ADDRESSING IMBALANCED CLASSES PROBLEM OF INTRUSION DETECTION SYSTEM USING WEIGHTED EXTREME LEARNING MACHINE

Mohammed Awad[1] and Alaeddin Alabdallah[2]

[1]Faculty of E&IT, Dept. of Computer Systems Engineering,
Arab American University, Palestine
[2]Faculty of E&IT, Dept. of Computer Engineering,
An-Najah National University, Palestine

ABSTRACT

The main issues of the Intrusion Detection Systems (IDS) are in the sensitivity of these systems toward the errors, the inconsistent and inequitable ways in which the evaluation processes of these systems were often performed. Most of the previous efforts concerned with improving the overall accuracy of these models via increasing the detection rate and decreasing the false alarm which is an important issue. Machine Learning (ML) algorithms can classify all or most of the records of the minor classes to one of the main classes with negligible impact on performance. The riskiness of the threats caused by the small classes and the shortcoming of the previous efforts were used to address this issue, in addition to the need for improving the performance of the IDSs were the motivations for this work. In this paper, stratified sampling method and different cost-function schemes were consolidated with Extreme Learning Machine (ELM) method with Kernels, Activation Functions to build competitive ID solutions that improved the performance of these systems and reduced the occurrence of the accuracy paradox problem. The main experiments were performed using the UNB ISCX2012 dataset. The experimental results of the UNB ISCX2012 dataset showed that ELM models with polynomial function outperform other models in overall accuracy, recall, and F-score. Also, it competed with traditional model in Normal, DoS and SSH classes.

KEYWORDS

Machine Learning, Weighted Extreme Learning Machine, Intrusion detection system, Accuracy, UNB ISCX2012.

1. INTRODUCTION

With the increase of the services that offered by the computational systems on computer networks, it's necessary to maintain the reliability, integrity, and availability of these systems, which makes the information security of these systems more important. A very important problem is the increasing of attackers on these systems [1]. The operations of cyber-attacks able to cause significant economic damage to companies and organizations, thus attacks the national security of any country [2]. There is also a greater complexity of Intrusion attacks due to the exponential growth of mobile devices and cloud environments. Intrusion detection (ID) in cyberspace is a multi-disciplinary problem. One side of the problem is a cyber-security problem, and the other side is the statistical, knowledge-based and machine learning fields that represent the factories that produce the pool of solutions. This paper focuses on the machine learning solutions of the ID problem.

The security problem becomes more complicated because of the high connectivity of the world via the internet. Studying communication, computer network systems, protocols, and services fields, which represent the main parameters of the internet appear wide distributions of the faults for most computing components of the system. These faults caused the previous, current and future attacks. In paper [3] the authors present some of these facts where TCP protocols suffer from a list of security flaws.

As mentioned in [4], the ID solutions are classified into one of three common methodological classes. The first class called misused or signature-based IDS, in this approach, different normal and abnormal known rules or patterns are classified in the training phase from labeled data, and then the generated models are used to make a prediction for the unseen data. Although these models produce high accuracy for detecting known and some variant of unknown attacks, they fail in detecting zero-day attacks. The second class is the anomaly-based IDS, it depends on the closed world hypothesis [5], which supposes hat the model has the capability to capture all normal behaviors in the training phase, and then developed models are used to measure the deviation from the normal behavior in the testing phase to predict the unseen data as normal or anomalies. This model can detect the zero-day attacks but with total accuracy not better than the preceding one. The Third one is the hybrid approach which combines both previous approaches in one model.

The network ID field has a wide set of open issues, some of them will be illustrated in the following few paragraphs. Firstly, the scalability issue for ML algorithm or any other tool that used to solve the ID problem. Computer networks generate a huge volume of traffic which is increasing more and more due to the expansion of the Internet services, increasing the mobile devices and the movement toward the internet of things (IoT) technology [6] Secondly; it is related to labeling the records collected from the traffic correctly. This process needs extra efforts from experts to label the traffic correctly. It increases the need to benefit from the huge size of unlabeled records besides the correct labeled records. Third, this issue related to an anomaly detection method, it is about the inability of the data collector to aggregate a pure set that includes all variant of either normal traffic or abnormal traffic in case that the zero-day or newly attacks are renewable. This is summarized with the impossibility to have the close world in our domain. The question is if the incremental learning by the ML algorithm can address this dilemma that based on the closed world assumption which is impractically in our domain. Fourth, it is a multifaceted issue that this paper focused on; it is about the sensitivity of the IDS toward the errors. Most works in this field concerned about increasing the detection rate and decreasing the false alarm rate (FAR) in order to improve their system accuracy [7] [8]. Even the number of misclassified records is little, in huge traffic; it represents a big problem for the clients of network services if the normal traffic treats as an anomaly, and it makes a big headache for network administrators to treat a huge amount of false alarms. On the other hand, the exact detection of abnormal traffics helps the system administrator to solve the problem easily. Most studies performed the performance of their approaches using the NSL-KDD dataset [10] [12]; which had succeeded in improving the overall accuracy, this phenomenon called accuracy paradox [10]. The detection of the minor attacks will be a crucial issue [11] if it is related to minor attacks that have a high level of security.

In this paper, we are interested in improving the accuracy of IDS for the new attacks and mitigating the existence of accuracy paradox problem. So, a weighted algorithm which is Extreme Machine Learning (ELM) with different weight schemes stratified sampling and with optimizing for some parameters of these algorithms was consolidated to solve this problem. WELM is the fast and simple NNs that solve the time consuming iterative process in feedforward neural

networks (FFNN). Furthermore, the evaluation phase was processed inconsistent and fairway; it was taken into account the data selection reasons and the way of performing different tests. These experiments were performed on one benchmark dataset which is UNB ISCX2012. The UNB ISCX 2012 is a benchmark dataset that includes real-time contemporary traffic for normal and attack behaviors. It is generated systematically so this makes it modifiable, extendable, and reproducible dataset. It includes four types of attack scenarios which are inside network infiltration, Hypertext Transfer Protocol (HTTP) denial of service, IRC Botnet Distributed Denial of Service Attacks (DDoS), brute force SSH. These are some of the open issues in this field and there are others included in literature. They prevent a lot of these efforts, especially those developed using anomaly-based methods to deploy in operational real-world environments [5]. The awareness of the pressing needs to improve power and dynamics security tools that protect the contemporary computing systems emphasize the great interest of researchers of both communities to improve the IDSs.

