Using weighted Support Vector Machine to address the imbalanced classes problem of Intrusion Detection System

Alaeddin Alabdallah[1], Mohammed Awad[2]
[1]Computer Engineering Dept., Faculty of E&IT, An-Najah National University
Nablus, P.O. Box: 7, Palestine.
[e-mail: eng_alaeddeen@najah.edu]
[2]Computer Systems Engineering Dept., Faculty of E&IT, Arab American University
Jenin, P.O. Box: 240, Palestine
[e-mail: mohammed.awad@aauj.edu]
*Corresponding author: Mohammed Awad.

Received March 18, 2017; revised December 9, 2018; accepted April 16, 2018;
published October 31, 2018

Abstract

Improving the intrusion detection system (IDS) is a pressing need for cyber security world. With the growth of computer networks, there are constantly daily new attacks. Machine Learning (ML) is one of the most important fields which have great contribution to address the intrusion detection issues. One of these issues relates to the imbalance of the diverse classes of network traffic. Accuracy paradox is a result of training ML algorithm with imbalanced classes. Most of the previous efforts concern improving the overall accuracy of these models which is truly important. However, even they improved the total accuracy of the system; it fell in the accuracy paradox. The seriousness of the threat caused by the minor classes and the pitfalls of the previous efforts to address this issue is the motive for this work**.** In this paper, we consolidated stratified sampling, cost function and weighted Support Vector Machine (WSVM) method to address the accuracy paradox of ID problem. This model achieved good results of total accuracy and superior results in the small classes like the User-To-Remote and Remote-To-Local attacks using the improved version of the benchmark dataset KDDCup99 which is called NSL-KDD.

Keywords: Intrusion detection system, Weighted Support Vector Machine, Stratified sampling, Cost function, NSL-KDD

http://doi.org/10.3837/tiis.2018.10.027 **ISSN : 1976-7277**

1. Introduction

With the growth of computer networks and the increase of the services offered, the need to maintain the reliability, integrity, and availability of the computing systems is increased, this makes the security of these systems gain more importance. On the other hand, the attackers increased the directed attacks to these systems which become a serious problem [1]. The cyber-attacks operations capable of causing significant economic damage to both public and private companies and organizations, thus attacking 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 in cyberspace is a multi-disciplinary problem. One side of the problem is a cyber security problem. And the other side is the statistical and the artificial intelligent learning fields represent the factories that produce the pool of solutions. The security problem becomes more complicated because of the high connectivity of the world via the Internet. Deep looking for the communication, computer network systems, protocols, and services which represent the backbone of the Internet shows the wide distributions of the flaws for most computing components. These flaws represent the reason of previous, now and future attacks. Part of this fact presented in [3] which included a list of security flaws in the TCP stack protocols.

There are set of open points in the network intrusion detection. The first point is the huge volume of generated network traffic which will increase more and more because of the expansion of internet services, increasing the mobile devices and the movement toward internet of things technology [4]. This opens the scalability issue for machine learning community.

Before we continue to address other points, it is necessary to discuss the three common methodological categories of Intrusion detection system [5]. The first category called misused or signature-based IDS, in this approach different known rules or patterns - either it was normal or abnormal-capture in training phase from labeled data, then the generated models are used to make a prediction for unseen data. Although these models get high accuracy for detecting known and some variant of know attacks, they fail in detecting zero-day attacks. The second category is anomaly-based IDS, it is based on the closed world assumption [6], which assumes there are capabilities to capture the complete normal behaviors in the training phase, and then the 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 approach success in detection the zero-day attacks, but with total accuracy less than the preceding one. The Third one is the hybrid approach which combines both previous approaches in one model.

Now, and after explaining some facts about the ID problem, it is a good time to state the second point of our problem; it is the labor extensive efforts of experts needed to label the traffic correctly. The related open point here is the need to benefit from the huge size of unlabeled records. The third point is the inability to aggregate a set that includes all variant of normal traffic, while the zero-day or newly attacks are renewable, which could be summarized with the impossibility to have the close world in our domain. The question is if the incremental learning of the unknown normal behavior by the ML algorithm can address the dilemma of the impractically of existence the closed world in our domain.

In this work, we used NSL-KDD public benchmark dataset [7]; it is an improved version of the KDDCup99 dataset, which is the most used benchmark dataset in this field. It includes labeled records from five classes which are Normal, DoS, Probing, R2L, and U2R categories. Even there is a gap between the nature of traffic aggregated in this dataset and the contemporary real traffic, the points we are focusing on to address are existing in our set on par with the real traffic. For this work scope, this dataset is sufficient leaving behind the former open points to address in other locations.

