant de la DGRSDT (qui se réuni deux fois par an).
- **Catégorie C:** les revues scientifiques ayant un ISSN, un comité de lecture, et dont les abstracts sont accessibles sur le net (avec une régularité de publication bien établie).
- **Catégorie D:** les revues sans comité de lecture ou les prépublications enregistrées dans les bibliothèques avec des abstracts accessibles sur le net dans les deux cas.
- **Catégorie E:** les revues ou ouvrages de vulgarisation.

Il est à noter que les revues en «open access» sont considérées comme des revues non payantes, une revue en open access suit un autre modèle économique de diffusion de l'information scientifique et technique. Le cout d'un article est financé par l'auteur ou son institution pour le rendre accessible et

Fig. 2: Extract from an official text

– Article Processing Charges: The APC, also known as publication fees, is charged to authors to make their work available as OA.
– Open access: Open Access (OA) is a model for publishing where articles are freely accessed, as the publishing is funded through means other than subscriptions.
– Age of the journal: is the number of years since the journal was first published.
– Impact Factor (IF): The impact factor is the most widely used indicator for journals. It is a citation-based measure indicating the per-year average citations articles published by a journal.
– Indexations: Indexing in eminent databases gives credit to scientific journals and gives journals wide coverage and accessibility and shows its power. This study is based on databases recommended by official texts and used by official academic libraries: Proquest, Scopus, and Web of Science (WoS) [20].
– Identifier: Journals have registered identifiers: ISSN and EISSN numbers. These numbers are helpful to identify journals with very similar titles.

In our study, we label the dataset by adding a column for the target called Y that contains zero or one (0 = non-predatory, 1 = predatory) as illustrated in Figure 3. The size of the dataset is 1000 items with 657 non-predatory and 343 predatory journals. The considered journals belongs to computer science and related fields and applications such as economics, biology and education. The correlation matrix among the features is presented on Figure 4. The chosen

	feature	type	descritpion
1	ISNN	Logical	1 = Journal with ISSN 0 = Journal without ISSN
2	EISNN	Logical	1 = Journal with EISSN 0 = Journal without EISSN
3	APC	Logical	1 = Journal with required APC 0 = Journal without APC
4	Age	Numeral	Age of the journal
5	IF	Numeral	the value of the impact factor
6	Database (Scopus, WoS, or Proquest)	logical	1 = Journal in the database 0 = journal not in the database
7	OA	Logical	1 = Open access journal 0 = Restricted access journal

Table 1: Description of the features

correlation threshold is 75%, therefore, from the correlation matrix, the selected features are ISSN, EISSN, impact factor, age, APC, WoS, and Proquest.

ISSN	E-ISSN	IF	Age	OA	APC	Scopus	WoS	Proquest	Y
1	0	3.425	45	0	0	1	1	1	0
1	0	8.443	46	0	0	1	1	1	0
1	1	2.114	14	1	1	1	1	0	0
1	0	5.514	34	0	0	1	1	0	0
1	1	0,000	16	0	0	1	1	1	0
1	1	0,000	12	0	0	1	0	1	0
1	0	1.576	25	0	0	1	1	1	0
1	1	0,000	4	0	0	1	1	0	0
1	1	0,000	14	1	1	0	0	0	1
1	0	0,000	11	1	0	1	1	0	0
0	1	0,000	3	1	1	1	0	0	0
1	0	2.929	9	0	0	1	1	0	0
1	1	0,000	6	1	0	0	0	0	1
1	0	4.94	42	0	0	1	1	0	0
1	1	2.659	29	0	0	1	1	1	0
0	0	0,000	1	1	1	0	0	0	1
1	1	3.182	50	0	0	1	1	1	0
0	0	0,000	4	1	1	0	0	0	1
0	0	0,000	4	1	1	0	0	0	1
1	1	1.656	30	0	0	1	1	1	0
1	1	0,000	2	0	0	1	1	1	0
0	1	0,000	6	1	1	1	0	0	0
1	0	0,000	29	1	0	1	1	1	0

Fig. 3: Extract from labeled dataset

4.2 Models Implementation and Testing

This work is based on Python libraries for machine learning and data science, Scikit-learn, pre-installed with Anaconda. We used the three most successful clas-

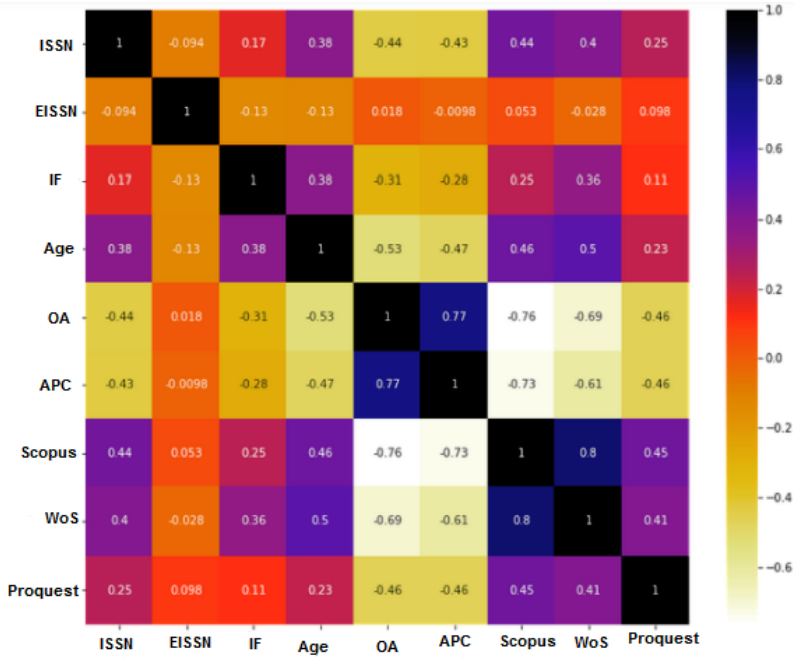


Fig. 4: Correlation matrix

sification algorithms, with regard to the literature review: K-Nearest Neighbor (KNN), Decision Tree (DT), and Support Vector Machine (SVM). Each model has a different approach to datasets. SVM is a supervised learning method that uses the principle of risk minimization to estimate a classified hyperplane to make the boundary between the two classes maximized so that they can be distinguished clearly. KNN calculates the distance between the target data and each data. Then, it considers K minimum distance of data and counts the amount of label to which data belong to. Finally, it predicts target data to the maximum number of labels. The Decision Tree algorithm organizes data continuously on the basis of selected attributes to predict the outcome of a test. The main objective is to divide the data into smaller subsets and by selecting the best feature that divides the training set. This feature is used as the test at the decision node of the tree. A branch of this decision node is then created for each possible value of this feature to partition the training set. The process ends when all the examples in the current subset belong to the same class and then a leaf node is constituted.

The application of the three algorithms on the constructed dataset provided the results illustrated in Table 2 . It shows the precision, the recall, and the F1-score values obtained by the three algorithms for different sizes of the dataset. Three datasets are considered in the experiences:Dataset1 , Dataset2

and Dataset3. The first dataset contains 1000 journals with 343 predators and 657 non-predators journals. The second dataset contains 857 journals with 343 predators and 514 non-predators journals. The third dataset contains 686 journals with 343 predators and 343 non-predatory journals. In each dataset, the ration 80% for training set and 20% for test set is respected.

	KNN			SVM			DT		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Dataset1	0.98	0.93	0.95	0.98	0.94	0.96	0.97	0.94	0.96
Dataset2	0.95	0.97	0.96	0.96	0.97	0.97	0.97	0.94	0.96
Dataset3	0.96	0.90	0.93	0.94	0.92	0.93	0.99	0.90	0.94

Table 2: Algorithms and obtained results

The obtained results, shown in Table 2, presents high values for evaluation metrics for the three datasets. Precision value is stable in the three datasets whereas the recall value tends to get better with a higher dataset size. A more insightful reading of the results is provided in Figure 6 where the details about every metric is graphically represented.

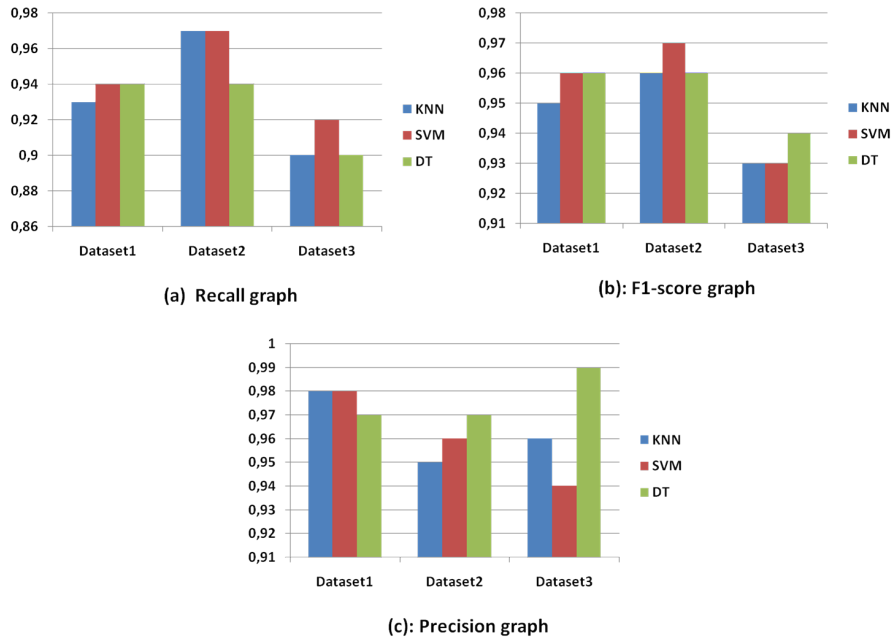


Fig. 5: The metrics in different algorithms and datasets

Part (a) of the figure shows the results of the recall metric. Part (b) shows the results of the F1-score metric whereas part (c) is related to the precision metric. The Recall and F1-score graphics show that the SVM algorithm gives the best results in Dataset2 for the two metrics. For the precision metric, the graph shows that SVM gives better results in the dataset Dataset1, but it is outperformed by the DT algorithm in the others datasets.

At the light of the obtained results, some observations are made. First of all, the obtained results illustrate the soundness between officially recommended criteria and proposed lists for Algerian Scientists. With the continual growth of the number of scientific journals, this tool based on Machine Learning is interesting for institutions in categorizing new journals to update the official lists. Secondly, the presented results show the importance of considering indexing in recognized databases as important features for scientific journal classification. This consideration is a novelty of our proposal, and the carried experiences showed the impact of eminent indexations to guarantee journal quality. Finally, the use of several datasets sizes establishes the stability of the proposed model.

5 Conclusion

This paper presents a successful application of machine learning techniques for the classification of scientific journals. This topic is studied by recent researches and attracts attention because of its importance to both individual researchers and institutions. The proposal used the most successful algorithms, namely, SVM and KNN, and Decision Tree, and is applied on a dataset constructed by the authors based on official lists and information. Besides, the considered features have been carried out using academic recommendations from the Algerian Ministry of Higher Education and Scientific Research. The early results of this on-going work showed a promising effectiveness. As future work, in immediate term, we aim to use larger dataset including other scientific fields, considering more interesting features such as editorial board. We also plan to apply other Machine Learning algorithms and propose to use Deep Learning techniques.

Acknowledgements

This research was supported by the Algerian General Directorate for Scientific Research and Technological Development (DGRSDT).

References

1. O Tomyuk, A Shutaleva, M Dyachkova, A Fayustov, and A Novgorodtseva. University positioning in modern world. In *Proceedings of the International Conference on" Humanities and Social Sciences: Novations, Problems, Prospects.*, 2019.
2. Hugo Horta and João M Santos. The impact of publishing during phd studies on career research publication, visibility, and collaborations. *Research in Higher Education*, 57(1):28–50, 2016.

3. B Hammouti. Comparative bibliometric study of the scientific production in maghreb countries (algeria, morocco and tunisia) in 1996-2009 using scopus. *Journal of Materials & Environmental Science*, 1(2):70–77, 2010.
4. Philip Ball. Index aims for fair ranking of scientists. *Nature*, 436(7053):900, 2005.
5. Fytton Rowland. The peer-review process. *Learned publishing*, 15(4), 2002.
6. William H Guilford. Teaching peer review and the process of scientific writing. *Advances in physiology education*, 25(3):167–175, 2001.
7. Sara L Rizor and Robert P Holley. Open access goals revisited: How green and gold open access are meeting (or not) their original goals. *J. of Scholarly Publishing*, 45(4), 2014.
8. Bo-Christer Björk and David Solomon. Open access versus subscription journals: a comparison of scientific impact. *BMC medicine*, 10(1):1–10, 2012.
9. Selcuk Besir Demir. Predatory journals: Who publishes in them and why? *Journal of Informetrics*, 12(4):1296–1311, 2018.
10. Dalowar Hossan. Predatory journals are indexing in reputed databases: a case study of unsolved issues. *Int. J. of Social Science Research*, 2(3), 2020.
11. Allison A Lewinski and Marilyn H Oermann. Characteristics of e-mail solicitations from predatory nursing journals and publishers. *The Journal of Continuing Education in Nursing*, 49(4):171–177, 2018.
12. Michaela Strinzel, Anna Severin, Katrin Milzow, and Matthias Egger. Blacklists and whitelists to tackle predatory publishing: a cross-sectional comparison and thematic analysis. *mbio*, 10(3):e00411–19, 2019.
13. Awais Adnan, Sajid Anwar, Tehseen Zia, Saad Razzaq, Fahad Maqbool, and Zia Ur Rehman. Beyond beall's blacklist: Automatic detection of open access predatory research journals. In *2018 IEEE 20th International Conference on High Performance Computing and Communications*, pages 1692–1697. IEEE, 2018.
14. Li-Xian Chen, Kai-Sin Wong, Chia-Hung Liao, and Shyan-Ming Yuan. Predatory journal classification using machine learning. In *202020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*. IEEE, 2020.
15. Manas Satish Bedmutha, Kaushal Modi, Kevin Patel, Naman Jain, and Mayank Singh. Predcheck: detecting predatory behaviour in scholarly world. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, 2020.
16. Mona Andoohgin Shahri, Mohammad Davarpanah Jazi, Glenn Borchardt, and Mehdi Dadkhah. Detecting hijacked journals by using classification algorithms. *Science and engineering ethics*, 24(2):655–668, 2018.
17. Willi Richert. *Building ML systems with Python*. Packt Publishing Ltd, 2013.
18. Categorization of scientific journals (Edition 2021) , howpublished = `http://www.dgrsdt.dz/v1/index.php?fc=news_a&id=333`, note = Accessed: 2021-12-30.
19. Direction Générale de la Recherche Scientifique et du Développement Technologique. `https://www.crbt.dz/images/valorisation/Cat%C3%A9gorisation%20des%20revues-DGRSDT.pdf`. Accessed: 2021-09-30.
20. SNDL Systeme National de Documentation en Ligne bases de données. `https://www.sndl.cerist.dz/index.php?p=3`. Accessed: 2021-12-30.

Towards formal modeling and verification of an autonomous car

KACEM Islam[1*] and HAMMAL Youcef[1]

[1]Departement of computer science, USTHB university, Bab Ezzouar Algiers, 16111, Algeria.

*Corresponding author(s). E-mail(s): ikacem@usthb.dz
Contributing authors: yhammal@usthb.dz

Abstract

A self-driving vehicle having the ability to recognize its surroundings and maneuver by itself is referred to as an autonomous car. To accomplish autonomous driving, there are many different kinds of sensors, actuators, and computers , as well as complex algorithms needed for autonomous vehicles to achieve perception, location, planning, and control. In addition to relieving human drivers from driving, autonomous or self-driving cars are emerging as a solution to a number of issues on roads that are mostly brought on by people, such as accidents and traffic jams. However, such benefits are accompanied by significant challenges in the verification and validation for safety assessment, and reliability. Furthermore, due to the potentially unpredictable nature of artificial intelligence utilized in self-driving automobiles. This raises issues that require attention. The degree of adaptation in self adaptive systems has been rising, and such rise and improvement makes it more challenging for control and find errors. With high dynamism in the environment, the challenge is to ensure that the system performance is operating correctly. For that we use a modeling method based on probabilistic timed automata, and in order to verify and analyze the performance evaluation of the autonomous cars, we use statistical model checking of UPPAAL SMC tool.

Keywords: self adaptive systems, autonmious driving cars, verification, Modeling, MAPE loop, Model checking, Requirements analysis, Reliable systems

1 Introduction

The technological systems of these days are growing fast and becoming more and more independent from human assistance. Some of them make use of artificial intelligence, among these system there is the self adaptive systems. The self adaptive system is capable of altering and modifying its behavior at running time responding to any unpredicted change of it environment by themselves[1], following the mechanism of MAPE[1] loop [2].
The main task and goal of the technology is to assist human in their daily living activities, some of them are critical, since any bugs or errors in the software or hardware can lead to loss of money, time or even lives. The autonomous vehicles are a type of self adaptive systems which are spreading over all the world in last few years, they are used in many areas, like manufacturing, medicine , urban transportation, or the exploration of the space,.. In this paper we are interested in autonomous vehicles more precisely self-driving cars.

The autonomous vehicles [3, 4] are constituted of many subsystems, the interconnection and the communication between these subsystems are the factor key in the overall performance of the full system of the autonomous vehicles, such as the controller sub system, the path planner subsystem, the localizer subsystem [5] (see Section 3). To be sure that the systems behavior of autonomous vehicles is reliable and safe, we make use of formal verification to model the subsystems and verify the reliability of the global system. The contribution of our work include: propose a modeling of several subsystems of autonomous cars based on the mechanism commonly used MAPE loop, then verify some qualitative and quantitative properties based on the proposed model. The paper is organized as follows : Section 2, presents some related work. In section 3, we outline autonomous car definition and architecture. In Section 4, behaviors models using Probabilistic timed automata. Section 5, gives the evaluation part. Section 6 concludes this paper.

2 Related work

Formal method is one of the principal method used by engineers to identify inconsistencies and defects in the software. It can only prove the presence of mistakes, not the absence of them. This means that testing can help reduce the number of defects, but considered alone it is not sufficient to prove compliance against certification requirements. In the literature, many works studied the performance of self adaptive systems such as Robots, Drones, Cyber-Physical Systems (CPSs) which are defined as systems in which the physical behavior of the vehicle is controlled by computer-based algorithms without human intervention [8]. In the literature we can find a lot of research and application of verification of self adaptive systems, such as: The authors of [10] presented a decentralized self-adaptive system, an intelligent transport system (ITS). It uses the data collected from the communication to improve the traffic. The

[1] Monitor,Analyze,Plan,Execute

authors used the model checker UPPAAL to verify behavioral properties of the system. In article [11], the authors introduced an integrated framework using a self-adaptive mechanism MAPE loop and decentralized functionality. They show an example of an isolated city from all communication due to a disaster, unmanned aerial vehicles (autonomous drones) are dispatched to increase security and locate wounded people. In the article [12], the authors used a model checking technique to verify the safety of Autonomous unmanned aircraft systems.

3 Autonomous Cars

Autonomous cars [3, 4], also known as driver-less cars, have been studied and developed since 1980s by many universities, research centers, car companies and others. An autonomous car is able to drive and navigate by itself without human assistance or intervention. The autonomous car system is divided into a group of subsystems (Figure 1). Each subsystem has a role to play, the collaboration between them, achieve the self driving mechanism. We can outlines the following: *perception subsystem*: its role consists of collecting data, by sensing the environment surrounding the autonomous car, using many types of sensors such as: *LIDAR* (stands for light detection and ranging, use laser to estimate the distance between objects to avoid collision). *RADAR* (radio detection and ranging, it utilizes electromagnetic waves to locate things, estimate their location, and determine their speed). *CAMERAS* (Records multiple still shots (frames) to capture a two dimensional motion picture). The *localization subsystem* (satellites are used to provide autonomous geo-spatial location). The *planing subsystem*: it makes a plan in relation to the collected information. The *vehicle control subsystem*: it follows the instruction of the plan subsystem by steering, accelerating, and braking the autonomous car. The *management subsystem*: supervises the overall autonomous car system *(See Figure 1)*.

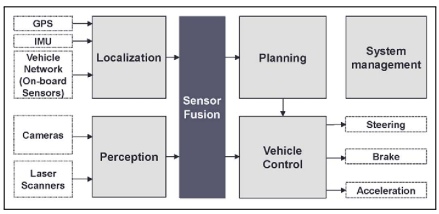


Fig. 1: The architecture of The subsystems of autonomous car [5]

3.1 levels of autonomy of Autonomous Cars

According to the published classification in 2014 by Society of Autonomous Engineers (SAE), the autonomous cars is divided into 6 levels [9]:

- Level 0: All major systems are manually controlled by the driver.

- Level 1: Vehicles can take control of steering, braking and acceleration, but not in all circumstances, the driver must remain aware of the car's doing.
- Level 2: Vehicles can handle steering, acceleration/brake pedals, but it lets the control to the driver immediately if he detects objects or events, the driver must monitor surrounding environment like weather, traffic, etc.
- Level 3: The vehicles have environment detection capabilities and can handle steering and acceleration/brake pedals but in certain environments, the driver is still required, he must remain alerted and ready to take control.
- Level 4: The vehicles can handle steering, accelerate/brake pedals, and can supervise the surroundings environment, the human driver is not needed at most of the circumstances, but can drive within limited area, and can take control at any time.
- Level 5: The vehicles at this level can handles all the tasks, the driver has only to set the destination and start the car. The biggest difference is that, starting from level 3 is the capability of monitoring the environments.

3.2 how does the autonomous car work

An autonomous car can navigate and drive on its own without help from a human, but first of all the driver must start the car, and enter the destination. The car localize itself (GPS) and then calculates the optimal route to the destination. The car goes on its way, the car creates a 3D map of the surrounding environment using its sensors, to calculates its position in the street or the road to avoid collisions. The car makes use of google maps and google street to get notified of the challenges that it might encounter on the road, such as traffic congestion, traffic signs, red lights, or pedestrians, etc.

In real life, a human can drive late, or when it's dark, it's the same for the self-driving car it can drive down a narrow country road, or urban and can depict obstacles and dangers that suddenly appear. Before it can navigate through obstacles, the car must first detect their shape and position. In order for its control algorithms to plot the safest course, given the absence of a human behind the wheel, the car needs smart eyes, sensors, which will resolve these details, no matter the environment, weather, or darkness, all in a fraction of a second. The self-driving car uses a miniature version of the communication technology that keeps the internet going, called embedded photonics. LIDAR using a narrow invisible infrared laser to detect objects. LIDAR fires a train of ultra-short laser pulses to give depth resolution [5, 9] .

The autonomous car follows a MAPE loop mechanism: *Monitoring phase*: collect information about itsself (to check if its circuits work well)and environmental factors (supervise the environment continuously, using Lidar, camera). *Analyze and Plan phase* : After analyzing the information, the autonomous car plan an adaptation process (accelerate, decelerate, stop, change the road, send notification, ..). *Execution phase*: the car executes the plan, but it will never stop monitoring, it can happen that the car changes the plan because of other information, the autonomous car is a system that reacts instantly to changes in its environment. Figure 2 shows the behavior of an autonomous

car, in the input the car collects information about position, traffic signs, road information, then it will behave and react to these informations.

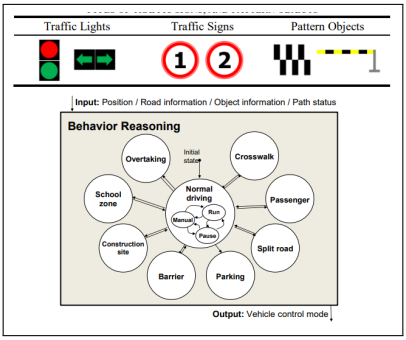


Fig. 2: behavior of autonomous car [5]

3.3 Major challenges and Disadvantage of autonomous cars

The idea of safe and comfortable vehicles has emerged in the automotive sector, which has accelerated the development of various intelligent vehicles technologies. A fully autonomous car can maximize comfort and safety, but it must first be able to recognize its surroundings and carry out course navigation and control without human assistance. To do this, the car must overcome the biggest obstacles that may be in its way on the road, such as : Drive in urban environment. Communication with road users such as normal cars, pedestrian. Understanding pedestrian behaviors depending to age, density of people, gender, walking pattern. Understanding the road (Traffic density, obstacles, understand the panels, cars type and size, street width, lighting , weather, day or night, type of road, etc.). Even though the self-driving car can overcome these obstacles, there are still a number of drawbacks that we should point out: The reflection of humans can't be programmed, ex : a kid traversing the road, a parked car with the driver inside and the engine on). Absence of communication with normal cars. vulnerabilities of hacking and remote control. Can't communicate with pedestrians. problems with one or more of detector and/or executor (such as cameras, brakes, lidar, communication system, etc. Due to these challenges and disadvantages, the autonomous car must be passed through a verification step to avoid any risks and be sure of its safety.

4 Behaviors modeled by using probabilistic timed automata

The modeling uses a formalism to represent the uncertain behavior of the system and communication among its subsystems. For that, we use UPPAAL

SMC [6]. It is based on the traditional version of UPPAAL, it takes stochastic properiters into consideration, for this it uses Probabilistic Timed Automata (PTA) in the modeling process. The properties are also specified by UPPAAL SMC using a temporal logic formula as the query language.

The SMC[2] is a suggested technique that adds the ability to verify the properties with the probabilistic features in comparison to the classical model checking. It generates enough simulations before making a probabilistic determination of whether the verified property is satisfied or not.

Probabilistic Timed Automata (PTAs) [7] are regarded as a modeling formalism devoted to systems whose primary features of behavior are real-time and non-determinism. A probabilistic timed automaton is defined as a tuple (L,$\bar{l}$,χ,Σ, inv, prob) where:

- L: Represents a finite set of localities, where $\bar{l} \in$ L is the initial location;
- χ: A set of clocks (variables), $\chi \in \mathbb{T}$ and $\mathbb{T} \subseteq \mathbb{R}$, $\mathbb{N}$; Σ: A set of events;
- Zone$\{\chi\}$ Is the set of guards (logical expressions), they are in the form: x $\sim$ c, such that x $\in \chi$,$\sim \in \{\leq, =, \geq\}$ et c $\in \mathbb{N}$;
- inv: L $\rightarrow$ Zone $\{\chi\}$is the invariant condition, associated for each location.
- prob: A finite set of probability transitions: prob $\subseteq$ L x Zone(χ) $\times$ Σ $\times$ Dist(2^{χ} $\times$ L). Dist(2^{χ} $\times$ L) is the set of probabilistic distributions on all countable subsets of the product: $2^{\chi}\times$ L. To put it another way, each transition connects one locality to another, and it is identified by an event, a guard, a group of variables that need to be reset, and ultimately a transition probability. So the tuple (l, g, σ, p), represents a probabilistic transition, where p Is a probability function p = μ(X, l′) defined for each pair (X, l′) $\in$ 2^{χ} $\times$ L; with X $\subseteq$ χ.