2. RELATED WORKS

ID problem has great interest from the researchers; part of these efforts concentrated on review the problem from a different point of view, one of the recent surveys [16] studied different categories of anomaly ID methods, they are classifications, statistical, information theory, and clustering. The Machine learning and data mining community suggest many tricks to solve the deficiency of its models in predicting the small classes of ID problems. Different approaches suggested in [17] are to solve the imbalanced classes. The oversampling and undersampling are the common two resampling methods in literature while the cost function was added to different ML algorithms to address its sensitivity to imbalanced classes. Different cost functions suggested in related works and applied with different data mining and machine learning tools, such as ANNs, SVM, clustering, DATA, naive Bayes, EA [17]. In [18], an enhanced method called sample selected extreme learning machine (SS-ELM) is used to classify the ID of cloud servers. Then, the selected sample is given to the fog nodes/MEC hosts for training. This design can bring down the training time and increase detection accuracy. Several Network ID models were proposed and tested in the last decade. These models were built based on a well-known dataset called NSL-KDD [15]. Most of these efforts concentrated on either making normal or abnormal record prediction or multi-class classification predictions.

On the other hand, a few efforts tried to build sub-models like in [19]. This paper develops a new hybrid model consists of Meta Pagging, Random Tree, REP Tree, AdaBoostM1, Decision Stump, and Naïve Bayes, that can be used to estimate the intrusion scope threshold degree based on the network transaction data's optimal features that were made available for training. The results revealed had a significant effect on the minimization of the computational and time complexity involved when determining the feature association impact scale. The authors in [20] proposed a hybrid model for detecting different classes of DoS attacks. In this model, the Particle Swarm Optimization algorithm used as feature selection methods, then it used SVM to build a model for predicting the different classes of DoS attacks. These efforts and others go with the advice that recommended narrowing the scope of the ID problem in order to reduce the FAR when building the ML models. To compare the performance of the supervised or unsupervised ML models as ID solutions, the authors in [21] have built a framework and made a number of experiments. They demonstrate that the supervised learning model does better if the test data contain known or a variant of known attacks. While both have close performance in the dataset contains unknown attacks.

Many efforts performed to generate benchmark contemporary and real-time traffic datasets, one of these done by ISCX. A systematic approach was used to generate modifiable, extendable, and

reproducible dataset [14] which is known as UNB ISCX2012. It includes real traffic related to FTP, HTTP, IMAP, POP3, and SMTP and SSH protocols. UNB ISCX2012 dataset includes four types of attacks in addition to the normal traffic; these attacks are inside network infiltration, HTTP denial of service, IRC Botnet DDoS, Brut force SSH. In [22] the author used a supervised ML method to detect DDoS depends on network Entropy estimation, Co-clustering, Information Gain Ratio, and Extra-Trees algorithm. The unsupervised phase of the approach allows reducing the irrelevant normal traffic data for DDoS which allows reducing false- positive rates and increasing accuracy. Experiments performed using datasets NSL-KDD, UNB ISCX 12 and UNSW-NB15.The authors in [23] applied a hybrid scheme that combines deep learning and support vector machine to improve accuracy in ISCX IDS UNB dataset classes. The result indicated the combined model outperforms SVM alone in terms of both accuracy and run-time efficiency.

Another kind of hybrid model was introduced in the literature for our problem, but at this time, it was combined with multiple kernels together [24]. Multiple Adaptive Reduced Kernel Extreme Machine Learning Model (MARK-ELM) was proposed. This work proposed a framework that used the AdaBoost method to combine each set of Reduced Kernel Multi-class ELM models in order to increase the detection accuracy and decrease the false alarm. Twelve combined models were performed, seven of them got greater than 99% accuracy in total, but only one of them got greater than 30% for U2R class and it got 60.87%, which confirms the existence of accuracy paradox problem in these experiments. Another multi-level ID model was proposed in [9]. It passed through three phases. In the first phase, the categorical records were used to generate a set of rules to binary normal, abnormal prediction using the well-known Classification and Regression Trees (CART) algorithm. The second phase included building three predictive models using SVM, Naïve Bases, and NNs in order to determine the exact attacks categories for only three of the attack, while U2R attacks excluded because of the insufficient amounts of records, this confirms the existence of the imbalanced class problem. In this phase, it used both the row data features once and the features were generated using Discrete Wavelet Transformation (DWT) methods in again, the models were built using the last set of features performed better than the features of raw data. In the last phase, it deployed a visual analytical tool called iPCA to perform visual and reasonable analysis of the results. This is a remarkable suggestion or solution for the recommendation assigned in [5] about the clearance of the interpretation of the result at the evaluation step of our problem.

The author in [25] used the UNB ISCX2012 dataset to build multiple class classification solution for the ID problem. The SVM with Gaussian radial base function (RBF) and polynomial kernels, MLPNN and Naïve Based algorithms are deployed to build different models. The SVM with polynomial kernel had the best performance than others. There are two remarks related to this work, the first, the number of records of this dataset as it is included in this paper is inconsistent with the real number of records of the UNB ISCX dataset. They assumed that the number of records of Botnet and DoS attacks equals 5 and 40 sequentially, while the correct number of these classes is 37460, 3776 sequentially. Second, “All the tests were carried out on the same training and testing dataset” which a subset was selected randomly with respect to the huge classes. The performance of these experiments is not fair to reflect the correct performance of that algorithm on this dataset or on any other subset else. A lot of ML algorithms were used, and many tricks and enhancements also were deployed in order to improve the ID solutions, they could increase the detection rate and also decrease the false alarms in total but they failed to detect the rare but serious attacks.

In the research presented in [26], the authors present a framework for anomaly detection depending on the Bayesian Optimization technique to tune the parameters of SVM-RBF, Random Forest, k-Nearest Neighbor algorithms. The performance of the proposed model evaluated using the ISCX 2012 dataset. The produced results are effective in accuracy, precision, low-false alarm rate, and recall. This paper [27] depends on 2 stages; building prediction models for each type of attacks separately and optimizing the model with the highest accuracy. Then, build a prediction model for all attacks together using deep learning with the smallest number of features and we optimize the model to achieve the highest accuracy. The model applied UNB ISCX 2012 dataset.
In this work, we have deployed Extreme Machine Learning (ELM) with different weight schemes, stratified sampling and with optimizing for some parameters of these algorithms are consolidate to improve the accuracy of IDS for the new attacks and mitigate the existence of accuracy paradox problem.

3. METHODOLOGY

In this chapter, the proposed method is illustrated, it aims to improve the predicting accuracy of the small and serious classes of the ID problem co-occurrence with preserve the overall accuracy. It starts with emphasizing the datasets selection considerations. Then, it illustrates different preprocessing steps which include data type portability, data cleaning, feature selection, and stratified sampling. Next, it illustrates the deployed WELM model; they used to implement ID solutions. Finally, it includes the evaluation process.

