Arab American University- Jenin

Faculty of Graduate Studies

Intelligent Solution for New Cyberspace Attacks

By

Ala'Eddin Minwer Saleh Alabdallah

Supervisor

Prof. Dr. Mohammed Awad

This thesis was submitted in partial fulfillment of the requirements for the Master's degree in Computer Sciences.

December 2017

Intelligent Solution for New Cyberspace Attacks

By

Ala'Eddin Minwer Saleh Alabdallah

This thesis was defended successfully on **23/12/2017** and approved by:

Committee Members	Signature
1. Supervisor: **Prof. Dr. Mohammed Awad**	….....
2. Internal Examiner: **Dr. Amjad Rattroot**	.....
3. External Examiner: **Dr. Mohammad AL-Dasht**	.....

Dedication

It is our genuine gratefulness and warmest regard that we dedicate this work to my parents, my wife, my children, my brothers, my sister and all my friends.

Acknowledgments

Dr. Mohamed Awad has been the best thesis supervisor. His sage advice, support, and patient encouragement aided the writing of this thesis in best ways.

Abstract

The main issues of Intrusion Detection Systems (IDS) are the sensitivity of these systems toward the errors and the inconsistent and inequitable ways in which the evaluation processes of these systems were often performed. Most of the previous efforts concerned about improving the overall accuracy of these models via increasing the detection rate and decreasing the false alarm which is truly important. However, even they improved the overall accuracy of these systems; they almost fell in the accuracy paradox phenomena. Machine Learning (ML) algorithms mostly classifies all or most the records of the minor classes to one of the main classes with negligible impact on performance. The seriousness of the threats caused by the minor classes and the short coming 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 thesis, stratified sampling method and different cost-function schemes were consolidated with both Support Vector Machine (SVM) and Extreme Learning Machine (ELM) methods to build competitive ID solutions that improved the performance of these systems and reduced the occurrence of the accuracy paradox problem. This, while ensuring a consistent and fair evaluation of the performed experiments. The main experiments were performed on NSL-KDD dataset while that the UNB ISCX2012 dataset was used to proof the concept. The experimental results of NSL-KDD dataset showed that the ten-fold Gaussian radial base function (RBF) kernel WSVM model was better than Ji et al. Multi-Level ID method models, it was the most stable one and it had better performance than the multi-level SVM model in all rounds and the multi-

level neural network (NN) model in most rounds. They also showed that the optimized Gaussian RBF kernel with two-fold SVM model was better performance than Al-Yaseen at el. Multi-level hybrid SVM and ELM models in overall accuracy, recall and F-score. Also, it competed the best model of Fossaceca et al. MARK-ELM in DoS and R2U classes and it had better performance in the Probing and U2R classes. While the experimental results of UNB ISCX2012 dataset showed that the optimized Gaussian RBF with WSVM was better than the polynomial kernel SVM model in the recent thesis in the overall accuracy in addition to all F-score values except the Botnet F-Score. The better F-score of the botnet that achieved by the previous thesis experiments on a random selected subset did not reflect better performance on that set because the weakness of the experiments.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CART	Classification and Regression Trees
C-SVC	C Support Vector Classification
DDoS	Distributed Denial of Service Attacks
DoS	Denial of Service Attacks
DWT	Discrete Wavelet Transformation
ELM	Extreme Learning Machine
FAR	False Alarm Rate
FFNN	Feedforward Neural Networks
FN	False Negative
FP	False Positive
FTP	File Transfer Protocol
G-mean	Geometric mean of Recall and Precision
HTTP	Hypertext Transfer Protocol
ID	Intrusion Detection

IDS	Intrusion Detection System
IMAP	Internet Message Access Protocol
IP	Internet Protocol
iPCA	interactive system for Principal Component Analysis
IRC	Internet Relay Chat
ISCX	Information Security Centre of Excellence
KDD	Knowledge Discovery and Data Mining
KKT	Karush Kuhn Tucker
LBNL	Lawrence Berkeley National Laboratory
MARK-ELM	Multiple Adaptive Reduced Kernel Extreme Machine Learning
ML	Machine Learning
MLPNN	Multilayer Perceptron Neural Network
NN	Neural Network
PCA	Principal Component Analysis
POP3	Post Office Protocol 3

R2L	Remote to Local attacks
RBF	Radial Base Function
SLFFN	Single hidden Layer Feedforward Neural Network
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
SVM	Support Vector Machine
SVMLIB	A Library for Support Vector Machines
TCP	Transmission Control Protocol
TN	True Negative
TP	True Positive
U2R	User to Remote attacks
UNB	University of New Brunswick
UTF	Unicode Transformation Format
WELM	Weighted Extreme Machine Learning
WSVM	Weighted Support Vector Machine

Chapter 1

Introduction

1 Introduction

With the growth of computer networks and the increase of the services that offered by the computing systems, the need to maintain the reliability, integrity, and availability of these systems is increasing, this makes the security of these systems more important. On the other hand, the attackers increased the directed attacks on these systems which become a serious problem [1]. The operations of cyber-attacks able to cause significant economic damage to both public and private 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 multi-disciplinary problem. One side of the problem is a cybersecurity problem, and the other side is the statistical, Knowledge-Based and ML fields that represent the factories that produce the pool of solutions, this thesis interests in the ML solutions of the ID problem. 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 for previous, current and future attacks. Part of this fact presented in [3] which included a list of security flaws in the TCP stack protocols.

The ID solutions are categorized into one of three common methodological categories [4]. The first category called misused or signature-based IDS, in this approach, either different normal and different abnormal known rules or patterns are captured in training phase from

labeled data, and then the generated models are used to make a prediction for the 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 [5], which assumes the capability 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 detecting 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.

The network ID field has 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 tools that used to solve the ID problem. Computer networks generate 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 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 beside the correct labeled records.

Third, this issue related to 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 thesis 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 either fail to predict like [9] or predict with an insufficient accuracy of the minor classes like user to remote (U2R) and remote to local (R2L) classes in the NSL-KDD dataset; even they 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 high level of security as U2R and R2L in the NSL-KDD dataset.

In this thesis, we are interested in improving the accuracy of IDSs for the new attacks and mitigating the existence of accuracy paradox problem. So, two weighted algorithms which are SVM and Extreme Machine Learning (ELM) with different weight schemes, stratified sampling and with optimizing for some parameters of these algorithms were consolidated to solve this problem. WSVM is an effective algorithm than any other algorithm when dealing with any training subset contains many more samples than the others. Also, 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 in consistent and fair way, it was taken into account the data selection reasons and the way of performing different tests.

These experiments were performed on two benchmark datasets which are the NSL-KDD and UNB ISCX2012. The NSL-KDD dataset is used to apply the main experiments while the UNB-ISCX2012 dataset is used to apply the support experiments. The NSL-KDD is public benchmark dataset [12]; it is an improved version of the KDDCup99 dataset, which is the most frequently used benchmark dataset in this field. It includes labeled records from five classes which are Normal, Denial of Service Attacks (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, it is still the most important general benchmarked dataset on one hand and the points we are focusing on to address are existing in our set comparting with the real traffic on the other hand. For this work scope, this dataset is sufficient leaving behind the former open points to address in other locations. The UNB ISCX 2012 was suggested to perform the verification experiments, it 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 type of attack scenarios which are inside network infiltration, Hypertext Transfer Protocol (HTTP) denial of service, IRC Botnet Distributed Denial of Service Attacks (DDoS), Brut 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 that developed using anomaly-based methods to deploy in operational real-world environments [5]. The awareness of the pressing needs to improve powerful and dynamics security tools that protect the contemporary computing systems emphasis the great interest of researchers of both communities to improve the IDSs.

1.1 Objective

The objectives of this thesis are to improve the accuracy of IDSs for the new attacks and to mitigate the existence of accuracy paradox problem. This phenomenon appears as a result of existing small classes in any dataset. Some of these minor classes have serious effects on the security of computing systems as U2R and R2L attacks. As well we concerned in performing the evaluation of the performed experiments in consistent and fair way.

1.2 Contribution

This thesis presented several ID solutions on NSL-KDD and UNB ISCX2012 datasets, these solutions were evaluated in consistent and fair way with a set of new preceding works that intersect with our interest and very close to our tests. With regard to NSL-|KDD dataset, both algorithms had better performance than two of three works, even that the third work got better results in overall accuracy, G-mean, and F-score of the normal class; our models can compete them in DoS and R2U classes and it does better in the Probing and U2R classes. Regards to UNB ISCX2012 dataset, both algorithms with best-optimized parameter had better performance in the overall accuracy and in F-score of all classes except the botnet. This does not reflect better performance in this class, because the experiments on the previous works performed once on a randomly selecting subset while our results represent the average of ten round on different ten randomly selecting subsets.

1.3 Overview

The remainder of this thesis is arranged as the following. In Chapter 2, will present a background that includes the dataset description of the NSL-KDD and UNB ISCX2012

datasets, then a literature review that includes works in the multiple class classification solutions of the ID problem and others included some techniques used to treat the unbalanced class problem. Chapter 3 presents the methodology that includes the dataset selection considerations firstly. Secondly, a brief description of the preprocessing phase is included; the stratified sampling method is illustrated as a part of the preprocessing phase. Thirdly, Both WSVM and WELM algorithms are explained, different weight schemed are illustrated to be combined with both algorithms. Finally, different concepts and metrics of the accuracy are introduced. Then in Chapter 4, all experiments on both NSL-KDD and UNB ISCX2012 datasets are illustrated, and the summary of results are included. They include the selection considerations of normalization method, the selection of best kernel, regularization parameter C and weight scheme that were used with WSVM to perform our experiments, in addition to the selection of best number of hidden layer neurons L, C, activation function and weight scheme that were used with WELM to perform our experiments. The best models for each dataset are evaluated with new other works that intersect in concern with this thesis. After that, some conclusion and future works will be presented in Chapter 5. Finally, the appendix part includes the results of all experiments that were performed on NSL-KDD and UNB ISCX 2012 datasets to optimize some parameters of WSVM and WELM models.