For properties specification, UPPAAL SMC uses the same query language defined in standard model checking. They question if a particular state formula ϕ, can be satisfied using a path formula, such as:
A <> ϕ $\Rightarrow$ for liveness, A[ ] not deadlock $\Rightarrow$ for safety and E <> ϕ $\Rightarrow$ for specifying, the reachability properties. Additionally, in UPPAAL SMC, new queries are now available for stochastic interpretation of PTA, such as:

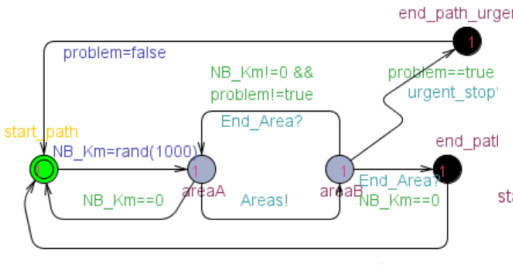
– Simulate N [<= bound] where N is the overall number of runs, bound denotes the time limit for simulations, and E1... Ek stands for the monitored and visualized expressions of k states.

– Pr [<= bound](<> ϕ), where Pr stands for the estimated probability, bound is the time bound and ϕ stands for an expression.

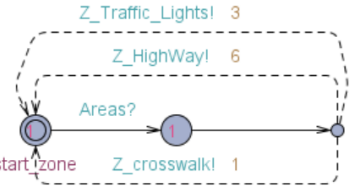
Using this formalism, the formal models of the subsystems that participate in the adaptation of autonomous cars are shown in the Figures 3 and 4.

The system is modeled based on its functioning, following the MAPE loop mechanism. In this modeling we suppose that the system is one autonomous car (the system), and all other entities (other cars, humans, obstacles, etc) are supposed to be the environment. As we mentioned in section 3, the system is composed of subsystems, every subsystem has it own responsibilities, every

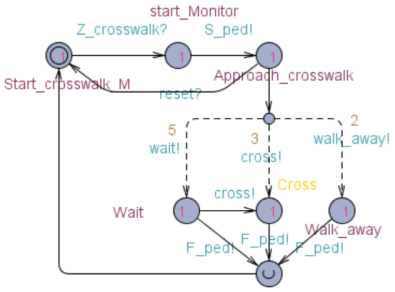
[2]Statistical Model Checking



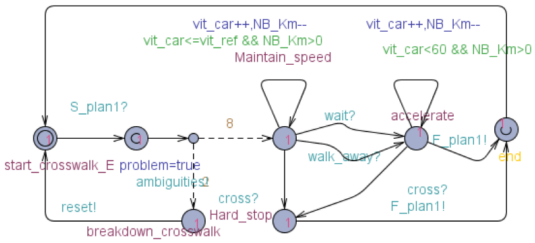
(a) path of cars



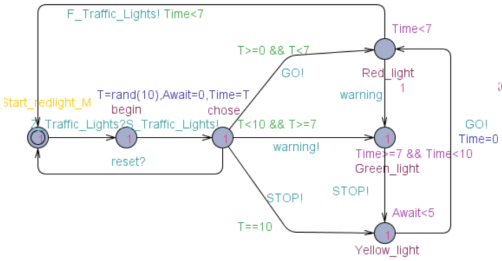
(b) Zone of path



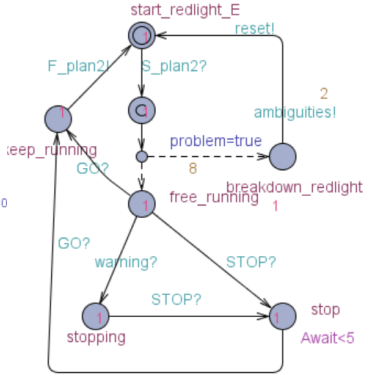
(c) crosswalk monitor



(d) crosswalk executor



(e) redlight monitor

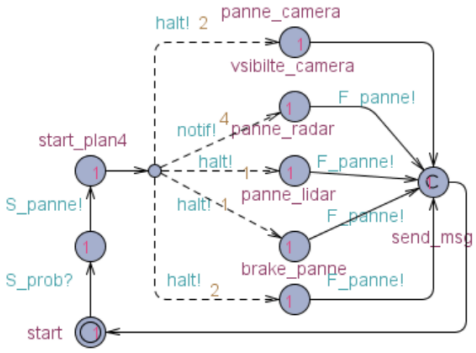


(f) redlight executor

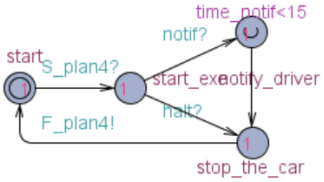
Fig. 3: behavioral models of autonomous car subsystems part i

subsystems of the autonomous car uses information depicted by sensors and detectors, to adapt itself of the change in the environment.

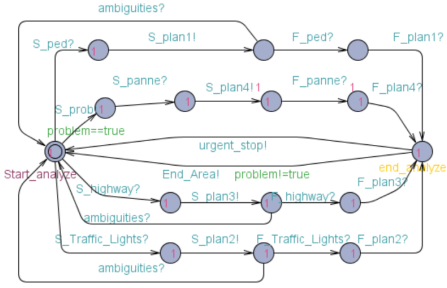
The *Crosswalk-Monitor* (Fig3.c) inspects the status of the crosswalk, every time the car encounters one, it triggers signals based on the behavior of the pedestrians. It uses probabilistic distribution to represent the behavior that can be taken by the pedestrians when reaching the pedestrian because it is



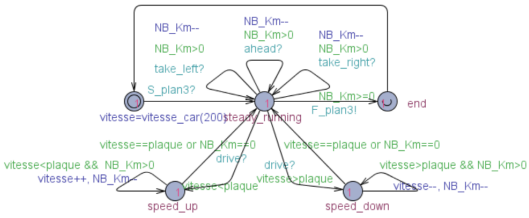
(a) system monitor



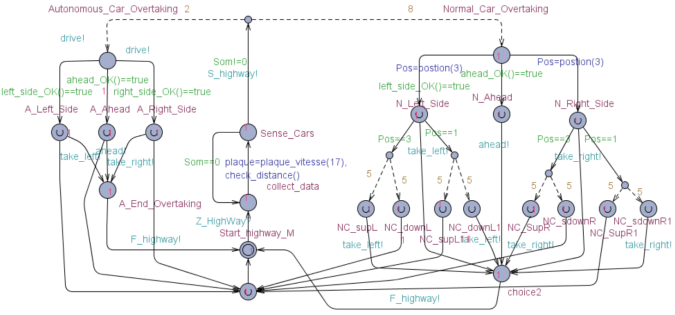
(b) system executor



(c) planner



(d) highway executor



(e) highway monitor

Fig. 4: behavioral models of autonomous car subsystems part ii

not deterministic. It is hard to predict the behavior of humans, but given the situation, it is limited. The pedestrian can wait, cross, or walk away.

Red-light-Monitor(Fig 3.e) inspects the status of the red-lights every time the autonomous car finds one on the road. It detects the current color of the red-light(red, green, or yellow), as well as the timer of the lights. *Highway-Monitor*(Fig 4.e) inspects the status of the highway (which line is safe), it can

depict the behavior of two types of cars, the status of normal car (with the driver inside) or autonomous car. In case the car is autonomous the two cars can communicate and detect each other, but the problem arises when encountering normal car, where the driver can speed up, slow down, change the lane, stop suddenly, In real life the decisions are chosen by the autonomous cars based on calculations of distances and the probabilistic of the behavior of the driver (for example, the probabilistic distribution for a driver to speed up based on the current situation), in the model, the behavior is represented by probabilistic (the driver will speed up or slow down) and calculations (which line is safe ahead, left or right) *Highway-Monitor*(Fig 4.e), inspects every situation that the autonomous car may encounter on the highway.

There is also the risk of hardware malfunction, the *System-Monitor*(Fig 4.a) automaton, inspects the status of internal component used by the autonomous car (cameras, radars, sensors, etc). In case of problem, it notifies the passenger and stop the car in urgent using an emergency plan *system-executor*(Fig 4.b), every possible breakdown is represented by probabilistic such as camera breakdown, camera visibility, lidar breakdown,... (the probabilities are chosen based on the probability of their occurrence in normal life). The *Path-of-Cars*(Fig 3.a) is executed first it simulate the behavior of the GPS of the car, first the passenger had to choose the destination. It is represented by variable KM (selected randomly), and there after the car localize itself, and drives passing through different zones. The *Zones*(Fig 3.b) represents an automaton that chooses in a non-deterministic probabilistic way a zone (supposed only the zones red-light, crosswalk, or highway). Every collected data by the monitoring subsystems passe through the *Planner*(Fig 4.c), the latter decides the necessary plan to execute. The car planner calls the *Crosswalk-Executor*(Fig 3.d), in order to stop the car if a pedestrian is crossing the crosswalk, or slow down if the pedestrian is near the crosswalk to stop in case he decides to cross, or keep running if he walks away. *Redlight-Executor*(Fig 3.f), stops the car if the light is red, slows down if the car is yellow, or keep running if it is red. *Highway-Executor*(Fig 4.d), speeds up, slows down, or maintains the speed based on detected panels in the road, to change the line and overtake cars. There is a transition in Executors automata in order stop urgently the car in case of system dysfunction and notify the passenger during the execution of the plan.

In real life, self-driving cars use their sensors, radars, and cameras to measure destinations, detect signs, and observe the path that people take. Based on this information, it calculates the probabilities, and makes the decisions. In order to simulate this with UPPAAL SMC, we assigned probabilities and random variables. functions that choose the panel speeds and the speeds of the other cars, ... in a random way (random values).

Figure 5 shows the communication between the subsystems automata, the signals send/receive are represented by arrows. All the automaton can be synchronized through binary channels between two automata or broadcasting channels among corresponding multi automata.

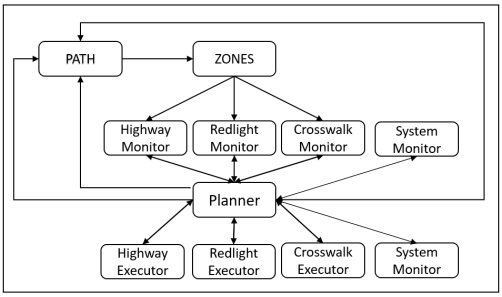
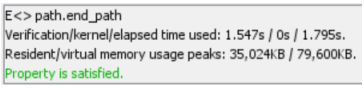
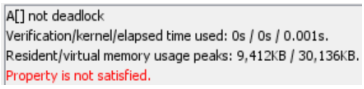


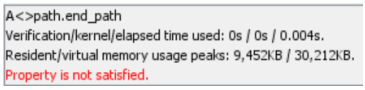
Fig. 5: Interactions between component



(a) Reachability



(b) deadlock



(c) liveness

Fig. 6: Verification of qualitative properties

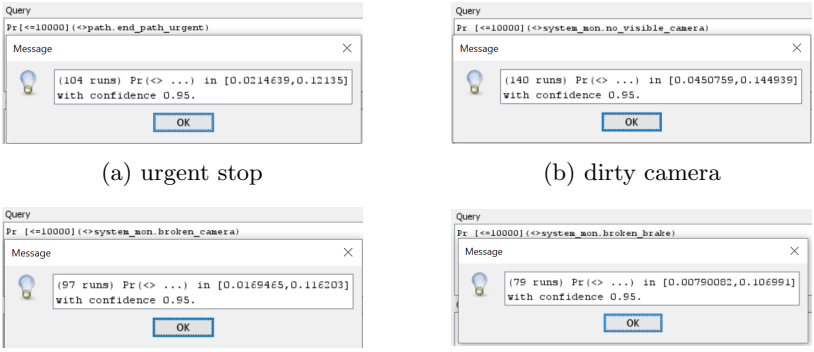
5 Evaluation

This section presents the evaluation of our models based on two types of verification qualitative and quantitative verification.

5.1 Qualitative Verification

We opt first to verify some properties. Among the main classes of Qualitative properties are: reachability, safety, liveness.

- Reachability Properties : They inquire as to whether any reachable state could be able to satisfy the provided state formula ϕ, We express that "a state satisfying ϕ" is reachable by the use of the path formula, E <> ϕ. For example: E <> path.end-path means that the car can make it to destination. The Figure 6.a displays that this formula is satisfied.
- Safety Properties: they take the following form: " something bad will never happen". For example, the formula A [] not deadlock determines that the model is deadlock-free. The Figure 6.b displays that this formula is not satisfied, which means the car can get a hardware dysfunction.
- Liveness Properties: These properties have the following form:"Something will eventually happen". In the model of autonomous car, any car should eventually reach the destination. In its simplest form, the liveness is



(a) urgent stop

(b) dirty camera

(c) camera breaks down

(d) brakes break down

Fig. 7: Verification of quantitative properties

expressed using the path formula A <> ϕ, meaning that ϕ is always eventually satisfied. For example, A <> path.end-path. The Figure 6.c displays that this formula is not satisfied which means that the car wont always reach the state end-path, it can had problems in the way to destination.

5.2 Verification of quantitative properties

verification of quantitative properties. The goal is to calculate the probability that a system will satisfy a specific property prior to a constraint being violated. In other words, with what probability the autonomous car will get to a state is more important to us than whether it gets there or not. We assigned the probability for the transitions based on the probability of happening in real life (for example the probability for camera to get dirty is more likely to breakdown). The following set of queries will be tested:

- The query: Pr[<=10000](<>path.end-path-urgent), which allows to know the probability that the car reaches the state end-path-urgent during the run of the system, which means a dysfunction had occur. Figure 7.a displays the UPPAAL answer to this request.
- The query: Pr[<=10000](<>system-mon.vsibilte-camera), which allows to know the probability that the camera gets dirty during the run of the system, which means a problem with sensors had occur. Figure 7.b displays the UPPAAL answer to this request.
- The query: Pr[<=10000](<>system-panne-camera), which allows to know the probability that the camera breaks down during the run of the system, which means a dysfunction had occur. Figure 7.c displays the UPPAAL answer to this request.
- The query: Pr[<=10000](<>system-mon.brake-panne), which allows to know the probability that the brakes breaks down dirty during the run of the system, which means a dysfunction had occur. Figure 7.d displays the UPPAAL answer to this request.

6 Conclusion

In this paper, we have discussed how to analyze and evaluate the self-driving car using statistical model checking. Using probabilistic timed automata, we have modeled stochastic uncertainties in the behavior of self-driving car systems, pedestrian crossing the crosswalk, redlight intersection, highway, and the most common dysfunction of components of the car.

In the UPPAAL tool, we have employed statistical model checking to assess the effectiveness and the performance of the self-driving automobile. To find more anomalies in the suggested model, further evaluation and verification of self-driving cars, in order to correct and update the model, and then conduct a thorough verification procedure.

References

[1] S.Mahdavi, P.Avgeriou, D.Weyns,"A Classification Framework of Uncertainty in Architecture-Based Self-Adaptive Systems With Multiple Quality Requirements","Managing Trade-Offs in Adaptable Software Architectures", 2017.
[2] IBMCorporation,"An architectural blueprint for autonomic computing","Published in the United States of America 06-05",2006.
[3] U.Nunes, J.A.Fonseca, L.Almeida, R.Arujo,"Using distributed systems in real-time control of autonomous vehicles",Robotica. 21,PP 271-281, 2003.
[4] Kichun Jo, Junsoo Kim, Dongchul Kim, Chulhoon Jang, "Development of Autonomous Car Part I: Distributed System Architecture and Development Process","IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS",2014.
[5] Kichun Jo, Junsoo Kim, Dongchul Kim, Chulhoon Jang, "Development of Autonomous Car Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture","IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS",2015.
[6] Alexandre David, Kim G. Larsen, Axel Legay, and Danny BOgsted Poulsen. Uppaal smc tutorial. Int. J. Softw. Tools Technol. Transf, 397–415, August 2015.
[7] Marta Kwiatkowska, Gethin Norman, and Jeremy Sproston. Symbolic Computation of Maximal Probabilisti Reachability, pages 169–183. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.
[8] A. Platzer, The logical path to autonomous cyber-physical systems,Quantitative Evaluation of Systems(Lecture Notes in Computer Science), vol. 11785. Cham, Switzerland: Springer, 2019, pp. 25–33, 2019.
[9] Kichun Jo; Junsoo Kim; Dongchul Kim; Chulhoon Jang; Myoungho Sunwoo, Development of Autonomous Car—Part I: Distributed System Architecture and Development Process, IEEE Transactions on Industrial Electronics (Volume: 62, Issue: 8, August 2015).
[10] M.Usman Iftikhar and Danny Weyns,"A Case Study on Formal Verification of Self-Adaptive Behaviors in a Decentralized System","FOCLASA 2012 EPTCS 91, 2012, pp. 45-62", 2012.
[11] N. Li Di Bai and Y. Peng and Z. Yang and W. Jiao,"Verifying Stochastic Behaviors of Decentralized Self-Adaptive Systems: A Formal Modeling and Simulation Based Approach", "IEEE International Conference on Software Quality, Reliability and Security (QRS)", 2018.
[12] M. Webster and M.Fisher, N. Cameron and M. Jump,"Formal Methods for the Certification of Autonomous Unmanned Aircraft Systems","International Conference on Computer Safety, Reliability, and Security SAFECOMP 2011: Computer Safety, Reliability, and Security pp 228-242",2011.

The Impact of Attention Mechanism on Arabic Dialect Neural Machine Translation

Amel Slim[1], Ahlem Melouah[1] Usef Faghihi[2], and Khouloud Sahib[1]

[1] Laboratory of Research in Computer Science (LRI), Department of Computer Science, University of Badji Mokhtar, 23000, Annaba, Algeria
First Author Email: selimamel79@gmail.com,
[2] University of Quebec, Trois-Rivières, Trois-Rivières, Canada

Abstract. Although the attention mechanism has been shown to improve the quality of neural machine translation (NMT) in translating long sentences in foreign languages and dialects, its impact on Arabic Dialects NMT has not been proven. In this paper, we study the impact of the attention mechanism in NMT for Arabic dialects, as well as its impact on large and small datasets. This is accomplished using sequence-to-sequence and attentional sequence-to-sequence NMT models. The translation results were shown to be better with large dataset, whereas the attention mechanism performed better with small dataset.

Keywords: Neural Machine Translation, Arabic Dialects, Attention Mechanism, Sequence-to-sequence

1 Introduction

Machine translation using deep neural networks has experienced considerable success with sequence-to-sequence (seq2seq) models [1], [2], [3] that used long short-term memory (LSTM) cells with recurrent neural networks (RNNs) [4]. Linguistic translation has gained a lot of attraction recently, Despite the diversity of languages, dialects have arisen and are seen to be more accessible than languages. The majority of people often employ NMT to translate dialects.

Traditional phrase-based systems have significant shortcomings, which might potentially be addressed by NMT, an end-to-end learning approach to autonomous translation. Tragically, NMT can usually produce subpar translations of lengthy phrases, and it is thought to be computationally expensive for training and translation inference [5]. The solution to this issue was to concentrate on certain portions of the source text when translating, as suggested by [2], a strategy that has since been used to improve NMT.

In this research, we concentrate on the impact of the attention mechanism in Arabic dialect translation. Two databases one small and the other large were also used. The influence of attention on translating both small and large datasets is another aim of this work. And, no work that is identical to or comparable to our work could be found. In order to explore the influence of the attention mechanism in this context, two datasets MADAR (the large dataset) and PADIC

(the small dataset) were used, using the two most popular NMT models seq2seq and attentional seq2seq.

The body of the article is structured as follows: In section 2, relevant work in the field of NMT with attention mechanism is addressed. The data used to assume this work is described in Section 3. We also present the Arabic dialects datasets in section 4. section 5 summarizes all our experiments. Section 6 presented the obtained results and a discussion of this work. The paper is concluded in Section 7.

2 Related Work

The accuracy of machine translation has been greatly improved in a number of language performances thanks to NMT, which has become increasingly popular. One typical enhancement is the attention mechanism which is demonstrated its exceptional capacity to improve the translation's quality.

In this section, we present various works that included an attention mechanism in their translation system, and a comparative study is presented in table 1). this comparison is based on a variety of factors, including the task, data size used, and score bleu [6] result.

[7] proposed supervised attention for NMT, which is inspired by standard statistical machine translation supervised reordering models and incorporates input from external typical alignment models. [8] proposes a neural alignment model and combines it with a lexical neural model in a log-linear system, the models are employed in an automated word-based decoder that directly hypothesizes search alignments. [9] added an explicit coverage vector to the attention mechanism to alleviate the over and under translation that is inherent in NMT. To capture long-term dependencies, [10] developed an extra recurrent structure for attention.

[11] developed a bidirectional NMT model based on agreement for symmetrizing alignment. [12] improved alignment by including various structural alignment biases (structural biases from word-based alignment models such as spatial bias, Markov conditioning, fertility, and agreement over translation directions) into attention learning. They all enhanced the attention models that were learnt unsupervised.

For Arabic, the best effort is shown in dialectal translation. Several papers on translations between Modern Standard Arabic (MSA) and Arabic dialects and between Arabic dialects and foreign languages such as [13], [14], [15], [16], but in dialectal NMT with attention, we found just two works, [13] and [14], [13] is unsupervised dialectal NMT, attentional Seq2seq model proposed for translate Arabic dialect into MSA with unsupervised learning in this work. [14] used the attention mechanism in the dialectal NMT model, this work translated the Algerian dialects by applying the transfer of learning to an attentional seq2seq model to improve the quality of the translation of Algerian dialect. In addition to being concurrent with our work, none of these systems consider similar of our work. Our work is considered the first of its kind, as far as our knowledge is

concerned, particularly because we study the impact of attention mechanism in Arabic dialectal NMT.

Table 1. Present related works.

Work	Year	Task	Data	Bleu score result
[7]	2016	Chinese into English (language)	1.8 M sentences from NIST dataset	40.0
[8]	2016	German into English (language)	4.32M sentences from IWSLT 2013	30.0
		Chinese into English (language)	4.08M sentences from BOLT	16.0
[9]	2016	Chinese into English (language)	-	32.73
[10]	2016	Chinese into English (language)	(Train : 0.5 M sentences, Test : 1082 sentences, Validation : 913 sentences) From NIST dataset	33.14
[11]	2016	Chinese into English (language)	2.56 M sentences from NIST	43.50
[12]	2016	Romanian into English (language)	(Train: 100 K sentences, Test: 2K sentences) from Europarl corpus	40.6
		Estonian into English (language)	(Train: 100 K sentences Test: 2K sentences) from Europarl corpus	17.0
		Russian into English (language)	(Train: 100 K sentences Test: 2K sentences) from web derived corpus	6.35
		Chinese into English (language)	Train: 44016 sentences from BTEC	6.24
[13]	2020	Arabic Dialect (Jordanian, Saudi, and Egyptian) into modern standard Arabic	143518 sentences	32.14
[14]	2022	Algerian dialects into MSA	12800 sentences	35.87

3 Background

Machine translation advanced significantly through deep learning and later came to be known as NMT. In addition, the attention mechanism has improved NMT, the idea behind the attention in NMT allows important bits of the source phrase to be encoded at each translation stage. As a result, attention is also regarded as an alignment model[17].

Early NMT models often provide imprecise translations for extended phrases. [18, 19] suggested that fixed-length source sentence encoding is to blame for this problem. Sentences of varying lengths convey varying amounts of information. As a result, while a fixed-length vector is appropriate for short phrases, it does not have the potential to represent a long sentence with a complicated structure and content [19]. Fortunately, a NMT can now improve its translation by focusing on sections of the text that are interesting to it. This method is known as attentional translation.

A parallel corpus C consisting of a series of parallel sentence pairs (x, y) is given to the NMT where $x = x_1, \ldots, x_n$ is an input sentence and $y = y_1, \ldots, y_n$

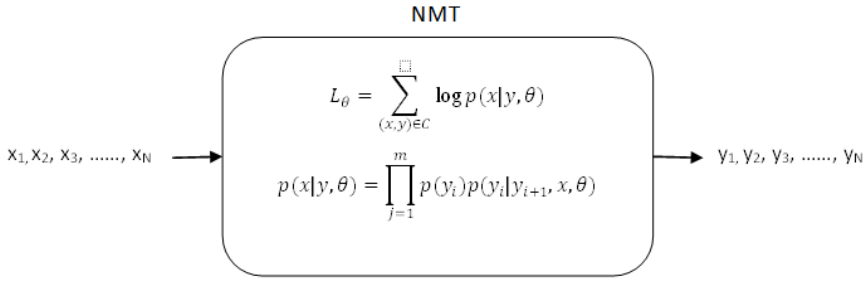


Fig. 1. Present the principle of machine translation.

is its translation as shown in Fig.1, NMT's training objective is to optimize the log-likelihood L_θ, which measures a model's goodness of fit to a sample of data for given values of unknown parameters, θ is a set of parameters to be learned and $p(x|y,\theta)$ presents the equation to measure the probability of a target sentence given the source sentence.

Where:

- m is the number of words in y.
- y_i is the current generated word.
- and $y_(i+1)$ are the previously generated words.

Beam search is typically used at inference time to find the translation that maximizes the above probability $p(x|y,\theta)$.

The first model that was suggested in the context of NMT is the Seq2seq model proposed in [20], after that several advances have been made, such as using subword information [21] and adding residual connections [22] and bidirectional attention based-encoder [23]. an attentional mechanism has been used to enhance NMT by concentrating selectively on parts of the source sentence during translation [21].

The seq2seq architecture is the base architecture for the state-of-the-art NMT models. the seq2seq is the embedded encoder-decoder LSTM, and the attentional seq2seq is the embedded encoder-attention-decoder LSTM, the encoder begins by converting the source sentence's words into word embeddings.

The neural layers then process the word embeddings and translate them into representations that collect semantic knowledge about the words. These representations are referred to as encoder representations. An attention mechanism, the encoder representations, and previously generated words are used by the decoder to generate the decoder representations (states) that in turnover are used to generate the next target word. An attention mechanism used to increase translation quality. Fig.2 gives the architecture of attentional seq2seq.

α is the alignment model that measures the similarity degrees between input at position i and output at position j. In e_{ij} formula, s_{i-1} correspond to the value of the previous word given by the decoder hidden layer and, h_j correspond to the value given by the encoder's hidden layer to the word at the position i.

