International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Data Clustering Optimization using Support Vector Machines

Ichrak Lafram, Siham El Idrissi, Aicha Marrhich, Naoual Berbiche, Jamila El Alami

Abstract: ***The analysis of massive data is becoming more and more critical. One of the systems that process real-time data are computer networks. The data flowing through these networks is enormous and requires technicality to manage it better, and the most central characteristics of these systems is to ensure security. To ensure this task, administrators use intrusion detection systems (IDSs). The major problems with these systems are the false positive and the speed of the system to process data and analyze it. For this, we present an optimization of the existing methods based on artificial neural networks, through combining two machine learning procedures; unsupervised clustering followed by a supervised classification framework as a Fast, highly scalable and precise packets classification system.***

Index terms: Intrusion Detection, Machine Learning, Traffic Classification, Artificial Neural Networks, support vector machines, x-means

I. INTRODUCTION

The explosive growth of the Internet and the continued expansion of data systems is a major achievement. But the internet ecosystem involves many menaces of attacks, different programs have been placed in action in order to block attacks on the Internet. In particular, Intrusion detection systems (IDS) help the network withstand external attacks.

IDSs ' goal is to provide a defense wall to confront intruders and limit the danger of their attacks trials.

IDSs are generally segmented in two types: Anomaly identification and misuse detection, depending its detection styles. The detection of anomalies attempts to evaluate whether intrusions can be labeled as deviations from known normal use patterns. While on the other hand, detection of abuse uses well-known signatures to detect intrusions.

Revised Manuscript Received on 30 July 2019.

***** Correspondence Author

Ichrak lafram, LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco

Siham El idrissi, LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

Aicha Marrhich, LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

Naoual Berbiche, LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

Jamila EL Alami, LASTIMI Laboratory, Superior School of Technologies of Sale, Mohammadia School of engineering, Mohamed V University city of Rabat, Morocco.

Static detection approaches are experiencing a myriad of issues and usually lead to unsatisfactory detection systems [1]:

- New and unknown attacks cannot be identified
- Unable to detect variations of known attacks
- Require constant regular updates, therefore a loss of detection in real time
- High false-alarm rates

Machine learning techniques have been proposed to start breaking further into the field of intrusion detection as a redress to these issues. These techniques are imprecision and uncertainty - tolerant which make them very useful where optimization and precision are most needed [2]. The use of these techniques aims to train algorithms to learn from existing data so that the class of each entry can be predicted. To carry out this process of training and testing then predicting, researchers use publicly available data sets to train and evaluate detection models. A Dataset like the KDDCup99 is frequently used [3] because it is based on DARPA'98 data captured in DARPA's IDS evaluation program. But this is an outdated dataset which doesn't cover new and important characteristic of real traffic of current networks [4]. This paper presents a detection approach based on two-level classification for fast detection and high accuracy. While traditional approach relies on entire packets inspection, we use an optimized clustering algorithm to cluster the incoming traffic. Once the cluster is set using x-means algorithm and optimized by a support vector machines algorithm, a specific artificial neural network classifies its entries in order to detect the abnormal traffic. The model presented here has been trained and evaluated using the CICIDS2012 and CICIDS2017 the recently published dataset from the Canadian Institute for Cyber security [5]. This paper is arranged in the next way: After introducing our work, we will give an overview of the techniques used. Then some of the works linked to the matching background will be conferred in a third section. In the fourth unit, we will present the methodology we actually followed in our research study and the reasons for opting for a new dataset. The proposed detection model is proposed in the fifth section. Finally, the last section will be dedicated to our experimental results validating this approach.

II. TECHNICAL OVERVIEW

Unsupervised machine learning is a branch of machine learning where we deal with unlabeled data.

Supervised learning is where a model is trained from labeled data and then makes predictions or provides any other output by using this model. The model is trained without labels (labeled data) in unsupervised learning and a trained model selects new or abnormal observations from a dataset based on one or more resemblance metrics to normal data.

A. *Unsupervised Learning – Clustering*

In order to figure out groups or forms of similarities in a given dataset without having a prior knowledge of the data, we simply use a set of clustering techniques.
Without this prior knowledge of data characteristics, clustering algorithms aims toward uncovering smaller groups present within a dataset and to identify structures that could be in it [6].
There exist a large set of clustering algorithms, and we will focus on the simplest and frequently used one: x-means.
We choose this algorithm because it is a simple one and very fast executing real time clustering. Another major advantage is that it runs on large bulky datasets. Also, we don't have to specify the number of clusters to start with [7].

1. *K-means clustering algorithm*

X-means algorithm is a variant of the well-used *k-means* algorithm which process different sets of clusters formed by *k-means*, the cluster having the lower sum of squared error is the best one to consider. Then every entry is allocated to just one cluster.
Sum of squared error is defined by:

$$SSE = \sum_{i=1}^{k} \sum_{C_i} dist(c_i, x)^2 \ \ where \ \forall x \in C_i \qquad (1)$$