Chapter 2

Background

2 Background

The fact that both internal and external users to the context of the network they can connect locally or remotely increases considerably the probability of attacks, for this they have been developed different tools and strategies, both in hardware and software, to detect intrusions. As example firewalls tool restrict service of unknown traffic [4]. IDSs were used to detect the attacks, in this work we will use ML methods to detect attacks with a certain percentage of accuracy. So, we will conduct experiments on some well-known ID datasets

2.1 Datasets Description:

There are still shortages in the available datasets in ID domain in spite of the great efforts were exerted in this field [13] [14], Some important datasets in the ID field are KDD-CUP99, NSL-KDD, UNB ISCX 2012 and Kyoto University dataset. we selected the NSL-KDD and UNB ISCX2012 dataset to perform our experiments due to some reasons that will be illustrated in section 3.1.

2.1.1 NSL-KDD Dataset

NSL-KDD dataset [15] is an improved version of KDDCUP99 public simulated benchmark dataset, it includes only the distinct records of the mother set. This selection solves the biasing problem that appears during training and evaluating any learning method. Furthermore, this makes the dataset size is reasonable; this facilitates performing the experiments on the complete dataset without the need to get some subsets randomly. This increases the ability to evaluate the different models in consistent and fair manner.

It consists of five categories. The first category is the normal traffic. The others represent abnormal traffic which falling into one of the following categories:

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

All these types of traffic are represented by 41 features that fall into three groups. They are basic features which were extracted from TCP/IP protocols records, the time-based 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.

The records of both files were used as one big set in this thesis. The Figure 2.1 shows the numbers of records for each category in this complete set of data and clarify the existence of the minor classes in this set.

2.1.2 UNB ISCX 2012 ID Evaluation Dataset

It is an ID evaluation dataset; it was created by Information Security Centre of Excellence (ISCX) [14]. A systematic approach was used to generate the modifiable, extendable, and reproducible dataset. In this approach, two different profiles α and β were generated; each one included a presentation set of normal or attack behaviors or event of the real and modern networks. They were the key to make these set modifiable, extendable and producible. They have included a description for generating real traffic related to FTP, HTTP, IMAP, POP3, SMTP and SSH protocols.

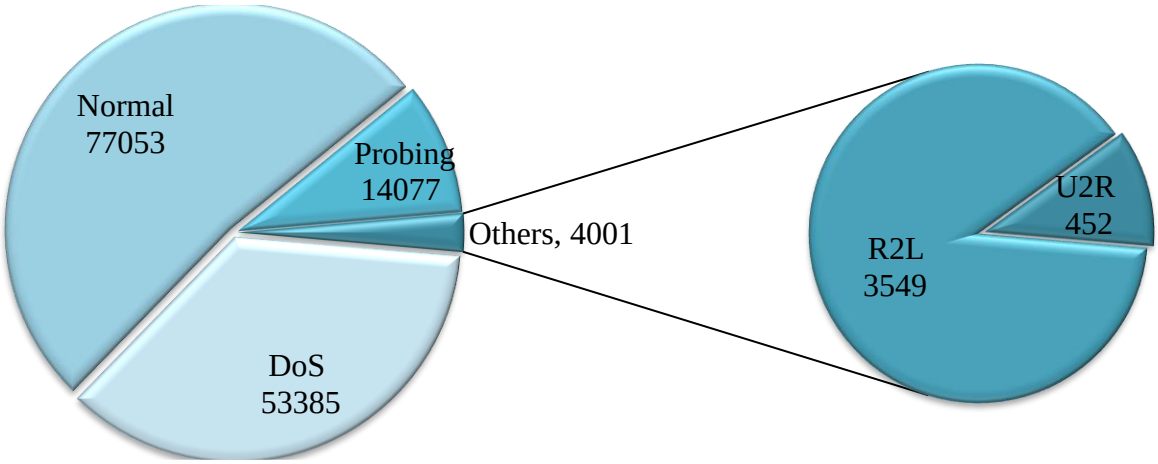


Figure 2.1: A chart illustrates the number of records for each category in the NSL-KDD dataset.

The dataset was generating during seven days, three days of them included only normal traffic while remain days included one scenario of attack for each in addition to the normal traffic. The four types of attacks that scenarios were deployed are inside network infiltration, HTTP denial of service, IRC Botnet DDoS, Brut force SSH. The traffic represented by 19 features, these features were listed in Table 2.1, the Tag feature was used for dataset labeling, it was used to distinguish between the normal and the attack traffic.

The record will classify by distinct the day that the attacks appeared in. 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. The distribution of the records in the days which included attacks scenarios illustrates in the Table 2.2.

Table 2.1: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	

It is clear that there are a sufficient number of records for each attack class, and there is a tremendous number of normal records.

Table 2.2: The distribution of the records in the UNB ISCX 2012 set 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

2.2 Related works

ID 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 [16] studied four categories of anomaly ID methods, they are classifications, statistical, information theory, and clustering. It founds that classifications and clustering outperformed in detecting DoS attacks, while statistical technique outperformed in U2R and R2L attacks. It focused on the lack of the public datasets that used in network intrusion system. The Mchine 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 was mentioned in section 1. Different approaches suggested [17, 18] 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 ML algorithms to address it's sensitivity to imbalanced classes. Different cost functions suggested in both works and applied with Neural Networks [18] and Extreme machine learning (ELM) [17] algorithms. In the last work, ELM neural network solved the time-consuming process on the FFNNs by initializing the weight of the hidden layer randomly, the proposed algorithm is fast, simple and has the capability to support different kernels with the small size dataset. Both preceding works concluded that using different weight scheme will improve the prediction of the small classes in the used dataset. In this thesis, different weight schemes will be tested with two different algorithms, one of them is the WELM that referenced in [17].

Several Network ID models proposed and tested in the last decade, these models were 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 or abnormal record prediction or multi-class classification prediction. On the other hand, a few efforts tried to build sub-models like in [19], where 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 [20] 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 [5] in order to reduce the FAR 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 ID solutions, the authors in [21] have built a framework and made a number of experiments. They demonstrate that supervised 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 [22] 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 heavy 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 [23] have built multi-level hybrid classification model based on an improved set include non-redundant 10% KDDCup99 subset. They have 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, they have deployed the ELM algorithm and they have sampled the data using a modified version of K-means clustering algorithm to get the best representative data. Although they have made some improvement of total prediction accuracy which was 95.75%, they have 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 introduced in literature for our problem, but at this time, it was combined multiple kernels together [24]. 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 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 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 [5] about the clearance of the interpretation of the result at evaluation step of our problem.

Many efforts performed to generate benchmark contemporary and real-time traffic dataset, 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, 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. The new thesis [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 thesis is inconsistent with the real number of records of the UNB ISCX dataset. The thesis assumed that the number of records of Botnet and DoS attacks equals 5

and 40 sequentially, while the correct number of these classes are 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 this work, we have deployed two weighted algorithms, which are SVM and 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 IDSs for the new attacks and mitigate the existence of accuracy paradox problem.

Chapter 3

The Proposed Method

3 The Proposed Method

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 datatype portability, data cleaning, feature selection and the stratified sampling. Next, it illustrates the deployed models which are the WSVM and WELM, they used to implement ID solutions. Finally, it includes different metrics that were used in the evaluation process.

3.1 Dataset Selection Considerations

There are still shortages in the available datasets in ID domain in spite of the great efforts 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 to preserving the 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. KDDCUP99 is the main public simulated benchmark

dataset which is still used for a lot of recent researches [20] [23] [24], although the gap between the characteristics of the contemporary traffic and the records were included in this dataset. 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 the unwanted biasing problem in the mother dataset. Several improved versions of KDDCUP99 were selected like 10 percent KDDCUP99 and NSL-KDD dataset. The NSL-KDD dataset was generated as an improved version of it is origin which includes only distinct records with reasonable size. The reasonable size of NSL-KDD dataset improves the evaluation consistency and efficiency for this set than any other dataset. Furthermore, the dataset includes two small classes, this is evident from Figure 2.1. It is an important and sufficient selection at this scope, so it was suggested to perform the primary experiments in this thesis.

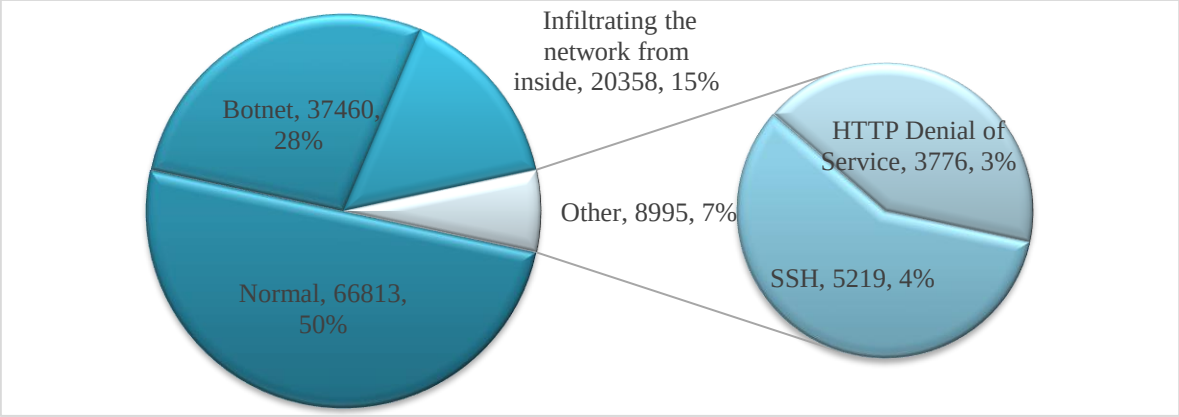


Figure 3.1: The records distribution of the selected sets of UNB ISCX 2012 dataset to build the secondary experiments.