The fourth point which is multifaceted and is the point we focus on, it is the sensitivity of the intrusion detection system toward the errors. Most works concerned with [8] [9] increasing the detection rate and decreasing the false alarm rate in order to improve their system accuracy. Even the error rate 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. Most studies [10] either fail to predict or predict with an insufficient accuracy of the minor classes as U2R and R2L*;* even they gain high in the overall accuracy, this phenomenon called accuracy paradox [11]. The Interesting point is that the different categories of attacks belong to the different level of security while the minors or U2R and R2L are the most serious [12].

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 that developed using anomaly-based methods to deploy in operational real-world environments [6]. The awareness of the pressing needs to improve powerful and dynamics security tools that protect the contemporary computing system emphasis the great interest from researchers of both communities to improve the Intrusion detection systems.

To address the accuracy paradox of imbalance classes and to develop intrusion detection model that takes care in the serious classes of attacks, viz., the minor classes U2R and R2L in NSL-KDD dataset, we proposed weighted Support Vector Machine algorithm WSVM with a striated sampling of data. The remaining of this work organizes as the following: section 2, presents an overview of the related works. Section 3 presents in detail the methodology used. Then, section 4 shows some experiments that confirm the performance of the proposed model with evaluation and discussion. Finally, some conclusion and future works will be presented in section 5.

2. Related Works

Intrusion detection problem has a great interest from the researchers; part of these efforts concentrated on review the problem from different point of view, one of the recent surveys [13] studied four categories of anomaly ID methods, they are classifications, statistical, information theory, and clustering. It found that classifications and clustering are outperformed in detecting DoS attacks, while statistical technique outperforms in U2R and R2L attacks. It focused on the lack of the public datasets that used in network intrusion system.

Machine learning community suggests many tricks to solve the deficiency of its models in predicting the small classes like U2R and R2L classes of ID problem, it is the problem mentioned in the previous review. Different approaches suggested [14, 15] to solve the imbalanced classes like resampling techniques and algorithmic approach. The oversampling and undersampling are the common two resampling methods in literature while the cost function was added to different machine learning algorithms to address it’s sensitivity to imbalanced classes. Different cost functions suggested in both works and applied with Neural Networks [15] and Extreme machine learning (ELM) [14] algorithms. This improves the prediction for the small classes in all used data set. One of these cost functions was deployed in this work with WSVM algorithm which gained good result in the minor classes such as U2R and R2L.

Several Network Intrusion detection models proposed and tested in the last decade, these models built based on the KDDCup99 or one of an improved set from KDDCup99 like NSL-KDD. Most of these efforts concentrated on either making normal abnormal record prediction or multi-class classification prediction. On the other hand, a few efforts tried to build sub-models like in [16], the proposed model distinguished between the Scanning networks threats and normal traffic based on selected records of NSL-KDD. It used PCA as statistical feature reduction method and Multilayer Perceptron Neural Network (MLPNN) as a binary class classification model. The Authors in [17] proposed a hybrid model for detecting different classes of DoS attacks. In this model, 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 [6] in order to reduce the false alarm rate when building the ML models. Our work aimed to solve the complete problem which makes these works out of the scope.

To compare the performance of the supervised or unsupervised ML models as intrusion detection solutions, the authors in [18] built a framework and made a list of experiments. They demonstrate that supervising learning model do better if the test data contain known or variant of known attacks. While both have close performance in dataset contains unknown attacks. The suggested semi-supervised learning as promise solution. With the same hypothesis, the authors in [19] proposed a semi-supervised model based on NLS-KDD dataset as an enhanced version of the KDDCUP'99 dataset. The main goal of this efforts is to evade from the heavily and extensive works need from experts to correctly label the complete traffic while they preserve the good performance that caused by using the sufficient amount of Label data.

The authors in [20] built multi-level hybrid classification model based on an improved set include non-redundant 10% KDDCup99 subset. It combined between Extreme Machine Learning (EML) and SVM algorithm in order to improve the accuracy of the model and reduce the execution time. In order to decrease the execution time, it deployed the ELM algorithm and it sampled the data using modified version of K-means clustering algorithm to get the best representative data. Although it made some improvement of total prediction accuracy which was 95.75%, it occurred in the accuracy paradox which clearly shown in the bad prediction accuracy result achieved for the minor classes U2R and R2L which was 21.9% and 31.39% sequentially.

Another kind of hybrid models was deployed in literature for our problem, but at this time, it was combined multiple kernels together [21]. Multiple Adaptive Reduced Kernel Extreme Machine Learning model (MARK-ELM) was proposed. This work proposed a framework which used 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 confirm the existence of accuracy paradox problem in these experiments.