C_i : i^{th} cluster
c_i : centroid of C_i

$$SST = \sum dist(C_m, x)^2 \ where \ \forall x \ in \ Dataset \ D \qquad (2)$$

2. *X-means clustering algorithm*

Despite its advantages and its performance dealing with big datasets, k-means algorithm suffers from a key weakness: The user must provide the initial number of clusters in order to initialize the k-means algorithm. We can perform this task by giving multiple values of k and observe changes happening to the objective function.
Unfortunately, when dealing with data of big size such as the case of network traffic data, this method is inefficient. To overcome this, we will use a variant of k-means called x-means.
With *k-means* algorithm, we have to specify an array of the potential values of k. Then we set "two" as the min value and the value of the square root of half the data size as the max range [8].

$$k = \sqrt{\frac{n}{2}} \qquad (3)$$

This equation (3) is called the rule of scan for approximating potential k number of clusters.
Following, the algorithm improves its parameters starting with the min value of k and continues until finding its maximum.

$$BIC(M_j) = l_j(A) - \frac{p_j}{2} \times logS \qquad (4)$$

At the same time, it keeps improving its structure until the best centroids are found based on the calculation of the BIC values (Bayesian information criterion) (4).
Every cluster will then be divided into two slices in order to move the newly formed centroids in the opposite direction with distances proportionate to the size of the area.
The k-means are applied in each original clustered area through k equal to two and the BIC calculated for k equal 1 and 2.
Using Eq. (4): If: $BIC(k = 1) > BIC(k = 2)$,
The cluster then regains his original centroid.
Otherwise, the divided form is retained. Where:
l_j(A): log-likelihood of the dataset A for the j^{th} model in the point of max probability,
p_j : Number of parameters within M_j (a space of alternative models).
S: Size of the dataset *A.*

3. *The SVM algorithm:*

SVM algorithm aims toward finding a hyperplane in a N - dimensional space (N: number of data features), which clearly classifies the data points [9].
There is a big set of probable hyperplanes that can be chosen in order to separate two different classes of data points. The SVM algorithm attempts to discover a hyperplane where margin between classes is optimal. Maximizing this margin provides some strengthening so that new data points could be more confidently classified.
a) Linear SVM
Support Vector Machines consist on discovering the optimal separating hyperplane which maximizes the distance amongst the hyperplane and the two classes [10]. This distance is called Margin.
For a given learning set(x_i, y_i), where $x_i \in R^m$ is a data space and $y_i = \pm 1$ is the classes space.
We define the hyperplane by: $\omega.x + b = 0$
(ω, b)Are the hyperplane parameters:
ω A normalized Vector
b The Bias
The classifier is defined as follow:

$$f: x_i \in R^m \rightarrow Sign(\omega.x + b) \in \{\pm 1\}$$
$$\omega.x_i + b \geq 0 \ if \ y_i = +1$$
$$\omega.x_i + b \leq 0 \ if \ y_i = -1$$

The OSH should maximize the margin defined as:

$$\frac{2}{\| \omega \|}$$

To resolve the margin maximization problematic under predefined conditions is equivalent to resolve a quadratic optimization problem defined as:

$$min_{(\omega,b)} \frac{\| \omega \|^2}{2}$$

While: $y_i(\omega.x_i) \geq 1, \forall x \in [1, M]$

It constitutes a convex quadratic optimization problematic under linear constraints.
We then obtain a final classification function:

$$f(x) = \omega.x + b = (\sum_{i=1}^{M} \lambda_i y_i x_i).x + b$$

With $\omega = \sum_{i=1}^{n} \lambda_i y_i x_i$
$b = y_i - \omega . x_i \quad , i, \lambda_i \neq 0$

In the case of not linearly separable data, we may violate some constraints in order to optimize the margin. We then focus more on the robustness over the learning error rate.
To make such a thing happen, we use the relaxation variables ξ_i and regularization parameter C (see fig.1).
Then, the primal optimization problem becomes:

$$min_{(\omega,b)} \frac{\| \omega \|^2}{2} + C \sum_{i=1}^{M} \xi_i$$

With:

$$y_i(\omega . x_i + b) \geq 1 - \xi_i \quad , \xi_i \geq 0 \quad , \forall\, i \in [1, M]$$

We use Lagrange multipliers [11] to solve this optimization problem:

$$max_\lambda \omega(\lambda) = -\frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \lambda_i \lambda_j y_i y_j . x_j x_j + \sum_{i=1}^{M} \lambda_i$$

With:

$$\sum_{i=1}^{M} \lambda_i y_i = 0 \quad , \qquad 0 \leq \lambda_i \leq C \, , \forall\, i \; in [1, M]$$

The kernel technique is an efficient method to treat the problem [12]. We transform the existing data into another dimension that has a clear dividing margin between classes of data.
In our case, we transformed the learning set $x_i \in R^m$ into $\Phi(x) \in R^M$ in which the data become linearly separable.

We then replace the scalar product $x_j x_j$ by its corresponding $\Phi(x_i)\Phi(x_j)$.
The kernel function is defined as follow:

$$K(x_j , x_j) = \Phi(x_i)\Phi(x_j)$$

We then obtain a new classification function:

$$f(x) = \sum_{i=1}^{N} \lambda_i \, y_i \, K(x_i , x) + b$$

To choose the suitable Φ function, it should satisfy the Mercer conditions [13] [14].
In our study we used cross-validation to choose the best kernel and parameters for the SVM algorithm.

B. Supervised Learning - Classification

Artificial Neural Networks

a.1) Artificial Neuron

A neuron is a neural network's basic processing unit. It is linked to information sources as input and returns output information.
The neuron receives a number of input data, each data point is processed and transmitted with its weight.