To verify the proposed model, it was nessessary to select other dataset. But, we are interested in selecting a contemporary and real-time dataset. So, the UNB ISCX 2012 suggested to perform the verification 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 the secondary experiments based on the general method in this thesis using the complete records of all attacks in addition to some randomly selected normal subsets that have the same size of attacks records, these subsets have small classes. This is evident from Figure 3.1 which is shown the distribution of the records for that subsets.

3.2 Preprocessing Phase

Data preprocessing includes many steps [26] that depend on the nature of the data. Different preprocessing sub-steps were used; they included data-type portability, data cleaning, feature selection and the stratified sampling.

3.2.1 Main Preprocessing Steps

The NSL-KDD dataset consists of set of features that fall into three types which are Nominal, Numerical and Binary. None of them were excluded. It was observed that it did not have missing data, the numeric features did not follow balanced scale and the data was labeled into five classes which are Normal, DOS, Probing, U2R, and R2L.

UNB ISCX 2012 ID Evaluation Dataset consists of 19 features, they listed in Table 2.1. 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 balanced scale, so, they treated using data cleaning method. The generated tag feature refers to one of the following classes which are

Normal, inside network infiltration, HTTP DoS, IRC Botnet DDoS and Brut force SSH classes. Knowing that, this dataset was collected in seven days, only four of them included attacks scenarios, one class per day.

This phase started with converting the nominal or categorical data to sequential numeric values as data-type portability step. Then the imbalance scale of the features was addressed using two common methods [26] in data cleaning phase:

1. 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:

$$z_i^j = \frac{x_i^j - \mu_j}{\sigma_j} \qquad 3.2.1$$

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.

2. Min-Max Scaling method: It scales all attributes into $[0,1]$ range and it is represented using the following equation:

$$y_j^i = \frac{x_j^i - \min_j}{\max_j - \min_j} \qquad 3.2.2$$

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 feature. The selection considerations will illustrate in section 4.1.

3.2.2 Stratified Sampling

Stratified sampling is a statistical sampling method [27]. 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 [28] of their classes as in the main corpus. The following equation illustrates the sample fraction:

$$f_i = \frac{n_i^j}{N^j} \qquad 3.2.3$$

Where f_i is the fraction of the 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 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.

3.3 Building Models Phase

Both WSVM and WELM algorithms were used to build ID solutions on both NSL-KDD and UNB ISCX2012 dataset. WSVM is an effective algorithm than any other algorithm when dealing with any training subset contains many more samples than the others, while WELM is the fast and simple neural networks that solve the time consuming iterative process in FFNNs. This section will describe both algorithms in details.

3.3.1 Weighted Support Vector Machine

SVM is a powerful classification method. The robustness of SVM doesn't come from the search for the hyperplanes that correctly separates the data, but the search for the maximal margin hyperplane. It is used directly for binary classification, and with some tricks, it is used as a multiple class classification algorithm. One of these tricks which is used to solve multi-class classification problem is one-against-one trick, so if there is n classes, it 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 [29]. 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 and voting strategies for multiple class classification was used to build our models. Now we can see the problem as a set of binary class classification sub-problems which is solved using C-SVC.

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 [30]. 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 [31]. Now, suppose there are N records of training data denoted by x_i where $x_i \in R^n, i = 1, \dots N$, 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 $1/\|w\|$. The dual optimization problem which make tradeoff between maximizing the margin and decreasing the error using C regularization parameter will be described using the following formulas:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{n}\xi_n \qquad 3.3.1$$

$$\text{subject to } y_i(w^2\emptyset(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, i = 1, \ldots\ldots.l$$

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

At the next step, the optimization problem represented by equation 3.3.1 is reformulated using the Lagrange Method then the Lagrange form of the optimization problem is solved [31]. Finally, the decisions are performed based on the following formula:

$$\text{sgn}(w^T\emptyset(x) + b) = \text{sgn}\left(\sum_{i=1}^{l} y_i\alpha_i K(x_i, x) + b\right) \qquad 3.3.2$$

Where sgn is the step function, α_i is a Lagrange constant which must be ≥ 0 and $K(x_i, x)$ is the kernel function.

Different Kernels are supported with SVM like sigmoidal, polynomial, and Gaussian RBF kernels. Gaussian RBF kernel is represented by the following formula:

$$K(x_i, x) = e^{\gamma * |x_i - x|^2} \qquad 3.3.3$$

While the sigmoidal kernel is represented by the following formula:

$$K(x_i, x) = \tanh(\gamma * x_i' * x + coef0) \qquad 3.3.4$$

The last kernel was Polynomial which is represented by the following formula:

$$K(x_i, x) = \gamma * x_i' * x + coef0^{degree} \qquad 3.3.5$$

In this work, default value of degree = 3 and coef0 = 0 parameters were used. γ will be calculated using the following equation:

$$\gamma = \frac{1}{number\ of\ feature} \quad 3.3.6$$

we are interested in finding the best kernel, C and weight scheme that will be used to build the SVM ID solution. The selection considerations according to the dataset will be illustrated in subsections 4.2.1 and 4.3.1.

3.3.2 Weighted Extreme Learning Machine

ELM is a feedforward neural network, but it is not suffering from the time consuming and iterative process in the feedforward backpropagation neural network. This is addressed via random selection of the hidden layers weights and biases, so it is fast and simple method. A set of other features related to ELM still need to be discussed. One of them, it is the ability of ELM to deploy different feature mapping and kernels. The other point is the ability to build multiple class classification solutions easily in one model, without the need to combine multiple binary classes together.

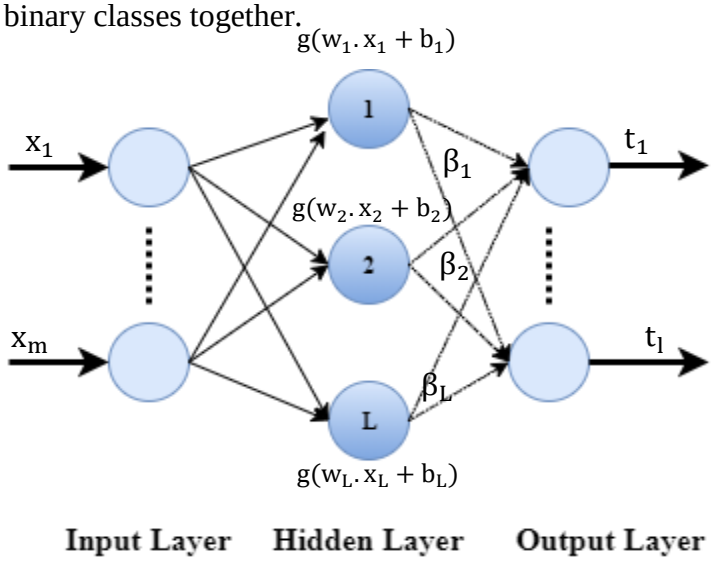


Figure 3.2: Extreme Learning Machine Network

The method illustrated in [17] was used in this thesis as 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 3.2. 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 $i = 1,2 \ldots\ldots, N$ and m features, while $\overrightarrow{T_i}$ is a target matrix. As 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 asking user to determine the activation function and the number of the hidden neurons which are denoted by $g(x)$ and L in sequential orders. Then the weight matrix W_{L*N} and bias vector B_{L*1} of hidden neurons are generated randomly, this saves the time for this algorithm and makes this algorithm faster. Different feature mapping can be used in ELM which represents different activation functions that can be used, examples of activation functions that can be used are:

- Sigmoid function

$$g(x) = \frac{1}{1 + e^{-x}} \tag{3.3.7}$$

- Gaussian function

$$g(x) = e^{-x^2} \tag{3.3.8}$$

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

$$H = g(W_{L*N} \,.\, X_{N*m} + B_{L*1}) \tag{3.3.9}$$

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:

$$\text{minimize} : \|H\beta - T\|^2 \text{ and } \|\beta\|. \tag{3.3.10}$$

The other form of the previous equation is:

$$\begin{gathered}\text{minimize} : Lp_{ELM} = \frac{1}{2}\|\beta\|^2 + CW\frac{1}{2}\sum_{i=1}^{N}\|\varepsilon_i\|^2 \\ \text{Subject to: } h(x_i)\beta = t_i^T - \varepsilon_i^T, \quad i = 1, \dots, N.\end{gathered} \tag{3.3.11}$$

Where W is 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 equation (3.3.11) using Lagrange and based on Karush Kuhn Tucker (KKT) theorem, it is being:

$$L_{D_{ELM}} = \frac{1}{2}\|\beta\|^2 + CW\frac{1}{2}\sum_{i=1}^{N}\|\varepsilon_i\|^2 - \sum_{i=1}^{N}\alpha_i(\ h(x_i)\beta - t_i^T + \varepsilon_i^T) \tag{3.3.12}$$

Where, α_i is the Lagrange Multiplier which is a constant.