3.1 Dataset Selection Considerations

There are still shortages in the available datasets in the ID domain in spite of the great efforts that were exerted in this field [14] [13]. These datasets are divided into two categories which are simulated-based datasets and real-time datasets. Most the considerable public benchmarked datasets are simulated-based datasets, they cannot reflect the nature of the contemporary traffic and there is no possibility to modify or extend or reproduce these old datasets. The public real-time datasets often subject to heavy anonymization in order preserve privacy. The dataset anonymization is a process of hiding the critical data of these sets like payload content, real IP-addresses, and others. CAIDA (2011), and LBNL are an example of public real-time datasets which they are heavily anonymized and totally removed payload. Furthermore, most datasets suffer from labeling problem regards the correctness or the completeness. Some of the important datasets in the ID field are KDD-CUP99, NSL-KDD, UNB ISCX 2012 and Kyoto University dataset. The UNB ISCX 2012 dataset includes two small classes, this is evident from figure 1it is an important and sufficient selection at this scope, so it was suggested to perform the primary experiments in this work. We are interested in selecting a contemporary and real-time dataset. So, the UNB ISCX 2012 suggested performing the experiments. It is a benchmark dataset, and it is included real-time contemporary traffic for normal and attack behaviors. It is generated systematically so this made it modifiable, extendable, and reproducible dataset. To proof the proposed idea, we performed experiments based on the general method in this paper using the complete records of all attacks in addition to some randomly selected normal subsets that have the same size of attack records, these subsets have small classes. This is evident from figure1which is shown the distribution of the records for that subsets.

3.2 Pre-processing Phase

Data preprocessing includes many steps [28] that depend on the nature of the data. Different preprocessing sub-steps were used; they included data-type portability, data cleaning, feature selection, and stratified sampling. TheUNB ISCX 2012 ID Evaluation Dataset consists of 19

features, they listed in table 1. As a feature selection step, the cumulative and redundant records were excluded. So, only the main 12 features were used in our experiments. The selected features fall into two categories which are nominal and numerical features. The nominal features converted to numeric features. Most features did not follow a balanced scale, so, they treated using a data cleaning method. The generated tag feature refers to one of the following classes which are Normal, inside network infiltration, HTTP denial of service, IRC Botnet DDoS and Brut force SSH classes.

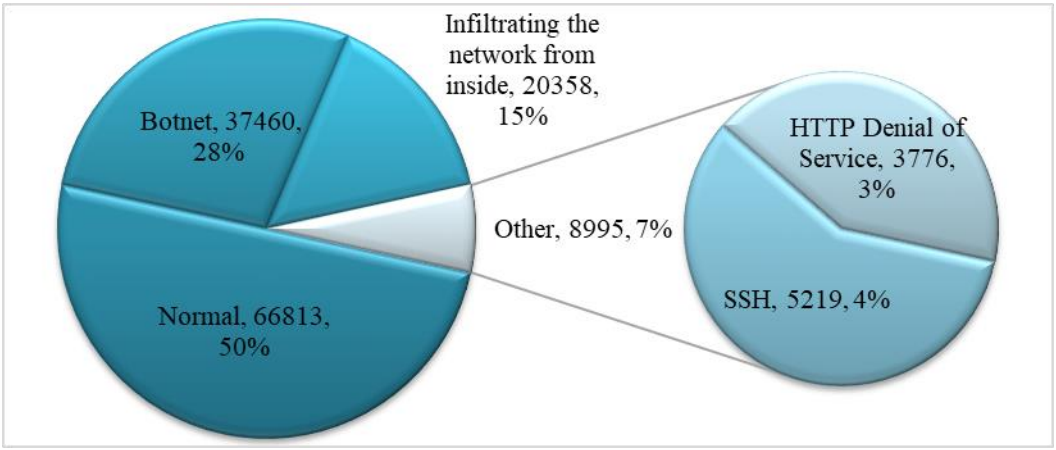

Figure.1. The records distribution of the UNB ISCX 2012dataset to build the experiments.

Table 1.The UNB ISCX 2012 features List.

Main features	Application Name	Total Source Bytes
	Total Destination Bytes	Total Destination Packets
	Total Source Packets	Direction
	Source TCP Flags Description	Destination TCP Flags Description
	Protocol Name	Source Port
	Destination Port	Tag
Accumulative and redundant features	Time Start	Time End
	sourcePayloadAsBase64	sourcePayloadAsUTF
	destinationPayloadAsBase64	destinationPayloadAsUTF
	dataroot_Id	

Knowing that, this dataset was collected in seven days, only four of them included attacks scenarios, one class per day. The record will classify by distinct the day that the attack appeared in table 2. So, all the attacks appeared on the day of the inside network filtration scenario and they were classified as attacks they will be classified as inside network filtration attacks and so on.

Table 2. The distribution of the records in the UNB ISCX 2012 dataset in the days which included attack scenarios

The days named by the attack scenarios	Attack	Normal
Infiltrating the network from inside	20358	255170
HTTP Denial of Service	3776	167604
Distributed Denial of Service using an IRC Botnet	37460	534238
Brute Force SSH	5219	392376
Sum of the records	66813	1349388

The distribution of the records in the days which included attack scenarios illustrates in table 2. It is clear that there are a sufficient number of records for each attack class, and there are a tremendous number of normal records. This phase started with converting the nominal or categorical data to sequential numeric values a data-type portability step. Then the imbalance scale of the features was addressed using two common methods [26] in data cleaning phase: Standardization: It is one of the common data transformation methods, it reproduces the data for each feature to have zero mean and unity variance, it is represented using the following equation:

$$z_i^j = \frac{x_i^j - \mu_j}{\sigma_j} \tag{1}$$

Where μ_j: is the mean of the feature j, σ_j is the standard deviation of the feature j and x_i^j is the j attribute of the i^{th} records. Min-Max Scaling method: It scales all attributes into $[0,1]$range and it is represented using the following equation:

$$y_j^i = \frac{x_j^i - \min_j}{\max_j - \min_j} \tag{2}$$

Where $\{\max_j, \min_j\}$represent the $\{\text{maximum, minimum}\}$value of the feature j and x_i^j is j attribute of the i^{th} records. The standardization method was selected to clean the imbalance scale of the feature.

3.3 Stratified Sampling

Stratified sampling is a statistical sampling method. It is an alternative to the known method called random sampling. It is used to generate new subsets of data that have the same sample fraction of their classes as in the main corpus. The following equation illustrates the sample fraction:

$$f_i = \frac{n_i^j}{N^j} \tag{3}$$

Where f_i is the fraction of the class i in the main set and any subset, N^j is the number of records in an arbitrary set j and n_i^j is the number of records belonging to the class i in the arbitrary set j. It guarantees that any generated subset will include records from all classes and the ratios of records of all classes in these subsets as they are in the main data-set, while the class-records selected each time randomly. It is clear that in the case where the minor classes present and the random sampling is used, some models will be built that do not learn anything about these classes. This was the reason for using this method.