A weight is a coefficient w_i calculated for every data point x_i. The i^{th} neuron receives the information $(w_i \times x_i)$.

This data is conceded to the activation function in order to get the final output [15].

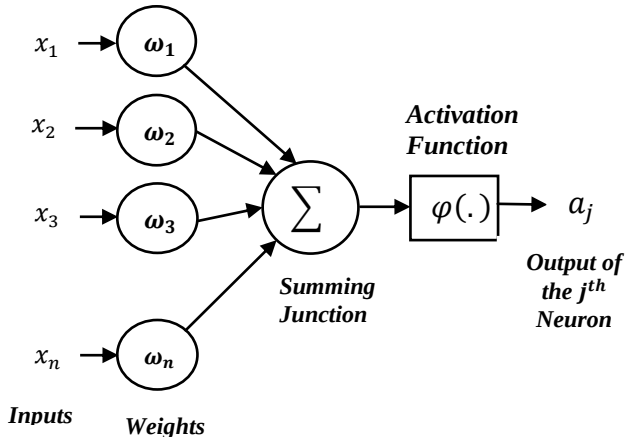

Fig.1. Single artificial neuron

We denote:

$\omega_{i,j}$ For $(1 \leq i \leq n)$ and $(1 \leq j \leq p)$, the weight connecting the information x_i and the neuron j

And

a_j the output of the j^{th} neuron presented with the following equation:

$$\forall\, 1 \leq j \leq p : \; a_j = \varphi(\sum_{i=0}^{n} w_{i,j} \times x_i) \qquad (5)$$

The output a_j may become a stimulus for neurons in the next layer.

We used the sigmoid as the activation function given by:

$$\varphi(\gamma) = \frac{1}{(1 + e^{-\gamma})} \qquad (6)$$

In a neural network, activation function is used to produce a non-linear decision boundary via linear combinations of the weighted inputs.
The non-linearity of the activation functions within hidden layers makes artificial neural networks one of the best universal approximator [16].

a.2) Multilayer perceptron

The multilayer perceptron process in a feed-forward way, it is shortened as (FFNN) feedforward neural network structure. It is the basic type of artificial neural networks which is able to approximate generic function classes. Feed-forward architecture contains an input layer, an output layer and one or more hidden layers made of processors called neurons. These neurons, each having a specific weight, are entirely interconnected with neurons of ensuing layer.

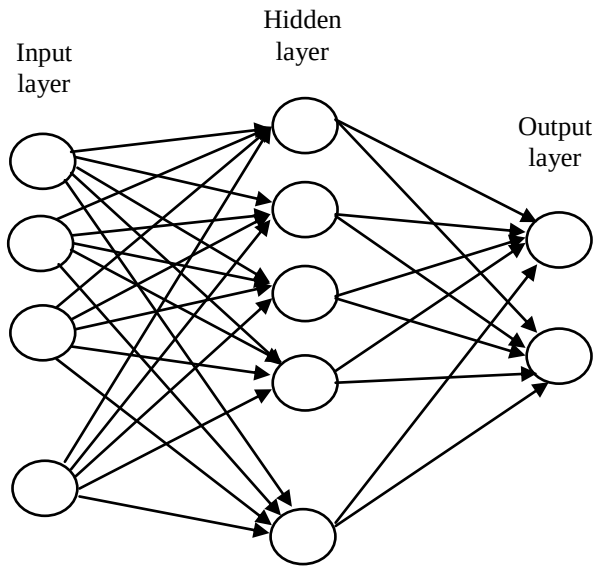

Fig.2. MLP architectural graph with one hidden layer

During the learning phase (process of finding an optimal set of weight parameters ω to approximate the original problem behavior), training data is given as pairs of $(x_k, d_k), k = 1, \ldots, P$ where d_k is the desired outputs for inputs x_k of P training samples.

The Backpropagation learning algorithm is applied for minimization of error function [15] defined by:

$$E = \frac{1}{2} \sum_{k \in T_r} \sum_{j=1}^{m} (y_j(x_k, \omega) - d_{jk})^2 \qquad (7)$$

d_{jk} : j^{th} element of d_k
$y_j(x_k, \omega)$: j^{th} neural network output for input x_k
T_r : A set of training data

III. RELATED WORKS

In our approach, we combine unsupervised and supervised techniques for packets traffic identification. The majority of existing work using either supervised or unsupervised techniques deal with intrusion detection through the use of the KDD'99 dataset. A key issue concerning an intrusion detection system is that traffic vectors to analyze in order to identify network attacks are high dimensional vectors [17]. Numerous research studies based on unsupervised detection methods use clustering and detection of outliers. In [18], Eskin et al. show good results with three partitioning algorithms: a fixed-width clustering, an optimized version of KNN, and a one-class support vector machine. Xiangmei Li [19] proposed to optimize IDS based on neural networks using multi-sub-classifiers system. It deals with the four types of attacks present in the dataset, and for every type of attacks he assigned a specific classifier. The four types of attacks are DOS, Probe, Remote 2 Local and User 2 Root. Even though the accuracy of the proposed system is optimized and better results have been shown, it has not shown how quickly the classification process is, since each sub-classifier should inspect the entire incoming traffic causing redundant and time - consuming process.