In the next step, the partial derivative is performed based on β, α, and ε.

$$\begin{gathered}\frac{\partial L_{D_{ELM}}}{\partial \beta} = 0, \rightarrow \beta = \sum_{i=1}^{N}\alpha_i\ h(x_i)^T = H^T\alpha \\ \frac{\partial L_{D_{ELM}}}{\partial \varepsilon_i} = 0, \rightarrow \alpha_i = CW\varepsilon_i, \quad i = 1, \dots, N \\ \frac{\partial L_{D_{ELM}}}{\partial \beta_i} = 0, h(x_i)\beta - t_i^T + \varepsilon_i^T = 0, \quad i = 1, \dots, N\end{gathered} \tag{3.3.13}$$

Two forms of equation produce the β, caused by solving (3.3.13) equation, 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.

$$For\ small\ N:\ \beta = H^T\left(\frac{1}{C} + WHH^T\right)^{-1} WT \quad 3.3.14$$

$$For\ Big\ N:\ \beta = \left(\frac{1}{C} + H^T WH\right)^{-1} H^T WT \quad 3.3.15$$

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

$$f(x) = fun(h(x)\beta) \quad 3.3.16$$

$$fun(h(x)\beta) = \begin{cases} sign, & in\ binary\ problms \\ arg\,max\ (h(x)\beta)_l, & l\ is\ the\ number\ of\ classes \end{cases}$$

Due to the large sets represent the ID problems; the equation (3.3.15) 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. The selection considerations according to the dataset will be illustrated in subsections 4.2.2 and 4.3.2.

3.3.3 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 negligible effect of the total accuracy. But the problem gets worse when these minor classes be crucial in the environment. 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, which are:

First scheme, the default weight scheme where all classes have the same weight value which equal 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 is used to calculate the weights for each class:

$$W_i = \frac{N}{n_i} \tag{3.3.17}$$

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 class i 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:

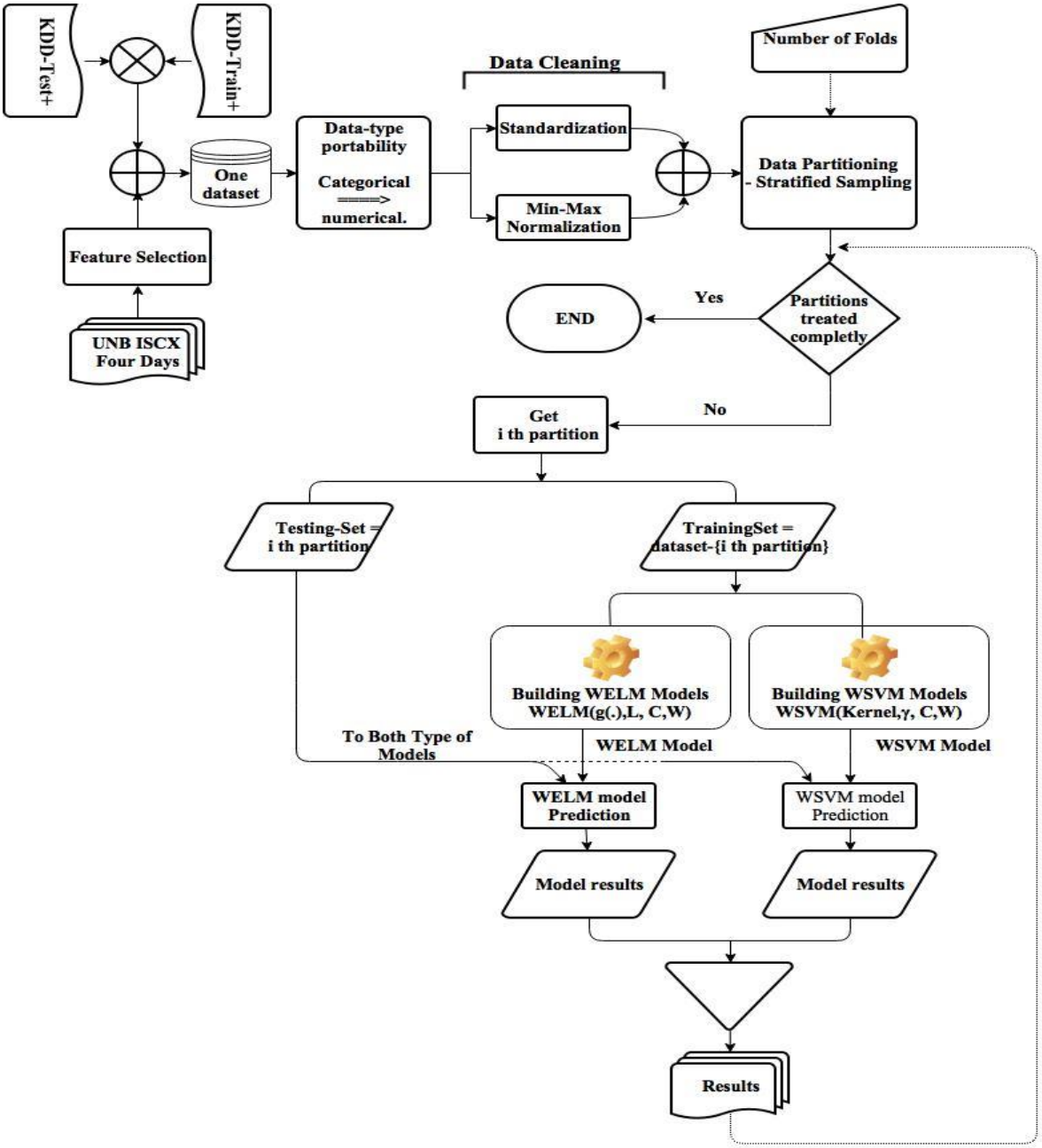
$$W_i = \begin{cases} \frac{0.618}{n_i} & if\ n_i > AVG(n_i) \\ \frac{1}{n_i} & if\ n_i \leq AVG(n_i) \end{cases} \tag{3.3.18}$$

Due to convergence issues of the deployed algorithms, the second method was used with the WSVM, while the third one was used with WELM in addition to the default weight scheme with both. But the default weight scheme with both algorithms can better improve the overall accuracy and mitigate the unbalanced class issue of the ID problem, this clearly appears in Experiments and Results chapter.

3.3.4 General Method Procedure

The general procedure that was used in performing all experiments on NSL-KDD dataset and the secondary experiments on UNB ISCX2012 dataset is shown in **Figure 3.3** and is illustrated by **Algorithm 1:** in detail.

Figure 3.3: The flow chart of the general method procedure



Algorithm 1: The general procedure that was used in building the ID models

Input: Dataset, the number of partitions method P, cost-function, regularization parameter C, number of hidden layer neurons L (only for ELM), ($kernel \mid activation\ function\ g(.)$);
Output: P number of models, ResultObject;

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 = **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*
  $f_i \leftarrow \frac{\text{number of records}_i}{\textbf{Size}(\text{dataset})}$
  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**
      $w_i \leftarrow \frac{\textbf{Size}(\text{dataset})}{\text{number of records}_i}$
    else If cost-function == third **then**
      if Size($class_i$) ≤ **Size**(dataset)/num_Classes
        $w_i \leftarrow \frac{\mathbf{1}}{\text{number of records}_i}$
      else
        $w_i \leftarrow \frac{0.618}{\text{number of records}_i}$

end if;
end if;
end if;

Main:

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

for i = 1 **to** P **do**
Train $\leftarrow$ dataset $-\{ i^{th}\text{partiton}\}$
Test $\leftarrow$ dataset[i^{th}partition]
If model == SVM **then**
$\text{model}_i \leftarrow$ **BuildWSVMModel**(Train, Kernel , C, default γ, W)
testing phase results = **TestWSVM**(model_i, Test, i^{th}partitionLabels)
else if model ==ELM **then**
$\text{model}_i \leftarrow$ **BuildWELMModel**(Train, g(.), L , C, default γ, W)
testing phase results = **TestWELM**(model_i, Test, i^{th}partitionLabels)
end if;
ResultObject = **computeAllMetrics**() /*Compute the overall accuracy and the accuracy for each class for the training data*/

return P number of models, Result.

3.4 Metrics Selection

Several metrics and definitions [32] [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).

It is important to clarify the concept of each term, these concepts will be 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. Table 3.1 illustrate the Confusion matrix description for IDS problem.

Accuracy: It is one of the main metrics that is used to measure the overall performance of the pattern recognition models; it is represented by the following formula:

$$Accuracy = \frac{\sum TP + TN}{\sum TP + \sum FP + TN + \sum FN + \sum MissCl_{-,-}} \quad 3.4.1$$

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:

$$Precision_i = \frac{TP_i}{TP_i + FP_i + MissCl_{(-,i)}} \quad 3.4.2$$

Table 3.1: Confusion matrix description for IDS problem

		Predicted Classes				
		DoS	Normal	Probing	R2L	U2R
Actual Classes	DoS	TP	FN	$MissCl_{(d,p)}$	$MissCl_{(d,r)}$	$MissCl_{(d,u)}$
	Normal	FP	TN	FP	FP	FP
	Probing	$MissCl_{(p,d)}$	FN	TP	$MissCl_{(p,r)}$	$MissCl_{(p,u)}$
	R2L	$MissCl_{(r,d)}$	FN	$MissCl_{(r,p)}$	TP	$MissCl_{(r,u)}$
	U2R	$MissCl_{(u,d)}$	FN	$MissCl_{(u,p)}$	$MissCl_{(u,r)}$	TP

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.

$$Recall_i = \frac{TP_i}{TP_i + FN_i + MissCl_{(i,-)}} \quad 3.4.3$$

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.

$$Specificity = \frac{TN}{TN + \sum FN} \quad 3.4.4$$

G-mean [33]**:** 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:

$$\text{G-mean} = \left(\prod_{1}^{m-1} sensitivity * specificity \right)^{\frac{1}{m}} \quad 3.4.5$$

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

$$\text{F-measuring} = \frac{2 * precision * recall}{precision + recall} \quad 3.4.6$$