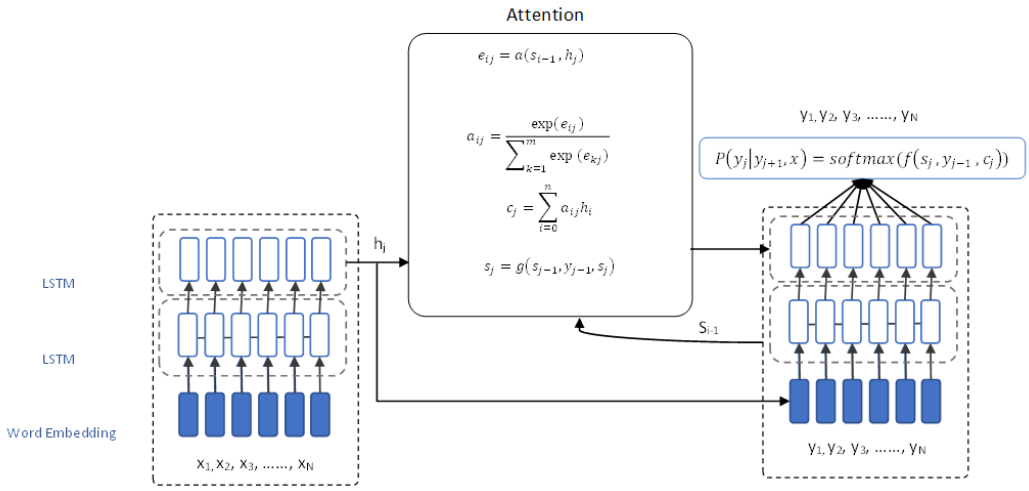


Fig. 2. Architecture of attentional seq2seq.

To obtain a c_j context vector, the calculated attention vector was used to weight the encoder's hidden states. This context vector, along with the previously produced word and its hidden state, is fed into the decoder to generate a representation for generating the current word decoder hidden state s_j, where:

- g is an activation decoder function,
- y_i is the previous decoder hidden state,
- y_{j-1} is the embedding of the previous word.

The current decoder hidden state s_j, the previous word embedding and the context vector are fed to f, which is a feedforward layer and a $softmax$ layer for calculating a score in order to generate a target word as output.

4 Datasets

The first problem which prevents the treatment of Arabic dialects is the serious lack of data on Arabic Dialects, and despite the availability of some data in recent years, most of them are considered very small, especially with regard to NMT.

However, we found some databases such as the parallel corpora PADIC [24] and MADAR [25], both works presented multidialectal Arabic parallel corpus. PADIC has three dialects from the Maghreb two from Algeria, one from Tunisia, and two dialects from the MiddleEast (Syria and Palestine), all these dialects are translated into MSA, PADIC has 6400 sentences for each of the 5 concerned dialects and MSA.

MADAR has two versions, MADAR-6 and MADAR-26, MADAR-6 has 5 dialects Beirut, Cairo, Doha, Rabat and Tunis translated into MSA, and it

has 12000 sentences for each dialect and MSA. MADAR-26 is a set of sentences are translated to 25 city dialects (Beirut, Cairo, Doha, Rabat, Tunis, Aleppo, Alexandria, Algiers, Amman, Aswan, Baghdad, Basra, Benghazi, Damascus, Fes, Jeddah, Jerusalem, Khartoum, Mosul, Muscat, Riyadh, Salt, Sanaa, Sfax, Tripoli), in addition to MSA.

5 Experimentation

In this work, we focus on the role of the attention mechanism in Arabic dialect translation, we also used two databases, one of which is small and the other large. Our Aim can be summed in two points:

- The impact of attention in translating the Arabic dialect.
- The impact of attention in translating small and large datasets.

For assume this study two seq2seq learning model architectures are used (Encoder-Decoder Model and attentional encoder-decoder model). we have two experiments:

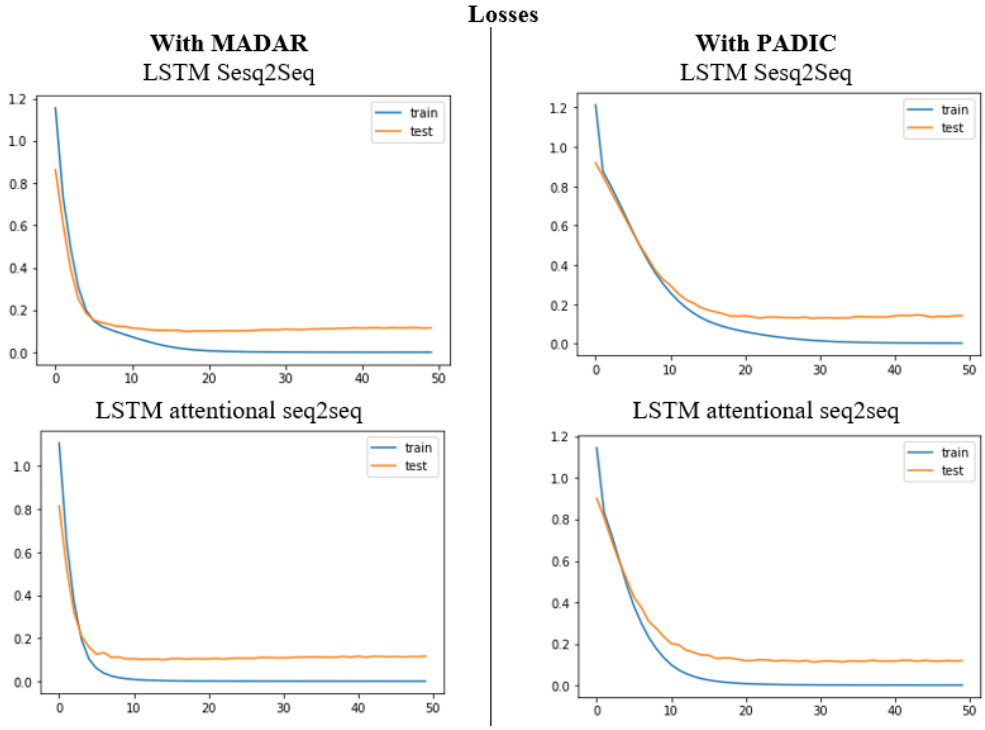


Fig. 3. Shown training and validation losses, with seq2seq and Attentional seq2seq.

- Experiment 1: train MADAR (large dataset) and PADIC (small dataset) with Encoder decoder model.
- Experiment 2: train MADAR and PADIC with Attentional Encoder decoder model.

In each experiment, accuracy metrics and categorical cross-entropy loss are used for the evaluation of the models, the performances are graphically represented with loss for each model Fig.3. The overall loss score and accuracy based on the test dataset was computed and used to determine the performance of the models. We train the models for 50 epochs with each dataset with a selected batch of size 312 sequences for each iteration. In preparation, all sequences within a lot must be padded to the longest sequence in that batch. We note that all models result, training, validation accuracy, and losses are converging. In Fig. Fig 3, we noticed that all the results were very close in all Experiments, except for the loss of training during the training of the MADAR dataset was very small compared to the PADIC dataset, and we observed that this is due to the existence of more sentences in MADAR dataset during a training step.

6 Result and Discussion

6.1 Result

We tested our system on 45000 sentences with MADAR and 34624 sentences with PADIC in the training step, and 10% of these sentences in each dataset were reserved for the evaluation. The rest, 5000 sentences with MADAR and 3848 sentences with PADIC, were exploited in the test step. We tested the model's efficiency in translating the sentences of test datasets, and for the best evaluation of our method, we use bleu score [6], Table 2 shows the result of our system. Overall, in this test we observe that:

Table 2. Score bleu result for each experiment.

	PADIC	MADAR
Experiment 1	33,58	54,49
Experiment 2	44,53	56,10

- With seq2seq (experiment 1) and attentional seq2seq (experiment 2) the highest blue values were with the MADAR dataset because it contains more sentences, While the lowest of the blue score values was with the database PADIC.
- We noticed that the attention property positively affected the two datasets, Where the blue score values increased from 33.58 to 44.53 with the PADIC dataset, and it also increased from 54.49 to 56.10 with the MADAR dataset.

Table 3. Present the translation samples result from PADIC dataset with seq2seq and Attentional seq2seq.

Num	Size	Original	MSA	Seq2seq	Attentional seq2seq	English
1	2	الشروق كدابين	الشروق كاذبون	الشروق كاذبون	الشروق كاذبون	El-Shourouk liars
2	12	فاش مشاو لي لقاوهم في لا دروك تما القصة ديل لا دروك	لما ذهبوا إلى المخازن وجدهم يحضرون المخدرات هناك إكتشف قصة المخدرات	عندما تذهب للفحص عند الطبيب في الجزائر أحضر لها بعض الملابس	لما جاء إلى تركيا أصبح هو نفسه العشيرة	When they went to the stores, he found them bringing drugs there. He discovered the drug story
3	13	شحال هادي كان نيقول لي وليد بلي ديل الشروق طريقهم نيشان لي جهنم	في الماضي كان يقول لي وليد صحفيو الشروق طريقهم معبدة إلى جهنم	كيف هي التي لم تذبح إلى غاية الثانية إلا أن تنفق	في الماضي كان غسيل الثياب مضنيا	In the past, Walid Al-Shorouk journalists used to tell me their road is paved to hell

Table 4. Present the translation samples result from MADAR dataset with seq2seq and Attentional seq2seq.

Num	Size	Original	MSA	Seq2seq	Attentional seq2seq	English
1	2	تحب تفاح،	أ أريد بعض التفاح، من فضلك	أريد بعض الخيط	أريد بعض الغسول لإزالة كريد ريفلون	I want some apples, please
2	7	مهم برشا إنو نستعملو الأساسي اللازم للمشروع	من المهم جدا أننا نستطيع الوصول إلى المواد الخام التي تحتاجها في المشروع	من الأفضل أن تسرع انهم يركبون بالفعل	من الأفضل لك أن تدخل مستشفانا	It is very important that we have access to the raw materials we need in the project
3	14	أكبر مبيعات عندنا الاربعة سنين الي ماكينة التصوير ايه اكس أربع مية وخمسين	أكثر منتجاتنا بيعت عبر السنوات الأربع الماضية هي الآلة الناسخة طراز أ أكس أربعة خمسة صفر	أكثر منتجاتنا بيعت عبر السنوات الأربع الماضية هي الآلة الناسخة طراز أ أكس أربعة خمسة صفر	أكثر منتجاتنا بيعت عبر السنوات الأربع الماضية هي الآلة الناسخة طراز أ أكس أربعة خمسة صفر	The A Four Five Zero copier is our most-sold product over the past four years

- We also got a valuable note, the blue score values increased by 10 points with the PADIC dataset, obverse to that with the MADAR data set, which increased by only 2 points. In our case, we noticed that the attention mechanism (experiment 2) worked better with smaller datasets.
- We note that some of the sentences have been translated incorrectly, although the bleu result is good, and this is due to the fact that the sentences have been rewritten more than once, and in all dialects in the database. The similarity in words in dialects creates a similarity between the sentences, which raises the bleu score value.

Tables 3 and 4 give the results of the translation of certain sentences from the Arabic dialect into MSA. the three first sentences are from PADIC, and the three last sentences are from MADAR.

- With PADIC: in the first and second sentences, the translation is wrongly with seq2seq and attentional seq2seq, but in the translation of last sentence is correct with the two models.
- With MADAR: in the first sentence, the translation is correct with seq2seq and attentional seq2seq, but with the last sentences, with the two models the translation is wrongly.

6.2 Discussion

This work studies the impact of attention mechanism used in NMT for Arabic dialects translation, two datasets are used, MADAR the large dataset and PADIC is small dataset, we devised the database according to the length of sentences and calculate the bleu score for each sentence length. After the dataset's division, we have 5 categories of sentences: 0 to 5 words, the 6 to 10 words, the 11 to 15 words, the 16 to 20 words, 21 to 25 words, and the last category contains sentences that have a length greater than 25 words, the table.5 presented the number of the sentence contains in each category for MADAR and PADIC datasets. We calculated the blue score of model translation for each category as shown in fig.4, and we noticed that:

Table 5. The number of sentences compared to the length of the sentence.

	0-5	6-10	11-15	16-20	21-25	Greater than 25
PADIC	91	253	439	490	464	2130
MADAR	9	78	366	545	637	3357

- For each category, we noticed that the results were very close for MADAR. In contrast, with the PADIC datasets, the blue score result of the sentence categories translated by the attention model was higher than the others.

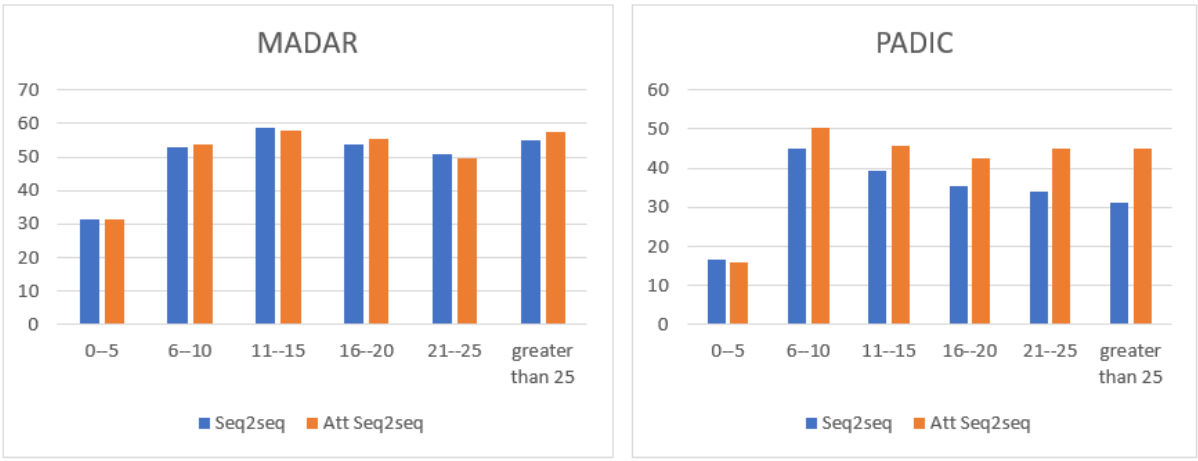


Fig. 4. The length of the sentence compared to bleu score result.

– Attention mechanism works well with large sentences, but in our case, we noticed that it appears to play an active role with the small dataset, with MADAR, the bleu score of translation of the sentence's length grater then 25 words with seq2seq is 54.92 and with attentional seq2seq is 57.36. with PADIC, the bleu score of translation of the sentence's length grater then 25 words with seq2seq is 31.24 and with attentional seq2seq is 45.20.

7 Conclusion

In this work, we interested on the study of the impact of attention in NMT, and we apply this study to the translation of Arabic dialects, we have also investigated the effect of the attention mechanism on the small and large Arabic dialects dataset. Two NMT models seq2seq and attentional seq2seq are used to assume this study. We observed that the translation results were best for the large dataset, but the attention mechanism performed better for the small dataset.

Acknowledgments

We are grateful to the Direction Générale de la Recherche Scientifique et du Développement Technologique (DGRSDT) which kindly supported this research, as well as to the Laboratoire de Recherche Informatique (LRI) where this study was conducted.

References

1. I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks," arXiv preprint arXiv:1409.3215, 2014.
2. D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.
3. K. Cho, B. Van Merriënboer, D. Bahdanau and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.
4. S. Hochreiter and J. Schmidhuber, Long short-term memory Neural computation 9, MIT Press, 1997.
5. P. Sountsov and S. Sarawagi, "Length bias in encoder decoder models and a case for global conditioning," arXiv preprint arXiv:1606.03402, 2016.
6. K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002.
7. Liu, L., Utiyama, M., Finch, A., and Sumita, E. (2016). Neural machine translation with supervised attention.
8. Alkhouli, T., Bretschner, G., Peter, J. T., Hethnawi, M., Guta, A., and Ney, H. (2016, August). Alignment-based neural machine translation. In Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers (pp. 54-65).
9. Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural coverage for neural machine translation. In Proceedings of ACL.
10. Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In Proceedings of ACL.
11. Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Agreement-based joint training for bidirectional attention-based neural machine translation. In Proceedings of IJCAI.
12. Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In Proceedings of NAACL-HLT.
13. W. Farhan, B. Talafha, A. Abuammar, R. Jaikat, M. Al-Ayyoub, A. B. Tarakji and A. Toma, "Unsupervised dialectal neural machine translation," Information Processing and Management, vol. 57, p. 102181, 2020.
14. Slim, A., Melouah, A., Faghihi, U., and Sahib, K. (2022). Improving Neural Machine Translation for Low Resource Algerian Dialect by Transductive Transfer Learning Strategy. Arabian Journal for Science and Engineering, 1-8.
15. Slim, A., Melouah, A., Faghihi, Y., and Sahib, K. (2020, December). Algerian Dialect Translation Applied on COVID-19 Social Media Comments. In International Conference in Artificial Intelligence in Renewable Energetic Systems (pp. 716-726). Springer, Cham.
16. L. H. Baniata, S. Park et S.-B. Park, A Neural Machine Translation Model for Arabic Dialects That Utilizes Multitask Learning (MTL), Computational intelligence and neuroscience, vol. 2018, 2018.
17. Ghader, H., and Monz, C. (2017). What does attention in neural machine translation pay attention to?. arXiv preprint arXiv:1710.03348.
18. P. Sountsov and S. Sarawagi, "Length bias in encoder decoder models and a case for global conditioning," arXiv preprint arXiv:1606.03402, 2016.
19. K. Cho, B. Van Merriënboer, D. Bahdanau and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.

20. I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks," arXiv preprint arXiv:1409.3215, 2014.
21. P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135-146, 2017.
22. Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey and others, Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint arXiv:1609.08144, 2016.
23. H. Ghader and C. Monz, "What does attention in neural machine translation pay attention to?," arXiv preprint arXiv:1710.03348, 2017.
24. K. Meftouh, S. Harrat, S. Jamoussi, M. Abbas and K. Smaili, "Machine translation experiments on PADIC: A parallel Arabic dialect corpus," in The 29th Pacific Asia conference on language, information and computation, 2015.
25. H. Bouamor, N. Habash, M. Salameh, W. Zaghouani, O. Rambow, D. Abdulrahim, O. Obeid, S. Khalifa, F. Eryani, A. Erdmann and others, "The madar arabic dialect corpus and lexicon," in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.

A Taxonomy of Formal Methods used in Verification of Self Adaptive Systems

KACEM Islam[1*] and HAMMAL Youcef[1]

[1*]Departement of computer science, USTHB university, Bab ezzouar, Algiers, 16111, Algeria.

*Corresponding author(s). E-mail(s): ikacem@usthb.dz
Contributing authors: yhammal@usthb.dz

Abstract

The technology of the current era is following an impressive, rapid, and complex development curve. It is moving significantly away from human control and becoming independent. The formal methods used for self-adaptive systems is the main subject of this article. They are effective at making decisions on their own. They know what to do, when, and how. However, they have started to be deployed in various critical areas such as security, business, exploration, supervision, etc. The software core of the self-adaptive system must perform in-depth and accurate data analysis, which means that the self-adaptive system's decision could be critical. Dynamic change in such systems can lead to architectural violations and/or transgressions of functional or non-functional system requirements, this can cause a change in the system's purpose and its services. Thus, it can cause major risks to humans or the environment. So, how can we be sure of the reliability and safety of these systems after the dynamic change, and how should we analyze them? In this article, we'll present a taxonomy of formal techniques that have been used on self-adaptive systems to validate and demonstrate their reliability after the adaptation. To build a mathematical model of a system, technical formal approaches are employed. With formal methods, it is possible to verify the characteristics of the system more comprehensively than with empirical tests.

Keywords: self adaptive systems, formal methods, modeling, MAPE loop, verification, model checking, theorem proving, run-time verification.

1 Introduction

In our modern era, the idea of self-adaptation is emerging in information systems [12, 16, 18, 21, 23]. This increases the complexity of systems and the difficulty of predicting their behavior over time. So it's very important to make sure that there is no dangerous or unexpected behavior. It is thus necessary to identify errors at the beginning of the life cycle of an autonomous system, as well as during their execution. In this context, formal methods have been found to be the most appropriate to ensure the absence of errors or faults. To define, reason about, and verify the properties of dynamic and complex systems, formal techniques rely on formal logic and mathematical notations. To implement formal methods in system verification, we need to translate the system into a mathematical structure (i.e, a collection of equations), next logic is applied as mathematical rules to ask questions about the system and obtain answers about whether or not certain outcomes occur. Formal methods did not attract a lot of interest until the 1990s. Formal methods used to be primitive and difficult to implement. Computers and softwares were relatively simple, and the concept of self-adaptation still did not exist at the time.

IBM was the first to use the term "autonomous computer" [1], where it introduces the idea of a self-adaptive system inspired by the autonomous nervous system of the human body, which regulates heart rate and respiration independently. Adaptive systems can cope with any change that can happen in their environment without any external interference. The system needs to remain available and offer services at all times and under all circumstances.

There are several steps involved in verifying system integrity and effectiveness. To obtain a satisfactory answer, it is necessary to answer the following questions: (1) What kind of system can self-adapt, and why ? (2) What are the formal methods used for describing and depicting most of the aspects of behavior for this kind of systems ? (3) Which formal methods specify the properties that this type of systems should fulfill before and after adaptation ? (4) What techniques and tools are used to verify and guarantee the validity of such systems ? The contributions of our work include: Characterization of each step of the verification and evaluation process of systems . Taxonomy proposes to consistently and comprehensively classify and criticize the formal methods used in each step of the analysis and verification of self-adaptive systems. A thorough comparison between formal verification techniques used to verify SAS [1]. We highlights some challenges which require further research.

The document has the following structure: Section 2 presents related works dealing with other taxonomies of such systems. Section 3 introduces the self-adaptive systems definition, the adaptation levels, the MAPE loop, the adaptation triggers. Section 4 outlines the approach we take to classify formal methods proposed in this area. Section 5 introduces the languages which describe the model of system. Section 6 provides the specification languages for specifying properties. Section 7 describes the techniques and tools used in the testing of self-adaptive systems. Section 8 presents the conclusions and upcoming challenges.

2 RELATED WORK

The formal methods are powerful when it comes to analysis and verification. In the literature, we find many approaches and methods, which present the most used formal methods in the field of SAS. The adaptation of self adaptive system whether behavioral or structural can be described and discussed using formal techniques. Several surveys have been introduced for the verification of self-adaptive systems and re-configurable systems, we mention some of them in the following: The authors in [2] presented a survey about run time models used for analysis and evaluation of quality attributes of self-adaptive software, they focus on run time model. The authors in [3] presented a survey and a comparison of selected approaches on formal verification of self-adaptive systems from some selected papers according to reviewing guidelines. The authors in [4] presented a taxonomy about self protecting

[1] Self adaptive system

software systems, which are a class of dynamic systems, and capable of mitigating and detecting threats at run-time. They applied a taxonomy that classifies the self protection systems based on two categories: what (objectives of self protection) and how (Technique Characterization). The authors in [5] presented a survey of a structured overview of self-adaptation and approaches for engineering SAS. The authors in [6] presented a survey of approaches to adaptive application security. They presented an evaluation scheme from two perspectives covering critical security and reconfiguration requirements. They then use this scheme to evaluate some approaches. The authors in [7] presented some statistics about formal methods used in SAS between 2000 and 2011. The authors in [46] presented a survey of existing tools that have been used for formal verification. As we can see, self-adaptive systems have been evaluated and verified using formal verification. So many surveys have been published, some of them have presented a particular type of self-adaptive systems[5], some of them have presented modeling system languages, some of them present the tools used[7], but none of them depicts an integrated/overall view of the verification steps of a self-adaptive system, or asses the effectiveness of the formal methods used to verify a given system.

The aim of our work is to overcome this drawback by presenting a complete extended taxonomy which tackles each step of the verification process, by answering the questions asked in the introduction.

3 THE KIND OF SYSTEMS THAT CAN PERFORM ADAPTATION

3.1 Self-Adaptive Systems

An adaptive system [8] is a system capable of changing and adjusting its behavior or structure, at run time, due to unexpected changes in the system environment or any system requirements. To face unforeseen changes, the system uses a mechanism known as the MAPE loop [2]. Adaptation may take place at two different moments, and the changes at two distinct levels : 1) *Time stages*: *Conception(design) time*: during the process of developing the system when the adaptation mechanism is chosen by the designer. *Run Time*: When the system is completely operational, and the effectiveness of the alterations must be continuously monitored. 2) *Implementation levels*: *Structural Change*: occurs when the system changes its architecture and internal structure. *Behavioral* change: the system changes behavior as a result of unforeseeable and unpredictable changes in the environment.

Unforeseeable circumstances could be system failures, new requirements and/or changes in priority of requirements, defective design, damage to portions of the system during execution, environmental changes, modifying the purpose of the system. These are a few of the factors that might prompt an adaptive system to take adaption actions.

3.2 MAPE Loop [9]

The MAPE loop is a mechanism that SAS employs, to check if any adaptation is needed. It comprises four stages: ***Monitor***: the system gathers details about itself and its surroundings. Next, it correlates and filters these details in order to find a detail that needs to be analyzed, or needs improvement, to enhance the performance. ***Analyze***: it consists of an in-depth analysis of the details collected in the previous phase. If any change or improvement is needed, it moves to the planning phase, when the system anticipates an adaptation to fix the previously detected problem. ***Plan***: the system sets its objectives. It goes on to outline the measures to be taken. It selects between multiple predefined actions or creates new ones, to change the behavior of the system to make the desired fit. ***Execute***: the system implements measures from the planning phase.

[2] MAPE-loop: Monitor-Analyze-Plan-Execute

3.3 Levels of Adaptation

An intelligent system can influence and can be influenced by the objects and the persons that it interacts with, in its environment. We can observe that the barriers and the borders between it and its environment are lightened. In this section, we outline the various levels of adaptation that an intelligent system can take. From the paper [10], we can observe four levels of adaptation, from the simplest to the most complex: ***Level 1***: this type is the simplest, starting with a thorough analysis of the system environment by the designer in order to extract any possible states that the environment will take. Then, the designer specifies the required modifications that need to be done by the system in every state. Next, the designer defines the behavioral model with statements 'if . else'. The adaptation in this level is a set of statements. ***Level 2***: this type is more complicated and requires more knowledge that the system needs to gather in order to identify the strategies it needs to implement so that it adapts to changes. These strategies are anticipated at design time. There is a predefined strategy for each environment change. Each strategy has an impact on non-functional requirements (scalability, capacity, reliability, maintainability, etc.). The adaptation at this level is thus a set of strategies. ***Level 3***: this type is intended for advanced intelligent systems which may experience situations where they are required to use uncertain knowledge (data already collected), situations where there is no predefined strategy. This type of system has features that make it possible to develop new strategies based on the data collected to solve an unexpected problem. Adaptation at this level is a set of new strategies which do not form part of a predefined strategies. ***Level 4***: this type is the most complex. The system will monitor itself and the environment. This type of system has the ability to learn on its own from previous experiences for the purpose of modifying their own specifications, source code, components, to achieve adaptation. For example, we can mention self-programming software, self-driving cars, etc.