Another multi-level intrusion detection model was proposed in [10]. 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 CART algorithm. The second phase included building three predictive model using SVM, Naïve Bases, and NNs in order to determine the exact attacks categories for only three of the attacks, 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 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 [6] about the clearance of the interpretation of the result at evaluation step of our problem.

A lot of machine learning 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. We deployed one of the cost functions with weighted SVM to address the accuracy paradox of intrusion detection problem, it was improved the detection of the rare but serious attacks and it was preserved the overall accuracy of the system.

3. Methodology

In this section, we will cover set of points. The first point includes the description of the dataset which was used to address some points of the intrusion detection problem and the reason for this choice. The other points represent the main steps of the data mining process [22]. These steps started with converting the categorical features to numeric features as data type portability then the data was cleaned via standardization method. Both previous steps represent the data preprocessing phase. The second phase is the analytical component. The WSVM combined with cost function was used to develop multi-class classification models for the distinct classes of network data.

3.1. Dataset Selection and Description

KDDCUP99 is the main benchmark data set which still used for a lot of recent researches [21] [17] [20], although the gap between the characteristics of the contemporary traffic and the records were included in this data set. KDDCUP99 dataset suffers from a large number of redundant records in both training and testing sets. They are the cause of unwanted biasing in both training and evaluation processes. To overcome these points NSL-KDD dataset was generated as an improved version of it is origin, which includes only distinct records. NSL-KDD was used to perform our experiments. The aim is to solve the accuracy paradox, viz., to improve the detection rate of the minor and serious attacks. The minor attacks exist in the selected dataset as same exist in the contemporary real-traffic, that is obvious in **Fig. 1**. So, it is sufficient at this scope.

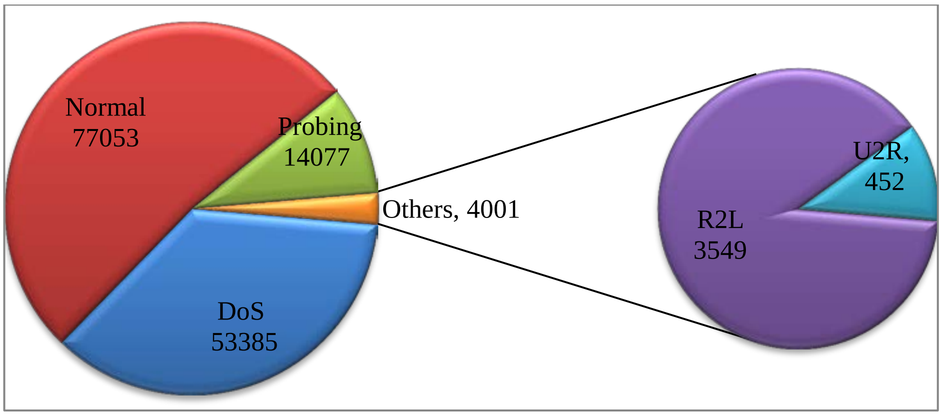

Fig. 1. A chart illustrates the number of records for each category in the data set.

NSL-KDD data set [23] consists of five categories. The first category is normal traffic. The others are abnormal traffic which falling in the following categories:

1. Denial of service attacks: it is the most frequent attacks, which based on generation huge amounts of offensive traffics in order to saturate the targeted computing components; this would be lost the rights of legitimate users.
2. Probing attacks: it represents the first step of any adversary behaviors. At this process, the efforts concentrate in gathering information about a different component of the targeted cyberspace.
3. Remote to local attacks (R2L) is made to get illegal root privilege in the targeted component.
4. User to remote attacks (U2R) is a remotely accessing of the target by a penetrative local account via internal flaws like operating system flaws.

NSL-KDD consist of 41 features falling into three groups Basic features were extracted from TCP/IP protocols records, traffic features or time base features and content features which important for detect R2L and U2R attacks like login status.

The complete NSL-KDD records are included in the following files:

1. KDD-Train+: This file includes records suggested as a training set.
2. KDD-Test+: This file includes records suggested as a testing set.

To be precise, and for evaluation and comparison reasons, the records of both files are used as one set then we used the cross-validation method to evaluate the proposed models. **Fig. 1** illustrates the number of records for each category in the complete set of data.

3.2. Data Preprocessing

Data preprocessing performed in many steps that depend on the nature of the data [22]. Our dataset consists of 41 features that fall into three types which are Nominal, Numerical and Binary. It is observed that it does not have missing data, the numeric features do not follow balanced scale and the data was labeled into five classes which are Normal, DoS, Probing, U2R, and R2L.

This phase started with converting the nominal or categorical data to sequential numeric values. Then the problem of imbalance scale of features is addressed using two common methods [22]:

1. Standardization: It is one of the common data transformation methods, it reproduces the data for each feature to have zero mean and unit 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 j attribute of the i^{th} records.