Chapter 4

Experiments and Results

4 Experiments and Results

The primary experiments were performed on the NSL-KDD dataset, while the secondary experiments were performed on the UNB ISCX2012 dataset to justify the occurrence of thesis goals. The WSVM and WELM algorithms were used to perform multiple class classification experiments on both datasets. Before starting in the issues of the algorithms, it should be determined the best data normalization method which should be used later in all experiments. This selection depends on the properties of the data in the datasets, the existence of outlier records is the crucial properties of the ID datasets. This step will be addressed in the next subsection. WSVM was used to build pattern recognition models as ID solution. The MATLAB version of well-known library which is called LIBSVM [29] was used to build that models. It uses C-SVC algorithm with one against one scheme and voting strategy to build multiple classes classification solutions. The algorithm supports different kernels which are polynomial, sigmoidal and RBF kernel. The RBF kernel has $\boldsymbol{\gamma}$ parameter. The sigmoidal kernel has coef0 parameter in addition to $\boldsymbol{\gamma}$. The polynomial kernel has degree parameter in addition to coef0 and $\boldsymbol{\gamma}$ parameter. To make tradeoff between the distance of the separating margin and the training error, WSVM algorithm uses the regularization parameter C. A 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. Parameters optimization is an

important step to improve the algorithms performance knowing that the results of this step depend on the dataset used. The default values of $coef0 = 0$ and $degree = 3$ parameters for WSVM kernels parameters were used, while γ was calculated using $\frac{\mathbf{1}}{\mathbf{number\ of\ features}}$ formula [29].

The next point was testing the effect of different weight schemes on the performance of both algorithms. Both default weight scheme and first weight scheme were combined with WSVM, while the default weight and the second weight schemes were used with WELM.

The objective of the set of experiments conducted using the WSVM algorithm was finding the best kernel, C and weight scheme that will be used to build the SVM ID solution on. As well, the objective of the set of experiments conducted using the WEVM algorithm was finding the best L, C, activation function and weight scheme that will be used to build the ELM ID solution. All combinations of both algorithms parameters are listed in Table 4.1.

The cross-validation method called leave-one-out [26] was mainly used to build, evaluate and validate all models. Based on 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.

This thesis took into account making consistent and equitable assessment. Some of recent works which proposed multiple class classification solutions for ID problem on complete NSL-KDD dataset were selected to evaluate our models. Some of these works used twofold and others used tenfold cross-validation. Twofold cross-validation is more generalized than tenfold. In twofold cross-validation, both training and testing phases are performed on distinct 50 percent of dataset records, while in tenfold cross-validation the training phase

are 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 on the first dataset. The stratified sampling method was used to maintain the same ratio of number of records for any class to the number of records for all classes in all partitions and in the complete set.

Table 4.1: Various combinations of parameters were used to build thesis models.

<table>
<tr><th>Algorithm</th><th>Kernels, Activation Function</th><th>C</th><th>Weight</th><th colspan="3">The optimized Parameters, Number of Hidden Neurons</th></tr>
<tr><td rowspan="3">WSVM</td><td>Sigmoidal</td><td rowspan="4">$\begin{cases}1\\10\\50\\10^2\\300\\500\\10^2\\10^3\\10^4\\10^5\\10^6\end{cases}$</td><td rowspan="3">$\begin{cases}Default\ scheme:\\ w_i = 1\ \forall\ i\\ Second\ scheme:\\ w_i = N/n_i\end{cases}$</td><td rowspan="3">NSL-KDD
$\gamma = 0.0244$

UNB ISCX2012
$\gamma = 0.0909$</td><td rowspan="3">$coef = 0$</td><td rowspan="3">$Degree = 3$</td></tr>
<tr><td>Gaussian RBF</td></tr>
<tr><td>Polynomial</td></tr>
<tr><td rowspan="2">WELM</td><td>Sigmoidal</td><td rowspan="2">$\begin{cases}Default\ scheme: w_i = 1\ \forall\ i\\ Third\ scheme: w_i =\\ \quad\begin{cases}0.618/n_i, if\ n_i > avg(N)\\ 1/n_i, \quad if\ n_i \le avg(N)\end{cases}\end{cases}$</td><td colspan="3" rowspan="2">$\begin{cases}500\\700\\10^3\\1500\\2000\end{cases}$</td></tr>
<tr><td>Gaussian</td></tr>
</table>

The recent thesis which is referenced by [25] was used to make 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 inconsistency between the number of records as that thesis 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 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.

4.1 Standardization Method Selection Considerations

The NLS-KDD dataset was used to select one of the data cleaning methods which were standardization and Min-Max normalization. The cross-validation leave-one-out with $n = 2$ method was used to build WSVM models; these models used default C and γ parameters of the Gaussian RBF Kernel. The experiments were performed twice, at first time the data was cleaned with standardization method. In the second round the data was cleaned using MIN-Max normalization method.

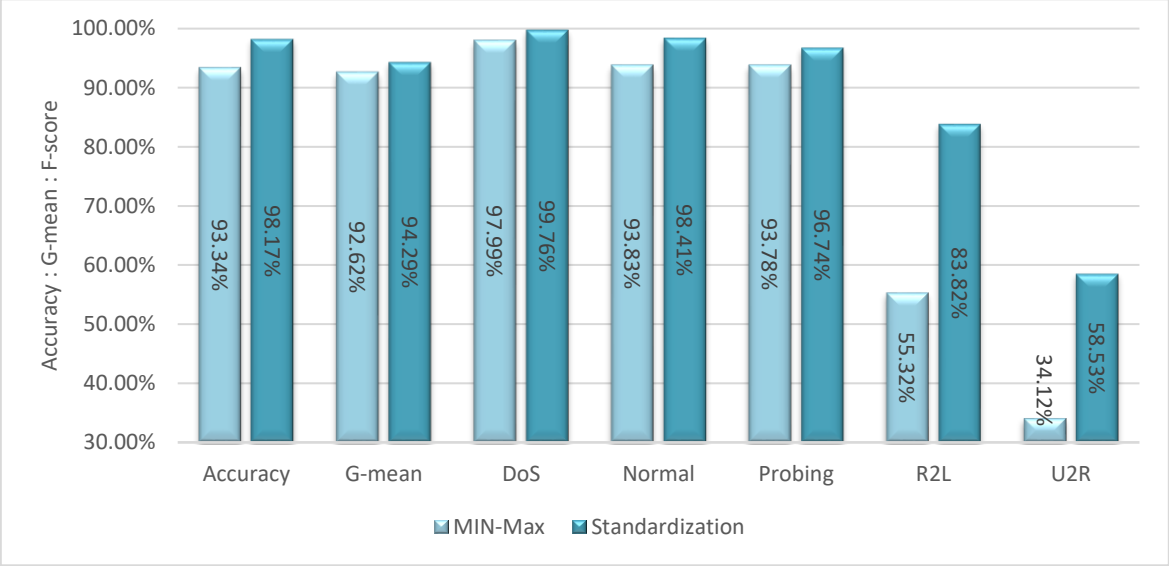


Figure 4.1: The effect of the Min-Max and Standardization normalization on the performance of WSVM models**.**

The models were built using 50% of NSL-KDD records which were selected based on stratified sample method then they were tested using the remainder records. These preliminary experiments were performed using the WSVM method showed that the

standardization method got better performance at all, this is shown in Figure 4.1. It shows the comparison between the two methods with respect to three different metrics, two of them are overall metrics which are the overall accuracy and the G-mean while the third is the F-score for each class. It is known that the standardization method does better in any environment that includes outlier records [26]. So, Standardization method was used to clean the data for the following experiments.

4.2 NSL-KDD dataset Experiments

Both WSVM and WELM algorithms were used to build ID solutions on NSL-KDD dataset. The best WSVM model with certain kernel, C and weight scheme and the best WELM model with certain L, C, activation function and weight scheme were proposed to address our thesis problem. Several experiments were performed to look for the optimized models, they are listed in the following subsections.

4.2.1 WSVM Experiments on NSL-KDD Dataset

The objective of the set of experiments conducted using WSVM algorithm was finding the best kernel, C and weight scheme that will be used to build the SVM ID solution. All combinations of these parameters are listed and shown in Table 4.1. The list that included all experiments that were performed to achieve this goal are:

Test 1: Several two-fold cross-validation models of WSVM with sigmoidal kernel were built to find the C value and the weight scheme that achieved the optimized model. Table A.1 includes the complete results of these models. A weak result appeared of applying the

sigmoidal kernel with WSVM as an ID solution. So, WSVM with sigmoidal kernel was excluded.

Test 2: Several two-fold cross-validation models of WSVM with Gaussian RBF kernel were built to find the C value and the weight scheme that achieved the optimized model. Table A.2 includes the complete results of these models. It shows that the best C value with the default weight scheme and the first weight scheme was 10^3. It is clear that when the C values increased; most results of assessment metrics increased until certain value then stabilized. The results of the optimized first weight scheme model were adequate while the results of optimized default weight scheme model were the winner. Table 4.2 shows the optimized models results for this test in detail.

Test 3: Several two-fold cross-validation models of WSVM with Polynomial kernel were built to find the C value and the weight scheme that achieved the optimized model. Table A.3 includes the complete results of these models. It shows that the best C value with the default weight scheme was 10^3 while the best C value with the first weight scheme was 500. It is clear that when the C values increased; most results of assessment metrics increased until certain value then stabilized. The results of the optimized first weight scheme model were adequate while the results of optimized default weight scheme model were the winner. Table 4.3 shows the optimized models results for this test in detail.

Test 4: A ten-fold cross-validation models of WSVM with Gaussian RBF kernel, $C = 100$ and first weight scheme were built on the NSL-KDD dataset. Table A.4 includes the complete results of these experiments. This test was added to compare with the work referred by [9].