PCA is used for a large set of purposes, including subspace understandings, identification systems, abnormality detection and classification. PCA is actually processed in batch mode and all data must be accessible for the calculation of the PCA. Thus, batch PCA is impractical for stream data applications such real time intrusion detection, as it must be retrained whenever new data comes. To remedy this situation, several iterative algorithms were used, like the least square recursive, stochastic gradient, etc. [20] Other studies proposed a hybrid approaches for intrusion detection by combining supervised with unsupervised neural networks, or by using combinations of neural networks and other data mining methods.

Chen, et al. [21] proposed a hybrid intrusion detection approach based on a flexible neural tree combined to an evolutionary algorithm and particle swarm optimization. The result shown, has proved the efficiency of the method. Jirapummin et al. [22] employed a hybrid approach to visualize intrusions using Kohenen's SOM and also to classify intrusions using a resilient propagation neural network. Vladimir Bukhtoyarov et al. [22] proposed a neural network ensembles approach in which they joined many trained neural networks in order to combine outputs to get a solution for the classification problem. But the approach is hard to implement because of the complexity of the network topologies in real systems. Moreover, the effectiveness of the knowledge exchange between the ensemble members has not been proved.

Admittedly, different methods for constructing a hybrid IDS have been proposed, but how to design and implement this system greatly influences performance when detecting intrusions. Our motivation for using the hybrid ANN is to overcome the limitations of individual ANN discussed below in the first part.

IV. METHODOLOGY

A) Dataset

a) CICIDS 2017 dataset

The lack of suitable free datasets to evaluate anomaly detection systems is the biggest challenge an evaluation can face. To measure the efficiency of any detection approach, we must study it and experiment it with data that replicates the actual traffic of modern networks at an adequate level.

Data sets available in the field of network intrusion detection systems for machine learning are limited. DARPA dataset (KDDcup99, NSL-KDD) is one of the few yet at the same time widely used datasets. Even though they are the most comprehensive datasets available, they may not be a great representative of currently existing realistic networks and yet still suffer from the problems discussed in [23].

Most of the proposed works, in the field of intrusion detection using machine learning techniques, focus on accuracy over productivity and are tested on the old DARPA dataset.

There exist other datasets for the same purpose, but some of these datasets actually lack volumes and variety of traffic[24] and are composed of packets that do not reflect modern trends[5].

Based on the evaluation framework elaborated by [25], there are critical standards necessary to build a consistent benchmark dataset. In the following, here are these criteria [5]:

- **Complete Traffic:** different machines and real-world attacks.
- **Labeled Dataset:** benign and attack labels
- **Complete Interaction:** separated networks
- **Complete Capture:** All traffic
- **Attack Diversity:** Most common attacks [26].

In our study, we wanted to test our approach on data close to the one generated in our network. To gauge the performance of any detection approach, we need to really experience it with data that simulates real traffic to an adequate level in modern networks. To this end, we have opted for a fairly current IDS assessment dataset containing real world traffic data.

The CICIDS2017 dataset [5] given by the Canadian Institute for Cyber security and UNB (University of New Brunswick) in 2017.

In a physical testbed implementation, the CICIDS2017 data set was generated using real devices that interactively generate real traffic reflecting complex and realistic network traffic of nowadays infrastructures [27].

b) Traffic composition of the CICIDS2017 dataset

CICIDS2017 Dataset comprises fairly benign, most current, common attacks [28], which reflects real-life data. Moreover, it comprises labeled flows of traffic analysis based on timestamp, source and destination IPs, source and destination ports, protocols and attacks.

Table1. Characteristics of CICIDS2017 dataset

Total number of instances	2830540
Number of features	83
Number of classes	15

The data collection period began on Monday and ended on Friday for a total of 5 days. Monday only benign traffic generated. The table below shows the attacks which were performed in mornings and afternoons of the next four days. Below is the traffic distribution of the captured data, it follows a normal distribution.

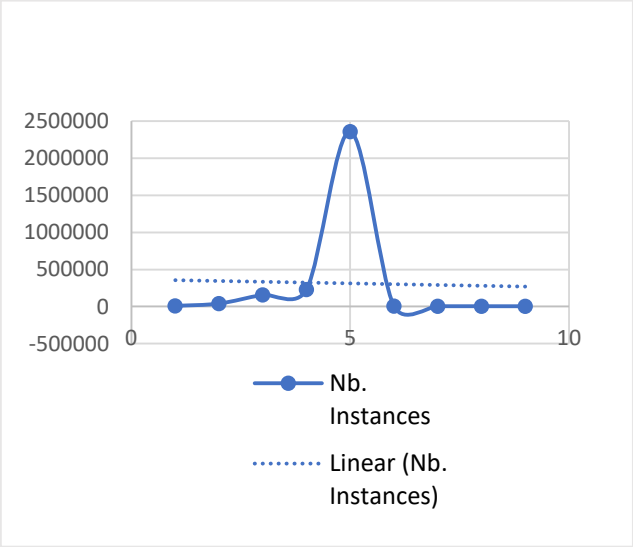

Fig.3. Traffic Distribution of the CICIDS2017

B) B) CICIDS 2012 dataset

Available datasets for machine learning in the field of network intrusion detection systems is limited. One of the few but at the same time, widely used datasets is the DARPA datasets (KDDcup99, NSL-KDD). Although, they are the most comprehensive existing datasets, they don't represent actual computer networks and still suffer from the problems discussed here [4]. In our study, we wanted to test our approach on data close to the one generated in our network. For this, we opted for a relatively current IDS evaluation dataset containing real-world representative traffic data. The UNB ISCX 2012 dataset [28] generated by UNB (University of New Brunswick) and collected from modern complex networks.