3.4 Weighted Extreme Learning Machine

ELM is a feed-forward neural network, but it is not suffering from the time consuming and iterative process in the feed-forward back propagation neural network. This is addressed via a random selection of the weights and biases of the hidden layer, so it is a fast and simple method. A set of other features related to ELM still needs to be discussed. One of them, it is able to deploy different feature mapping and kernels. The other is the ability to build multiple class classification solutions easily in one model, without the need to combine multiple binary classes together. The method illustrated in [17] was used in this paper a WELM solution to address the unbalanced classes in ID problem, a single hidden layer feedforward neural network (SLFFN) was used; it is architecture is shown in figure 2. For any dataset $(\overrightarrow{X_i}, \overrightarrow{T_i})$, where $\overrightarrow{X_i} = x_1 + x_2 + \cdots .. + x_m$ is the feature matrix which includes N records called$i = 1,2 \ldots ., N$ and m features, while $\overrightarrow{T_i}$ is a target matrix. Like any neural network algorithms, both feature and target matrixes are numerical matrices which are obtained from the output of the preprocessing phase. The algorithm starts with ask user to determine the activation function and the number of the hidden neurons which is denoted by $g(x)$ and$L$ in sequential orders. Then the weight matrix W_{L*N} and the bias vector $B_{L*1}$of hidden neurons are generated randomly, this saves the time for this algorithm and makes this algorithm faster.

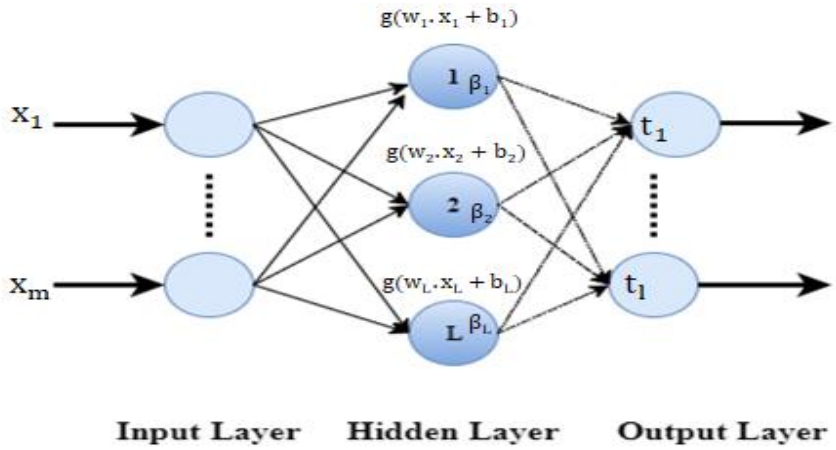

Figure. 2. Extreme Learning Machine Network

Different feature mapping can be used in ELM which represents the different activation functions can be used, an example of activation function can be used are:

- Sigmoid function

$$g(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

- Gaussian function

$$g(x) = e^{-x^2} \tag{5}$$

Then the output of hidden neurons H is computed using the following equation:

$$H = g(W_{L*N} \cdot X_{N*m} + B_{L*1}) \tag{6}$$

The output layer consists of l neurons, while l is the number of classes in the problem, and the weight matrix of the output layer donated by β_{l*L}. Now, solving the problem means finding the value of β which maximize the marginal distance and minimize the weighted and accumulative error, this represented the following equations:

$$\text{minimize} : \|H\beta - T\|^2 \text{ and } \|\beta\|. \tag{7}$$

The other form of the previous equation is:

$$\begin{aligned} &\text{minimize} : Lp_{ELM} = \frac{1}{2}\|\beta\|^2 + CW\frac{1}{2}\sum_{i=1}^{N}\|\varepsilon_i\|^2 \\ &\text{Subject to: } h(x_i)\beta = t_i^T - \varepsilon_i^T, \quad i = 1, \dots, N. \end{aligned} \tag{8}$$

W, itis a diagonal matrix with $N * N$ size, W_{ii} is the weight of the x_i record.ε_i , is the error of the sample x_i, which equal to the difference between the target value and the actual output. Reformulate the equations using Lagrange and based on Karush Kuhn Tucker (KKT) theorem, it is being:

$$L_{D_{ELM}} = \frac{1}{2}\|\beta\|^2 + CW\frac{1}{2}\sum_{i=1}^{N}\|\varepsilon_i\|^2 - \sum_{i=1}^{N}\alpha_i(\, h(x_i)\beta - t_i^T + \varepsilon_i^T) \tag{9}$$

Whereα_i is the Lagrange Multiplier which is a constant. In the next step, the partial derivative is performed based on β, α, and ε.

$$\begin{aligned} &\frac{\partial L_{D_{ELM}}}{\partial \beta} = 0, \rightarrow \beta = \sum_{i=1}^{N}\alpha_i\, h(x_i)^T = H^T\alpha \\ &\frac{\partial L_{D_{ELM}}}{\partial \varepsilon_i} = 0, \rightarrow \alpha_i = CW\varepsilon_i, \quad i = 1, \dots, N \\ &\frac{\partial L_{D_{ELM}}}{\partial \beta_i} = 0, h(x_i)\beta - t_i^T + \varepsilon_i^T = 0, \quad i = 1, \dots, N \end{aligned} \tag{10}$$

Two forms of equation produce the β, caused by solving equations, the first one has $N * N$ dimension, and the second has $L * L$ the dimension of the inverse matrix. The first one is better when the size of the dataset is small and it is able to reformulate in kernel form, while the other is better for huge datasets.

$$ForsmallN: \ \beta = H^T\left(\frac{1}{C} + WHH^T\right)^{-1} WT \tag{11}$$

$$ForBigN: \ \beta = \left(\frac{1}{C} + H^TWH\right)^{-1} H^TWT \tag{12}$$

Finally, the output for the complete network calculates using the following equation:

$$f(x) = fun(h(x)\beta) fun(h(x)\beta) = \begin{cases} sign, & inbinaryproblms \\ \arg\max(h(x)\beta)_i, & listhenumberofclasses \end{cases} \quad (13)$$

Due to the large sets represent the ID problems; the equation was used to solve this problem. We are concerned with finding the best L, C, activation function and weight scheme that will be used to build the ELM ID solution.

3.5 Accuracy Paradox and Cost-Function Scheme

The paradox of accuracy occurs frequently when most pattern recognition models were built using unbalanced classes, it is easier for the ML algorithms to classify either all or most the records of the small classes into one or more of the major classes, this happens with the negligible effect of the total accuracy. But the problem gets worse when these minor classes are crucial in the environment. The cost function is one of the methods were suggested to address this problem [17], [18] it affects the learning process by giving different weights to the records that belong to different classes. Different cost function methods were used. First scheme, the default weight scheme where all classes have the same weight value which equals to 1. Second scheme [17], it depends on the ratio between the numbers of records in the corpus to the number of records for each class, the following equation used to calculate the weights for each class:

$$W_i = \frac{N}{n_i} \quad (14)$$

We used W_i to represent the weight for all records which belong to $i^{th}class$, N to represent the number of records in the corpus and n_i to represent the number of records belong to a class $i$the in the corpus for all equation in this sub-section.