2. 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 $\{maximum,\ minimuim\}$value of the feature j and x_i^j is j attribute of the i^{th} records.

The result of the preliminary experiments appeared that standardization method does better in our case; this is shown in the **Fig. 2** which represents the gained accuracy for two WSVM models, one of them was preprocessed using Min-Max normalization method while the other was preprocessed using the standardization method. The overall accuracy and the accuracy for each separated class were calculated for both models. The experiments performed using 50% of NSL-KDD records that selected using stratified sample method which will describe later, then the model tested using remainder records. This result expected in any environment includes outlier records, so standardization method was used in later experiments.

3.3. Stratified Sampling

Stratified sampling is a statistical sampling method [24]. It is an alternative to known methods called random sampling. It is used to generate new subsets of data that have the same sample fraction [25] of their classes as in the main corpus**.** The following equation illustrates the sample fraction:

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

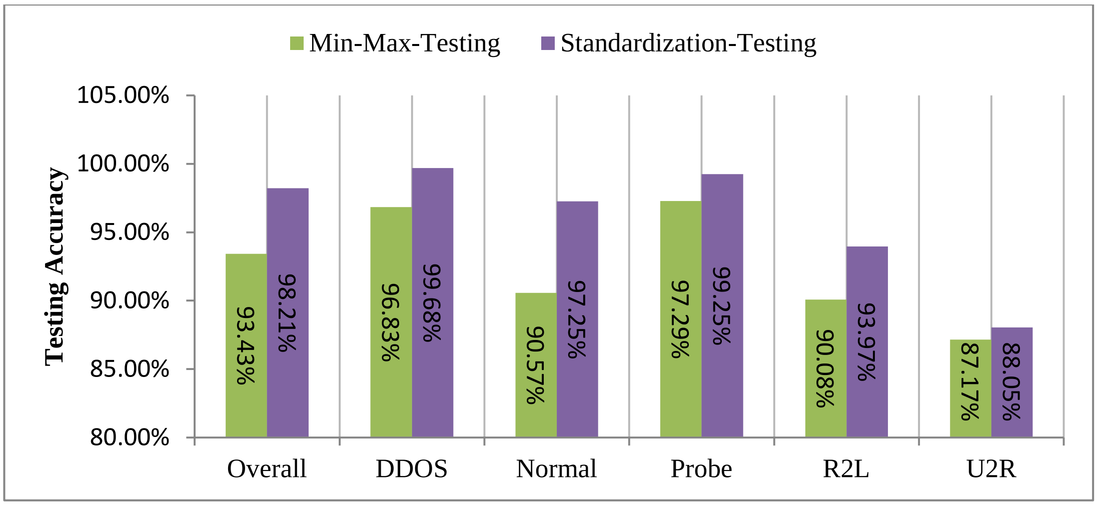

Fig. 2. The effect of the Min-Max and standardization normalization on the accuracy of WSVM models.

Where f_i is the fraction of class i in 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 belong to class i in the arbitrary set j .

It guarantees that any subset includes records from all classes and in the same ratio of the main corpus, 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 is the reason for using this method.

3.4. Support Vector Machine

SVM is a powerful classification method. It is used directly for binary classification, and with some trick, it is used as a multiple class classification algorithm. One of the tricks used to solve multi-class classification problem is building a set of one-against-one models, so if we have n classes we will build $n(n+1)/2$ models. The final result of all models is made based on voting strategy. This approach is used in the SVMLIB [26]. It is a Library of SVM which includes set of SVM algorithms for different purposes such as binary and multiple class classification, regression and distribution estimation. It supports several interfaces and extensions for different programing environments like MATLAB, Java, R, Python, C++ and C#. The MATLAB version of C Support vector classification (C-SVC) algorithm with one against one trick for multiple class classification was used to build our models.

The robustness of SVM doesn't come from the searching for the hyperplane that correctly separates the data, but the searching for the maximal margin hyperplane. Now we can see the problem as a set of binary class classification sub-problems. The binary class classification SVM was solved using C-Support Vector Classification algorithm (C-SVC*)* in the used library.

Looking for SVM shows that SVM is based on mapping the problem into high dimensional features space, and then the linear hyperplane is constructed in that space [27]. It is important to understand the fact of the existence of few points that do not site on the correct side of the plane after building the models, this emphasis the use of a slack parameter to move this point to the correct side, this model called Soft-margin SVM [28]. Now, suppose there are l records of training data denoted by x_i where $x_i \in R^n, i = 1, \dots l$, these records belong to one of two classes donated by y, while $y \in R^l$, and the normalized margin separate the two classes equal $1/\|w\|$.