UNB ISCX 2012 dataset has been generated in a physical testbed implementation using real devices that dynamically generate real traffic which reflects network traffic and intrusions [28].

The UNB ISCX 2012 Intrusion Detection Evaluation Dataset characteristics are:

- Realistic network and traffic
- Labeled dataset
- Complete capture
- Diverse intrusion scenarios

a) Traffic composition of the UNB ISCX dataset

The traffic generated by UNB center is based on real traffic for HTTP, SMTP, FTP, SSH, IMAP and POP3 which is vital for the realism and effectiveness of the data set. Malicious activities were generated with multiple scenarios to make them sophisticated and hardly detectable [28].

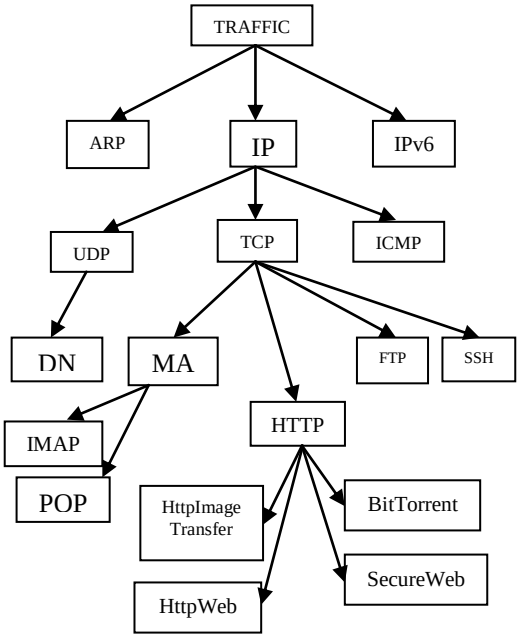

Fig.4. UNB dataset traffic composition diagram

Scenarios based on predefined user profiles were executed to generate normal traffic, besides various multi-stage attacks to mimic malicious behavior.

Our dataset is composed of the whole flows generated during the experiment on data collection at the UNB laboratory. Data was collected continuously during seven days and flows are distributed as follow:

Table 2. UNB-ISCX dataset major flows distribution

Type of flow	*Number of records*
TCP	1,941,454
UDP	498,032
ICMP	10,689
Normal	2,381,532
Attack	68,792
Total	2,450,324

The traffic observed in the dataset contains the internet most used services; this reflects the realism of the test bed network and shows that the experiment of generating a representative flow was conducted rigorously.

Traffic composition shows the network protocols and services most present in the dataset, the majority of it is IP traffic which contains mainly TCP packets as shown below.

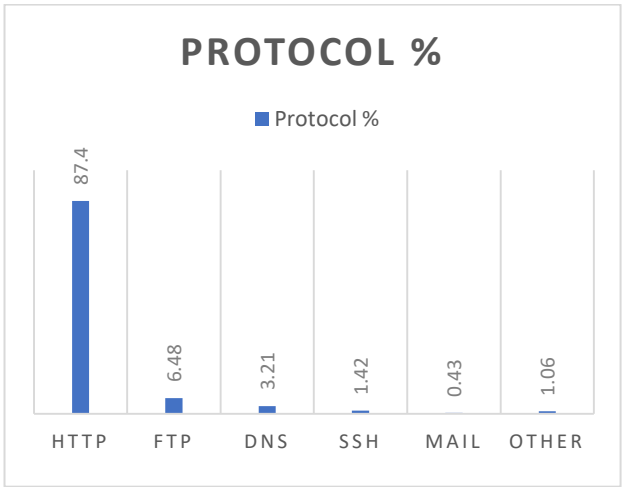

Fig.5. Protocols and applications percentage of the dataset

V. PROPOSED WORK

We will present an algorithm and a diagram in this part to better illustrate our proposed approach to the classification of network traffic. While most of the related existing studies propose a single machine learning classification model, we build a two-level network traffic identification framework with a combination of x-means and support vector machines and n-artificial neural network classifier.

The first stage of our framework is a combination of two algorithms; X-means, we will use this algorithm for a first clustering step, and then we will optimize this clustering using the SVM algorithm. This whole process aims to determine the optimal cluster for every network entry. The next step, is to apply an artificial neural network to every cluster, the ANN is trained only for each cluster in order to reduce the training and testing time. So the final process is a classification optimization process.

The system obtained clearly shows very good performance: better accuracy and a reduced training and testing time.

a) Unsupervised clustering

Our study approach for anomaly detection was validated in three steps.

First step is vector attributes normalization including all traffic data (features): the incoming traffic data is normalized and each input vector is constructed using all packet features.

Second step: X-means based cluster detection: the clustering algorithm x-means, will determine to which cluster the incoming vector will be directed.

The third step: Cluster identification optimization using SVM. In this step, the SVM optimizer, while maximizing the margin between clusters, will confirm the optimal position of the incoming packet. Then each entry of every cluster will pass through a specific artificial neural network trained to classify it.

b) Supervised classification

Coming to this level, each entry of the determined clusters will be evaluated by its specific ANN.