4 FORMAL VERIFICATION APPROACH OVERVIEW

Formal Verification (FV) is the process of verifying the conformity of a design against a specific properties. Figure 1, outlines the approach we took, which is based on modeling and simulation and encompasses four phases[42, 44]. The modeling provides a robust and effective basis for verifying the SAS behaviors. At the same time, the simulation implements an intuitive method for predicting and validating the adaptability of a SAS within less time. ***Stage one***: consists of formally specifying the behaviors and relations between the system entities and the environment. The components of SAS and the various aspects of the environment have their own behavior. The components of SAS are in constant communication, as well with the environment. At this stage, we use formal methodologies and approaches for modeling these communications/behaviors. ***Stage two***: formally defines the local objectives of the subsystems and the global objectives of the entire SAS, and evaluates the performance and adaptability of the system through simulation. The system objectives are specified as logics (TCTL, CTL, etc.) properties, such as security, reach-ability, correctness and liveliness. These properties can be checked on an automatic basis. ***Stage three***: evaluates the performance of adaptability under unexpected changes within the system itself. In this stage, we introduce some of the tools that have been used in the literature to study the performance of the system in a given environment, by performing simulation analyses across the system model designed in stage one, for checking specified properties (stage Two). ***Stage four***: concerns the verification step. Either a part or all properties may not be satisfied. As the results of the analysis are not reasonable (i.e, space state explosion, out of memory, time calculation very long), it may be necessary to adjust the system design artifacts. In the case of negative response the tool provides a counter-example. On the other hand when the system meets the properties, it gives statistical data and results.

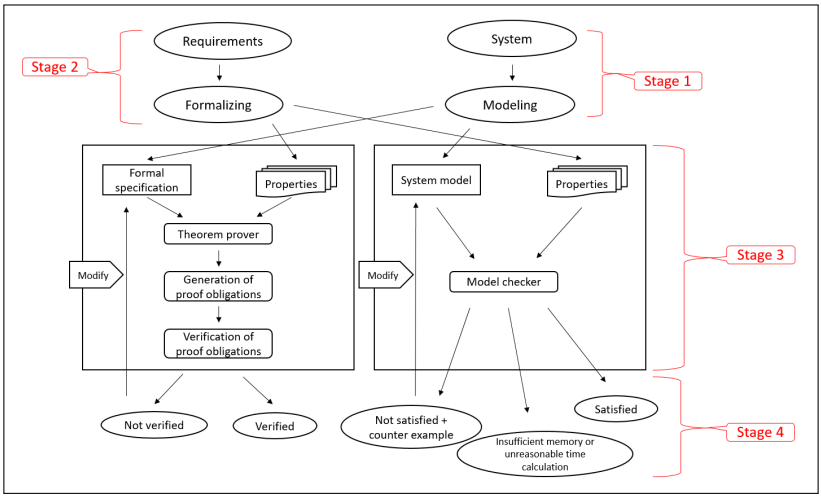


Fig. 1 Stages of formal verification for self adaptive system.

5 DESCRIPTION LANGUAGES OF ADAPTIVE SYSTEMS

Designers may represent very complicated systems as mathematical entities with the use of formal approaches. By creating a mathematically accurate model of the system, the designers may thoroughly test the system's properties. Applying the formal methods at the design phase can help designers to flesh out the system specifications (all the states that the system may take). Based on these specifications, the designers can develop theorems and rules about how a system should behave. At run time, it is difficult and complex to build a model of a SAS, because of the following reasons: (A new requirement may appear at any time. It is impossible to anticipate and predict all behavioral cases (system states), as well the changes in the environment.)

The SAS has many levels of adaptability, as was indicated in (section 3.3). The adaptation can occur at design or run time. So, this rise the question of which formal approaches are appropriate to build a model of each type according to (levels of adaptation) of SAS. In this section, we will discuss the formal methods that have been used to build up the mathematical models of some types of SAS. Based on observation from Table 1, we note that the majority of existing works that have been used to model SAS of level 2 (section 3.3) of adaptation. That means the adaptation has to be anticipated at design time.

5.1 Petri Nets based Modeling languages

Petri net is a mathematical modeling language which describes a system in graphical form. It is a powerful language, language that may be utilized the majority of the time to represent parallel or simultaneous activities in a system. Almost every type of the Petri net family can only model the SAS of low level of adaptation (section3.3), because it belongs to traditional models, that are incapable of modeling adaptive systems. Indeed, they just deal with fixed requirements, and they cannot handle the behavior that change at run-time, due to environmental changes. Petri nets include several temporal or probabilistic extensions that address specific adaptive aspects. However, there is a lack of tools that analyze the performance of SAS, such as High-level PN [13], plausible PN[16],Intelligent PN [17], Time-basic PN [18], context PN [20], learning PN[19], etc. For example the authors in the article [29], were forced to translate the language of colored Petri nets to Mu calculus, to achieve the

Table 1 Describing SAS using formal languages and semi formal languages

Language	Reference	case study	Adapt Level
Ordinary PT	[11]	GSM oriented audio stream encoding and decoding protocol	1
Colored PT	[12]	Traffic Monitoring System	2
High-level Petri nets	[13]	cloud infrastructure	1
Timed Colored Petri Net	[14]	IOT production logistics systems	2
Stochastic Petri Net	[15]	Environment-aware Self-Adaptive strategy	2
Plausible Petri net	[16]	railway track asset management	2
Intelligent Petri net	[17]	manufacturing system	2
Time-Basic Petri nets	[18]	gas burner system	2
Learning Petri Nets	[19]	manufacturing system	2
Context Petri net	[20]	-	-
Timed automata	[21]	intelligent transport system	2
Stochastic TA	[23]	unmanned aerial vehicles	3
Self-adaptive Automata	[24]	Unmanned vehicles	2
I/O automata	[25]	cyber physical robotics	2
DTMC	[26]	znn.com case study	2
PI calculus	[27]	self-controlling system	3
CSP	[28]	communicated nodes	3
CSP	[48]	software architecture	3
Mu CALCULUS	[12]	SMACS framework	2
Adaptive CSP	[29]	art gallery building	2
UML Based Language	[30]	Forest Fire detection system	2
B method	[31]	autonomic rover protection system	2
Z notation	[32]	mobile robotic system	2
pi - ADL	[33]	flood monitor system	3
Brahms	[47]	Robot House	2

verification. Adaptation in Petri nets is mostly represented as a transition, which transits from the old Petri net (old behavior) to the new Petri net (new behavior, after adaptation) such as in [18]. Sometimes a new formalism is presented as an improvement to cope with adaptation, such as [16, 17, 20], etc

5.2 Automata based Modeling languages

Automata are mathematical objects, widely used in computer science, which allow representing the dynamic aspects of a system. In the automata family there are many types, that allow us to model a system whose behavior incorporates both probabilistic and real-time characteristics, such as stochastic timed automata [23], timed automata [21], input/output automata [25], DTMC[26], UML [30], B-method [31]. That means with automata family we can model high level (level 3, section 3.3) adaptive systems. The adaptation is modeled most of the time in the automata family by a transition that changes the state of the system due to events from the environment, or a network of automata that operate simultaneously, depending on the required adaptation. The B method is a formal method that allows reasoning about complex systems as well as software development. The B method makes it possible to model in an abstract way the behavior and the specifications of a software in the formal language of B.

5.3 Algebric based Modeling languages

Algebraic modeling languages are advanced computing specification languages to describe and deal with very complex issues or systems. They are strict mathematical languages with clear semantics, which can be used to describe and check the properties of simultaneous communication systems, such as CSP [28, 48], Mu calculus [12], Adaptive CSP [29], PI-ADL [33], Z notation [32], etc.

5.4 Discussion

Formal verification is the process of checking whether a model satisfies some requirements, It is divided into two groups: ***Static verification*** (model checking/theorem proving /deductive methods): It is performed during the design time in the sense that we do not need any run time information to perform the verification. It aims at exhaustively check every single possible state of the system to find bugs and errors, which could end up being a serious treats in the future. To do so, it requires a lot of time and a lot of computing resources. It takes a long time to produce outputs in order to explore every state and that could lead to state explosion like in model checking technique. SAS are systems that require the ability to adapt, they are often affected by changes that modify their behavior in ways that may not or cannot be captured by a static model. ***Dynamic verification*** (run time verification): It uses run time information to perform verification, but it comes with a price, a limited coverage. Run time verification analyzes small snapshots of system behavior to issue fast results instead of trying to cover everything at once as static verification does.

As discussed in (Section 5) and from the Table 1, we can see that formal methods are most of the time used to model systems where in some sort we have an idea about their behavior taking after the adaptation. None of them had been used to model a SAS of level four (Section 3.3) of adaptation. Here are some of the limitations of model languages presented in this survey: Lack of tools for new emerged formalisms, since the tools for old ones are limited in verifying complex systems and real time applications. Most of languages can not handle run time adaptation since they are destined to design time verification only. Most of techniques used take long time to produce results and consume a lot of computing resources. In order to verify SAS, we most use run-time verification technique (called also run-time monitoring) [56] to get instantly, fast results. The model of SAS may need to change often, and sometimes building a model may be complex or impossible with static formal languages.

6 SPECIFICATION LANGUAGES OF PROPERTIES FOR SELF-ADAPTIVE SYSTEMS

Formal methods can assist in verifying the satisfaction, after adaptation process, of functional requirements, such as certain calculations, data manipulation, interaction with the environment, and any other features which the system has been able to perform prior to adjustment. We may also check non-functional requirements such as durability, maintainability, safety, etc, to make sure that the behavior of the system is logically coherent and is in fact, the most desirable. The early stage verification will ensure that costly errors do not occur in the final stages of development. In order to ensure that properties are guaranteed in all states in which the system will be, we use a formal method that specifies those properties. The authors in [26] choose a Znn.com to be a case study, they used Probabilistic Computation Tree Logic (PCTL) for expressing the property of resilience. In [34] the authors used Linear Time Logic (LTL) to specify the properties (such as correctness, reachability, safety)

of a light control system. The authors in [35] present a SOTA[3] as general goal-oriented modeling framework for analysis and design of SAS, the E-mobility (electric vehicles) is chosen to validate the approach and Fluent Linear Temporal Logic (FLTL) to specify the properties. The Timed Computation Tree Logic (TCTL) is selected to be used by [22] and [15]. Another formalism which is an extension of LTL, called Adapt operator-extended (A-LTL) is presented in [36]. The authors in [37] used another formalism DynBLTL [38], a new logic to express both structural and behavioral properties, which is an extension of LTL, whereas LTL targets the system's infinite behavior, a limited form of LTL called BLTL targets finite sequences of execution states. The authors in [39] used the "graphegen" software tool to verify safety, robustness, timed properties, trough an exploration of TRG. In [30] the authors verify the safety and reachability properties along with temporal properties using Event-B method.

7 Verification techniques for Self-Adaptive Systems

Formal modeling, analysis and verification techniques play a growing role in the development of high assurance systems, since the traditional testing and debugging methods are limited to locate errors within a complex system. This section aims to present the formal approaches that have been most frequently used to validate SAS's requirements specification, model checking and theorem proving are the most popular.

7.1 Model checking

In the model checking the properties of the system are expressed in temporal logic. Model checker explores all the states reachable by the system, through an exhaustive enumeration over a given model, to prove that the design meets the specified properties. The model checker will return one of the three results automatically [46]: - the properties are satisfied by the design. - the properties are not satisfied, for that a counter example will be provided. - undetermined, the calculation time of the result is not reasonable by the tool because of state spaces. We can list a few works that employ model checking technique such as [12, 29], etc. The majority of model checking tools are created with existing or specialized languages, examples of some of these tools include: Spin (utilized to model asynchronous or concurrent software.[42]), Uppaal (applied to model real-time systems), SMV [44], NuSMV [43] (is applied to model synchronous digital logic), FDR (employed to model asynchronous systems [45]).

7.2 Theorem prover

Theorem proving is used for system verification. It gives mathematical reasoning for the accuracy of system properties. It simply uses restrictions to reason about the state space of the system, not all possible states of the state spaces of the system in order to prove properties [40]. We mention next a few works that use proof theorem such as [31, 34]. Theorem provers are called also "proof assistants" or "mechanical reasoning", that is a mixture of automated techniques and manual guidance to prove correctness. It uses a mathematical logic and artificial intelligence techniques [41]. The tools for model verification [31] are used to verify the correctness of the model (conception). That means the model is created then verified. The theorem prover tools involve the use of formal techniques in the creation of the design itself, and maintaining the required properties at every step of the design process. At the end, the design is known to be correct by construction. Some of the tools used in this topic are : Rodin [46], Rebicca [34], Isabelle/Hol Higher Order Logic (HOL) [49], etc.

[3] State Of The Affairs

7.3 Run-time Verification

In this case the systems will be abstracted in form of events (observation terms) and then they will be compared to formal specification of the expected behavior. In case the system behavior differs from the expectation then the monitor can take mitigation action, notify a human controller and can save logs of previous runs (behavior) for further studies on offline mode [56].

Table 2 Stages of verification of self adaptive systems

	Formal Approach				
Self adaptive system	Stage 1	Stage 2	Properties	Stage3,4	Ref
Robot house and care-o-bot	Brahms	LTL	Respond requirements	Spin[1]	[47]
ZNN.com	DTMC	PCTL	Resilience	Prism [1]	[26]
Traffic monitoring	CPN	M calculus	Safety,liveness	TAPA[1]	[12]
E-mobility vehicles	LTS model	FLTL	Safety, liveness	LTSA[1]	[35]
Art gallery building	-	AdaptiveCSP	-	FDR [1]	[29]
Multi-Mode traveler system	-	PCTL	Safety, performance	ProM [2]	[50]
Automatic rover protection	Event B	Event-B	Safety,liveness	Rodin[2]	[31]
Light control	PobSAM	LTL	correctness	Rebecca [2]	[34]
Building automation	GME	CTL, LTL	deadlock, stability	Isabelle[2] /Hol [2]	[49]
Robot scenario	Probabilistic State machine	PCTL	Reachability	Prism	[51]
Swarm robotics	Bio-PEPA	CTMC	Reachability	Prism [1]	[52]
Self adaptive robots	Z notation	Z model	respond requirements	CZT [1]	[53]
HTTP server architecture	B Method	FTPL,LTL	performance	AtelierB[2], ProB [1]	[54]
Fire fighting	-	LTL	safety	LTLMop[1]	[55]

[1]Model checking technique.

[2]Theorem proving technique.

Table 2 represents the summary of the papers used to presents this taxonomy, these papers were collected using a search strategy, which includes automatic and manual search. We used the following keyworks as a search string: "verification" and "formal methods" and "self-adaptive" and "model checker" and "theorem prover". In order to improve the search and find direct related works to the asked questions in the introduction, we used a manual search. The research strategy has two dimensions based on two questions when and where, selected papers are published between 2006 and 2020 in relevant scientific sources such as IEEE, springer, ACM. we have included studies that use formal methods in any stage of the verification of self-adaptive systems (Section 4), and exclude the rest of the papers that deal only about formal methods or self-adaptive systems. As well, studies that do not provide a reasonable amount of information were excluded. Table 2 introduces the formal methods used in each stage of verification, in each selected paper. We notice that the model checking and theorem proving are the most used techniques by most of the existing works to verify the various properties of the SAS. So, what are the differences between the two techniques, as well as their strengths and weaknesses? In model checking, we describe a shortened version of our system and we can automatically check certain properties. We must have a sufficiently

small model (number of states), to be treated using the tool, and the range of formulas (logic) we can express needs to be limited. In another way, using a theorem prover, can work on more precise representations of the system and express whatever properties. But most tests (proofs) must be hand-made, and this takes time and expertise. Model verification attempts to use brute force to answer the question and does not require any human interaction. The check is done through an exhaustive state space search to check if the wrong thing happens. In proving theorem, we try to give justification for why things can't go wrong in the form of theorems. However, we must also persuade the theorem prover that our reasoning is right. So, we must first understand which reasoning methods we use specifically. Therefore the advantages of the model checking are: Automatic execution (does not require any human interaction). It generates counter-examples, which usually reveal subtle design errors that would otherwise be difficult to find. The disadvantages however the followings are it can only deal with "small" models (models can become huge because of the problem of state space explosion). It can an only check properties expressed in logic, and not always effective while checking some properties, such as the stability of the adaptation process. The advantages of theorem proving are: Ability to handle arbitrary size model. It is able to verify any property. On the other hand, the disadvantages are: Manually (hand made). It may only be used by specialists. The advantages of Run-time verification (RV) are: RV is a "lightweight formal method". RV It may produce incredibly precise information about the behavior of the monitored system at run time. The disadvantage are: limited execution coverage (not the entire system, only snapshots)

8 CONCLUSION

Software runs our world, and is becoming sophisticated and infinitely complex. Our lives depend more and more on software ranging from cell phones, our cars to our medical devices. The effects of faulty software can be catastrophic, this is called mission critical system when failure or interruption may cause high financial loss, injury or even death .

The goal of formalization is to lower the possibility of major specification and design mistakes during the design of daily programs. Such mistakes can be discovered early on through analysis, while they are still affordable and simple to correct. Although comprehensive verification would be expensive and impractical, early mistake identification is a more practical aim. Formal methods appear to be a very powerful instrument. The cost of the specification must be significantly decreased, and the analysis itself must be automated, for the analysis to be economically viable. Until the advantages of formalization are instantly recognized, with an analysis that does not demand enormous extra effort, there will be no rationale for industry to embrace formal methods. At least when applied conventionally, current formal approaches are unable to accomplish these objectives. Due to the fact that formalization is promoted in highly expressive languages, its advantages are widely distributed. Using the formal approaches described in (section 5), it is exceedingly difficult and occasionally impossible to create a model for a self-adaptive system for which we do not know the behavior. In contrast, a light approach that prioritizes partiality and targeted application might result in higher benefits at lower costs and immediate outcomes.

Based on the questions mentioned in the introduction, our taxonomy provides a clear view of the existing approaches and formal methods that deal with adaptation at different levels, (see Section 3.3). The automation is the strength of formal methods, particularly model checking technique. Because of the need for a strong expertise to exploit the theorem proving technique and the fact that some proofs may need to be proven by hand, it makes it less used than the model checking techniques. The difficulties in theorem proving represent a challenge to researchers, to enhance the automation of it. The new emerged properties in SAS are consistency, responsiveness, mismatch, and loss-tolerance, which need a thorough analysis and verification. The verification techniques presented in this topic threat SAS mostly at design time. To conclude, we presented the limitations of the formal methods used, and described some research challenges in order to improve the formal verification of

self-adapting systems, because of SAS criticality and in order to get rapid outcomes while verifying them, the use run-time verification techniques (lightweight formal methods) is mandatory.

References

[1] IBM corporation, "Practical Autonomic Computing: Roadmap to Self Managing Technology," (2006), enterprise Management Associates, Inc.

[2] T. Gu, M. Lu, "Run time Models for Analysing and Evaluating Quality Attributes of SAS: A Survey,"12th International Conference on Relia-bility, Maintainability, and Safety",2018.

[3] M. Hachichaa, R. BenHalimaa, A. Hadj Kacem,"Formal Verification approaches of Self-adaptive Systems: A Survey,"23rd International Conference on Knowledge-Based and Intelligent Information and Engineering Sys-tems(2019).

[4] E. Yuan, S. Malek, "A taxonomy of self-protecting software systems,"7th International Symposiumon Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2012).

[5] C. Krupitzera, F. Maximilian, R. Sebastian, V. Gregor, S. C. Becker, "A survey on engineering approaches for SAS,"Pervasive and Mobile Compt-ing, vol. 17, pp. 186–206 (2015 February).

[6] A. Elkhodary, J. Whittle, "A Survey of Approachesto Adaptive Application Security,"International Workshop on Software Engineering for SAS (SEAMS '07)(2007 May)

[7] D. Weyns,M.U.Iftikhar, D.GildelaIglesia, T.Ahmad,"A survey of formal methods in self-adaptive systems","C3S2E '12: Proceedings of the Fifth International Conference on Computer Science and Software Engineering", june 2012.

[8] S.Mahdavi-Hezavehi, P.Avgeriou, D.Weyns,"A Classification Framework of Uncertainty in Architecture-Based SAS With Multiple Quality Requirements","Managing Trade-Offs in Adaptable Software Architectures", 2017.

[9] IBM Corporation,"An architectural blueprint for autonomic computing","Published in the United States of America 06-05",2006.

[10] L. Sabatucci, V. Seidita,"The Four Types of Self-adaptive Systems: A Metamodel","International Conference on Intelligent Interactive Multimedia Systems",2018.

[11] J. Zhang, Betty H.C. Cheng,"Model-based development of dynamically adaptive software","International Conference on Software Engineering 2006:371-380", 2006.

[12] M.I.Fakhir, S.Asad Raza Kazmi,"Formal Specification and Verification of Self-Adaptive Concurrent Systems"," IEEE volume 6", 2018

[13] M.Camilli, C.Bellettini, L.Capra," A high-level Petri net-based formal model of distributed self-adaptive systems", "the 12th European Conference on Software Architecture", 2018.

[14] Z. Guo, Y. Zhang, X. Zhao, X. Song,"A Timed Colored Petri Net Simulation-Based Self-Adaptive Collaboration Method for Production-Logistics Systems","Modeling, Simulation, Operation and Control of Discrete Event Systems", 2017.

[15] L. Ge, B. Zhang,"A Modeling Approach on Self-Adaptive Composite Services", "International Conference on Multimedia Information Networking and Security (MINES)", 2010.

[16] J. Chiachio, M.Chiachio, D.Prescott,"Modelling adaptive systems using plausible Petri nets","REC2018 Papers. Institute for Risk and Uncertainty, Liverpool University, p.103-109".

[17] Z.Ding, M.Zhou,"Modeling Self-Adaptive Software Systems by Fuzzy Rules and Petri Nets","IEEE Transactions on Fuzzy Systems (Vol: 26, Issue: 2, April 2018",2017.

[18] M.Camilli, A.Gargantini, P.Scandurra,"Specifying and verifying real-time self-adaptive systems", "IEEE 26th International Symposium on Software Reliability Engineering", 2016.

[19] Z.Ding, M.Zhou," Modeling Self-Adaptive Software Systems With Learning Petri Nets", "IEEE Transactions on Systems, Man,ernetics: Systems (Volume: 46, Issue: 4", April 2016.

[20] N.Cardozo, S.González, K.Mens, R.Van Der Straeten,"Modeling and Analyzing Self-Adaptive Systems with Context Petri Nets","Conference: Proceedings of the Symposyum on Theoretical Aspects of Software Engineering", July 2013.

[21] M.Usman Iftikhar, Danny Weyns,"A Case Study on Formal Verification of Self-Adaptive Behaviors in a Decentralized System","FOCLASA 2012 EPTCS 91, 2012, pp. 45-62", 2012.

[22] F. Cicirelli, L. Nigro, F. Pupo,"Formal Modelling and Verification of Real-Time Self-Adaptive Systems"," 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)", 2020.

[23] N.Li.Di Bai, Y.Peng, Z.Yang, W. Jiao,"Verifying Stochastic Behaviors of Decentralized Self-Adaptive Systems: A Formal Modeling and Simulation Based Approach", "IEEE International Conference on Software Quality, Reliability and Security (QRS)", 2018.

[24] A.Borda, V.Koutavas,"Self-Adaptive Automata", "FormaliSE ', Gothenburg, Sweden", 2018.

[25] J.O.Ringert, B.Rumpe,"A Requirements Modeling Language for the Component Behavior of Cyber Physical Robotics","Modelling and Quality in Requirements Engineerin", 2012.

[26] J. Cámara and R. de Lemos,"Evaluation of resilience in self-adaptive systems using probabilistic model-checking", " 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)",2012.