The third scheme [16], it is used the golden ratio $0.618/1$ multiplied with the inverse of the number of the records belongs to each class; it is illustrated by the following equation:

$$W_i = \begin{cases} \frac{0.618}{n_i} if n_i > AVG(n_i) \\ \frac{1}{n_i} if n_i \leq AVG(n_i) \end{cases} \quad (15)$$

Due to the convergence issues of the deployed algorithms, the first method was used with the WSVM, while the third one was used with WELM, which performed well to address the imbalanced class of ID problem.

3.6 General Method Procedure

The general procedure that was used in performing all experiments of WELM with on different Kernels, activation functionon UNB ISCX2012 dataset is shown in the pseudo code(Algorithm) below in detail illustrated the general structure of the proposed model.

Algorithm: The general procedure that was used in building the ID model

Input: Dataset **UNB ISCX2012**, cost-function, number of hidden layer neuron, other parameters.

Output: P number of models, Result Object;
Data Preprocessing: // *Converting nominal fields into numerical*
for a field **in** the dataset, **do**
if is_nominal (field**) then**
uniqueList ← **uniqueElements**(field) ;
sortedList ← **sort**(uniqueList) ;
for k =1 **to size(**field**), do**
index ← **compare**&$findSortedlistIndex$(sortedList, field[variable]); /* *Find the index of unique element that have the same value of the* k^{th} *element of field column**/
numfield[variable] ← index ;
dataset = **replace**(dataset, numfield, field); /* replace the old field with the new numeric field*/
end if; // *Applying the standardization method on the dataset fields*
for a field **in** the dataset, **do**
for variable = **1 to size(**field**), do**

$$\text{field}[\text{variable}] \leftarrow \frac{\text{field}[\text{variable}] - \text{mean}(\text{field})}{\text{standerd deviation}(\text{field})}$$

// *Partitioning the dataset into P sub-sets* and // *Calculate the fraction of each class i*

$$f_i \leftarrow \frac{\text{number of records}_i}{\text{Size}(\text{dataset})}$$

DatasetList ← **StraifiedSampling**(dataset, N, f)
// *Generating the weight array which elements represent a weight for distinct class i*
if cost-function == default **then**
W = ones(num_Classes)
else
for $class_i$**to** num_Classes, **do**
If cost-function == second **then**

$$w_i \leftarrow \frac{\text{Size}(\text{dataset})}{\text{number of records}_i}$$

elseIf cost-function == third **then**
if Size($class_i$) ≤ **Size**(dataset)/num_Classes

$$w_i \leftarrow \frac{1}{\text{number of records}_i}$$

else

$$w_i \leftarrow \frac{0.618}{\text{number of records}_i}$$

end if;
end if;
end if;
Main:

$$\gamma = \frac{1}{\text{NumofFeature}(\text{Dataset})}$$

for i = 1 **to** P **do**
Train ← dataset − {i^{th}partiton}
Test ← dataset[i^{th}partition]
model ==**WELM then**
$model_i$ ← **BuildWELMModel**(Train, g(.), L, C, default γ, W)
testing phase results = **Test WELM** ($model_i$, Test, i^{th}partitionLabels)
end if;
ResultObject = **compute All Metrics**() /*Compute the overall accuracy and the accuracy for each class for the training data*/
return P number of models, Result.

4. EXPERIMENTS AND RESULTS

The experiments were performed on the UNB ISCX2012 dataset. The WELM algorithm is used to perform multiple class classification experiments on the dataset. Before starting in the issues of the algorithm, it should be determined the best data normalization method which should be used later in experiments. This selection depends on the properties of the data in the dataset, the existence of outlier records is the crucial properties of the ID datasets. The MATLAB version of the proposed WELM algorithm in [17] was used to perform parts of our experiments. The performance of the WELM algorithm depends on the selection of the various parameters of this algorithm; these parameters are the number of hidden layer neurons (L), the activation function of the hidden layer neurons and the regularization parameter$C$. Two activation functions were used, they are sigmoidal and Gaussian activation function.

The activation function and weight scheme that will be used to build the ELM ID solution. All combinations of the proposed model parameters are listed in table 3. The cross-validation method called leave-one-out [7] was mainly used to build, evaluate and validate all models. Based on the leave-one-out method the experiments repeated n times, for each round i the data will divide to n partitions, the i^{th} partition is used for testing phase while the $(n-1)$ other partitions are used for building the model. Twofold cross-validation is more generalized than tenfold. In twofold cross-validation, both training and testing phases are performed on distinct 50 percent of the dataset records, while in tenfold cross-validation the training phase is performed on 90 percent of the dataset records and the remaining 10 percent of dataset records are used to perform the testing phase. So, two-fold cross-validation was mainly used to perform the experiments. The stratified sampling method was used to maintain the same ratio of a number of records for any class to the number of records for all classes in all partitions and in the complete set. The recent paper which is referenced by [25] was used to make an evaluation with our models that were built on UNB ISCX2012 dataset. It assumed that it used 1 percent from each attack class records randomly to build the models, and 10 percent to test the built models. There is an inconsistency between the number of records as [25] mentioned and the correct number of records. So, the same number of records was used to build our primary experiments on this dataset. The number of records was chosen in this way to make a consistent and equitable assessment. The overall accuracy and F-score evaluation metrics were used with parameter optimization phase, while the confusion matrix, recall, precision, F-score, FP, miss-detection, miss-classification in addition to the overall accuracy were used to measure the performance of the proposed models. Several metrics and definitions [24] are used in evaluating the multi-class pattern recognition models. Some of these are Confusion matrix, True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), Accuracy, Precision, Recall, Detecting Rate, Misclassification, G-mean, F-measuring and Receiver Operating Characteristics curve (ROC).

Table 3. Various combinations of parameters were used to build a paper model.

Algorithm	Kernels, Activation Function	C	Weight	The optimized Parameters, Number of Hidden Neurons
WELM	Sigmoidal Gaussian	$\{1, 10, 50, 10^2, 300, 500, 10^2, 10^3, 10^4, 10^5, 10^6$	$\{Defaultscheme: w_i = 1 \forall i$ $Thirdscheme: w_i = \{0.618/n_i, if n_i > avg(N)$ $1/n_i, \quad if n_i \leq avg(N)$	$\{500, 700, 10^3, 1500, 2000$

It is important to clarify the concept of each term, these concepts oriented toward the ID problem, that is included in the following paragraphs.