So, the main optimization problem will be described using the following formulas:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{n} \xi_n \tag{4}$$
$$subject\ to\ y_i(w^2 \emptyset(x_i) + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0, i = 1, \dots \dots . l$$

Where $\emptyset$ the mapped feature of x_i , ξ is the slack parameter and C is the regularization parameters.

At the next step, the optimization problem (4) is reformulated using the Lagrange Method then the Lagrange form of the optimization problem is solved [28]. Finally, the decisions are performed based on the following formula:

$$sgn(w^T \emptyset(x) + b) = sgn\left(\sum\nolimits_{i=1}^{l} y_i \alpha_i K(x_i, x) + b\right) \tag{5}$$

Where α_i is a Lagrange constant it must be ≥ 0, and $\mathrm{K}(\mathrm{x_i}, \mathrm{x})$ is the kernel function.

Different Kernels are used with *SVM* like sigmoidal, polynomial, and Gaussian RBF basis kernels. Gaussian RBF basis kernel was used in this work which represented in the following formula:

$$K(\bar{X}_i, \bar{X}_j) = e^{\frac{-\|\bar{X}_i - \bar{X}_j\|^2}{2\sigma^2}} \tag{6}$$

There are two parameters C and σ in our problem, the default values were used without optimization.

3.5. Accuracy Paradox and Cost-Function Schema

Accuracy paradox is the behavior of most machine learning algorithms toward learning imbalanced classes datasets, it is easier for these algorithms to classify either all or most the records of the small classes into one or more of the major classes, this happen with negligible effect of the total accuracy. But the problem gets worse when these minor classes be crucial in the environment. Cost function is a scheme that affect the learning process by changing the weight of records in the learning process. In this scheme, the records of minor classes have greater weights than the biggest classes, this will improve the learning process for these minor classes. SVM, as mentioned, is a powerful and robust classification algorithm, but it is like most pattern recognition models make accuracy paradox when training using imbalanced classes. Cost-function is one of the methods were suggested to address this problem [15] .It gives different weights to the records that belong to different classes, many cost-function options were suggested by Alejo, et al [15], the option 2 was used here which illustrated in the following equation:

$$W_i = \frac{N}{n_i} \tag{7}$$

Where W_i is the weight for all records which belong to class i , N is the number of records in the corpus and n_i is the number of records belong to class i in the corpus.

The complete procedure which used in performing the experiments is illustrated by **Algorithm 1:** in detail.

Algorithm 1: The proposed algorithm for building the Intrusion detection models

Input: NSL-KDD dataset, the number of partitions used with cross-validation method P.
Output: P number of WSVM models, the testing-phase results on these models which include the overall accuracy and the accuracy for each class.

Data Preprocessing:

```
// Converting nominal fields into numerical
for field in 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 = $\mathbf{replace}$ (dataset, numfield, field) // replace the old field with the new
                numeric field
        end if

    // Applying the standardization method on the dataset fields
    for field in dataset, do
        for variable = 1 to size(field), do
            field [variable] ← $\frac{\text{field[variable]} - \text{mean(field)}}{\text{standerd deviation(field)}}$

    // Partitioning the dataset into P sub-sets
        // Calculate the fraction of each class i
        $\mathrm{f_i} \leftarrow \frac{\text{number of records}_\mathrm{i}}{\mathbf{Size}(\text{dataset})}$
        DatasetList ← $\mathbf{StraifiedSampling}$(dataset, N, f)

    // Generating the weight array which elements represent a weight for distinct class i
    $\mathrm{w_i} \leftarrow \frac{\mathbf{Size}(\text{dataset})}{\text{number of records}_\mathrm{i}}$

Main:

    for variable = 1 to P do
        $\text{model}_\text{variable}$ ← $\mathbf{BuildWSVMModel}$(dataset − { $\text{i}^\text{th}$partiton},
        RBF Kernel , default C,  default γ, w)
        testing phase results = $\mathbf{TestWSVM}$($\text{model}_\text{veriable}$, $\text{i}^\text{th}$partition, $\text{i}^\text{th}$partitionLabels)
        Compute the overall accuracy and the accuracy for each class for the training data.

    return P number of WSVM models, the testing-phase results on these models.
```

4. Experiments and Results

We used weighted SVM to build pattern recognition models as intrusion detection solution. Matlab version of well-known LIBSVM library [26] is used to build the models. RBF kernel C-Support Vector algorithm was used for building multi-classes classification models with default values of *C* and gamma. These experiments performed on NSL-KDD dataset, the benchmark dataset which includes four type of network attacks which are DoS. Probing, R2L, U2R in addition to normal. As mentioned before, the complete records in both files KDD-Train+ and KDD-Test+ are used as one set. The cross-validation method called leave-one-out [22] was used to build, evaluate and validate the models. In this method, the data will divide into a number of partitions, one of them used for testing while the others used for training these models. According to that and for the purpose of comparison, two forms of experiments were performed; one of them used two folds and the other used ten folds.