Summary: the default weight scheme was the best tested weight scheme, the WSVM with Gaussian RBF was the best model with $C = 10^3$. Its assessment was 99.19 percent in the overall accuracy and 99.87, 99.30, 98.79, 92.50, 70.46 percent in the F-score of DoS, Normal, Probing, R2L, U2R classes.

Table 4.2: Test 2 optimized models results.

WSVM, Gaussian RBF Kernel, $C = 10^3$, Default weight scheme.											
	Accuracy	G-mean	FAR	Round 1							
	99.19%	93.73%	0.73%								
	Confusion matrix					Precision	Recall	F-score	FP	Miss Classification	Miss Detection
DoS	26657	34	2	0	0	99.89%	99.87%	99.88%	0.09%	0.01%	0.13%
Normal	25	38265	71	92	74	99.27%	99.32%	99.30%	99.27%	0.00%	0.68%
Probing	1	87	6946	5	0	98.95%	98.68%	98.81%	1.01%	0.09%	1.24%
R2L	2	125	1	1602	44	94.01%	90.30%	92.12%	5.40%	2.65%	7.05%
U2R	2	34	0	5	185	61.06%	81.86%	69.94%	24.42%	3.10%	15.04%
	Accuracy	G-mean	FAR	Round 2							
	99.20%	93.39%	0.76%								
	Confusion matrix					Precision	Recall	F-score	FP	Miss Classification	Miss Detection
DoS	26654	35	1	2	0	99.85%	99.86%	99.85%	0.12%	0.01%	0.13%
Normal	31	38281	72	69	73	99.24%	99.36%	99.30%	99.24%	0.00%	0.64%
Probing	6	89	6940	3	0	98.93%	98.61%	98.77%	1.03%	0.13%	1.26%
R2L	3	132	1	1609	30	95.21%	90.65%	92.87%	4.08%	1.92%	7.44%
U2R	0	37	1	7	181	63.73%	80.09%	70.98%	25.70%	3.54%	16.37%
WSVM, Gaussian RBF Kernel, $C = 10^3$, First weight scheme.											
	Accuracy	G-mean	FAR	Round 1							
	99.06%	94.16%	0.55%								
	Confusion matrix					Precision	Recall	F-score	FP	Miss Classification	Miss Detection
DoS	26657	28	6	0	2	99.88%	99.87%	99.87%	0.08%	0.03%	0.10%
Normal	22	38096	108	237	64	99.45%	98.88%	99.16%	99.45%	0.00%	1.12%
Probing	4	74	6956	5	0	98.33%	98.82%	98.58%	1.53%	0.13%	1.05%
R2L	2	81	2	1671	18	86.90%	94.19%	90.40%	12.32%	1.24%	4.57%
U2R	3	29	2	10	182	68.42%	80.53%	73.98%	24.06%	6.64%	12.83%

	Accuracy	G-mean	FAR	Round 2							
	98.93%	93.86%	0.61%								
	Confusion matrix					Precision	Recall	F-score	FP	Miss Classification	Miss Detection
DoS	26655	32	3	2	0	99.84%	99.86%	99.85%	0.11%	0.02%	0.12%
Normal	30	38012	123	293	68	99.39%	98.67%	99.03%	99.39%	0.00%	1.33%
Probing	7	90	6936	4	1	98.16%	98.55%	98.36%	1.74%	0.17%	1.28%
R2L	4	75	2	1681	13	84.51%	94.70%	89.32%	14.73%	1.07%	4.23%
U2R	1	35	2	9	179	68.58%	79.20%	73.51%	26.05%	5.31%	15.49%

Table 4.3: Test 3 optimized model results.

WSVM, Polynomial Kernel, $C = 10^3$, Default weight scheme.											
	Accuracy	G-mean	FAR	Round 1							
	99.13%	93.90%	0.68%								
	Confusion matrix					Precision	Recall	F-score	FP	Miss Classification	Miss Detection
DoS	26651	38	2	2	0	99.86%	99.84%	99.85%	0.11%	0.01%	0.14%
Normal	29	38236	80	107	75	99.32%	99.24%	99.28%	99.32%	0.00%	0.76%
Probing	5	86	6936	10	2	98.80%	98.54%	98.67%	1.14%	0.24%	1.22%
R2L	2	110	2	1603	57	92.61%	90.36%	91.47%	6.18%	3.44%	6.20%
U2R	2	28	0	9	187	58.26%	82.74%	68.37%	23.36%	4.87%	12.39%
	Accuracy	G-mean	FAR	Round 2							
	99.12%	93.83%	0.75%								
	Confusion matrix					Precision	Recall	F-score	FP	Miss Classification	Miss Detection
DoS	26650	36	3	3	0	99.76%	99.84%	99.80%	0.19%	0.02%	0.13%
Normal	52	38240	75	76	83	99.25%	99.26%	99.25%	99.25%	0.00%	0.74%
Probing	7	96	6926	9	0	98.84%	98.41%	98.63%	1.07%	0.23%	1.36%
R2L	4	132	2	1600	37	94.06%	90.14%	92.06%	4.47%	2.42%	7.44%
U2R	0	25	1	13	187	60.91%	82.74%	70.17%	27.04%	6.19%	11.06%

WSVM, Polynomial Kernel, $C = 500$, First weight scheme.											
	Accuracy	G-mean	FAR	Round 1							
	98.96%	94.70%	0.48%								
	Confusion matrix					Precision	Recall	F-score	FP	Miss Classification	Miss Detection
DoS	26651	26	8	8	0	99.84%	99.84%	99.84%	0.14%	0.06%	0.10%
Normal	37	38023	121	250	96	99.52%	98.69%	99.10%	99.52%	0.00%	1.31%
Probing	2	70	6962	5	0	98.11%	98.91%	98.51%	1.71%	0.10%	0.99%
R2L	1	65	3	1658	47	85.91%	93.46%	89.52%	12.95%	2.87%	3.66%
U2R	3	23	2	9	189	56.93%	83.63%	67.74%	28.92%	6.19%	10.18%
	Accuracy	G-mean	FAR	Round 2							
	98.83%	95.06%	0.53%								
	Confusion matrix					Precision	Recall	F-score	FP	Miss Classification	Miss Detection
DoS	26651	35	1	5	0	99.75%	99.85%	99.80%	0.21%	0.02%	0.13%
Normal	57	37936	142	274	117	99.47%	98.47%	98.97%	99.47%	0.00%	1.53%
Probing	6	85	6937	9	1	97.94%	98.56%	98.25%	2.00%	0.23%	1.21%
R2L	4	61	1	1674	35	84.85%	94.31%	89.33%	13.89%	2.25%	3.44%
U2R	0	21	2	11	192	55.65%	84.96%	67.25%	33.91%	5.75%	9.29%

4.2.2 WELM Experiments on NSL-KDD Dataset

The weighted ELM algorithm was used to perform several experiments on NSL-KDD dataset. Three parameters were adjusted to improve the performance of the algorithm. The first parameter was the number of the hidden layer neurons (L). The second parameter was the activation function of the hidden layer neurons. The third parameter was the value of C parameter. The objective of the set of experiments conducted using WEVM algorithm was finding the best L, C, activation function and weight scheme that will be used to build the ELM ID solution. All combinations of these parameters are listed and shown in Table 4.1. The list that included all experiments that were performed to achieve this goal are:

Test 5: The default weight scheme with WELM were used to perform several experiments on the NSL-KDD dataset. The used L values are [500 700 1000 1500 2000] and the used C values are $[1\ 10]$. All combinations of Various L and C values were used in performing the experiments. Table A.5 shows the algorithms results based on the activation functions and different L and C values. As it shown, the number of hidden neurons was an important parameter of improving the performance of the algorithm, the performance increased when L increased, the best results were obtained when $L = 2000$. The default C ($C\ equal\ 1$) was sufficient with the default weight scheme. When default value of C was used which equal to one, the overall accuracy of the WELM algorithm with sigmoidal activation function increased in ascending order when L increased. The overall accuracy values of various models were 98.06, 98.78, 98.35, 98.59 and 98.89 percent when these models were built using the following values of L which were 500, 700, 1000, 1500 and 2000 respectively. The same behavior appeared when the Gaussian RBF was used as an activation function of

hidden layer neurons, the overall accuracy values of these tests were 97.61, 97.97, 98.41, 98.67 and 98.78 percent when L values were 500, 700, 1000, 1500 and 2000 respectively. The sigmoidal activation function of hidden layer neurons was better than the Gaussian RBF. The two-fold cross-validation of WELM with the combination of sigmoidal activation function, $L = 2000$ and $C = 1$ has the best accuracy in case of using the default weight scheme. Table 4.4 shows the average of overall accuracy and the average of F-score for each class for the optimized WELM model.

Test 6: The third weight scheme was used with WELM algorithm to build multiple class classification solutions of ID problem on NSL-KDD dataset. All combinations of L and C as they appear in Table 4.1 were used to perform the experiments. Table A.6 shows the algorithms results based on the activation functions and different L and C values. As it shown, both the number of hidden neurons and C parameters represent important parameter of improving the performance of the algorithm, the performance increased when L and C increased, the best results were obtained when $L = 2000$ and $C = 10^5$. The overall accuracy with sigmoidal activation function increased in ascending order {L = 500: 97.07, L = 700: 95.66, L = 1000: 96.37, L = 1500: 97.63, L = 2000: ($C = 1$: 87.59, $C = 10^5$: 97.96, $C = 10^6$: 97.95)} percent when L and C increased. The same thing appeared with Gaussian RBF, the overall accuracy values were {L = 500: 95.67, L = 700: 96.43, L = 1000: 96.87, L = 1500: 97.39, L = 2000: ($C = 1$: 86.90, $C = 10^5$: 97.70, $C = 10^6$: 97.67%)} percent.