First, we need to choose features that will give us the best accuracy and the fastest prediction model since we have vectors of the same cluster. So, in order to reduce dimensionality, we used the recursive feature elimination algorithm [29] where different subsets of features are evaluated and compared to each other in order to determine the best subset based on the model accuracy.

Depending on the first stage clustering, the incoming packets are identified by their network features. Incoming classes have their specificities and the input vector of every ANN-classifier of the second stage classification will be treated based on its cluster specificities. Some behavioral features may be included, based on header information of the incoming packets (such as the TCP window size, TCP flag bits and packet directions) [30].

Some applications use IP layer encryption, making it impossible to check the traffic payload. Our approach is non – payload based, so it is robust to encryption. Traffic packets are classified by evaluating the similarities between it and groups formed of packets having similar traffic patterns.

A) Proposed Framework Algorithm

The following is the algorithm:

- Traffic clustering with X-means and SVM: by inspecting the whole incoming information of packets, the X-means and SVM combination determine the best cluster for every packet.
- Feature dimensionality reduction: Once clustered, the vector is set of the best subset of features and then directed to its proper ANN classifier.
- Traffic classification: Finally, the incoming traffic is classified whether malicious or benign.

ALGORITHM: PACKETS CLASSIFICATION

INPUT: incoming packet Pi
OUTPUT: Class Pi: (normal/attack)
BEGIN
/*/**
FOR i **FROM** 1 **TO** n **DO**
FOREACH Pi **DO**
/* rapid packets identification*/
Create an input vector V_i
Normalize vector attributes
/* First-level: Clustering
X-means clustering: V_i is clustered
/* SVM cluster optimization*/
Maximize margin between clusters
Set corresponding identifier to V_i
V_i Becomes V_i [+id = cluster identifier]
/* Input feature selection */
FOREACH V_i (id) **FROM** Cluster_id ∈[1, N] **DO**
Select best features subset for V_i [id]
V_i [id] becomes U_i [id]
Send U_i [id] to ANN [id]
END
/* Artificial neural network classification */
Classify U_i [id]
U_i [id] classified: Normal/Attack
END
Return Class (Pi)
END
/*/**
END

B) The diagram

The diagram below illustrates the proposed framework of network traffic anomaly detection. First, we proceed by normalizing the vector attributes including all data features. Then, we apply the X-means algorithm to find clusters and after that, the cluster identification optimization is performed using the support vector machine algorithm. Once an input vector is clustered, it undergoes a feature selection step before it is evaluated by the artificial neural network trained on the same cluster's data which finally determines whether it is normal or attack traffic.

The diagram below illustrates the proposed framework:

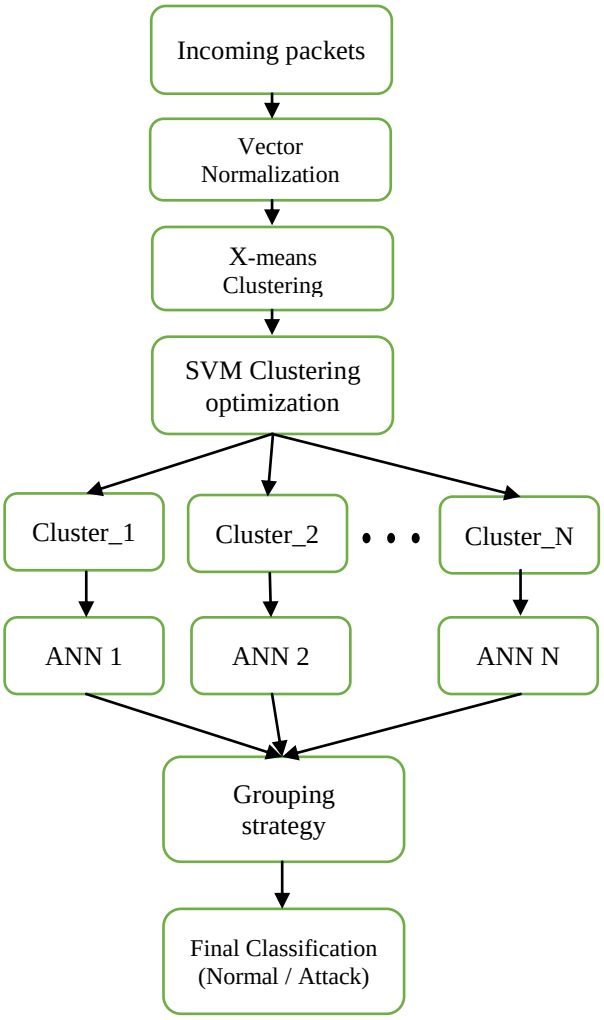

FIG.6. PROPOSED CLASSIFICATION FRAMEWORK

Incoming packet: Whole packet information passing by the network
X-means clustering: The algorithm specifies the cluster to which the incoming packet belongs

SVM: The SVM algorithm for cluster optimization
Vector V_i[id]: The clustered incoming vector with whole set of features and labeled with proper id (with its cluster identifier)
ANN id: An artificial neural network trained only for one cluster.
Clusters are constructed based on the incoming data, the number of clusters is not fixed. It may increase or even decrease over time depending on the learning process which allows our model to be adaptable and evolutive. This is mainly for the unknown traffic which may not be classified correctly in order to refine the overall accuracy of our model.