[27] Guo-You Zhang, Yin-Zhang Guo," A formal description of self-controlling software based on pi-calculus", " IEEE International Conference on Systems, Man and Cybernetics, 2004.
[28] S.JASKÓ, G.SIMON, K.TARNAY, T.DULAI, D.MUHI,"CSP-based modelling for self-adaptive applications"," info communication journal", 2009.
[29] A. Borda and L. Pasquale and V. Koutavas and B. Nuseibeh,"Compositional verification of self-adaptive cyber-physical systems","the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems", 2018.
[30] M.Hachicha, R.Ben Halima, A.H.Kacem,"Modelling, specifying and verifying self-adaptive systems instantiating MAPE patterns","International Journal of Computer Applications in Technology volume 57 issue 1", 2018.
[31] N.k.Singh, Y.Aït-Ameur, M.Pantel, A.Dieumegard, E.Jenn,"Stepwise Formal Modeling and Verification of Self-Adaptive Systems with Event-B. The Automatic Rover Protection Case Study","21st International Conference on Engineering of Complex Computer Systems", 2016.
[32] G.Edwards et al,"Architecture-driven self-adaptation and self-management in robotics systems","Workshop on Software Engineering for Adaptive and Self-Managing Systems, 2009.
[33] E. Cavalcante, J. Q.Louis-Marie,"Statistical Model Checking of Dynamic Software Architectures","European Conference on Software Architecture", Novermber 2016.
[34] N. Khakpour, R. Khosravi, M. Sirjani, S. Jalili,"Formal analysis of policy-based self-adaptive systems", "ACM Symposium on Applied Computing", March 2010.
[35] Dhaminda, B.Abeywickraman F.Zambonelli,"Model Checking Goal-Oriented Requirements for Self-Adaptive Systems", "19th International Conference and Workshops on Engineering of Computer-Based Systems", May 2012.
[36] Ji Zhang, Betty H.C.Cheng,"Specifying adaptation semantics","ACM SIGSOFT Software Engineering Notes 30:1-7",July 2005.
[37] E. Cavalcante, J.Q-Marie, T.F.Oquendo," Statistical Model Checking of Dynamic Software Architectures"," European Conference on Software Architecture",2016.
[38] J.Quilbeuf and E. Cavalcante and L.M. Traonouez and F.Oquendo,"A logic for statistical model checking of dynamic software architectures","Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques pp 806-820", 2016.
[39] M.Camilli, A.Gargantini, P.Scandurra,"Specifying and verifying real-time self-adaptive systems","26th International Symposium on Software Reliability Engineering", January 2016.
[40] P.Bagade, A.Banerjee,"Validation, Verification, and Formal Methods for Cyber-Physical Systems","AFSE-CIDSE: Computer Science and Engineering Computing, Informatics and Decision Systems Engineering, School of GIOS: Sustainability Initiative", 2017.
[41] M. Saqib Nawaz, Moin Malik, Yi Li, Meng Sun, M. Ikram Ullah Lali,"A Survey on Theorem Provers in Formal Methods", December 2019.
[42] G. J. Holzmann,"The Spin Model Checker","Addison Wesley", December 2004.
[43] A.Cimatti,E.Clarke ,F.Giunchiglia,M.Roveri,"NuSMV: a new Symbolic Model Verifier", "Conference on Computer-Aided Verification (N. Halbawchs and D. Peled, eds.), no. 1633 in Lecture Notes in Computer Science, (Trento, Italy), pp. 495–499, Springer",July 1999.
[44] K.McMillan,"Symbolic Model Checking","Kluwer Academic Press",1993.
[45] P.Broadfoot, B.Roscoe,"Tutorial on FDR and its applications","K. Havelund, J. Penix, and W. Visser, eds.), vol. 1885 of Lecture Notes in Computer Science, pp. 322–322", 2000.
[46] R.J.Punnoose, R.Armstrong, M.Wong, M.Jackson,"Survey of Existing Tools for Formal Verification","https://www.osti.gov/biblio/1166644",December 2014.
[47] M.Webster, C.Dixon, M.Fisher, M.Salem,"Formal verification of an autonomous personal robotic assistant", "Formal Verification and Modeling in Human-Machine Systems", 2014.
[48] Abdelfetah Saadi, Youcef Hammal, Mourad Chabane Oussalah, "A CSP-Based Approach for Managing the Dynamic Reconfiguration of Software Architecture", "International Journal of Information Technologies and Systems Approach (IJITSA)", june, 2021.
[49] R.AdlerIna, S.Tobias, S.Vecchié,"From Model-Based Design to Formal Verification of Adaptive Embedded Systems","International Conference on Formal Engineering Methods: Formal Methods and Software Engineering pp 76-95" ,2007.
[50] S.Jahan, A.Marshall, R.Gamble,"Embedding Verification Concerns in Self-Adaptive System Code","IEEE 11th International Conference on SA and Self-Organizing Systems",2017.
[51] Savas Konur, Clare Dixon, Michael Fisher,"Formal Verification of Probabilistic Swarm Behaviours","International Conference on Swarm Intelligence ANTS 2010.
[52] M Massink, M.Brambilla, D.Latella, M.Dorigo,"On the use of Bio-PEPA for modelling and analysing collective behaviours in swarm robotics "," Swarm Intell 7, 201–228 ",April 2013.
[53] D.Weyns, S.Malek, J.Andersson,"FORMS: a formal reference model for self-adaptation","the 7th international conference on Autonomic computing", June 2010.
[54] A.Lanoix, J.Dormoy,"Combining Proof and Model-checking to Validate Reconfigurable Architectures","Electronic Notes in Theoretical Computer Science",2011.
[55] V.RamanHadas, Kress-Gazit,"Analyzing Unsynthesizable Specifications for High-Level Robot Behavior Using LTLMoP","International Conference on Computer Aided Verification CAV 2011: Computer Aided Verification pp 663-668",2011.
[56] Ezio Bartocci, Ylie's Falcone, Adrian Francalanza, and Giles Reger Introduction to Runtime VerificationPart of the Lecture Notes in Computer Science book series (vol 10457)

Cloud Computing: Concepts and architecture

RAOUIA ELNAGGER[1] Med AMINE RIAHLA[2] and BOUGHACI DALILA[1]

[1] University Of Science and Technology Houarie Bouediene, Algeria
r.elnagger@gmail.com, dalila.boughaci@gmail.com
[2] University M'HAMED BOUGUARA of BOUMERDES, Algeria
ma.riahla@univ-boumerdes.dz

Abstract. The cloud computing is becoming increasingly popular in distributed computing environment. The success of Cloud Computing is mainly due to its on-demand and self-service nature. The main idea of cloud computing is to deliver remotely a vast supply of computing power, storage space, and fast network connections as computing utility just like how our essentials such as gas, water, and electricity get delivered to homes. Many papers in literature have presented the major benefits of the cloud, its services and its deployment models. However, there is a lake of detailed analysis of the underling architecture. As an attempt to fill this gap, we propose in this article an in-depthstudy of the cloud architecture. We address the internal architecture of a single node where we present the internal modules such as the hardware, the Operating system and the hypervisor as well as the communication between them to explain the provisioning process inside a virtual environment.

Keywords: Cloud Computing, Cloud Models, Virtualization, Cloud Computing Architecture.

1 INTRODUCTION

Lately internet has known a major evolution that led to the emergence of new types of network architectures witch are highly decentralized and provide self-organized services. This feature offers great advantages such as: the possibility to deploy and provide imperative applications and frameworks immediately, powerful and distributed computing resources as well as great storage capacities. The cloud computing is a new merging solution that allows users and companies to store and access to data or even applications using the Internet instead of a complete real hardware infrastructure. It has a distributed infrastructure of heterogeneous resources provided to the end users to be virtually accessible, easily usable as services. This flexible nature of the cloud is attracting more and more attention of many researchers who have tried to design different models allowing end-users to benefit from the power of this new paradigm. These models differ from one another according to: types of the delivered services, communication strategies as well as their physical architectures. It is therefore very important to carry out a deep study on the cloud computing to better understand its components, the interconnection between these components, how they can be managed, etc.

This article focuses on explaining the cloud infrastructure mainly the technical aspect of the communication between the Cloud components inside a virtual environment. The remainder of this article is as follows: Section 2 provides a definition of cloud computing models (business model, deployment models). Section 3 explains the global architecture of cloud computing and the interconnection between its components as well as the internal architecture of a single node. Finally, section 4 provides conclusions drawn from this study and presents our future works.

2 THE CLOUD COMPUTING MODELS

The cloud computing offers many features that distinguish it from other technologies which are as follow: On-demand Self-Services [1], Elasticity [2], Resource pooling [3], Broad network access[4], Measured services and Multi tenacity. It is often described as a stack of three services. Software as a Service (SaaS) is the top layer of the business model pyramid. It provides to clients applications as services which are accessible from any end-device over the network using only a web browser [5]. Platform as a Service (PaaS) is the second layer of the business model. It supplies customers' program development tools, Application Programming Interface (API), platforms and frameworks to develop their own applications and serves them by the upper model [6]. Infrastructure as a Service (IaaS) is the lowest layer. It provides the infrastructure to support SaaS and PaaS layers. Basically, IaaS offers to their customers the possibility to provision the claimant resources such as processing, network and storage in a flexible manner. Tenants can install everything they need but they should take the responsibility to configure and maintain the consumed resources. Since the cloud computing is designed to provide better utilization of resources using virtualization technology and to lighten the burden of work from client, many solutions are offered in the market. Each of them provides several benefits to enterprises such as elasticity, one- demand and low cost. Thereby, customers should choose the adequate cloud deployment model according to their requirements. Basically, cloud computing deployment models are classified into three main models which are: public [6], private and hybrid [7][8].

3 CLOUD ARCHITECTURE

The cloud computing is based on several concepts which are: virtualization, compute node, storage node, networking components and management nodes. The *Virtualization* is one of the most basic concepts of the cloud computing which provides an abstraction of hardware resources such as: CPU, memory, storage, network and software resources like: services, applications and operating systems [8]. We distinguish several virtual components such as: Virtual Machines (VMs) [9], Virtual Machine Image (VMI) [10] and the Hypervisor (Virtual Machines Monitor (VMM)) which can

consolidate physical resources into a virtual environment and shared them amongst virtual machines running on the same Host. Technically, there are two sorts of hypervisors:1)Hypervisor type I also known as “native hypervisor” or “bare metal” is installed directly on top of the hardware as an operating system of the physical host which facilitates provisioning resources and delivering them to the guest VMs. 2) Hypervisor type II (Hosted hypervisor) runs on top the operating system of the physical host [12]. Using virtualization allows reducing the costs significantly. However, it introduces new security challenges. In fact, by compromising the virtualization layer, all guests VMs will be infected. The *Compute nodes* are hosts that provide the computing resources required to run the guest VMs such as: CPU, memory, storage, and networking. Each compute node has its own hypervisor, either type 1 or 2, installed on it to host multiple VMs, enable resource pooling and multi tenancy [12]. The *Storage nodes* provide the storage capacities to the Cloud from the virtualization layer [13] such as such as instance storage, volume, template, snapshot and file system [14], [15]. The *management node* is the brain of the cloud that handles different types of requests. It takes place behind the load balancer to ensure a high availability to process clients’ requests [15].

The distributed architecture of the cloud relies on the use of networks which play a preeminent role in provisioning resources , which are located in different nodes scattered around the world, remotely to cloud tenants. Depending on the size and the purpose of the deployment of the aforementioned nodes; several architectures have been proposed. As case of study, an overview of the CloudStack deployment architecture is discussed.

3.1 CloudStack deployment architecture overview

The cloud computing resources are grouped in several levels which are: regions, zones, pods, clusters, hosts and the guest VMs. These levels provide the logical segregation of resources and facilitate their management [12], [26]. A guest VM is the smallest unit in the cloud computing deployment architecture. Each VM is an instance identified by an ID, address, etc. It can be assigned to tenants’ account remotely on demand. A Compute Node is a physical host that provides the computing resources to the guest VMs running on it. A Cluster is a group of Compute Nodes that possess the same hardware configuration, run the same hypervisor, are on the same subnet and share the same primary storage. The Pod is a dedicated rack of clusters. It can contain one or more clusters. The use of Pods is for administration purposes where the Pod network is used only by the CloudStack management servers. A Zone is the highest level of the CloudStack architecture. It is an aggregated endpoint for accessing hosts. It contains several Pods that share the same secondary storage. Secondary storage is dedicated to store Templates, VMI and Snapshots. It is associated to a zone and available to all Pods of that zone. Unlike the primary storage, secondary storage use only NFS to ensure its availability to any host in the zone. A Region is an optional level provided by CloudStack to group a community of zones that are close geographically.

Each region is controlled by a cluster of Management Servers running on one of its zones (see Fig.1). The advantage behind controlling a group of zones belonging to one region by their own nearby Management Servers is to decrease the latency of communications within the cloud compared to managing widely-dispersed zones from a single central Management Server.

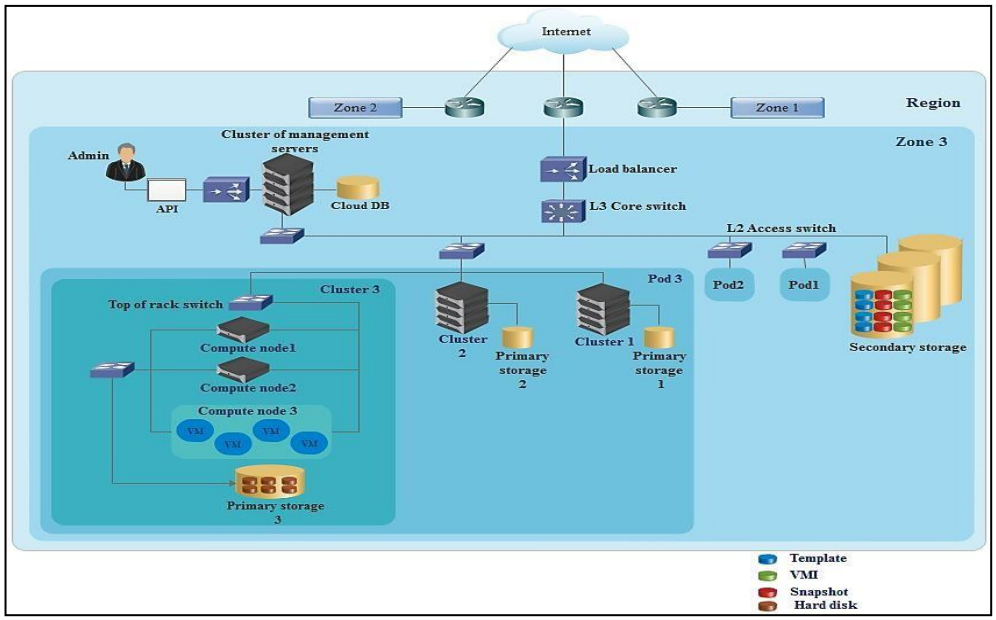


Fig. 1. Basic CloudStack deployment architecture

3.2 Logical architecture

The cloud computing relies on many sorts of nodes to provision IT resources to its tenants. The Computing node is one of them. Fig.4 illustrates a detailed architecture of the Compute Node which permits us to understand perfectly the communication mechanism between the Hypervisor and the guest VMs running on it.

Compute node architecture

In the following, we describe the different modules of a physical compute node such as the hardware module, the operating system module and the hypervisor module as well as the various connections linking between them for better understanding of the communication inside a virtual environment.

- Hardware module

The hardware module contains three main components: CPU, Memory and Storage resources.

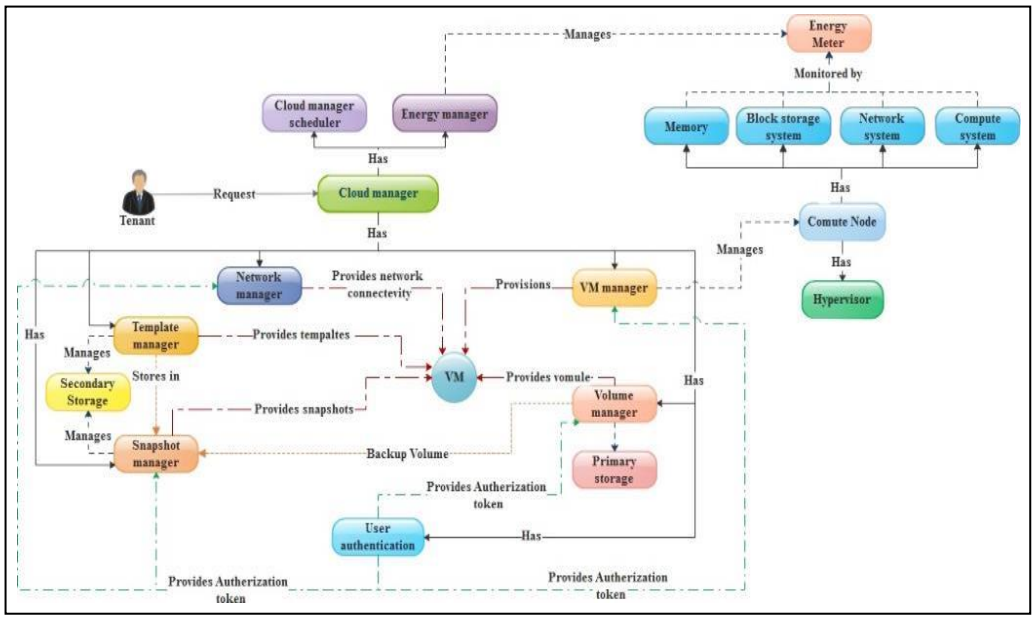


Fig. 2. Logical architecture of the cloud computing

- CPU module

The cloud computing relies on multi-processing where each node must have a powerful CPU to process data. Each CPU is composed of several cores assigned to VMs according to their needs. The CPU module communicates with the hypervisor through two interfaces: 1) FromOS[i]: receives the VM Operating System (VMOS) requests. 2) ToOS[i]: send the processing results to the claimant VMOS. Since the same CPU is shared amongst several VMs, each request will be labeled by an identifier composed of the process ID and the VM ID <ID number, ID VM>. When the request reaches the CPU module, it will be redirected to one of the available Cores. Each core receives VMOS's request through the interface In and sends its responses to the claimant VMOS through the interface Out.

- Memory module

The memory module is a temporary area where data are loaded for data processing. This module communicates with the hypervisor through four interfaces which are: 1) FromInput[i] and ToInput[i]: they permit to manage the memory like changing the current state (run, idle, off), allocating space into the memory or even release space from the memory. 2) FromOutput[i] and ToOutput[i] manage the Hard Disk requests which will be redirected to the File System of the VM in order to forward it to the storage module.

- Storage module

The storage module is responsible for saving data permanently on the hard disk of the current VM. In fact, the storage module is composed of several disks which are moni-

tored by a controller. The communication between the hypervisor and the storage module is guaranteed via two interfaces which are: 1) FromOS[i] and ToOS[i]: Each request sent to any hard device, it has to pass through the cache module (see Fig.4).

- Operating System modules

The operating system (OS) is the main brain of any node of the cloud computing. It is composed from several sub modules where each of them is responsible for a specific task. Whenever the OS receives/sends a request, it will be redirected to the corresponding sub modules according to the desired resource. If the physical node is managed using type1 hypervisor, the OS module is represented by the VM OS. However, if type2 hypervisor is deployed, the OS module is represented by the Node OS and VM OS.

- Node Operating System module

In the following a description of each sub module will be presented.

❖ *System Call Manager Module*

The System call manager module is responsible for redirecting requests to the corresponding sub modules. Each sub module represents a service such as CPU service, I/O service and Network service. The communication between the node System call manager and the other services is guaranteed through eight interfaces which are as follows:

- Communication between the system call manager and the CPU service: whenever an application requests the CPU, this request will be redirected to the CPU service using the interface ToCPU[i]. Each request is labeled by an <ID number, ID VM> since the CPU is shared between several VMs. The results of each processed request will be collected at the system call manager module through the interface From CPU[i].
- Communication between the system call manager and the I/O service: the communication between the system call manager module and the I/O redirector is guaranteed through two interfaces: FromIR[i] and ToIR[i]. These later are used whenever a local data is requested.
- Communication between the system call manager and the Network service: From Net[i] and ToNet[i] are two interfaces that insure the connection between the different nodes of the cloud computing. Using these interfaces, the System call manager can receives remote requests that need resources on the current physical node. Also, it allows the running OS to request remote resources on other nodes of the cloud infrastructure.
- Communication between the system call manager and the application: whenever and application attempts to requests any kind of resources, it sends a request to the system call manager through the interface Fromapp[i] and receives responses through the interface Toapp[i].

❖ *CPU scheduler module*

Each time an application needs data processing, it sends a request to the System call manager module which will redirect it later to the CPU scheduler using FromOS[i]

interface. When the request reaches the CPU scheduler module, it will be stored into a waiting list with other requests. Then, according to the scheduling policy, the pending requests are redirected to the Node CPU module using the interface ToCPU[i]. The results of each processed data are received through the interface FromCPU[i] and then redirected to the System call manager through the interface ToOS[i].

- *Volum manager module*

This module calculates the location of the storages involved in an I/O operation. Basically, each request will be fragmented into N sub-requests which are sent later to different storage servers. The volume manager must wait for all responses from the N sub-requests to be collected, and then sends a request to the corresponding file system. The volume manager module contains: 1) The Storage manager which calculates the location of the storages involved in an I/O operation, and sends requests to the corresponding storage server. 2) The storage scheduler permits to schedule the storage requests. 3) The Cache is used as a cache memory.

- *Node file system module*

This module translates I/O file requests expressed as (file name, offset, or request size) to a list of disk blocks. This module is able to distinguish between the different tenant accounts at the file system.

- Node Virtual file system module

This module acts as an I/O redirector in which user requests are redirected to the file system module (FS) or to an application module. The requests are redirected based on a path table. Each path is associated with an index and a module type. An index corresponds to one of the file systems (FS) stored at the physical node or to one of the applications running on that node. When a local I/O operation is required, the I/O redirector sends the request to the local file system. However, if a remote operation is required, the remote operations are sent to the corresponding tenant application. This module supports tenant management.

- *Node state module*

This module implements an application system which is responsible for managing the state of the hardware devices: running, pending and off.

- *Remote storage module*

This module is an application system that requests the system call manager to create a connection between the compute node and a remote storage server, loading the file system structure and tenant files from preloaded files (files that are going to be loaded before a user application execution).

- Virtual Machine Operating System module

The VM Operating System is a guest host installed on top of the physical host OS. It possesses the same modules as the operation system of the physical host except *the Node state module* and *the Remote storage module*. These later are replaced by the *VMMS Controller module* (Fig.3). The only difference between the VMOS and the Host OS resides into the fact that the *VM System call manager* must pass by the *VMMS Controller module* to communicate with the host OS.

- *VMMS Controller module*

VMMS Controller is the VM message controller. It is responsible for monitoring every message generated by the Virtual machine. This module possesses four interfaces where two of them are used to communicate with the VM Syscallmanager. Whenever and application request any hardware resources, this query will be redirected to the VMSyscallmanager using the interface ToOS[i] and the response will be received from the VMSyscallmager through the interface FromOS[i]. Each request will be labeled by an ID in order to make the difference between the different applications. The response corresponding to each request will be send to the VMMSController via the interface ToApp[i] and received at the VMMSController module through the interface FromOS[i].

The two other interface which are FromApp[i] and ToApp[i] are used if a remote application requests a resource from the current VM.

- Hypervisor modules

The hypervisor as mentioned above is a critical component of the virtualization environment. It manages the communication between the hardware and the VMs operating systems when it takes form of a native hypervisor. As well as the role of intermediate software between the physical node operating system and the VMs operating systems if the hosted virtualization approach is deployed. The hypervisor aims to provision the needed resources for the guest VM Operating system. In the following, an explanation of the hypervisor behavior according to type 1 hypervisor deployment approach.

- *CPU manager module*

This module manages the communication between the OS CPU scheduler and the Node CPU. The CPU manager has four interfaces where two of them are used to communicate with the CPU scheduler module and the others insure the communication with the node CPU module. Whenever a node OS or a VMOS sends a CPU request to the Hypervisor, it will be received at the interface FromVMCPU[i]. Later, this request will be redirected to the Node CPU using the output interface ToNodeCPU[i]. Each request will be labeled by an <IDnumber, ID VM>. When the request is processed, the hypervisor receives the response from the CPU module through the interface FromNodeCPU[i]. Then, the received response will be redirected to the OS using the interface ToVMCPU[i].

- *Network manager module*

This module is composed of two sub-modules: local net manager and network manager. The Local net manager is responsible for translating the virtual ports and virtual IP addresses to real physical ports and IP addresses. However, the Net manager manages the order of the arrival messages from VMs to access to physical resources. In fact, the net manager can communicates with VMs through six interfaces which are as follow:

- FromVMNet[i],ToVMNet[i],FromNodenet[i] and ToNodeNet[i]: these four interfaces are used in order to connect the VMs network with the Network service that will be explained later.

- From Hstorageserver[i] and ToHstorageserver[i]: they are interfaces used to communicate the Network manager with the storage manager. In fact, when the storage manager decides to send a request to the Node File System, it forwards its query to the net manager using the interface From Hstorageserver[i]. Besides, when the network manager receives a request to the Node File System, it will re-direct it to the storage manager using the interface ToHstoragemanager[i].

❖ *Network service module*

The network service is a module that controls the net device state and manages the remote communications with other VMs or physical nodes. It is able to establish, suspend or close a connection between nodes (VMs) that need distributed resources.

❖ *Memory manager module*

This module manages the memory component. It provides the possibility to know the number of VMs allocated on the same physical node, exchanging local data or even migrating the memory contents to other node. The memory manager communicates with other modules through eight interfaces:

- FromVMmemoryI[i], ToVMmemoryI[i], FromNodememoryI[i] and ToNodememoryI[i]: these four interfaces permit to exchange request/response between the Node memory and the VM System call manager module of the VMOS.
- FromVMmemoryO[i],ToVMmemoryIO[i], From-NodememoryO[i] and ToNodememoryO[i]: these four interfaces permit to exchange request/response between the Node memory and the VM I/O redirector module. Each request will be labeled by an identifier <ID number, ID VM>.

❖ *Storage manager module*

The storage manager is one of the main modules of the hypervisor. It is responsible for monitoring the storage space on the physical node as well as on the Virtual machines. When the VMs are linked between each other, they create storage cells. These later are controlled by the storage manager. Moreover, this module provides the possibility to used remote storage cells of the cloud computing infrastructure.

Whenever a node OS or a VMOS sends a storage request to the Hypervisor, it will be received at the interface FromVM Storage Server[i]. Later, this request will be redirected to the Node storage using the output interface ToNode Storage Sever[i]. Each request will be labeled by an <IDnumber, ID VM>. When the request is processed, the hypervisor receives the response from the storage module through the interface From Node Storage Server[i]. Then, the received response will be redirected to the OS using the interface ToVM Storage Server[i].

4 CONCLUSION

The cloud computing is a promising paradigm for delivering IT resources as computing utilities. These resources are provided to end-users on-demand and through a pay- as-you-go model. This innovation permits to the cloud users as known as tenants to benefit from a huge computing capability, storage capacities and network connec-

tion without taking care of the management burden. In this paper, we present a comprehensive taxonomy of the cloud computing, its features that distinguishes it from other technologies, its services commonly referred to as SAAS, PAAS and IAAS and its deployment models that differs according to the users' needs from public to private or even hybrid cloud. Furthermore, we present the architecture of one of the most important layer that the cloud computing relies on which is the underlining infrastructure.

Despite the large amount of studies about the cloud architecture, most of them has encompass the general architecture where the essential components have been presented such as compute nodes, storage nodes and management nodes as well as their inter-connection to build a distributed network. Yet, none of them has discussed the internal modules of a single node and how they can communicate to provision the requested resources. This is why, in this article, we present the different modules containing inside a single node which are the hardware module, the Operating System module in its different forms (host OS and VMOS) as well as the Hypervisor module which is a critical component in a virtual environment. We are convinced that this study will provide to researchers a good understanding of the underling infrastructure in order to improve the provisioning process.

Due to the various benefits of cloud computing, many organizations and companies migrate to this new technology. However, the increasing uses of the cloud computing reveals also new security risks. The new concepts provided by the cloud computing such as multi- tenancy, resource sharing and outsourcing, create new challenges to the security community. The security issues in the cloud computing platform can cause serious economic losses and will damage the reputation of its providers, especially if this platform is geared towards the large public. Addressing these challenges requires a heavy investment in IT risk assessment to ensure that customers' data are well protected and to establish reliable bases and standards for securing this infrastructure. Thereby, we focus our future work on the security issues in terms of the threats being faced by cloud platforms and their possible countermeasures. The detailed architecture presented in this paper, will help us to encounter the security risks and improve the security policies in such environment.

References

1. Z. Mahmood, "Cloud Computing: Characteristics and Deploy- ment Approaches", 11th IEEE International Conference on Computer and Information Technology, IEEE DOI 10.1109/CIT.2011.75, 978-0-7695-4388-8/11 $26.00 © 2011

2.W. Yassin, N.I. Udzir, Z. Muda, A. Abdullah4 and M.T. Abdul- lah, "A Cloud-Based Intrusion Detection Service Framework, Cyber Security", Cyber Warfare and Digital Forensic (Cyber- Sec), 2012 International Conference on, ISBN: 978-1-4673-1425- 1, 26-28 June 2012

3. P. Mell, T. Grance, "the NIST Definition od Cloud Computing, Recommendations of the National Institute of Standards and Technology",September 2011.