The WELM with Sigmoidal activation function and 2000 neurons for the first layer and with $C = 10^5$ was gained best result. Table 4.4 shows the average of overall accuracy and the average of F-score for each class for the optimized WELM models.

Table 4.4:The average results of the optimized models of Test 5 and Test 6.

	Default weight scheme, Sigmoidal, 2000 neurons and $C = 1$			Third weigh scheme, Sigmoidal, 2000 neurons and $C = 10^5$		
Accuracy	98.89%	G-mean	92.26%	97.96%	G-mean	94.59%
	Precision	Recall	F-score	Precision	Recall	F-score
DoS	99.74%	99.76%	99.75%	99.74%	99.69%	99.72%
Normal	99.16%	99.02%	99.09%	99.69%	96.86%	98.25%
Probing	97.84%	98.24%	98.04%	95.75%	98.81%	97.26%
R2L	91.05%	88.19%	89.59%	70.92%	94.39%	80.99%
U2R	58.94%	78.10%	67.17%	36.53%	84.07%	50.88%

4.2.3 Discussion of the Results

WSVM and WELM algorithms were applied on the NSL-KDD dataset, different weight schemes were employed and different parameters related to these algorithms were optimized, all these experiments appeared that both WSVM and WELM algorithm with default weight scheme has better performance with respect to the overall accuracy and the F-score for all classes. On the hand, the WSVM with default weight and its optimized parameters had better performance than the ELM with default weight and its optimized parameters. Although the suggested weight schemes with both algorithms 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 is associated with precision reduction. Table 4.5 shown the list of all tests that were performed on NSL-KDD dataset with the optimized parameters and results.

Table 4.5: The List of all Tests that were performed on NSL-KDD dataset with the optimized parameters.

Dataset	Algorithms	Test #	Kernels, Activation Function	C	Weight	The optimized Parameters, Number of Hidden Neurons			The overall accuracy	F-score				
										U2R	R2L	Probing	Normal	DoS
NSL-KDD	WSVM Two-Fold CV	Test 1	Sigmoidal kernel.	10	Default	$\gamma = 0.0244$			83.35%	53.54%	18.71%	49.30%	86.30%	90.04%
				1	2^{nd}				78.38%	29.10%	25.36%	57.49%	82.46%	90.47%
		Test 2	**Gaussian RBF kernel**	**10^3**	Default		$coef = 0$		99.19%	70.46%	92.50%	98.79%	99.30%	99.87%
				10^3	2^{nd}				99.00%	73.75%	89.86%	98.47%	99.10%	99.86%
		Test 3	Polynomial kernel	10^3	Default			$Degree = 3$	99.12%	69.27%	91.76%	98.65%	99.27%	99.83%
				500	2^{nd}				98.89%	67.50%	89.43%	98.38%	99.04%	99.82%
	WELM Two-Fold CV	Test 5	Sigmoidal	1	Default	2000			98.89%	67.17%	89.59%	98.04%	99.09%	99.75%
			Gaussian RB	10					98.78%	66.54%	88.49%	97.86%	99.02%	99.69%
		Test 6	Sigmoidal	10^5	3^{rd}				97.96%	50.88%	80.99%	97.26%	98.25%	99.72%
			Gaussian RB	10^5					97.70%	49.63%	78.28%	96.90%	98.07%	99.64%
	Test 4 WSVM Ten-Fold CV		Gaussian RBF kernel	100	2^{nd}	$\gamma = 0.0244$			98.98%					

Several works are published newly that intersect with this work of concern, three of closest ones are used to evaluate this work consistently which referred by [9] [23] [24]. As mentioned in the related works section, the works referred by [9] [23]built multi-level pattern recognition models. While CART algorithm was used in [9] to generate rule system to distinguish 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 [23] built a hybrid multi-level (SVM-ELM-SVM-SVM-SVM) model with selected records from the 10% KDD Cup99 using K-means clustering algorithm. The last work [24], proposed a framework that based on multiple kernels combination called MARK-ELM. The Table 4.7 presents our results compared with both works addressed by [23] [24]. As it is shown in the table, the proposed WSVM and ELM methods outperform the hybrid (SVM and EML) multi-level model in the overall accuracy and all sub-classes. The average accuracy for our twofold WSVM and twofold WELM were (99.19 and 98.89 percent sequentially) vs 95.75% for the hybrid multi-level model. On the other hand, the FAR average for our twofold WSVM and twofold WELM were (0.72 and 0.84 percent sequentially) vs 1.87% for the hybrid multi-level model. Final, detection rate for the minor classes which are R2L and U2R for our two-fold WSVM and twofold WELM were ((90.48, 80.97) percent and (88.19, 78.10) percent sequentially) vs (21, 93%, 31.39 %.) for the hybrid multi-level model.

The winner method of hybrid kernel MARK-ELM framework which is called F-Poly kernel is compared with the proposed weighted methods, even it got a better result in overall accuracy, G-mean, and normal class; our model can compete them in DOS and R2U classes and it does better in the Probing and U2R classes. Although our proposed algorithms have

higher FP degree than the poly-kernel MARK-ELM algorithm, they decrease the miss detection and the miss-classification of the DoS, Probing and U2R attacks, this means that the proposed methods make worse in classification the records to either normal or attacks but they succeeded in classifying the attacks to the correct attack.

On the other hand, Figure 4.2 shows the tested accuracy for our tenfold model and the multi-level models suggested in [9], both works used ten-fold cross-validation on the same data. To perform 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 U2R was excluded from the start in multi-level model's vs the superior accuracy was achieved by our model.

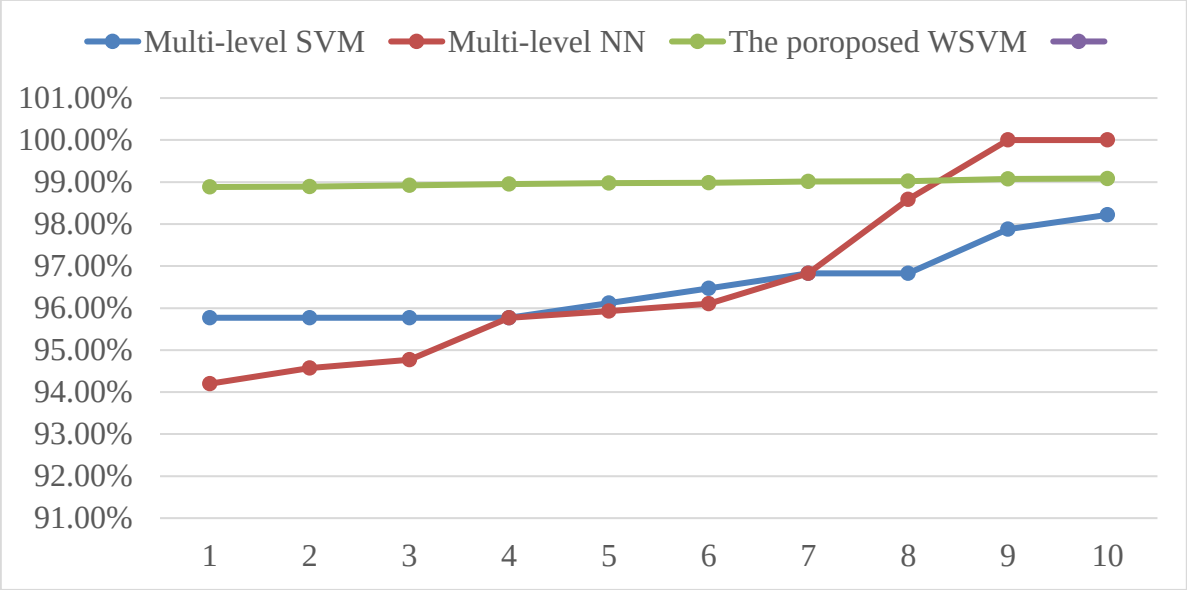


Figure 4.2: Comparison between The Testing result of our 10 Folds model and the Multi-level ID methods for abnormal network behaviors work [9].

Table 4.6: The average of overall accuracy for three models of the work assigned in **[9]**

	Multi-level SVM	Multi-level NN	NaïveBayes
Average of overall accuracy	96.54%	96.68%	89.02%

Based on the foregoing, our 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.

Table 4.7: Comparison among the optimized models in this thesis, Multi-level SVM & EML model and MARK-ELM F-Poly kernel set model.