VI. EXPERIMENTAL RESULTS

Results comparison using CICIDS2012/2017 datasets:
The majority of studies aiming to experiment the effectiveness of soft computing techniques in the detection of intrusions relies on data available in the internet such as DARPA 98 or KDD-CUP 99. But these datasets are old and have many weaknesses, this is why a recent study conducted by [31], that shows some inherent problems such as the high level of redundancy which may biases the learning algorithms.
We tested our framework on both datasets (KDDCup/NSL-KDD) in order to measure its performance and also to compare our results with studies based on these datasets.

1) Results comparison using KDD Cup99 dataset:

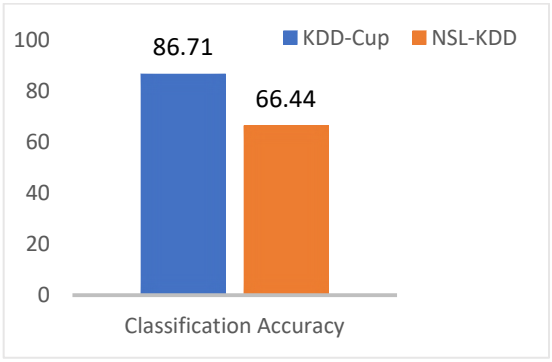

Fig.7. Performance of The model on KDDCup and NSL-KDD.

With regards to the (KDDCup / NSL-KDD) datasets, the results show that the performance of the classifier has declined clearly between the two datasets which confirm that the data used is the main important part of the learning process and affects the detection performance. This is to justify our choice of datasets.

The model performed well on the KDD-CUP99 dataset because of the high level of redundancy in it. It clearly declined when the same model was tested on the NSL-KDD which may be explained by an over fitting.

As a solution to these problems, a newer version dataset was proposed CICIDS2012 and CICIDS2017 where all records from each difficulty level are proportionally distributed.

We tested our framework on both datasets in order to be able to measure its performance and also to compare our results with studies based on these datasets.
The results show that the performance of the artificial neural network classifier has declined clearly between the two datasets which confirm that the data used is the main important part of the learning process and affects the detection performance.

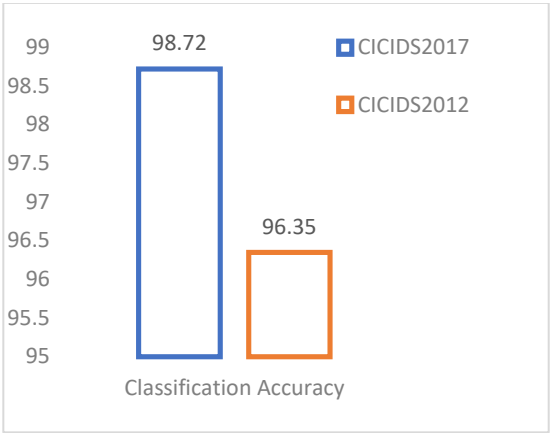

Fig.8. Performance of The model within Two different Datasets.

The detection model has to classify the incoming traffic whether normal or attack. We aimed to optimize the existing ANN based intrusion detection systems that classify the incoming traffic directly by analyzing the whole packets.

The first test applies to all packet features. Secondly, a specific artificial neural network (ANN [id]) is applied to a subset of the best features selected for the given network service after applying unsupervised learning with x-means and support vector machines.

Table 3. Classification accuracy results on both datasets

	CICIDS2012	CICIDS2017
Cluster 1	96.19	98.97
Cluster 2	93.11	96.62
Cluster 3	96.58	97.94

The first step was to apply artificial neural network classification on clusters before and after SVM optimization. As shown above, the classification accuracy has increased after applying the cluster optimization technique using SVM.

In this stage of analysis, we used common metrics to evaluate information retrieval in order to be capable of comparing results with other studies, especially the ones based on the same dataset:
Precision: Defined as the ratio of correctly classified attack flows TP (True positive), against all the classified flows TP+ FP (False positive)
Recall:
It is the ratio of correctly classified attack flows TP against all generated flows TP+FN (False negative)

F-Measure: A harmonic mixture of the precision and recall.

Here are the results comparison between our model and models used by [5] while performing the same task of network traffic classification.

Table 4. Classification results comparison

Optimized ANN	*Pr*	*Rc*	*F1*
CICIDS2012	0.96	0.96	0.96
CICIDS2017	0.99	0.98	0.98

The table above gives an overview of the performance evaluation for five commonly used machine learning algorithms, Multilayer perceptron (MLP), Adaboost, Naive-Bayes, K-Nearest Neighbors (KNN), Random Forests and our model we call it Optimized ANN.

Table 5. Results comparison for execution time

Algorithm	*Execution time (milliSecond)*
CICIDS2012	32.39
CICIDS2017	21.07

The detection time has decreased by applying our optimized model. The fact that the artificial neural network processes on one cluster makes it faster and accurate. The training time is reduced because the ANN doesn't need to make computational operations to the whole traffic.

VII. CONCLUSION

In this study, we presented an approach of intrusion detection optimization based on the combination of both unsupervised and supervised machine learning techniques in order to cover the whole process of traffic identification. In addition to these high performances, the proposed framework is tested on a newer dataset and thus more representative of current computer networks. We count on that to make sure that the proposed framework is best suited for modern network architectures and we are designing a cloud based distributed infrastructure for real time traffic analysis for better results.