4. SWAT, "SWAT White Paper Cloud Computing", 6-11-2012

5. M. Janssen, A. Joha, “CHALLENGES FOR ADOPTING CLOUDBASED SOFTWARE AS A SERVICE (SAAS) IN THE PUBLIC SECTOR”. European Conference on Information Systems, ECIS 2011 Proceedings, 10-6-2011
6. Q. Zhan, L. Cheng, R. boutaba, “Cloud computing: state of art and research challenges”, J Internet Serv Appl (2010) 1: 7–18, DOI 10.1007/s13174-010-0007-6
7. Z. Mahmood, “Cloud Computing: Characteristics and Deploy- ment Approaches”, 11th IEEE International Conference on Computer and Information Technology, 2011
8. Cloud Strategy Partners, LLC , “Cloud Service and Deployment Models”, IEEE Educational Activities and IEEE Cloud Compu- ting.
9. M. Kazim, R. Masood, M.A. Shibli, A.G Abbasi, “Security Aspects of Virtualization in Cloud Computing”,
10. D.A.B Fernandes, L.F.B Soares, j.V. Gomes, M.M.Freire, P.R.M Inacio, “Security Issues in Cloud Computing: Survey”, International journal of Information Security (ijis), ACM Volume 13, Issue: 2, pp: 113-170, April, 2014
11. M.Fenn, M.A. MURPHY, J.Martin, S.Goasguen, “An Evaluation of KVM for Use in cloud computing”
12. Apache CloudStack, “CloudStack Documentation”, release 4.8.0, Chapitre: Introduction, February 02, 2017
13. IEEE Cloud Computing, “Storage Virtualization in Cloud”
14. B. Salmon, “Understanding cloud storage models”, Jan 20, 2015
15. N. Sabharwal , “Apache CloudStack Architecture”, June 2013.

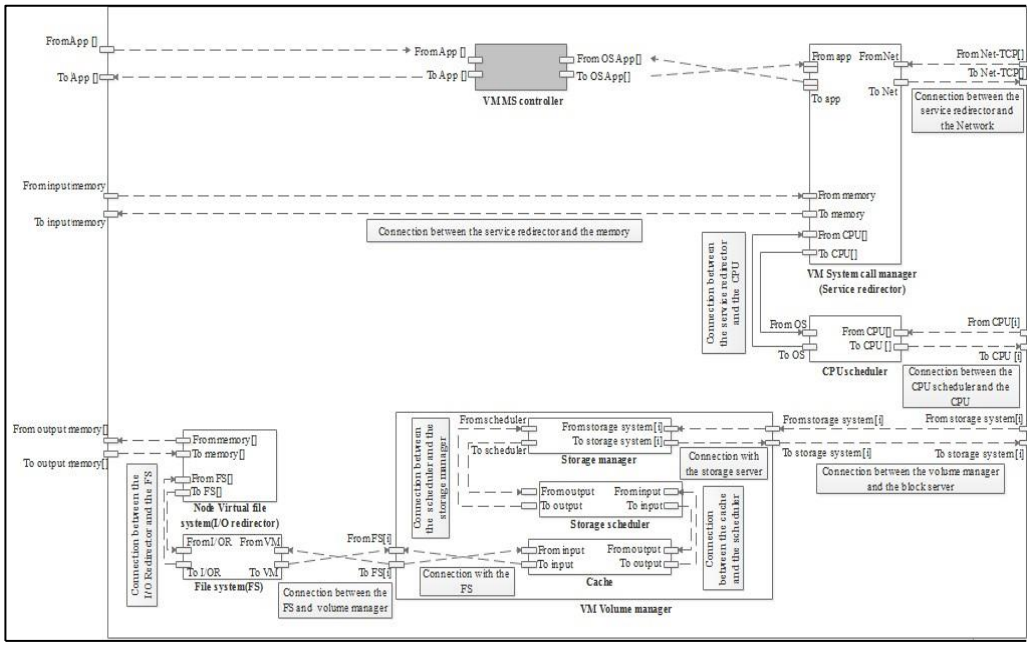


Fig. 3. VM operating system module

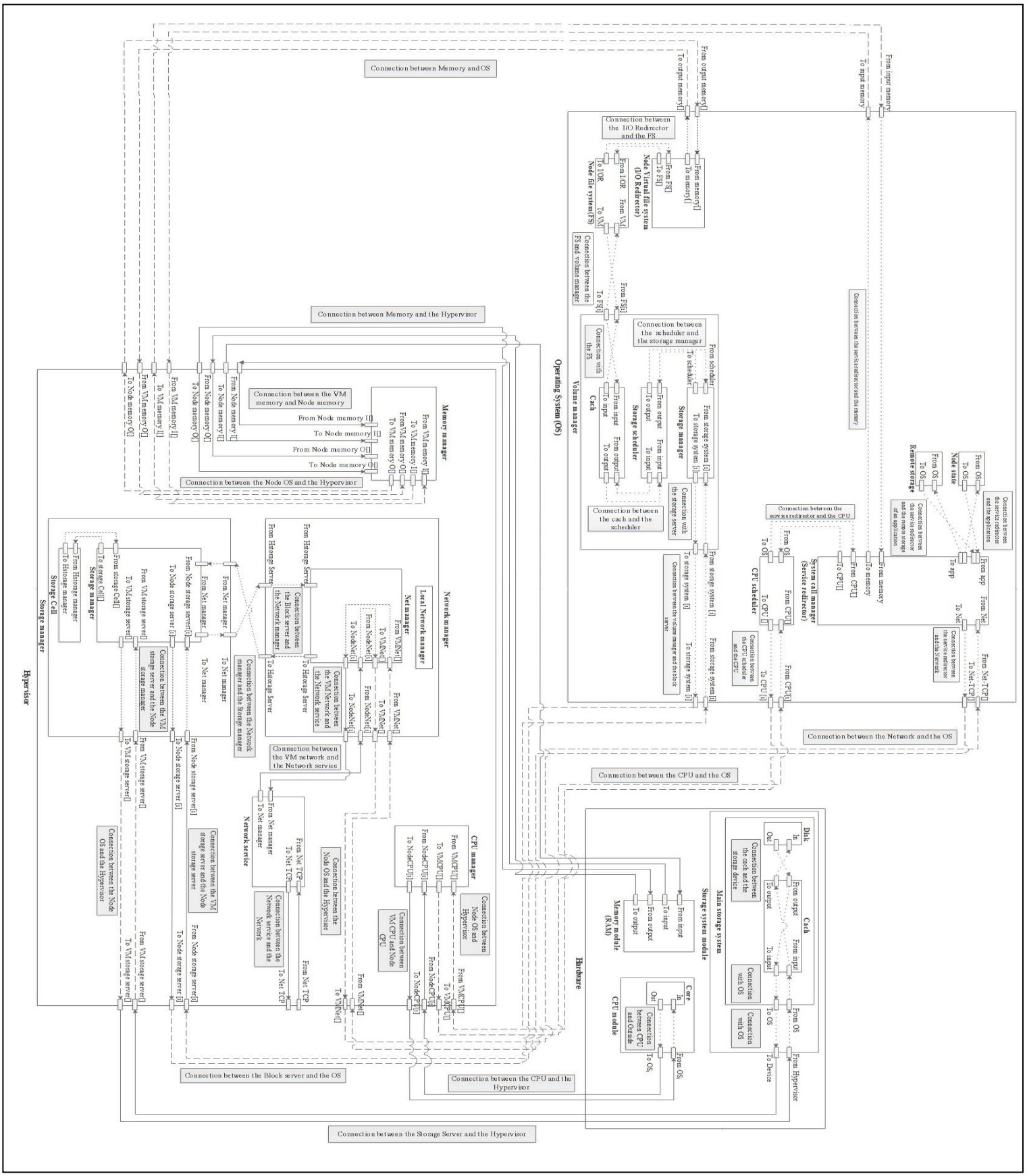


Fig. 4. The global architecture of a compute node

Internet Traffic Classification using Deep Neural Network

[1]Ahmed.KROBBA, [2]Mohamed. DEBYECHE,[3]Adil BAKRI
[1,2]Speech Communication and Signal Processing Laboratory University of USTHB, Algiers, Algeria
[3]Scientific Research and Technical Center for the Development of Arabic Language, CRSTDLA, Algiers, Algeria
akrobba@usthb.dz, mdebyeche@usthb.dz, adil.msilib@yahoo.fr

***Abstract*— Internet has become the global means of communication for transmitting all kinds of traffic data; this is what makes it develop rapidly, which means the network traffic data increases greatly. In order to manage the data circulating in the networks it is necessary to establish an analysis of the network traffic which makes it possible to provide detailed and explainable data on this traffic and the consumption of the bandwidth, generally by applications and their protocols. In this paper, we analyse and classify internet traffic using the «Deep Neurale Network » method and compare their performance to that of the artificial neural network. The results show that the proposed method is able to accurately and stably classify the content types of network traffic compared with to the ANN.**

Keywords-Internet traffic classification, deep neural network DNN, ANN.

1. Introduction

Internet use has become indispensable over the past decade and the enormous growth of users and data on the Internet poses a challenge to knowing the flow and movement of information. . So, traffic is something you shouldn't be without. In the past, there has been an evolution in the number of data streams. At the same time, there has been an evolution of procedures and tools for knowing the flow and classification of this data. [1]. There are many studies that have been done on this topic that have led to many different approaches, such as, the known port number based approach to identify the nature of internet traffic. Early work in ML- based traffic classification by Moore and Zuev [2] proposed to apply the supervised Naıve Bayes techniques using discriminators derived from packet headers for categorizing traffic. Previous research [11], [3], [4], [5] [6], found that, in traffic classification problems. In [7] Karagiannis et al presented a method that relies on network and transport layer behavior to identify P2P traffic. In [8], authors propose a method for classification of traffic based on statistical application signatures. In [9], [10] [12] used te bayesian trained neural network and machine learning to Internet classification. In [13] proposed the traffic classification, named ITCGAN based network traffic classification which can generate traffic samples for minority classes and train the optimal classifier. The multi-class traffic classification approach based on SVM was investigated by Peng, X et al [14]. The are many algorithms of machine learning techniques using the same set of features can achieve similar classification accuracy. Yu Wang et al [15] proposed the clustering scheme that makes decisions with consideration of some background information in addition to the observed traffic statistics while traffic classification techniques are improving in accuracy and efficiency. The new method was also proposed to effectively aggregate the correlation naive Bayes (NB) predictions a novel traffic masking method, called Generative Adversarial Network (GAN) tunnel, to protect the identity of applications that generate network traffic from classification by adversarial Internet traffic classifiers [16]. Recent works have applied CNN for traffic classification [17] [18]. Manuel L. M et al present the first application of the RNN and CNN models to network traffic classifier based on for Internet of things [19]. In the deep learning, especially deep neural networks (DNNs), are currently being used on many fronts such as speech recognition, facial recognition in social networks, automated cars and even some diagnostics in the field of Medicine [20]. Deep learning is considered a branch of machine learning, which is based on a group of algorithms that seek detailed information about the characteristics of data using a deep architecture with multiple layers of processing. In this study, we propose an approach based on deep learning or the artificial and deep neural network (ANN and DNN) which has served us to develop a lightweight model with high classification precision, low error and good generalization of data. This model is first built using training data and then is used for the classification phase of another dataset. Our work is aimed at the analysis and the characterization or the identification of the traffic by statistics of the flows of the TCP / IP headers in which the network traffic is classified in category (classes) based on the application protocol (for example: http, smtp… etc.). To improve our models, we used regularization to combat overfitting and reduce generalization error (validation error). One of the techniques of regularization is the dropout which we have used in this phase of improvement. At the end of this

study, we validated and evaluated the performance of our classifiers with four metrics which are precision, recall, f1-score and accuracy, respectively. The remainder of this paper is organized as follows. In Section.2, presents the classification du trafic and the transfer of the data or the traffic in the internet network. In Second.3 presents the experimental results. Finally, the conclusion and future work are presented in Section 4

.

2. Classification du traffic

The analysis of internet network traffic has become more and more vital and important nowadays for monitoring the traffic flow in the network. In recent years, administrators had been born monitoring only a small number of network devices or less than a thousand computers. The network bandwidth was maybe just less than or equal to 100 Mbps (Megabits per second). Currently, administrators have to deal with high speed wired network over 1Gbps (Gigabits per second) and various networks such as ATM (Network Asynchronous Transfer Mode) and wireless networks that require more traffic analysis tools. Powerful to manage them and solve their problems quickly. Thus, to avoid breakdowns, and take care of security. Analyzing network traffic has presented a number of challenges in recent days. The network is analyzed at different levels, namely packet level, flow level and network level for security management [11]. A common method of traffic analysis is to capture all the packets on a network for a certain length of time and analyze them later. Offline analysis allows for complex calculations because it does not need to run at line speed. Tcpdump [14] is the most common tool for acquiring packets entering and leaving a machine. It captures all packets on a specific network interface, and provides filters to store only specific packets. Therefore, an important parameter of Tcpdump is the number of bytes captured for each packet which are stored in a format called Pcap. The latter stores all packets with an additional header that includes a timestamp, the actual size of the captured packet, and the number of bytes

3. Experimental results

In the section, we present the different tests and results that are obtained from the implementation and evaluation of models based on artificial and deep neural networks, to classify or in other words predict certain application protocol the most present in an Internet network according to their characteristics. The first part describes the test environment and the tools used to manipulate the dataset. The second part presents the construction of the supervised classifiers and the tests carried out by characteristic vectors taken randomly from the packet stored in the trace. Tcpdump can capture traffic from each host, but it can also be used to capture traffic from an entire network. An administrator can run Tcpdump on a machine monitoring network that receives all packets entering and leaving the network. Traffic classification is an area of research that helps us classify flows carried over the internet or any network. It consists in examining and analyzing the IP packets to extract functionalities allowing to answer certain questions relating to its origin and its nature, to the transported content or to the user intentions. It often deals with packet streams (TCP and UDP streams) defined as sequences of packets uniquely identified by the same source IP address, same source port, same destination IP address, same destination port, and same transport layer protocol. However, the packets can be grouped in any way depending on classification needs. The classification of traffic or traces is important for the management of computer networks: for example, it is used for traffic shaping and packet filtering. Specifically, some methods classify traffic according to its category, i.e. whether the traffic represents bulk forwarding, peer-to-peer (P2P) content sharing, gaming, multimedia, web, or the attacks. The second group of methods classifies traffic according to the exact application that is driving the traffic, like Skype, eBay, Emule. The last group of methods aim to identify the protocol at the application level, such as FTP, HTTP, SSH, Telnet, also is the finer classification. Figure.1 shows typical classification objectives or in other words three different domains.

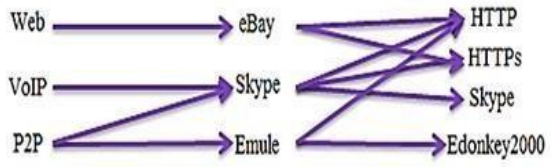


Fig.1: Objectives of the classification

3.1 Description of the database

The data we used for the two phases of the classification (learning and testing) comes from the traffic database maintained by the MAWI working group of the WIDE project. A working group has carried out the measurement, analysis and evaluation and verification of network traffic since the start of the WIDE project [21]. This database corresponds to daily captures lasting 15 minutes of the data flow connecting the research center of the WIDE project and their internet service provider (Japan). These are obtained with the TCPDUMP utility and saved in PCAP format. After downloading MAWI group traces, we have limited them in order to extract their characteristics because they are very large. These limited traces are analyzed using

Wireshark [21] to filter the packets corresponding to the protocols of interest in this study. The characteristics are calculated for each protocol with the CICFlowMeter utility [22].

3.2Features extractions

This step consists in calculating the characteristics for each PCAP file analyzed and filtered by Wireshark (DNS, HTTP, HTTPS, SMTP, SSH), to do this we used the CICFlowmeter application. CICFlowMeter is a downloadable Windows application that generates network traffic flow. It can be used to generate bidirectional streams from PCAP files, and extract features from these streams. So you just have to give in input the PCAP file that you want to know their characteristics and F1 score interpreted as a weighted harmonic mean of precision and recall, it is given by the following formula

$$F\ -score = 2*\frac{Precision*\ recall}{Precision + recall} \qquad (1)$$

Thus accuracy which represents the success rate and how correct an answer is, it is given by the following formula:

$$Accury = \frac{v_p + v_n}{v_p + v_n + f_p + f_n}$$

3.3Testing and evaluation

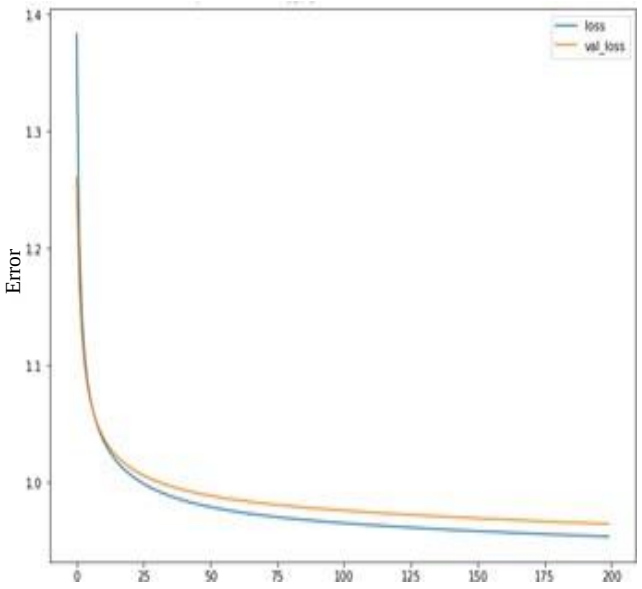
This step consists of classifying the characteristic vectors of unknown connections using our classifiers, thus evaluating the performance of our networks. As shown in the figure above, we used the predict function to make predictions on new data or test data. This is the most important step of the program because it will allow us to properly classify the characteristics to predict the protocols and therefore know the corresponding application (identification of internet traffic). The performance evaluation of our classifiers is done with four metrics which are respectively precision, recall, f1-score and accuracy. Precision is the ratio where is the number of true positives and the number of false positives, it represents the ability of the classifier not to label a negative sample as positive, it is given by the following formula:

$$Precision = \frac{v_p}{v_p + f_p} \qquad (3)$$

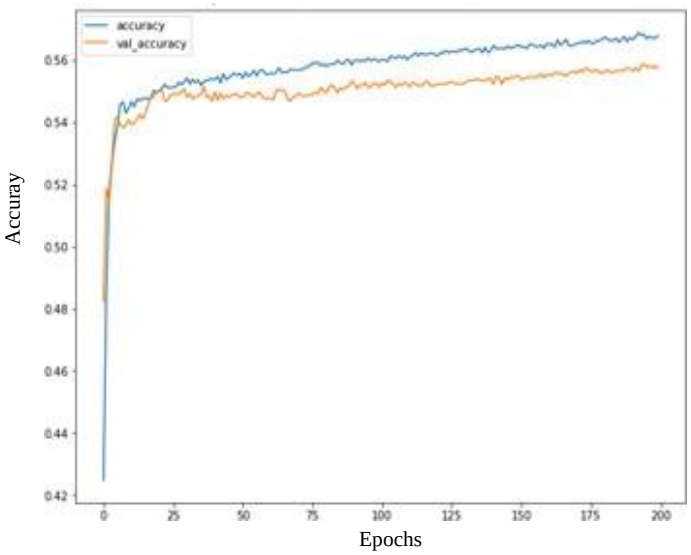
The recall is the ratio where is the number of true positives and the number of false negatives, represents the ability of the classifier to find all the positive samples and the rate of true positives, i.e. the proportion of positives that we correctly identified.

In the Figure.2 and Figure.3 presents the error and precision (accuracy) for ANN and DNN

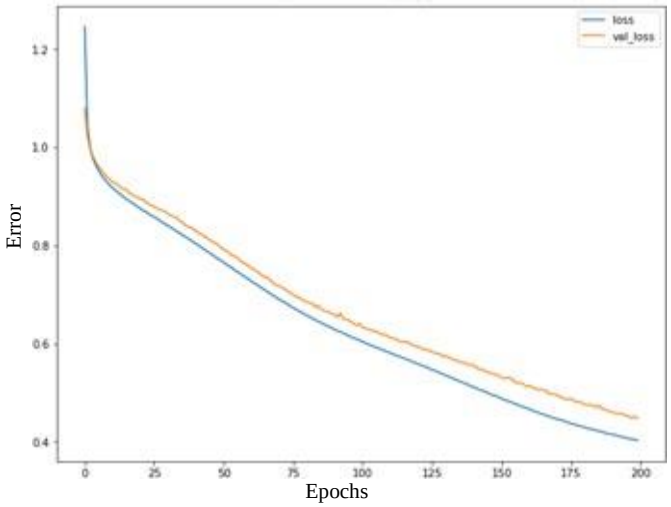
(Single-layer ANN model)



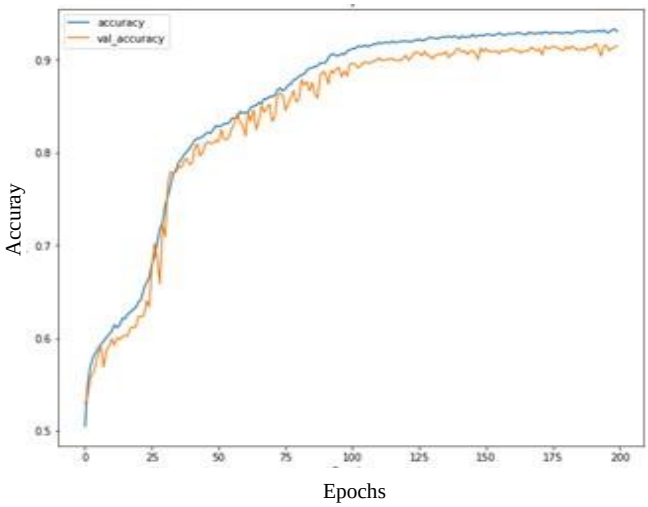
(single-layer ANN model)



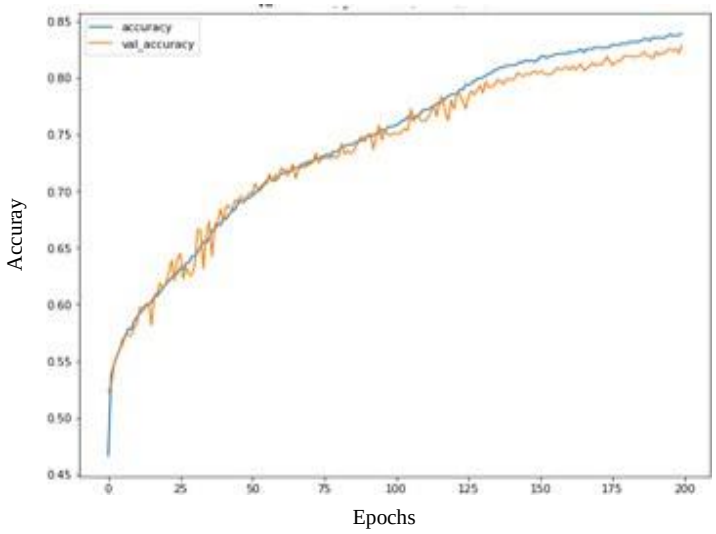
(single-layer ANN model)



(DNN-2)



(single-layer ANN model)



(DNN-3)

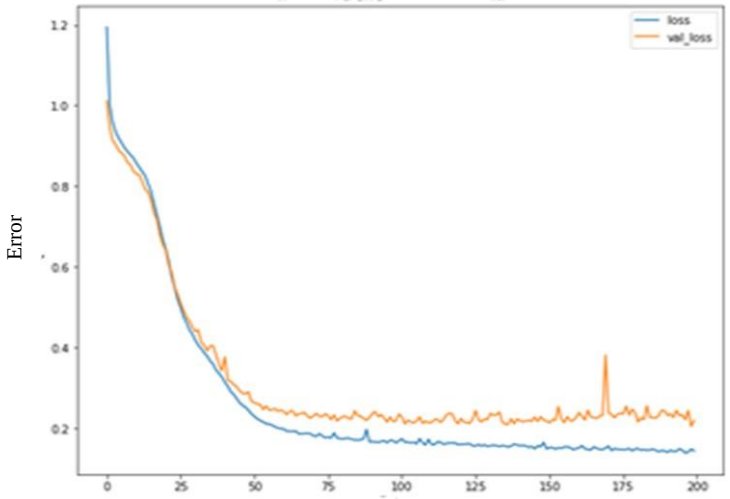
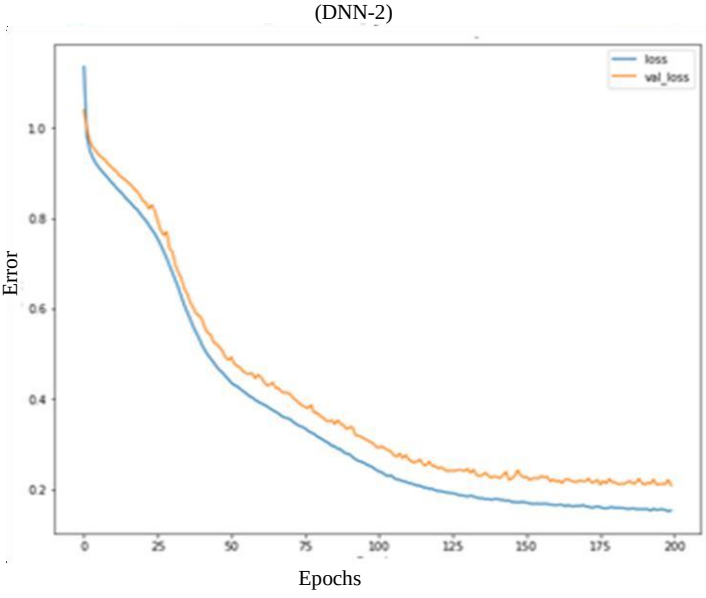


Fig.2. Error and precision (accuracy) for ANN model

(DNN-2)



(DNN-3)

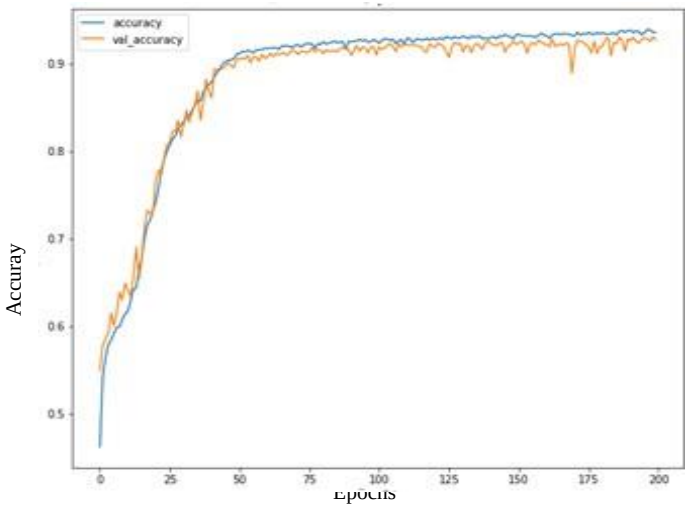


Fig.3. Error and precision (accuracy) for DNN model

From the Figure.2 and Figure.3, we see that for the single-layer ANN model, ANN-1 and DNN-2, the loss function continues to decrease and approach zero at epoch 200, which means that the model is well trained for s' adapt as best as possible to the training data, thus reducing the validation error (val-loss). From the DNN-2 model, we notice that after a certain number of epochs we see variations and error peaks return to the rapid increase of this error, such as the error of validation is pupil compared to the training error. We also noticed for the precision from the DNN-3 model, the appearance of certain peaks amounts to the decrease in this metric. Such as the training accuracy (accuracy) is high and the validation accuracy (val-accuracy) is low, and this is when the models start to overfit, i.e. they memorize training results and cannot generalize data that they have not seen before.