Ten models for tenfold and two for twofold scenarios were built. In each model, one portion was left for testing while the others were used in training phase. The cost function schema used to evaluate the cost for different classes using the equation (7). The results of twofold and tenfold cross-validation experiments illustrated in **Table 2** and **Table 1**: The result of WSVM in 10 folds cross-validation

. Both tables include the total accuracy for each phase and the accuracy for each class in all rounds.

Table 1. The result of WSVM in 10 folds cross-validation

	Round 1		**Round 2**		**Round 3**		**Round 4**		**Round 5**	
	Training Accuracy	**Testing Accuracy**	**Training Accuracy**	**Testing Accuracy**	**Training Accuracy**	**Testing Accuracy**	**Training Accuracy**	**Testing Accuracy**	**Training Accuracy**	**Testing Accuracy**
Total Accuracy	98.46%	98.40%	98.48%	98.40%	98.48%	98.29%	98.44%	98.44%	98.47%	98.24%
DOS	99.76%	99.76%	99.79%	99.74%	99.76%	99.72%	99.78%	99.74%	99.75%	99.76%
Normal	97.47%	97.51%	97.49%	97.55%	97.51%	97.30%	97.46%	97.66%	97.53%	97.21%
Probing	99.44%	99.36%	99.46%	99.36%	99.55%	99.36%	99.42%	99.15%	99.47%	99.43%
R2U	96.21%	94.65%	96.31%	93.80%	95.96%	95.77%	95.80%	94.08%	95.40%	94.37%
U2R	98.77%	88.89%	98.28%	93.33%	98.77%	84.78%	98.77%	89.13%	99.26%	86.67%

	Round 6		**Round 7**		**Round 8**		**Round 9**		**Round 10**	
	Training Accuracy	**Testing Accuracy**	**Training Accuracy**	**Testing Accuracy**	**Training Accuracy**	**Testing Accuracy**	**Training Accuracy**	**Testing Accuracy**	**Training Accuracy**	**Testing Accuracy**
Total Accuracy	98.48%	98.37%	98.49%	98.11%	98.47%	98.35%	98.45%	98.29%	98.44%	98.22%
DOS	99.76%	99.74%	99.76%	99.72%	99.75%	99.79%	99.75%	99.79%	99.79%	99.63%
Normal	97.53%	97.48%	97.53%	97.07%	97.49%	97.48%	97.47%	97.43%	97.44%	97.39%
Probing	99.48%	99.15%	99.44%	99.08%	99.57%	98.93%	99.46%	98.93%	99.46%	99.01%
R2U	95.90%	95.49%	96.40%	93.79%	95.99%	95.21%	96.27%	93.80%	95.59%	93.80%
U2R	99.02%	86.67%	98.28%	88.89%	98.77%	82.22%	98.53%	82.22%	99.26%	84.44%

Table 2. The result of WSVM in 2 folds cross-validation

	Round 1		Round 2	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
Total Accuracy	98.36%	98.18%	98.39%	98.12%
DOS	99.76%	99.68%	99.69%	99.71%
Normal	97.26%	97.22%	97.44%	97.13%
Probing	99.57%	99.13%	99.32%	98.96%
R2U	96.34%	94.64%	95.55%	93.46%
U2R	99.12%	82.30%	98.67%	89.38%

In order to evaluate this effort, it was compared with the newly published works [10] [20] [21]. As mentioned in the related works section, the works in [10] [20]built multi-level pattern recognition models. While CART algorithm was used [10] to generate rule system to distinct the normal from abnormal records, then a generated features by DWT was used with SVM and NN to build the predictive models for the abnormal classes, the work in [20] built a hybrid multi-level (SVM-ELM-SVM-SVM-SVM) model with selected records from the 10% KDDCup99 using K-means clustering *algorithm*. The last work [21], proposed a framework that based on multiple kernels combination called MARK-ELM.

The **Table 3** present our results compared with both works addressed by [20] [21]. The table shows that our model outperforms the hybrid (SVM and EML) multi-level model in the overall accuracy and all sub-classes except the normal class, this performed with significant improvement for the minor classes. The average accuracy for our twofold and tenfold were (94.05% and 94.48% sequential) vs 21.93% for R2U attacks. Our model got on average (86.72% and 85.84%) in twofold and tenfold experiment while the hybrid multi-level model achieved 31.39%. On the other hand, we compared with the winner hybrid kernel method of the MARK-ELM framework which is called F-Poly kernel, even it got a better result in overall accuracy and normal class; our model can compete them in DOS and R2U classes and it does better in the Probing and U2R classes.