The framework of the elaborated model will be combined with our previous work [32] to constitute a final complete architecture for network traffic classification. Both studies showed great results and encourage us to elaborate an optimal model based on a mixture of advanced machine learning techniques that will be tested with a platform for real traffic capture [33].

Our focus in the coming studies will be on stream data from real time network traffic.

REFERENCES

1. A.Jelsiana Jennet, Dr. J Frank Vijay. International Journal of Applied Engineering Research ISSN 0973-562 Volume 10, Number 17 (2015) pp.12635-12641.
2. Piero P.Bonissone, "Soft computing: the convergence of emerging reasoning technologies," Soft Computing Journal, vol 1, no 1, pp. 6-18, Springer-Verlag 1997.
3. S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Costbased modeling for fraud and intrusion detection: Results from the jam project," discex, vol. 02, p. 1130, 2000
4. J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000
5. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (ICISSP), Purtogal, January 2018
6. Jacob Kogan, "Introduction to Clustering Large and High-Dimensional Data", University of Maryland, Baltimore County
7. D Pelleg, Aw Moore, "X-means: extending k-means with efficient estimation of the number of clusters". School of computer science, Carnegie Mellon University, Pittsburgh. Icml, 2000
8. Mardia KV, Kent JT, Bibby JM (1979) Multivariate analysis. Academic Press, London
9. Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. ICM
10. web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf
11. P. bussotti, On the Genesis of the Lagrange Multipliers. journal of optimization theory and applications: Vol. 117, No. 3, pp. 453–459, June 2003
12. B. Scholkopf, K. Tsuda et J.P. Vert, Kernel Methods in computational biology. MIT Press, 2004
13. G. Kimeldorf , G. Wahba. Some Results on Tchebycheffian Spline Functions. Journal of mathematical analysis and applications Vol 33. No.1 January 1971
14. B. Scholkopf, R. Herbrich and A. Smola. A Generalized Representer Theorem, D. Helmbold and B. Williamson (Eds.): COLT/EuroCOLT 2001, LNAI 2111, pp. 416–426, 2001
15. http://alp.developpez.com/tutoriels/intelligence-artificielle/reseaux-de-neurones
16. Hornik, K., M. Stinchcombe, and H. White, ''Multilayer Feedforward Networks are Universal Approximators,'' Neural Networks, Vol. 2, 1989, pp. 359–366
17. Samira sarvari, et al., GA and SVM Algorithms for Selection of Hybrid Feature in Intrusion Detection Systems. IRECOS, Vol 10, No 3 (2015)
18. E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data," in Applications of Data Mining in Computer Security, Kluwer, 2002
19. 978-1-4244-5143-2 ©2010 IEEE DOI: 10.1109/ITAPP.2010.5566641 Internet Technology and Applications, 20-22 Aug. 2010
20. Bannour S. and M. R. Azimi-Sadjadi. Principal component extraction using recursive least squares learning, Neural Networks, IEEE Transactions on, 1995,6, 2. 457–469
21. Chen, Y. H., Abraham, A., & Yang, B. (2007). Hybrid flexible neural-tree-based intrusion detection systems. International Journal of Intelligent Systems, 22(4), 337–352
22. Jirapummin, C., Wattanapongsakorn, N., & Kanthamanon, P. (2002). Hybrid neural networks for intrusion detection system. Proceedings of ITC–CSCC, 928–931
23. J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000
24. Brown C, Cowperthwaite A, Hijazi A, Somayaji A. Analysis of the 1999 DARPA/Lincoln laboratory IDS evaluation data with netadhict. Computational intelligence for security and defense applications. Piscataway, NJ, USA: IEEE Press; 2009. p. 67e73
25. Gharib, A., Sharafaldin, I., Habibi Lashkari, A., and Ghorbani, A.A. (2016). An evaluation framework for intrusion detection dataset. In 2016 International Conference on Information Science and Security (ICISS), pages 1–6
26. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-dec-2016.pdf
27. Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security, 31(3): 357-374
28. Sharafaldin, I., Gharib, A., Habibi Lashkari, A., , and Ghorbani, A. A. (2017). Towards a reliable intrusion detection benchmark dataset. Software Networking, 2017:177–200

29. L. Bobrowski Feature subsets selection based on linear separbilty, In: Lecture Notes of the VII-th ICB Seminar: Statistics and Clinical Practice, ed. by H. Bacelar-Nicolau, L. Bobrowski, J. Doroszewski, C. Kulikowski, N. Victor, June 2008, Warsaw, 2008
30. T.T.T.Nguyen, G.Armitage, Asurvey of techniques for Internet Traffic Classification using Machine Learning. 4 th edition 2008 of IEEE
31. Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. Paper presented at the The Second IEEE international conference on Computational intelligence for security and defense applications CISDA'09, IEEE Press Piscataway, NJ, USA ©2009
32. Ichrak Lafram, N.Berbiche, J.Elalami, A random forest estimator combined with n-Artificial neural network classifiers to optimize network intrusion detection. IJAER, ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 5835-5843
33. S. El idrissi, N.Berbiche, F.Guerouate, Performance evaluation of web application security scanners for prevention and protection against vulnerabilities. IJAER Vol 12 n 17 (2017)