•For the single-layer ANN model, ANN-1 and DNN-2 val_loss continues to decrease and val_acc continues to increase, this means that the built model is learning and functioning correctly.

4. Conclusion

Analysis of internet traffic can provide important information to operators and network administrators, one of the main objectives is to identify the traffic mix conveyed by the network, this identification allows internet service providers to optimize resources as well to improve quality of service to customers. The objective of this thesis was on building a model based on deep machine learning methods, which identifies the main application protocols transported by the TCP protocol in internet traffic. This model is based on deep neuron networks, allow to determine the membership class for new connections. A MAWI group database was used in this study. Five daily catches are downloaded and analyzed to extract the different characteristics for each protocols. At total, the dataset for the five protocols contains 20,000 connections for training the model as well as 4000 connections for testing and evaluation. As part of our future work, we intend to implement the part of our framework in Software Defined Networks (SDN).

REFERENCES

[1] A. Sivanathan et al., (2019). Classifying IoT devices in smart environments using network traffic characteristics,‖ IEEE Trans. Mobile Comput., vol. 18, no. 8, pp. 1745–1759.

[2] Zuev, D., Moore, A. W. (2005). Traffic classification using a statistical approach. In International workshop on passive and active network measurement (pp. 321-324). Springer, Berlin, Heidelberg.

[3] .H. Shi, H. Li, D. Zhang, C. Cheng and X. Cao, (2018). An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification,‖ Computer Networks, vol. 132, no. 81–98.

[4] .C. Yu, J. Lan, J. Xie and Y. Hu, (2018). QoS-aware Traffic Classification Architecture Using Machine Learning and Deep Packet Inspection in SDNs,‖ Procedia computer science, vol. 131, pp. 1209–1216.

[5] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares et al., (2016). Machine learning in software defined networks: data collection and traffic classification,‖ in Network Protocols (ICNP), IEEE 24th International Conference on, pp. 1–5.

[6] Umair, M. B., Iqbal, Z., Bilal, M., Almohamad, T. A., Nebhen, J., & Mehmood, R. M. (2021). An Efficient Internet Traffic Classification System Using Deep Learning for IoT. arXiv preprint arXiv:2107.12193.

[7] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, (2004). ―Transport layer identification of P2P traffic,‖ in Proc. Internet Meas. Conf., Sicily, Italy, Oct. 2004, pp. 121–134.

[8] .M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, (2004). ―Class-of- service mapping for QoS: A statistical signature-based approach to IP traffic classification,‖ in ACM SIGCOMM Internet Meas. Conf., Sicily,
Italy, 2004, pp. 135–148

[9] .Auld, T., Moore, A. W., & Gull, S. F. (2007). Bayesian neural networks for internet traffic classification. IEEE Transactions on neural networks, 18(1), 223-239.

[10] Nguyen, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. IEEE communications surveys & tutorials, 10(4), 56-7.

[11] Salman O, Elhajj IH, Kayssi A, Chehab A (2020) A review on machine learning–based approaches for internet traffic classification. Ann Telecommun:1–38

[12] Salman O, Elhajj IH, Chehab A, Kayssi A (2019) A machine learning based framework for iot device identification and abnormal traffic detection. Trans Emerg Telecommun Technol 0(0):e3743

[13] Guo, Y., Xiong, G., Li, Z., Shi, J., Cui, M., & Gou, G. (2021). Combating Imbalance in Network Traffic Classification Using GAN Based Oversampling. In IFIP Networking Conference (IFIP Networking) (pp. 1-9). IEEE.

ACQAD: A Dataset for Arabic Complex Question Answering

Abdellah Hamouda Sidhoum[1*], M'hamed Mataoui[1], Faouzi Sebbak[1], and Kamel Smaïli[2]

[1] Computer Science Department, Ecole Militaire Polytechnique, Algiers, Algeria
[2] SMarT Group, Loria, University of Lorraine, France

Abstract. In this paper, we tackle the problem of Arabic complex Question Answering (QA), where models are required to reason over multiple documents to find the answer. Indeed, no Arabic dataset is available for this type of questions. To fill this lack, we propose a new approach to automatically generate a dataset for Arabic complex question answering task. The proposed approach is based on using an effective workflow with a set of templates. The generated dataset, denoted as ACQAD, contains more than 118k questions, covering both comparison and multi-hop types. Each question-answer pair is decomposed into a set of single-hop questions, allowing QA systems to reduce question complexity and explain the reasoning steps. We then provide a statistical analysis of the produced dataset. Afterwards, we will make the corpus available to the international community.

Keywords: Question answering, Arabic complex questions, QA dataset.

1 Introduction

Question Answering (QA) is a challenging task in natural language processing (NLP) that is used to evaluate machine reading comprehension (MRC). QA systems are designed to provide short answers to questions formulated in natural language. The majority of QA researches focus on single-hop QA, where a single paragraph is supposed to be sufficient to answer the question. Although, models' performance has been further boosted in recent years, particularly since the introduction of machine learning techniques such as BERT[5], they still lack the ability to perform multi-hop reasoning across multiple documents. A Multi-hop system has to aggregating dispersed pieces of evidence to predict the right answer (sentences highlighted in *blue italic* in Figure 1). In this example, the question can not be answered by matching its tokens with a single sentence in one paragraph. This area of research has recently received considerable attention, especially since the release of large-scale complex QA datasets, such as hotpotQA. [14] and ComplexWebQuestions [13].

Research in Arabic QA remains in its beginning stage. This delay is mainly due to the lack of datasets compared with those available for other languages, such as English [1],[10]. The existing Arabic QA corpora are either small datasets

Paragraph A, Algeria at the FIFA World Cup:
Algeria have appeared in the finals of the FIFA World Cup on four occasions in 1982, 1986, 2010 and 2014. They have once qualified for the knockout rounds.,...,. *In 2014, Algeria qualified for the first time into the round of 16.*
Paragraph B, 2014 FIFA World Cup:
The 2014 FIFA World Cup was the 20th FIFA World Cup, the quadrennial world championship for men's national football teams organized by FIFA. *It took place in **Brazil** from 12 June to 13 July 2014, after the country was awarded the hosting rights in 2007....*
Question: Where was the world cup hosted that Algeria qualified for the first time into the round of 16?
Answer: Brazil

Fig. 1: An example of the multi-hop questions in HOTPOTQA.

or unavailable publicly, and do not cover all categories of questions in terms of type, domain and complexity. Furthermore, these datasets remain limited to simple questions, where answers can be extracted from a single document. In contrast, complex questions, that require reasoning over multiple documents to infer the answer, have not been studied. To the best of our knowledge, there are no datasets for complex question answering in the Arabic language. Moreover, collecting complex questions is not trivial.

To address the above challenges, we propose a workflow to automatically generate questions in order to produce a dataset for Arabic complex QA. The approach covers both comparison and multi-hop questions, where the reasoning over more than one document to find the answer is required. We rely on a structured representation of information about a subject, named infobox, from Arabic Wikipedia articles as data source. Moreover, we use a set of predefined templates to create questions. The produced dataset provides, for each question-answer pair, a set of passages as context and a set of sub-questions along with their corresponding answers as a decomposition of the complex question into simpler questions.

The remainder of the paper is organized as follows: Section 2 reviews existing Arabic question answering datasets. The proposed methodology followed to build the new dataset is detailed in section 3. Several statistics about the generated dataset are presented in section 4. Finally, a conclusion and future work will conclude this paper.

2 Related Work

Arabic is one of the most spoken languages in the world, mainly in the Middle East and North Africa region. Despite the large community of speakers, research in Arabic QA is limited in terms of linguistic resources compared to other languages, with only a few datasets proposed. The investigated existing datasets, recently published in [1, 3] surveys, can be classified according to their construction approaches into three classes:

Machine Translation (MT) based datasets: It is a practical method to generate datasets by translating well established datasets from other languages. For instance, Arabic-SQuAD [8] which is a machine translation of SQuAD 1.1 [12], is composed of 48k paragraph-question-answer tuples. Atef et al. [2] presented AQAD, consisting of more than 17k questions and answers translated from the SQuAD 2.0 [11].Othman et al. [10] use Google translation to translate into Arabic a dataset released by [15]. The dataset was harvested from all categories in the popular Yahoo! Answers community platform. However, this class of datasets suffers from poor translation due to linguistic differences and complexity.

Crowd-sourced datasets: depend on hired crowd workers to create the dataset from scratch or to eliminate issues presented in existing resources. For this category, we cite the ARCD (Arabic Reading Comprehension Dataset)[8] and TyDiQA [4]. ARCD is composed of 1395 factoid questions asked by crowd workers on articles from Wikipedia. TyDiQA is a multilingual QA dataset that covers eleven typologically diverse languages including Arabic, with 204K question-answer pairs. However, Crowd-sourcing approach is time-consuming and requires funds to hire crowd workers.

Web scraping based datasets: rely on automatic process to retrieve QA resources from the Web such as community QA (cQA) sites. In this direction, a medical Arabic corpus for cQA named CQA-MD was proposed by Nakov et al. [9]. The corpus contains over 100k questions-answers pairs collected from Arabic Medical websites. Ismail and Homsi [7] introduced DAWQAS, a dataset for Arabic *Why* QA systems. The dataset contains 3205 *Why* question-answer pairs scraped from public Arabic Websites. The Web scraping based approach is preferable when the targeted question type or domain are available on the Web. However, it requires considerable efforts to annotate the crawled data.

In order to overcome the drawbacks of existing approaches, another method for constructing QA datasets could be based on **Automatic generation**. This method mainly relies on structured sources using logical rules and templates. This approach is more appropriate when the resources for the chosen question type are scarce. This motivate us, through the current work, to develop a method to automatically produce a dataset for Arabic complex QA.

3 Methodology

A carefully designed dataset has an impact on the robustness of the systems as well as the performance of the models built on it. For this reason, and due to the unavailability of a dataset for the Arabic complex QA task, we designed a simple workflow to automatically generate a dataset for the Arabic complex QA task. Figure 2 describes the main steps involved in generating two types of questions: comparison and multi-hop questions.

3.1 Comparison questions generation process

A comparison question is the type of questions that compares two or more similar entities in some aspects of the entity [14]. For instance, the compared entities

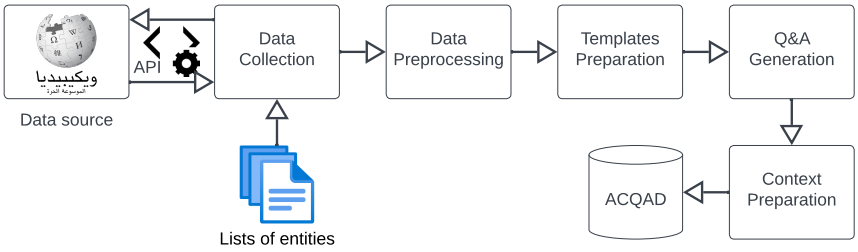


Fig. 2: The workflow of the proposed methodology to create ACQAD

from the question : ؟ من أصغر مساحة، السويد أم الأوروغواي (Which country has a smaller area, Sweden or Uruguay?), are *Sweden* and *Uruguay*, and the aspect of comparison is the area. The idea here is to find pairs of similar entities (A, B) that share common aspects or properties f, and then create the question Q using a template T. To find the answer a, property values $f(A)$ and $f(B)$ for entities A and B respectively are compared and the result determines the answer.

Data Collection. We start by manually curating lists of entities from the same category: animals (92 entities), Arabic cities (22 entities), and world countries (191 entities), totaling 305 entities. Then, we need to retrieve properties of these entities to be used as comparable aspects. To accomplish this task, we used the Wikipedia API[3] and BeautifulSoup[4] library to crawl and parse the infobox from the Wikipedia page of each entity (see Figure 3). Properties are presented in the infobox as (property; value) tuples. This structured representation facilitates the data collection process and eliminates the need for advanced NLP techniques to extract the properties of an entity from plain text.

Pre-processing and properties selection. Since the data gathered from Wikipedia infoboxes was entered by non-professional contributors and was not subject to any formal writing guidelines. Contributors can use words in different languages and introduce information using different styles and formats. Therefore, we performed pre-processing on the raw data to make it usable for question and answer generation. We first cleaned the data by removing special characters, diacritics, links, and non-Arabic words. Then, we normalized the writing of property labels and values in order to have common properties and comparable values.

For example, the property label الفصيلة (family of an animal), may also be found written as فصيلة. This is the same word as before, though without the prefix "ال". That prefix is the definite article in the Arabic language, typically translated as " the" in English. In this situation, we remove the prefix. Another case in numerical values, the expression "نسبة مئوية", or the symbol "%" both are used to describe a percentage. We chose to replace the expression by

[3] https://ar.wikipedia.org/w/api.php
[4] https://pypi.org/project/beautifulsoup4/

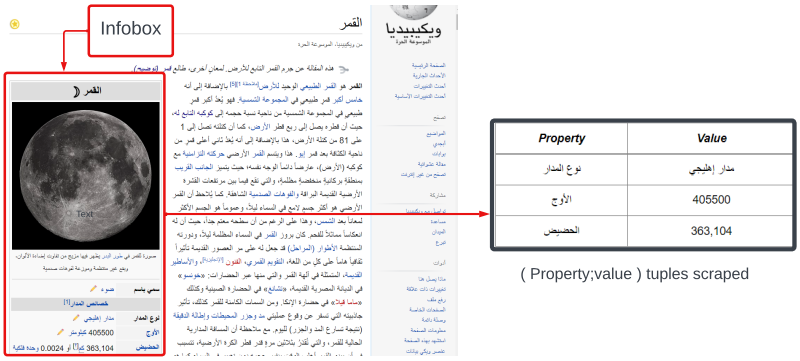


Fig. 3: An example of some data collected from a Wikipedia infobox.

the symbol. In numbers writing, hundreds and thousands parts of a number can be found separated by a dot ".", a comma "," or a space, and the same thing for decimal numbers, which causes ambiguity. For hundreds and thousands, we removed all separators, and for decimal numbers, we preserved the dot ".".
After normalization, we proceed to properties selection. We remove all uncompleted property-value tuples where the property or the value is an empty string (e.g., titles of property groups). We reject all non-comparable properties, such as الموقع الرسمي (official Website), and maintain only comparable properties with values for at least two entities. Finally, we categorized the selected properties into quantitative and qualitative.

Templates preparation. We define manually a set of templates to generate comparison questions and their sub-questions. For each property as an aspect of comparison, we create a list of templates that express the same question but in different ways. This will create more diversity while producing questions. Each template T has two variable tokens: X and Y, to be replaced by two entity labels. For instance, comparing two animals in terms of *gestation period*, ما الحيوان الذي لديه فترة حمل أطول ، X أم Y ؟ (Which animal has a longer gestation period, X or Y ?).

Templates are categorized according to the answer type into: Yes/No questions, or Choice questions where the answer is one of the compared entities. For more variety, we add templates with the opposite predicates of the ones used in the previously prepared templates. For the previous example, instead of أطول (longer), we use أقصر (shorter).

Furthermore, we provide each comparison question with its decomposition in the form of two sub-questions derived from the initial question. Each sub-question seeks the value of an entity's property. Similarly, a set of templates for the sub-questions is created for each property with one variable X to be replaced by each entity label. For the previous example, one of the sub-question templates would be: كم تدوم فترة الحمل عند X ؟ (how long is the gestation period of X?).

Generation method. From the set of entities, we make a list of combinations of two entities A and B from the same category. For each couple of entities, we generate questions about common properties in which they have values. For each property f, we select a random template and replace variable tokens in the pattern string with entity labels.

To produce the answer for the generated question, we compare the property values $f(A)$ and $f(B)$ considering that the question concerns superiority or inferiority, yes/no, or a choice between entities and whether the property is quantitative or qualitative. We obtain at the end four types of answers: yes/no, equality if $f(A) = f(B)$, and one of the compared entities A or B if it concerns a choice question. Subsequently, we generate the two sub-questions in the same manner, with the difference that the answer for each sub-question is the property value.

As our dataset is a text-based extractive dataset, we provide two paragraphs as context. Each paragraph is obtained by extracting the text in the infobox from the entity's Wikipedia page. These paragraphs serve also as a context for the sub-questions. The final structure of the dataset contains generated comparison question-answer pairs along with the two compared entities, the comparison aspect, two paragraphs as context, and two sub-questions with their answers. Eventually, we randomized and sampled the generated dataset.

3.2 Multi-hop questions generation process

Multi-hop questions require a model to reason using information taken from multiple documents to determine the answer [14]. Consider the following question, كم تبلغ مساحة المدينة التي نظمت الألعاب الأولمبية الشتوية لعام 1928 ؟ (How large is the area of the city that organized the 1928 Winter Olympics?). The model must identify, as a first hop, "*the city that organized the 1928 Winter Olympics*", and then "*its area*". From the example, we notice that it can be split into three parts: a hidden entity (a city), an unambiguous feature of this entity (organized the 1928 Winter Olympics), and a property of the hidden entity (the area). The idea to form a 2-hops question is to ask about a property of an entity that has an unambiguous feature. Therefore, the first hop is to find the entity that has the unambiguous feature, and the second hop is to determine the answer to the question concerning that entity's property. We propose below a formal method to generate multi-hop questions that enable to automatically produce a corpus. In the following, we note by $E = \{x\}$ a set of entities, $R = \{r\}$ a set of unambiguous features, and $F = \{f\}$ a set of properties.
With:

$$r(X) = x, x \in E \tag{1}$$

x is the hidden entity which is the answer to the first hop.

$$f(x) = y \tag{2}$$

Where y is the value of the property f for the entity x, considered as the answer to the second hop.

A 2-hops question is formed by replacing the entity x by $r(X)$ from equation 1 in equation 2.

$$f(r(X))) = y \tag{3}$$

Example:

let's take an unambiguous features r_1: أكبر بلد مساحة في أفريقيا (*the biggest country in Africa*).

$$r_1(X) = \text{الجزائر}\,(Algeria) \tag{4}$$

الجزائر (Algeria) is the hidden entity x. Let's take the property f_1 : الرئيس (*president of*).
Using the equation 2,

$$f_1(\text{الجزائر}) = \text{رئيس الجزائر}\,(president\ of\ Algeria) = \text{عبد المجيد تبون} \tag{5}$$

After substitution and adding an appropriate question word, the produced 2-hops question will be:
من هو رئيس أكبر دولة في أفريقيا؟ (*who is the president of the biggest country in Africa ?*), and the answer to this question would be: عبد المجيد تبون

Data collection. To generate multi-hop questions in the form described above, we need to collect entities having unambiguous features. We chose these features to be either unique, such as records, or time-related, such as events, to ensure that only one entity has the feature. Table 1 shows the entity classes collected accompanied with unambiguous features examples. Beside that, information such as the competition date, its round, the record set, the tournaments season, the start and the end dates are also collected to be used in the subsequent steps of the workflow. Next, we collect properties of the entities from Wikipedia infoboxes and we perform the same pre-processing steps as described in section 3.1.

Templates preparation. We propose a set of templates in order to produce a rich diversity of questions. Multi-hop questions templates are created as a concatenation of the triplet : question word, property label and unambiguous feature phrase. The question words are defined depending on the collected property types (number, date, etc.), and the property gender (masculine or feminine). Table 2 illustrates the appropriate question words used for some collected properties.

Regarding the unambiguous features phrase, we use the information available to formulate this part of the question template. For world and nature records category, the unambiguous features are in form of superlative expressions. We use these expressions as unambiguous feature phrases. Concerning Olympic records

Entity class	# Entities	Unambiguous features Type	Example
animals, countries, places, and buildings	50	world and nature records	أسرع الثدييات (The fastest mammal)
players	25	Olympic records	الرقم القياسي في سباق 100 م (100 meter world record)
players' countries	21		بلد اللاعب يوسين بولت (Usain Bolt's country)
host cities and countries	65	Olympic events	المدينة التي استضافت الألعاب الأولمبية الصيفية 2016 (The city that hosted the 2016 Summer Olympics)
tournaments	51		البطولة حيث سجل بولت رقما قياسيا في 100 م (The tournament where Bolt set 100m world record)

Table 1: Collected entity classes with unambiguous features examples

and Olympic events categories, we employ the available information, mentioned in section3.2, to formulate diverse, unambiguous features phrases. We use variable tokens that correspond to these information in the templates. For instance, اللاعب الذي حقق الرقم القياسي في منافسة C في دورة G "The player who set the record in competition C in tournament G". C denotes the competition, and G denotes the tournament where the record was set. The full template example will be:

كم يبلغ طول اللاعب الذي حقق الرقم القياسي في منافسة C في دورة G ؟

The decomposition of the multi-hop question consists of creating a sequence of sub-questions where each sub-answer solves a part or a hop of the question. In our case, we adapted the triplet used to construct the multi-hop questions templates for creating the sub-questions. The first sub-question template uses an appropriate question word (e.g., *who, what*) with the unambiguous feature phrase. The answer to this first sub-question is the hidden entity. The second sub-question template uses the question word and the property label from the triplet, in addition to the entity that is the answer to the first sub-question. The following sub-questions illustrate the case of the previous example:

- sub-question 1: من هو اللاعب الذي حقق الرقم القياسي في منافسة C في دورة G ؟
- answer to sub-question 1: entity P
- sub-question 2: كم يبلغ طول اللاعب P ؟

The final step is to generate the questions and their decomposition. For each entity's property, we select the appropriate question word. Then, we randomly select one of the unambiguous feature phrases, and replace the variable tokens with their corresponding information.

Question word	property
كم تبلغ (How much)	الكثافة السكانية (population) المساحة (area) نسبة المياه (water ratio)
ما هو (What is)	رمز الهاتف (phone code) السن القانونية (legal age) نظام الحكم (regime)
ما هي (what is, for female)	الكنية (surname) العاصمة (capital) العملة (currency)

Table 2: Examples of appropriate question words for collected properties

3.3 Collecting questions contexts

To answer multi-hop questions, models need more than one paragraph as a context. For each question, we retrieve two passages called gold paragraphs. The first paragraph is the summary of the entity's Wikipedia article where the unambiguous feature appears. The second paragraph is the text in the Wikipedia infobox where we find the properties and their values. In all cases, we ensure that the answer to the first sub-question appears in the first gold paragraph, and the answer to the second sub-question appears in the second paragraph.

For both comparison and multi-hop questions, we add distracting paragraphs to the gold paragraphs, following Yang et al. [14] and Ho et al. [6] setting. These paragraphs are used to make the dataset more challenging and test the model's ability to find the answer in the presence of noise. We use the retriever module from the SOQAL system [8]. We first retrieve the top 10 articles from Wikipedia, which are very similar to the question using the 1-gram TF-IDF formula. Then we use it again to select, from the retrieved articles, the top-8 paragraphs as distractor paragraphs. Finally, we mix the gold and the distractor paragraphs to obtain the context.

4 Dataset Analysis

In this section, we analyse the dataset by providing statistics regarding the number of questions generated, questions and answers length, and the types of answers. All results are presented for comparison and multi-hop questions separately.

4.1 Quantitative analysis of generated questions

The statistics of the generated ACQAD dataset are presented in Table 3, where Q denotes the question and A denotes the answer. We provide the number of instances produced per entity type for comparison questions and per unambiguous feature type for multi-hop questions. The number of questions generated depends on the number of entities within each type and the properties retrieved for these entities.

The dataset consists of 118841 questions in total. Comparison questions have the most instances, while multi-hop questions have fewer since it is difficult to find entities with unambiguous features. The list of entities' types can be extended in the case comparison questions to cover more topics (for instance : people, companies, events, etc.).

The average answer length of comparison questions is smaller than that of multi-hop questions. This is due to the fact that there are numerous yes/no answers for comparison questions.

Question type					
Comparison			Multi-hop		
Entity type	**#Entities**	**#Examples**	**Unambiguous features type**	**#Entities**	**#Examples**
Animals	92	8625	world and nature records	50	519
Arabic cities	22	462	Olympic records	46	1425
World countries	191	106769	Olympic events	116	1049
Total	305	115856	**Total**	212	2985
#Avg. Q	10.47		**#Avg. Q**	19.05	
#Avg. A	1.14		**#Avg. A**	3.77	

Table 3: Statistics per type related to the generated questions

Figure 4 shows the distribution of question length for comparison and multi-hop questions. The varied lengths of questions represent various complexity levels. We obtain almost the same range of questions length compared to the benchmark hotpotQA [14] where most questions contain between 10 and 40 tokens.

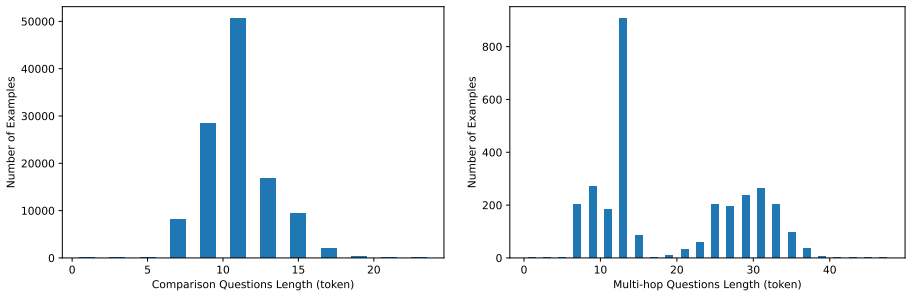


Fig. 4: Distribution of Questions lengths

4.2 Answer types

Answer types for comparison questions are restricted to yes/no, one of the compared entities for choice questions, or equality word تساوي if the property values being compared are equal. Statistics of answer types are given in Table 4. Yes/No type is abundant since most templates used while generating comparison questions are of type yes/no. As it is uncommon to find two equal values, the equality type is nearly non-existent.