TP: the records are predicted to the correct type of attacks.
FP: the records are predicted as attacks while they are normal.
TN: the normal records which are classified correctly.
FN: the records predicted as normal while they are attacks.
Misclassification: the hostile records are predicted to the wrong type of attacks.
FAR: the rate of normal records which classified as attacks.
Confusion matrix: It is one of the common methods used to view the result of the pattern recognition models; it represents a two-dimensional square matric. The fields of both dimensions are the classes of the problem, and the values of the cells represent the distribution of the predicted records on the target classes.
Accuracy: It is one of the main metrics used to measure the overall performance of the pattern recognition models; it is represented by the following formula:

$$\text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \sum \text{FP} + \text{TN} + \sum \text{FN} + \sum \text{MissCl}_{-,-}} \tag{16}$$

Precision: It is the percentage of records which are predicted to certain class correctly to all records predicted in that class. It is calculated using the following equation:

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{MissCl}_{(-,i)}} \tag{17}$$

Recall or Sensitivity: It is the percentage of the correctly predicted records of one of the attack classes to the number of records belonging to that class in the target table.

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i + \text{MissCl}_{(i,-)}} \quad (18)$$

Specificity: It is the percentage of normal records which are predicted as normal to the number of records belonging to the normal class in the target table.

$$\text{Specificity} = \frac{TN}{TN + \sum FN} \quad (19)$$

G-mean: It is an overall metric which measures a geometric mean of specificity for normal class and the sensitivity for all hostile classes, it is used to measure the performance in cases where the imbalanced classes exist. The following formula is used to measure the G-mean metric:

$$\text{G-mean} = \left(\prod_{1}^{m-1} \text{sensitivity} * \text{specificity} \right)^{\frac{1}{m}} \quad (20)$$

F-measuring or F-score: It is the harmonic mean of precision and recall and it is calculated using the following equation:

$$\text{F-measuring} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (21)$$

4.1 UNB ISCX2012 Dataset Experiments

Evaluation inconsistency is one of the important issues related to ID solutions, which is one of the points that have been taken into account in this work. The dataset was collected in seven days; three of them had normal records only while the other four days had a distinct attack scenario for each day as illustrated before. The number of normal and attack records in mentioned four days is shown in table 2. A newly published in [25] used mainly to evaluate our work on this dataset. Table 4 shows the number of records that were used in the experiments of [25]. The author used 11 percent of the more frequent classes (1 percent for training and 10 percent for testing phase) and all records for the low frequent attacks which are Botnet and DoS. As it is shown in table 2, the number of records for the Botnet and Dos attacks are (37460, 3776 records in sequential order), which shows inconsistency is the number of records between the dataset and the mentioned number of records in [25]. To overcome this problem, two sets of experiments were performed. The primary experiments had the same numbers of records as in [25] for each class, they showed in table 4, the number of records of UNB ISCX2012 dataset as they included in [24]. This enables us to make a consistent and fair comparison. The secondary experiments were performed using complete attacks records, in addition to 65000 normal records that randomly selected from the normal records of the day that included the scenarios of botnet attack, the number of normal records selected to be equal to the number of the attacks records; this based on the fact that most network traffic is normal. The secondary experiments were built on the general method of this work. To make the results of the experiments based on randomly selected records

more representative, each experiment was repeated ten times but using different sets of records each time then the average results were calculated.

Table 4: Number of records of UNB ISCX2012 dataset as they are included in [24]

Class Name	# of Train records	# of Test records
Infiltrating the network from inside	60	605
HTTP Denial of Service	4	36
Distributed DoS using an IRC Botnet	3	2
Brute Force SSH	46	463
Normal	1227	12285
Sum of the records	1340	13391

4.2 Discussion of the Results

WELM algorithm was applied to the UNB ISCX2012 dataset. In the primary experiments, different weight schemes were employed and different parameters related to these algorithms were optimized, all these experiments appeared that both the WELM algorithms with default weight scheme had better performance with respect to the overall accuracy and the F-score for all classes. Although the second weight scheme with WELM algorithm failed to improve the overall accuracy and the F-score for all classes, they succeeded in improving the recall for the vast majority of cases, but this was associated with precision reduction. Table 5 shows the results of the WELM Gaussian RB and Sigmoidal Functions for all classes of the dataset with the optimized parameters and result

Table 5: The results of the WELM Gaussian RB and Sigmoidal Functions on the UNB ISCX2012 during 2 tests

Test #	Kernels, Activation Function	C	W	optim Parmt. # of Hidden Neurons	The overall accuracy	F-score				
						Normal	L2L	DoS	Botnet	SSH
Test 1	Sigmoidal	1	Def t.	500	99.3%	99.7%	94.4%	39.3%	0.0%	98.9%
	Gaussian RB	1		500	99.1%	99.6%	93.1%	35.3%	0.0%	98.5%
Test 2	Sigmoidal	10^4	3^{rd}	2000	96.2%	98.9%	63.8%	15.2%	1.8%	98.9%
	Gaussian RB	500		2000	97.6%	99.1%	81.1%	23.5%	1.2%	98.9%

Figure 3 shows the accuracy for the 5 classes of the UNB ISCX2012 dataset in two tests applying WELM with Sigmoidal function and Gaussian RB function. It is obvious that the obtained accuracy results in Test 1 for the two applied Kernels Activation Function in the 5 classes have approximately the same result. On the other hand, as shown in figure 4 in Test 2 the obtained

accuracy result shows that the WELM with Gaussian RB function outperforms sigmoidal function in some classes of the dataset.