Table 3. Comparison between The Testing result of our models, Multi-level SVM & EML and MARK-ELM FPoly kernel set

	Our 10 Folds model results			**Our 2 Folds model results**			**Multi-level SVM & EML**	**MARK-ELM F-Poly kernel set**
	Max	**Min**	**Average**	**Max**	**Min**	**Average**		
Overall accuracy	98.44%	98.11%	98.31%	98.18%	98.12%	98.15%	95.75%	99.83%
DOS	99.79%	99.63%	99.74%	99.71%	99.68%	99.69%	99.54%	99.96%
Normal	97.66%	97.07%	97.41%	97.22%	97.13%	97.18%	98.13%	99.89%
Probing	99.43%	98.93%	99.18%	99.13%	98.96%	99.05%	87.22%	97.42%
R2U	95.77%	93.79%	94.48%	94.64%	93.46%	94.05%	21.93%	94.94%
U2R	93.33%	82.22%	86.72%	89.38%	82.30%	85.84%	31.39%	62.87%
Number of Samples	14852			74258			-	72793

On the other hand, **Fig. 3** shows the tested accuracy for our tenfold model and the multi-level models suggested in [10], both works used tenfold cross-validation on the same data. To perform the meaningful comparison, between these works which performed separately, the ten rounds results were sorted in ascending order and then the chart was made. It is obvious that our model is the most stable one; On the other hand, it outperforms the multi-level SVM in all rounds and multi-level Neural Network in the most rounds. Moreover, the less fortunate class which is U2R was excluded from the start in different multi-level ID models while our model achieved superior accuracy in all minor classes.

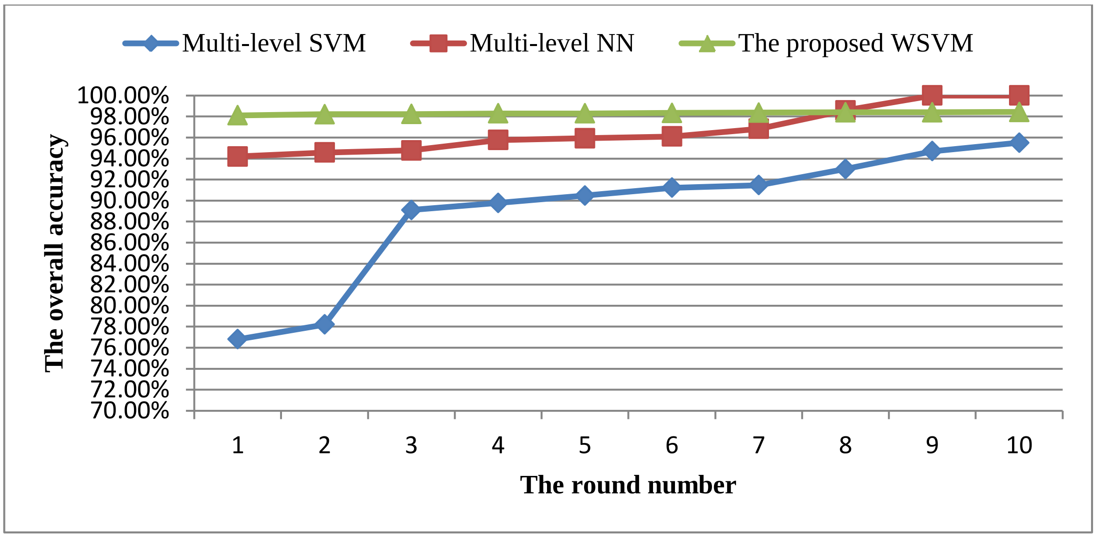

Fig. 3. Comparison between The Testing result of our 10 Folds model and the Multi-level ID methods for abnormal network behaviors work

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

5. Conclusion and Future Works

Intrusion detection system is a vital security tool. The daily Increase of the number of attacks encourages the development of the IDS. In this paper, a method was proposed for detecting the intrusion by a machine learning tool that combined the cost-function with SVM and proceeding with stratified samples method. Weighted SVM is effective when dealing with a subset of the training dataset contains many more samples than the 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 intrusion detection capable of maintaining better levels of accuracy as well as improving the detection of the most dangerous classes. The truth associated with this problem is that none of the open issues has been solved permanently and all points still opened. In the future work, we will deploy other algorithms from powerful family like feed-forward neural networks with set of cost-function schemes. We will use one of the optimization method to find the best values of the algorithms parameters. In addition, we will start using set of one-class classification methods which can be used in different manners. It is suggested to solve the imbalanced 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 detection models which may improve the detection of zero-day attacks.