Table 5 presents the answer types for the multi-hop questions. As is shown, the generated dataset covers a variety of questions centered around persons, organizations, locations, dates, and numbers, as well as other answer types. We notice clearly the domination of the number type because the majority of properties values are quantitative.

Answer Type	**%**
Yes/No	71.67
Entity	27.49
Equality	0.84

Table 4: Type of Answers for Comparison questions

Answer Type	**%**	**Examples(s)**
Number	53.9	62 سنة
Person	7.8	جو بايدن
Date	10.1	26 ديسمبر 1991
Location	5.7	منطقة التبت
Organization	1.3	جامعة ستانفورد
Other	21.2	دولار أمريكي

Table 5: Type of Answers for Multi-hop questions

5 Conclusion and future work

In this paper, we presented the creation process of ACQAD, an automatically generated dataset for Arabic complex question answering task. To the best of our knowledge, no dataset for Arabic language is available for this task. Our corpus consists of more than 118k questions. We relied on Wikipedia as data source and a set of predefined templates to generate high quality questions. The dataset provides with each question a set of sub-questions as decomposition. The proposed method can be adapted to any language that is lacking datasets for complex QA task. Future work will aim to establish baseline models with which researchers may compare their approaches and results. Furthermore, the focus will be on extending ACQAD by including more multi-hop examples. Since we now have a corpus, we will develop methods that leverage the data available to answer complex questions.

References

1. Alwaneen, T.H., Azmi, A.M., Aboalsamh, H.A., Cambria, E., Hussain, A.: Arabic question answering system: a survey. Artificial Intelligence Review pp. 1–47 (2021)

2. Atef, A., Mattar, B., Sherif, S., Elrefai, E., Torki, M.: Aqad: 17,000+ arabic questions for machine comprehension of text. In: 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA). pp. 1–6. IEEE (2020)
3. Biltawi, M.M., Tedmori, S., Awajan, A.: Arabic question answering systems: Gap analysis. IEEE Access **9**, 63876–63904 (2021)
4. Clark, J.H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., Palomaki, J.: Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. Transactions of the Association for Computational Linguistics **8**, 454–470 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Ho, X., Nguyen, A.K.D., Sugawara, S., Aizawa, A.: Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. arXiv preprint arXiv:2011.01060 (2020)
7. Ismail, W.S., Homsi, M.N.: Dawqas: A dataset for arabic why question answering system. Procedia computer science **142**, 123–131 (2018)
8. Mozannar, H., Hajal, K.E., Maamary, E., Hajj, H.: Neural arabic question answering. arXiv preprint arXiv:1906.05394 (2019)
9. Nakov, P., Màrquez, L., Moschitti, A., Mubarak, H.: Arabic community question answering. Natural Language Engineering **25**(1), 5–41 (2019)
10. Othman, N., Faiz, R., Smaïli, K.: Learning English and Arabic Question Similarity with Siamese Neural Networks in Community Question Answering services. Data and Knowledge Engineering (101962) (Dec 2021). https://doi.org/10.1016/j.datak.2021.101962, https://hal.archives-ouvertes.fr/hal-03500114
11. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad. arXiv preprint arXiv:1806.03822 (2018)
12. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
13. Talmor, A., Berant, J.: The web as a knowledge-base for answering complex questions. arXiv preprint arXiv:1803.06643 (2018)
14. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600 (2018)
15. Zhang, W.N., Ming, Z.Y., Zhang, Y., Liu, T., Chua, T.S.: Capturing the semantics of key phrases using multiple languages for question retrieval. IEEE Transactions on Knowledge and Data Engineering **28**(4), 888–900 (2015)

A Machine Learning Approach for Phishing URLs Detection using Lexical and Host-based Features

Samiya Hamadouche

LIMOSE Laboratory, Faculty of science, University M'Hamed Bougara of Boumerdes, Algeria,
hamadouche.samiya@univ-boumerdes.dz

Abstract. Malicious websites (phishing, spam, drive-by-download, etc.) represent a serious threat to security. Hence, their detection became imperative. Moreover, recent developments in the machine learning field have led to a renewed interest in its application to a wide range of cyber security issues. Machine learning based detection of phishing URLs relies on specific characteristics: lexical, host, content, and context features. The present paper focuses on the combination of two categories of features: lexical-based and host-based. The objective is to propose, implement, and evaluate a phishing URLs detection solution based on this set of features. After data collection, data pre-processing and features extraction phases, three algorithms (Support Vector Machine, Decision Tree and Random Forest) were applied on our dataset. The performance evaluation shows that the Random Forest is the best model for our problem with an accuracy of 96.90% and a false negative rate of 1.93%.

Keywords: Phishing URLs detection, Machine learning algorithms, Classification, Lexical-based features, Host-based features

1 Introduction

Phishing attacks can be defined as a cyber threat in which attackers take advantage of users through imitating legal original websites. They aim at stealing sensitive data like: passwords and bank statements. Phishing is carried out through various means: internet, short message service and voice. Email, instant messaging, smishing (short message phishing), vishing (voice phishing) and websites are some targeted vectors of these attacks [4].

As specified in the Anti-phishing Working Group (APWG) trend reports [1], the number of phishing websites identified between 2018 and 2019 is rather stable i.e. between 35000 and 100000. While in December 2021, 316747 phishing attacks are notified. This latter is the most evaluated monthly number in their history and six times the number of phishing attacks compared to early 2020 [1]. The principal reason beyond this notable rise is that malicious people exploited the widespread of COVID-19 pandemic. In fact, so as to restrain the periods of lockdown that foiled their ordinary daily tasks (work, shopping, studies, etc.),

people all around the world had to turn to the Internet. Therefore, the number of fake websites have been multiplied by cybercriminals so that they could deceive other victims. Hence, the detection of phishing websites became necessary. Blacklists, which are mainly a database of Uniform Resource Locators (URLs) that have been recognized to be malevolent, are traditionally used to detect phishing. But these lists are not complete and cannot identify new generated URLs [8]. Machine learning put forward a practical solution by new techniques to combat cybercrime through artificial intelligence using adequate algorithms. Machine Learning approaches use a set of URLs as training data, and based on the statistical properties, learn a prediction function to classify a URL as malicious or benign. This gives them the ability to generalize to new URLs unlike blacklisting methods [8].

Our work consists in using machine learning techniques to solve the problem of detecting phishing URLs. We have studied the impact of the combination of several types of features (lexical-based and host-based) on the performance of the proposed solution. We were able to conclude that, indeed, this combination allows to improve efficiently the obtained results.

The rest of the paper is structured as follows. Section 2 recalls the general context in order to position our work. The proposed approach is discussed in section 3. Section 4 provides details about the implementation and analysis of experimental results. Finally, a conclusion to conclude the paper in section 5.

2 Background

2.1 Phishing attack and URLs

Phishing is a common cyberattack done via sending an email or a message in order to defraud receivers visiting a bogus web page. After that, sensitive information of users, like: usernames, passwords and credit card numbers, are being gathered for financial profit [9]. There are different kinds of phishing attacks used to deceive the users.

Figure 1 illustrates the general life cycle of a phishing attack via e-mail. After designing a fake site (i.e. phishing site) very similar to the users' trusted site, the attacker tries to send the user a malicious link via e-mail (the choice of the victim can be made intentionally or randomly). Subsequently, the victim will click on the malicious link that has been received and redirected to the phishing site with a similar appearance to the real one that he knows. The user will be persuaded to disclose his information as he does on the legitimate site while he is on the fake site. Finally, this information will be sent to the attacker who will exploit it to manipulate the user's accounts.

URL is the abbreviation of Uniform Resource Locator, which is the global address of documents and other resources on the World Wide Web. A URL has two main components : protocol identifer (indicates what protocol to use) and resource name (specifes the IP address or the domain name where the resource is located) [8]. URL-based phishing attacks are mainly performed by embedding sensitive words or characters in a link that [2]:

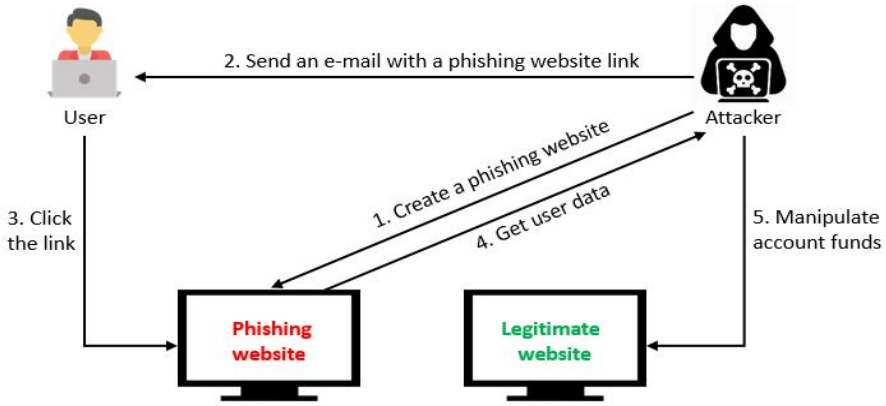


Fig. 1: Phishing attack life cycle (adapted from [9])

- Mimics similar but misspelling words.
- Contains special characters for redirecting.
- Uses shortened URLs.
- Uses sensitive keywords which seem reliable.
- Adds a malicious file in the link and so on.

2.2 Machine learning based phishing URLs detection

Because of the use of different techniques (URL obfuscation to shorten it, link redirection, manipulating links, etc.) so that phishing websites seem to be reliable, their detection have become demanding. Thus, moving from traditional detection methods to more intelligent ones become crucial. In machine learning based phishing detection, the characteristics of the URLs are extracted and fed into the used algorithms. In general, the classifier creates a model based on the information extracted from the training samples. The suspected URL is evaluated according to this model.

Several types of features can be used for malicious URLs detection (including phishing URLs). Mainly, they are categorized into [8]:

1. **Lexical-based features**: Obtained from the properties of the URL name (the URL string). It should be possible to identify the malicious nature of the URL based on its. The most commonly used lexical features include statistical properties of the URL string (length of URL, length of each component of the URL, number of special characters, etc.).
2. **Host-based features**: Obtained from the host-name properties of the URL. They allow us to know the location, identity, the management style and prop-

erties of malicious hosts. They may include: IP address properties, WHOIS information, domain name properties, connection speed, etc.

3. **Content-based features**: Obtained upon downloading the entire webpage. A lot of information needs to be extracted, and at the same time, safety concerns may arise. They include: HTML Document Level Features, JavaScript features, ActiveX Objects and feature relationships.
4. **Context features**: Represent the features of the background information where the URL has been shared (for examlpe: social media platforms like twitter, facebook, etc.).

2.3 Related works

The table 1 shows a synthesis of some related works in which we illustrate our present work in relation to previous ones dealing with the same context of phishing detection (lexical and host based features). However, we do not have pretension to compare results considering the differences of the used: databases, extracted features, applied algorithms, and the working environment.

Table 1: Related works synthesis

Authors	Algorithms	Dataset	Feautures	Accuracy	False Negative Rate
Catak et al [3]	Gradient Boosting, Random Forest	Public datasets (3231961 URLs).	Lexical + Host	98,60%	Not available
James et al[5]	Naïve bayes, SVM, KNN.	phishtank.com, alexa.com, dmoz.com, (37000 URLs).	Lexical + Host + Popularity	85.63%	14,37%
Korkmaz et al [6]	KNN, SVM, DT, Random Forest, ANN, Naïve Bayes, ADABOOST, XGBoost.	Phishtank.com, Common Crawl, Alexa.com, (83857 URLs).	Lexical + Host	94,59%	5,31%
Mahjan et al [7]	Decision Tree, Random forest, SVM.	alexa.com, phishtank.com, (36711 URLs).	Lexical + Host	97,14%	3,14%
Our work	Decision Tree, Random forest, SVM	4 datasets (see section 3) (11076 URLs).	Lexical + Host	96.90%	1,93%

3 Proposed Approach

3.1 General Process

The proposed approach consists of the following five steps (Figure 2), each of which will be detailed in the next sections:

1. Data collection.
2. Data pre-processing.
3. Feature extraction.
4. Model training.
5. Evaluation of the solution.

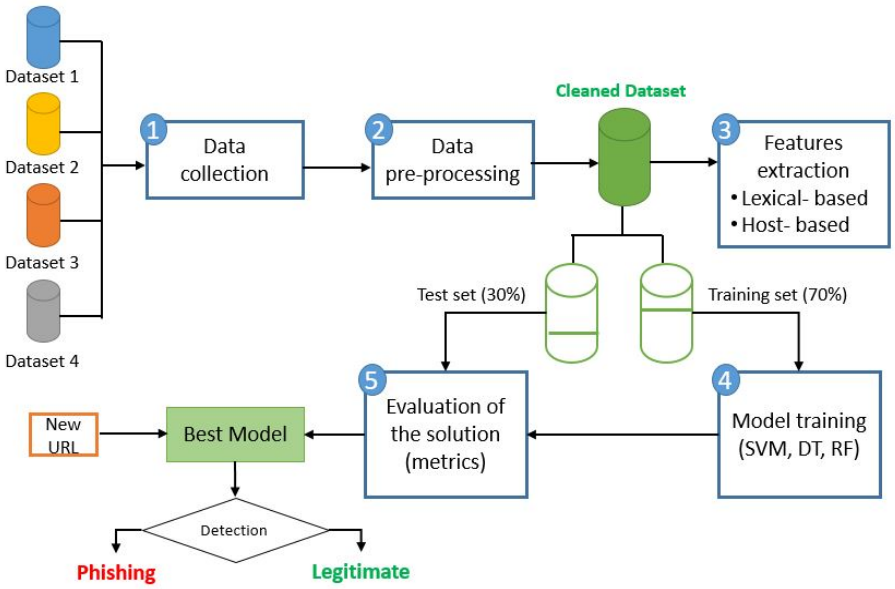


Fig. 2: General process of the proposed approach

3.2 Data Collection

The first step consists of building the dataset on which the rest of the process will be based. In order to build an acceptable and balanced list of URLs, two types of URLs are needed: legitimate and phishing. For this, we combined data from four datasets (see section 4).

3.3 Data Pre-processing

The pre-processing phase is imperative so that we can ensure the good quality of our data to avoid false results. It consists in our case of the following operations:

1. **Standardization of the Datasets format**: First, we had to organize the columns of the Datasets which contained several columns. But, we only need two of them. The first column, entitled "URL", contains URLs and the second one ,entitled "Label", can have two values: "0" when the URL is malicious and "1" in case the URL is a legitimate one.
2. **Removing inaccessible URLs**: Due to the short lifespan of phishing URLs, it was necessary to delete all URLs that are no longer accessible (not available in search engines).
3. **Removing redundant URLs**: As we used several datasets, there is a high probability of finding redundant URLs. Hence, datasets were examinated to delete the repeated ones.
4. **Combining URLs in the same Dataset**: Finally, we combined all the URLs obtained after cleaning in one Dataset. To prevent data bias, this dataset is balanced between phishing URLs and legitimate ones (the same number for each category).

3.4 Features Extraction

This step is based on the cleaned list of URLs (obtained in the precedent step). Initially, based on the related works existing in the literature and addressing the same research problem, we have opted for the selection of 30 features. We developed Python scripts for the extraction of different features. After that, based both on the correlation matrix (showing the dependance between the different features) and the *MostImportant()* function, we have selected the 18 most relevant features.

The set of features adopted in our approach can fall into two categories: host-based features (4 features: F1-F4) and lexical-based features (14 features: F5-F18). For extraction, we constructed the following feature vector:
$F = (F1, F2, ..., F18)$. These features are resumed in table 2

Table 2: List of the extracted features

	Feature	Corresonding function
Host-based	F1: Age of the domain	domainAge()
	F2: Website traffic	webTraffic()
	F3: Domain registration period	domainEnd()
	F4: DNS record	hasDnsRecord()
Lexical-based	F5: IP address usage	hasIpAdress()
	F6: URL length	isLong()
	F7-F12: Number of special characters (@, -, _, ?, &, .)	numberOf...()
	F13: Number of digits in the domain	hasNumberInDomain()
	F14: Depth of the URL	depth()
	F15: Number of points in the domain	hasPointInDomain()
	F16: Shortening the URL	isShortned()
	F17: Length of the domain	domainLength()
	F18: Length of the URL	urlLength()

3.5 Model Training

Splitting the dataset into training and test sets
After building a clean dataset ready to be used for machine learning algorithms, we split it into two datasets: one for training (train set) and one for testing (test set). When separating the data, usually, the majority of the data is used for training and the rest for testing in order to get a better performance of the learning models. Thus, we have separated our dataset of 11076 URLs according to the following ratio: training-test of 70-30 (i.e. 70% of the data for training and 30% for testing).

Choosing the learning algorithms Our problem consists of classifying the websites' URLs into phishing and legitimate. In other words, detecting whether they are malicious or not. Moreover the data we are using is labeled, hence, we used supervised machine learning techniques. There exists several algorithms for classification; in our case, we considered the following three algorithms (the most used ones in literature):

1. Support vector machine (SVM).
2. Decision tree (DT).
3. Random forest (RF).

3.6 Evaluation of the solution

The evaluation stage is a crucial step in choosing the best model. The selected models (algorithms) are trained on the training data to predict the results of the test data. Subsequently, the performance of each algorithm is evaluated

using several metrics to compare their performance. The evaluation is based on the confusion matrix that summarizes the result of each model and allows to calculate the recall, precision, accuracy and F1-score. This will allow choosing the best among the algorithms applied for phishing URLs detection.

4 Implementation and experimental results

After the design of the solution (section 3), the implementation phase leads to the realization of our models. Then, come the phases of test and evaluation that will allow us to compare our models to choose the most performing one among them. Python was used for the implementation of the proposed approach, using the following libraries: Scikit-learn, Scipy, Pandas, Matplotlib.

4.1 Dataset and its pre-processing

In our work we used three datasets with CSV files and one with a Json file. These datasets come from different sources which are summarized in the table 3. After the pre-processing phase, the obtained dataset (cleaned) gathers 11076 different URLs. To prevent data bias, we have used legitimate and phishing URLs in an equal proportion (i.e. 5538 for each).

Table 3: The used datasets

Dataset	Source	URL number	Type
1	https://www.phishtank.com/ developer_info.php (Phishtank)	14825 phishing	.CSV
2	https://data.mendeley.com/datasets/c2gw7fy2j4 (Mendeley)	11430 (5715 Phishing, 5715 legitimate)	.CSV
3	https://www.unb.ca/cic/datasets/url-2016.html (Canadian Institute for Cybersecurity)	35377 legitimate	.CSV
4	https://dataforseo.com/top-1000-websites	1000 legitimate	.Json

4.2 Results and performance evaluation

Each binary classifier has an output matrix known as the confusion matrix which summarizes the correctly and incorrectly predicted instances of each class. It has four outcomes namely true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

- **TP**: number of correctly predicted samples as phishing websites
- **TN**: number of correctly predicted samples as legitimate websites

– **FP**: number of incorrectly predicted samples as phishing websites
– **FN**: number of incorrectly predicted samples as legitimate websites

The metrics used to evaluate and compare the models are calculated from this matrix. They are summarized in table 4.

Table 4: Evaluation metrics

Metric	**Formula**	**Description**
Precision	TP/(TP+FP)	Total number of URLs detected as phishing out of total phishing URLs
Recall	TP/(TP+FN)	Total number of legitimate URLs classified as legitimate and phishing URLs classified as phishing.
F1-score	2*(precision*recall)/ (precision + recall)	The harmonic mean of precision and recall.
Accuracy	(TP+TN)/ (TP+TN+FN+FP)	Total number of overall correctly classified instances

We have used 3323 instances as testing data. Table 5 and figure 3 illustrate all the results obtained with the different classifiers. We can notice that:

– For the three models, the results obtained using lexical+host based features (F1-F18) are better than the results obtained using lexical-based features only (F5-F18).
– The best performing model is the Random Forest classifier (RF) with the highest accuracy of 96.90%.
– The Decision Tree classifier (DT) has scores that are significantly close to the ones of the best performing model.
– The Support Vector Machine classifier (SVM) has obtained the least performing scores.

Moreover, for performance evaluation, we have to analyze the true positive rate (TPR), true negative rate (TNR), false-positive rate (FPR),and the false-negative rate (FNR). These metrics and their calculation process are presented in Table 6. The obtained rates with different classifiers are illustrated in table 7 and figure 4.

In our work, we seek to reduce the false negative rate FNR. Indeed, the fact of wrongly classifying a phishing URL as legitimate constitutes the most dangerous case in terms of security. According to the results of table 7, the RF classifier is the one that allows obtaining the lowest FNR (1.93%). This confirms that it is the best performing model for this classification problem.

Table 5: Experimental results from different classifiers

	Model	Confusion matrix				Metrics (%)			
		TP	FN	FP	TN	Precision	Recall	Accuracy	F1-score
Lexical	SVM	1518	132	327	1346	82,28	92,00	86,19	86,87
	DT	1559	76	139	1549	91,81	95,35	93,53	93,55
	RF	1544	47	132	1600	92,12	97,05	94,61	94,52
Lexical+host	SVM	1387	263	98	1575	93,40	84,06	89,14	88,48
	DT	1538	58	80	1647	95,06	96,37	95,85	95,71
	RF	1681	33	70	1539	96,00	98,07	96,90	97,03

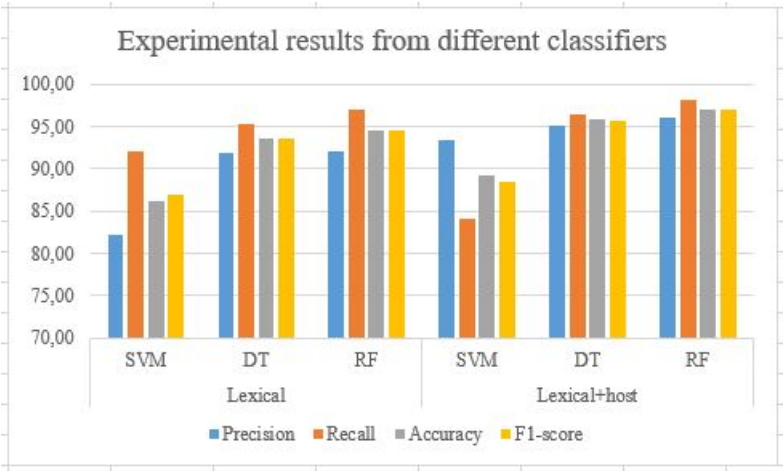


Fig. 3: Experimental results from different classifiers (lexical and lexical+host features)

Table 6: Description of the used rates

Rate	**Formula**	**Description**
TPR	TP/(TP+FN)	The rate of phishing URLs that are classified as phishing
TNR	TN/(TN+FP)	The rate of legitimate URLs that are classified as legitimate
FPR	FP/(FP+TN)	The rate of legitimate URLs that are classified as phishing
FNR	FN/(FN+TP)	The rate of phishing URLs that are classified as legitimate

Table 7: Rates of True/False positive and negative

	Model	Confusion matrix				Rates (%)			
		TP	FN	FP	TN	TPR	FNR	FPR	TNR
Lexical	SVM	1518	132	327	1346	92,00	8,00	19,55	80,45
	DT	1559	76	139	1549	95,35	4,65	8,23	91,77
	RF	1544	47	132	1600	97,05	2,95	7,62	92,38
Lexical+host	SVM	1387	263	98	1575	84,06	15,94	5,86	94,14
	DT	1538	58	80	1647	96,37	3,63	4,63	95,37
	RF	1681	33	70	1539	98,07	1,93	4,35	95,65

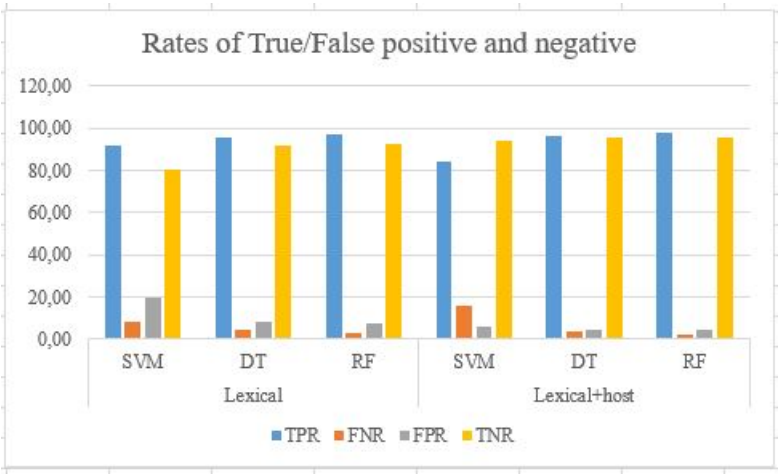


Fig. 4: Rates of True/False positive and negative (lexical and lexical+host features)

5 Conclusion

Phishing is a web problem that many research works dealt with and tried to circumvent because of its socio-economic impact. In this paper, we have shown that machine learning techniques are really able to overcome the problem of detecting phishing websites. The experimental results show the efficiency of the Random Forest (RF) algorithm compared to other machine learning algorithms (SVM and DT). Moreover, the combination of the lexical-based and host-based features with the Random Forest algorithm, give a better accuracy rate of 96.90%. Future work involves exploring other classification algorithms as well as deep learning on larger datasets with other feature sets (content and context).

References

1. APWG Phishing Trends Report: Year on Year Review, 2021. https://visua.com/apwg-phishing-trends-report-2021-review.
2. Eint Sandi Aung, Chaw Thet Zan, and Hayato Yamana. A survey of url-based phishing detection. In *DEIM Forum*, pages G2–3, 2019.
3. Ferhat Ozgur Catak, Kevser Sahinbas, and Volkan Dörtkardeş. Malicious url detection using machine learning. In *Artificial intelligence paradigms for smart cyber-physical systems*, pages 160–180. IGI Global, 2021.
4. Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, 106:1–20, 2018.
5. Joby James, L Sandhya, and Ciza Thomas. Detection of phishing urls using machine learning techniques. In *2013 International conference on control communication and computing (ICCC)*, pages 304–309. IEEE, 2013.
6. Mehmet Korkmaz, Ozgur Koray Sahingoz, and Banu Diri. Detection of phishing websites by using machine learning-based url analysis. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2020.
7. Rishikesh Mahajan and Irfan Siddavatam. Phishing website detection using machine learning algorithms. volume 181, pages 45–47, 2018.
8. Doyen Sahoo, Chenghao Liu, and Steven CH Hoi. Malicious url detection using machine learning: A survey. *arXiv preprint arXiv:1701.07179*, 2017.
9. Lizhen Tang and Qusay H Mahmoud. A survey of machine learning-based solutions for phishing website detection. *Machine Learning and Knowledge Extraction*, 3(3):672–694, 2021.