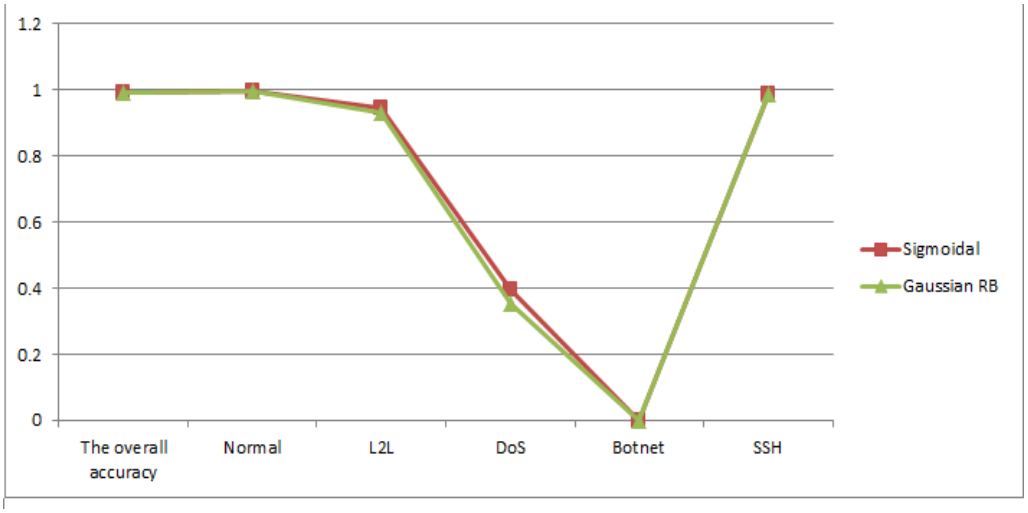

Figure. 3. The WELM Sigmoidal and Gaussian RB result in Test 1

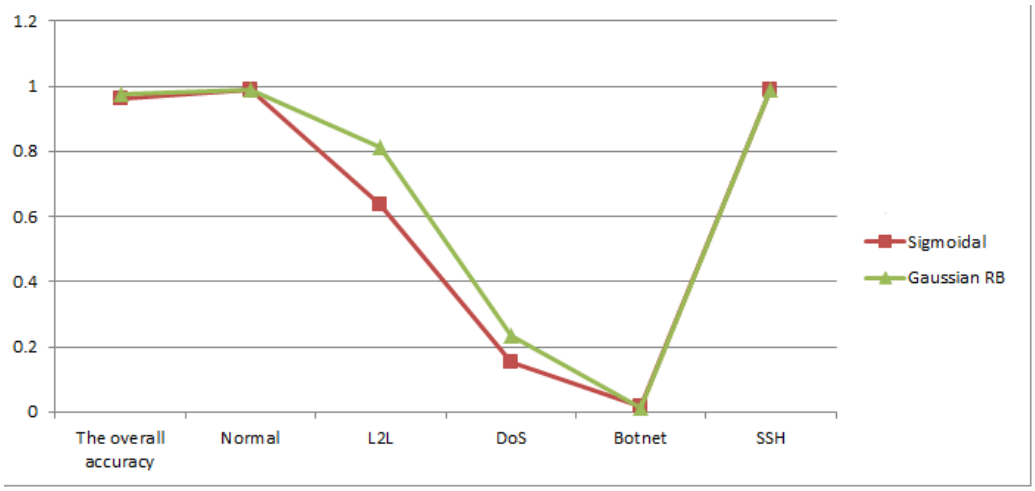

Figure. 4. The WELM Sigmoidal and Gaussian RB result in Test 2

The recently published work [25] which intersected with our work in concern was used to make a consistent and equitable evaluation. The results of applying the SVM algorithm on the UNB ISCX2012 dataset in the [25]. As shown in Table5, the polynomial kernel performs better than Gaussian RBF kernel with SVM; the results represented the value of applying the experiments on a random subset of data which was insufficient. The optimization step decreased the performance of the SVM algorithm with Gaussian RBF, which did not reflect the true behavior of the algorithm. On the other hand, the primary experiments on the UNB ISCX2012 dataset in our paper were repeated ten times for each scenario, and then the average of those rounds was used to evaluate the applied models.

The result of the WELM model outperforms the SVM model proposed in [25], when we applied the same activation function in the accuracy of Normal, SSH and DoS classes, as shown in table 6.

Table6: The results of the WELM with Polynomial Function on the UNB ISCX2012 Dataset compared with [25]

	WELM with Polynomial Function			**SVM with Polynomial Function [25]**		
	Accuracy	99.10%		**Accuracy**	99.11%	
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
SSH	**98.61%**	98.51%	**98.55%**	**95.9%**	100%	**97.9%**
Botnet	0.00%	0.00%	0.00%	18.2%	100%	30.8%
DoS	36.50%	**37.78%**	**35.33%**	60%	**25%**	**35.3%**
L2L	94.47%	91.98%	93.18%	95%	94.9%	95%
Normal	**99.63%**	**99.67%**	**99.65%**	**99.6%**	**99.5%**	**99.5 %**

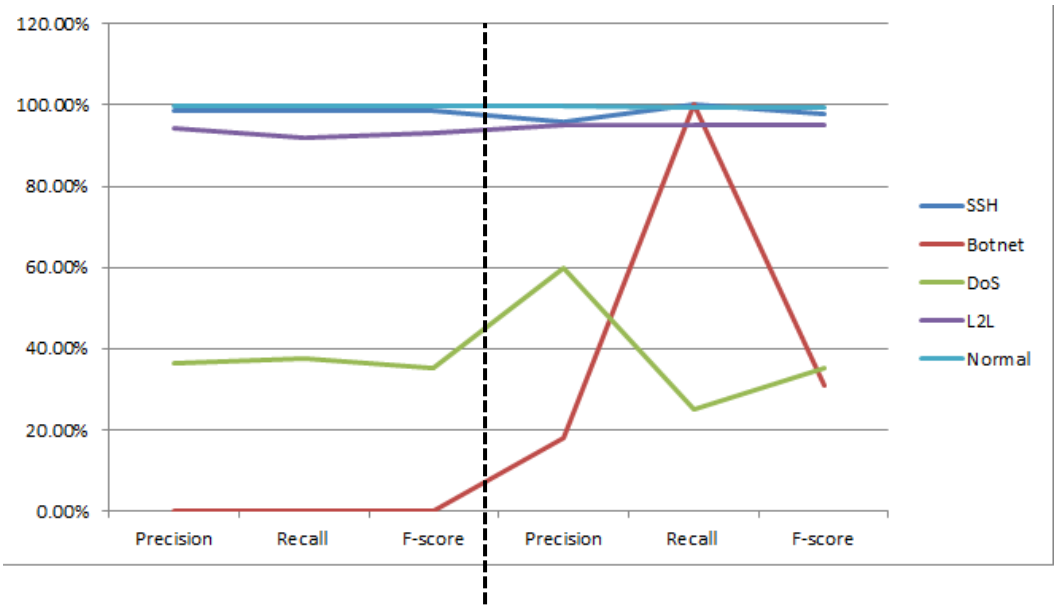

Figure. 5. The result comparison of WELM with Polynomial Function on the UNB ISCX2012 And the model presented in [25]

Based on the foregoing, the WELM model reaps the superior results in the minor classes and competitive results in overall accuracy and the accuracy of the major classes. It increases the ability to detect the most hazardous attacks.