References

[1] Cisco, "Cisco 2016 annual security report," Cisco, 2016.

[2] R. Walters, "heritage," The heritage Foundation, 27 October 2014 . [Online]. Available: Article (CrossRef Link). [Accessed 17 2 2017].

[3] S. M. Bellovin, "A look back at" security problems in the tcp/ip protocol suite," in *Proc. of Computer Security Applications Conference, 2004. 20th Annual*, 2004. Article (CrossRef Link).

[4] D. Munjin and J.-H. Morin, "Toward internet of things application markets," in *Proc. of Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, 2012. Article (CrossRef Link).

[5] M. H. Bhuyan, D. K. Bhattacharyya and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Ieee communications surveys \& tutorials*, vol. 16, pp. 303-336, 2014. Article (CrossRef Link).

[6] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. of Security and Privacy (SP), 2010 IEEE Symposium on*, 2010. Article (CrossRef Link).

[7] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research and Technology. ESRSA Publications*, 2013.

[8] P. Aggarwal and S. K. Sharma, "Analysis of KDD Dataset Attributes-Class wise for Intrusion Detection," *Procedia Computer Science*, vol. 57, pp. 842-851, 2015. Article (CrossRef Link).

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

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

[11] C. Thomas, "Improving intrusion detection for imbalanced network traffic," *Security and Communication Networks*, vol. 6, pp. 309-324, 2013. Article (CrossRef Link).

[12] J. K. Bains, K. K. Kaki and K. Sharma, "Intrusion Detection System with Multi Layer using Bayesian Networks," *International Journal of Computer Applications*, vol. 67, 2013. Article (CrossRef Link).

[13] M. Ahmed, A. N. Mahmood and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, 2016. Article (CrossRef Link).

[14] W. Zong, G.-B. Huang and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229-242, 2013. Article (CrossRef Link).

[15] R. Alejo, J. M. Sotoca and G. A. Casañ, "An empirical study for the multi-class imbalance problem with neural networks," in *Proc. of Iberoamerican Congress on Pattern Recognition*, 2008. Article (CrossRef Link).

[16] M. N. Abdurrazaq, B. Rahardjo and R. T. Bambang, "Improving performance of network scanning detection through PCA-based feature selection," in *Proc. of Information Technology Systems and Innovation (ICITSI), 2014 International Conference on*, 2014. Article (CrossRef Link).

[17] S. Anu and K. P. M. Kumar, "Hybrid Network Intrusion Detection for DoS Attacks," *Analysis (PCA)*, vol. 5, 2016.

[18] P. Laskov, P. Düssel, C. Schäfer and K. Rieck, "Learning intrusion detection: supervised or

unsupervised?," in *Proc. of International Conference on Image Analysis and Processing*, 2005. Article (CrossRef Link).

[19] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484-497, 2017. Article (CrossRef Link).

[20] W. L. Al-Yaseen, Z. A. Othman and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296-303, 2017. Article (CrossRef Link).

[21] 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. Article (CrossRef Link).

[22] C. C. Aggarwal, Data mining: the textbook, Springer, 2015. Article (CrossRef Link).

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

[24] I. Homoliak, D. Breitenbacher and P. Hanacek, "Convergence Optimization of Backpropagation Artificial Neural Network Used for Dichotomous Classification of Intrusion Detection Dataset," *Journal of Computers (JCP)*, vol. 12, pp. 143--155, 2017.

[25] "Wikipedia," July 2012. [Online]. Available: Article (CrossRef Link). [Accessed 9 2 2017].

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, p. 27, 2011. Article (CrossRef Link).

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273-297, 1995. Article (CrossRef Link).

[28] H. Daumé III, "A course in Machine Learning," *Publisher, ciml.info* , vol. 5, p. 69, 2012.

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

Mohammed Awad received the B.S. Degree in Industrial Automation Engineering from Palestine Polytechnic University in 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 Assistant Professor in Computer System Engineering Department, College of Engineering and Information Technology at Arab American University. At 2010 he has been Associate Professor in Computer Engineering. At 2016 he has been Full Professor in Computer Engineering. He worked for more than 12 years at the Arab American University in academic Position, in parallel with various Dean of Scientific Research and Editor-In-Chief, Journal of AAUJ). Through the research and educational experience, he has developed strong research record. His research interests include Artificial Intelligence, Neural Networks, Function Approximation and Complex Systems, Clustering Algorithms, Optimization Algorithms, and Time Series Prediction. He won a number of awards and research grants.