5. CONCLUSION AND FUTURE WORKS

The development of IDS in computer networks is a challenge for researchers because, with the growth of computer networks, new attacks appear constantly. IDS is a vital security tool. The daily increase in the number of attacks encourages the development of the IDS. In this paper, a

method was proposed for detecting the intrusions by using machine learning (ML) tools that consolidated stratified sampling and different cost function schemes with Extreme Learning Machine (ELM) method to build competitive ID solutions that improve the performance of these systems and deal with classes in the training set that contains many more samples than others in the same training set. The proposed method got a superior result than previous works in the accuracy paradox issue while preserved the accuracy improvement. In this way, the performance of ID capable of maintaining better levels of accuracy as well as improving the detection of the most dangerous classes. The WELM algorithm is a good competitor. The experiments that performed achieved competitive results of both overall accuracy and F-score per-class performance scale onthe UNB ISCX2012 dataset. The accuracy in this experiment SSH, DoS, Normal classes in outperform the SVM method. The truth associated with this problem is that none of the open issues have been solved completely and all points still opened although we covered some of ID the points through this effort. In future work, we will start using a set of one-class classification methods which can be used in different manners. It is suggested to solve the unbalanced class problem, to build novelty models and outlier detection, models. While the first way pours into solving the imbalanced classes, the others contribute to building anomaly models which may improve the detection of zero-day attacks.

REFERENCES

[1] Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials, 21(1), 686-728.

[2] Cashell, B., Jackson, W. D., Jickling, M., &Webel, B. (2004). The economic impact of cyber-attacks. Congressional Research Service Documents, CRS RL32331 (Washington DC).

[3] Bellovin, S. M. (2004, December). A look back at" security problems in the tcp/ip protocol suite. In 20th Annual Computer Security Applications Conference (pp. 229-249). IEEE.

[4] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security, 28(1-2), 18-28.

[5] Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access, 6, 33789-33795.

[6] Moustafa, N., Turnbull, B., & Choo, K. K. R. (2018). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet of Things Journal.

[7] Idhammad, M., Afdel, K., & Belouch, M. (2018). Semi-supervised machine learning approach for DDoS detection. Applied Intelligence, 48(10), 3193-3208.

[8] A.-C. Enache and V. V. Patriciu, "Intrusions detection based on support vector machine optimized with swarm intelligence," in Applied Computational Intelligence and Informatics (SACI), 2014 IEEE 9th International Symposium on, 2014.

[9] Ji, S. Y., Jeong, B. K., Choi, S., &Jeong, D. H. (2016). A multi-level intrusion detection method for abnormal network behaviors. Journal of Network and Computer Applications, 62, 9-17.

[10] Alabdallah, A., & Awad, M. (2018). Using weighted Support Vector Machine to address the imbalanced classes problem of Intrusion Detection System. KSII Transactions on Internet & Information Systems, 12(10).

[11] Bains, J. K., Kaki, K. K., & Sharma, K. (2013). Intrusion Detection System with Multi-Layer using Bayesian Networks. International Journal of Computer Applications, 67(5).

[12] Sharma, S., Gigras, Y., Chhikara, R., & Dhull, A. (2019). Analysis of NSL KDD Dataset Using Classification Algorithms for Intrusion Detection System. Recent Patents on Engineering, 13(2), 142-147

[13] M. H. Bhuyan, D. K. Bhattacharyya and J. K. Kalita, "Towards Generating Real-life Datasets for Network Intrusion Detection.," IJ Network Security, vol. 17, pp. 683-701, 2015.

[14] A. Shiravi, H. Shiravi, M. Tavallaee,and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Computers & Security, vol. 31, pp. 357-374, 2012.

[15] M. Tavallaee, E. Bagheri, W. Lu,and A.-A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009, 2009.

[16] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security, 28(1-2), 18-28.

[17] Buczak, A. L., &Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.

[18] An, X., Zhou, X., Lü, X., Lin, F., & Yang, L. (2018). Sample selected extreme learning machine based intrusion detection in fog computing and MEC. Wireless Communications and Mobile Computing, 2018.

[19] Aljawarneh, S., Aldwairi, M., &Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science, 25, 152-160.

[20] S. Anu and K. P. M. Kumar, "Hybrid Network Intrusion Detection for DoS Attacks," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 3, 2016.

[21] P. Laskov, P. Düssel, C. Schäfer and K. Rieck, "Learning intrusion detection: supervised or unsupervised?," in International Conference on Image Analysis and Processing, 2005.

[22] Idhammad, M., Afdel, K., &Belouch, M. (2018). Semi-supervised machine learning approach for DDoS detection. Applied Intelligence, 48(10), 3193-3208.

[23] Mighan, S. N., &Kahani, M. (2018, May). Deep Learning Based Latent Feature Extraction for Intrusion Detection. In Electrical Engineering (ICEE), Iranian Conference on (pp. 1511-1516). IEEE.

[24] J. M. Fossaceca, T. A. Mazzuchi, and S. Sarkani, "MARK-ELM: Application of a novel Multiple Kernel Learning framework for improving the robustness of Network Intrusion Detection," Expert Systems with Applications, vol. 42, pp. 4062-4080, 2015.

[25] E. Nyakundi, "Using support vector machines in anomaly intrusion detection," University of Guelph, Guelph, Ontario, Canada, 2015.

[26] Injadat, M., Salo, F., Nassif, A. B., Essex, A., &Shami, A. (2018, December). Bayesian Optimization with Machine Learning Algorithms Towards Anomaly Detection. In 2018 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

[27] Al Najada, H., Mahgoub, I., & Mohammed, I. (2018, November). Cyber Intrusion Prediction and Taxonomy System Using Deep Learning and Distributed Big Data Processing. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 631-638). IEEE.

[28] C. C. Aggarwal, Data mining: the textbook, Springer, 2015.

AUTHORS

Mohammed Awad received the B.S. Degree in Automation Engineering from Palestine Polytechnic University in the year 2000, master & Ph.D. degrees in Computer Engineering from the Granada University Spain (both are Scholarship from Spanish Government). From 2005 to 2006, he was a contract Researcher at Granada University in the research group Computer Engineering: Perspectives and Applications. Since Feb. 2006, he has been an Assistant Professor in the Computer Engineering Department, College of Engineering and Information Technology at Arab American University. At 2010 he has been an Associate Professor in Computer Engineering. At 2016 he has been a Full Professor in Computer Engineering. He worked for more than 12 years at the Arab American University in academic Position, in parallel with various Academic administrative positions (Departments Chairman and Dean Assistant, Dean of Scientific Research and Editor-In-Chief, Journal of AAUJ). Through the research and educational experience, he has developed a strong research record. His research interests include Artificial Intelligence, Neural Networks, Function Approximation of Structure and Complex Systems, Clustering, Algorithms, Optimization Algorithms, and Time series Prediction. He won a number of awards and research grants.

Alaeddin Alabdallah received the B.S. Degree in Computer Engineering from An-Najah National University in 2006, and a Master Degree in computer science at Arab American University in February 2018. From 2006 till now, he is Teacher and Research Assistance at the Computer Engineering Department at An-Najah National University. His research interests include Artificial Intelligence, computer networks, and Information Security.