Classes	The proposed methods on NSL-KDD dataset					MARK-ELM [24]			
	WSVM: 2-Fold cross-validation, Gaussian RBF, Default weight scheme, and $C = 10^3$			Accuracy	99.19%	F-Poly kernel set		Accuracy	99.77%
				G-mean	93.56%			G-mean	96.0%
				FAR	0.72%			FAR	
	Recall Average	F-Score Average	FP	Miss Classification	Miss Detection	Recall	FP	Miss Classification	Miss Detection
DoS	99.86%	99.87%	0.10%	0.01%	0.13%	99.96%	0.03%	0.03%	0.04%
Normal	99.34%	99.30%			0.66%	99.89%			0.15%
Probing	98.64%	98.79%	1.02%	0.11%	1.25%	97.42%	0.04%	2.52%	1.76%
R2L	90.48%	92.50%	4.74%	2.28%	7.24%	94.94%	0.07%	0.69%	5.05%
U2R	80.97%	70.46%	25.06%	3.32%	15.71%	62.87%	0.01%	13.89%	25.25%

Classes	The proposed methods on NSL-KDD dataset					Multi-level SVM & EML [23]			
	WELM: 2-Fold cross-validation, Default weight scheme, Sigmoidal, 2000 neurons and C=1			Accuracy	98.89%				
				G-mean	92.26%	Accuracy	95.75%	FAR	1.87%
				FAR	0.84%				
	Recall Average	F-Score Average	FP	Miss Classification	Miss Detection	Recall			
DoS	99.76%	99.75%	0.19%	0.09%	0.14%	99.54%			
Normal	99.02%	99.09%			0.98%	98.13%			
Probing	98.24%	98.04%	1.93%	0.23%	1.53%	87.22%			
R2L	88.19%	89.59%	6.60%	3.47%	8.34%	21.93%			
U2R	78.10%	67.17%	25.25%	8.63%	13.27%	31.39%			

4.3 UNB ISCX2012 Dataset Experiments

Evaluation inconsistency is one of the important issues related to ID solutions, which is one of the points that has 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 distinct attack scenario for each day as illustrated before. The number of normal and attack records in mentioned four days is shown in Table 2.2. A newly published thesis which referred by [25] that was used mainly to evaluate our work on this dataset. Table 4.8 shows the number of records that were used in the experiments of that previous thesis. The previous thesis 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 2.2, the number of records for the Botnet and Dos attacks are (37460, 3776 records in sequential order), which shows inconsistency in the number of records between the dataset and the mentioned number of records in the previous thesis. To overcome this problem, two set of experiments were performed. The primary experiments had the same numbers of records of the previous thesis for each class, they shown in Table 4.8: Number of records of UNB ISCX2012 dataset as they are 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 thesis.

To make the results of the experiments that 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.

4.3.1 WSVM Experiments on UNB ISCX2012 Dataset

The objective of the set of experiments conducted using WSVM algorithm was finding the best kernel, C and weight scheme that will be used to build the SVM ID solution. All combinations of these parameters are listed and shown in Table 4.1. **The list of the primary experiments that were performed to achieve this goal are:**

Test 1: the sigmoidal kernel was tested with the WSVM algorithm using the set of suggested parameters, its results were not competitive, so this kernel were excluded early.

Table 4.8: 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

Test 2: The default weight scheme was used with WSVM algorithm to perform some experiments on the UNB ISCX2012 dataset. All combinations of kernel and C parameters were tested and the results are shown in Table B.1. The upper part of the table included the result of WSVM algorithm with polynomial kernel while the lower part of the table included the result of WSVM algorithm with Gaussian RBF kernel, each column included the average of results of ten experiments that were repeated using different randomly selected subsets. The WSVM algorithm with Gaussian RBF kernel has best performance when $\mathrm{C} = 10$, the overall accuracy was 99.36 percent and the F-score for the SSH, Botnet,

DoS, L2L, Normal classes are 98.88, 6.67, 47.63, 95.14, 99.72 percent respectively. As well, the Polynomial kernel has better performance when $C = 1$. It was gained 99.32 percent in the overall accuracy besides the 98.80, 0.00, 46.92, 94.72, 99.70 percent as F-score for the SSH, Botnet, DoS, L2L, Normal classes respectively.

Test 3: The Second weight scheme was used with WSVM algorihtm to perform some experiments on the UNB ISCX2012 dataset. All combinations of kernel and C parameters were tested and the results are shown in Table B.2. The upper part of the table included the result of WSVM algorithm with polynomial kernel while the lower part of the table included the result of WSVM algorithm with Gaussian RBF kernel. The Gaussian RBF kernel had better performance when $C = 1$, while the Polynomial kernel had better performance when $C = 500$. The overall accuracy of the Gaussian RBF and Polynomial kernels were 97.30, 97.07 percent consecutively. Also, the F-scores for the SSH, Botnet, DoS, L2L and Normal classes respectively were 99.17, 1.33, 25.56, 84.08, 98.92 percent for the first kernel and they were 99.29, 0.29, 18.01, 54.50, 99.49 percent for the last kernel.

The result of the primary experiments on UNB ISCX2012 dataset using WSVM algorithm shown that the default weights improve the performance than the second weight scheme. This clearly shown in Table B.1 and Table B.2, the overall accuracy of WSVM with both Gaussian RBF and Polynomial kernels and with default weight scheme were 99.39, 99.32 percent respectively. On the other, the overall accuracy of WSVM with both kernels and with the second weight scheme were 97.30, 97.07 percent respectively. The default weight scheme with WSVM on the UNB ISCX2012 was selected to be used in the evaluation phase.

The secondary experiment that built on the general method of thesis was:

Test 4: In this test, the complete attacks records in the UNB ISCX2012 dataset were used with the same size of randomly selected normal subsets. The general method of our thesis was used which was one-leave-out the two-fold cross-validation. The reason for this test which not included in the evaluation phase was to perform more generalized experiments, it based on the correct number of the records in the dataset. Only WSVM algorithm with the Gaussian RBF kernel was used to perform the experiments of this test. All combinations of weight schemes and C parameters were tested and the results are shown in Table B.3. It clearly shown that the WSVM with Gaussian RBF and with default weight scheme had better performance when $C = 10^6$. The overall accuracy for this case was 99.22 percent and the F-score for SSH, Botnet, DoS, L2L and Normal classes were 99.94, 99.05, 98.31, 99.31 and 99.28 percent respectively. Also, the WSVM with Gaussian RBF and with the second weight scheme achieved best performance when $C = 10^6$, the overall accuracy for this case was 99.00 percent, and the F-score for SSH, Botnet, DoS, L2L and Normal classes were 99.76, 98.98, 96.05, 98.81 and 99.19 percent respectively.

4.3.2 WELM Experiments on UNB ISCX2012 Dataset

The weighted ELM algorithm was used to perform some experiments on UNB ISCX2012 dataset. Three parameters were evaluated to improve the performance of the algorithm. The first parameter was the number of the hidden layer neurons (L). The second parameter was the activation function of hidden layer neurons. The third parameter was the value of regularization parameter C. The objective of the set of experiments conducted using WEVM algorithm was finding the best L, C, activation function and weight scheme that will be used to build the ELM ID solution. All combinations of these parameters are listed and shown in Table 4.1.

The list of the primary experiments that were performed to achieve this goal are:

Test 5: In this test, the default weight scheme was used. Table B.4 shows that there is no benefit from increasing the number of L or increasing the value of C. Both activation functions had better performance when $\mathrm{C} = 1$ and with only 500 neurons. The WELM achieved better performance with the sigmoidal activation function. It achieved 99.32 percent as an overall accuracy with sigmoidal function vs 99.10 percent with Gaussian RB function.

Test 6: The third weight scheme was used with WELM on UNB ISCX2012 dataset. Table B.5 shown that algorithm performance was increased when L increased until certain value of L, then the performance was decreased. The WELM algorithm with sigmoidal activation function had the best performance when $\mathrm{L}, \mathrm{C} = (2000, 10^4)$ while the WELM algorithm with Gaussian RB activation function achieved the best performance when $\mathrm{L}, \mathrm{C} = (2000, 500)$. The WELM achieved better performance with the Gaussian RB activation

function. It achieved 96.29 percent as an overall accuracy with sigmoidal function vs 97.65 percent with Gaussian RB function.

Two weight schemes were tested with WELM algorithm, the default weight improveed both the overall accuracy and the F-score metrics values. So, it was selected in the final evaluation stage.

4.3.3 Discussion of the Results

WSVM and WELM algorithms were applied on 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 WSVM and 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 and third weight schemes with both algorithms 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 4.10 shown the list of all tests that were performed UNB ISCX2012 dataset with the optimized parameters and results.

The recent published work [25] which intersected with our work in concern was used to make consistent and equitable evaluation.

The results of applying SVM algorithm on the UNB ISCX2012 dataset in the previous thesis shown in Table 4.9. As shown, 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 insufficiently. 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 thesis were repeated ten times for each scenario, then the average of that rounds was used to evaluate the applied models.

Table 4.9: The results of SVM algorithm on the UNB ISCX2012 dataset as they appeared in the previous thesis [25].

Accuracy	99.11%	SVM-P Confusion Matrix Table.						
	Normal	L2L	SSH	Botnet	DoS	Precision	Recall	F-score
Normal	12224	27	19	9	6	99.56%	99.50%	99.53%
L2L	31	574	0	0	0	95.03%	94.88%	94.95%
SSH	0	0	463	0	0	95.86%	100.00%	97.89%
Botnet	0	0	0	2	0	18.18%	100.00%	30.77%
DoS	23	3	1	0	9	60.00%	25.00%	35.29%
Accuracy	94.88%	SVM-RBF Confusion Matrix Table.						
	Normal	L2L	SSH	Botnet	DoS	Precision	Recall	F-score
Normal	12164	37	84	0	0	95.72%	99.02%	0.973393
L2L	527	78	0	0	0	58.65%	12.89%	0.211382
SSH	0	0	463	0	0	84.18%	100.00%	0.914116
Botnet	0	0	2	0	0	0.00%	0.00%	0
DoS	17	18	1	0	0	0.00%	0.00%	0

The complete summary of the best-optimized results of the experiments which were included in the previous and this currents thesis are showed in Table 4.11. As shown, the optimized Gaussian RBF with WSVM in our work was better than the polynomial SVM in the previous work in the overall accuracy in addition to all F-score values except the Botnet F-Score. The better F-score of the botnet that achieved by the previous thesis experiments on a random selected subset does not reflect better performance on that set because the weakness of the experiments.

Table 4.10: The List of all Tests that were performed on UNB ISCX dataset with the optimized parameters.

Dataset	Algorithms		Test #	Kernels, Activation Function	C	Weight	The optimized Parameters, Number of Hidden Neurons			The overall accuracy	F-score				
											Normal	L2L	DoS	Botnet	SSH
UNB ISCX2012	Primary Experiments	WSVM	Test 2	**Gaussian RBF kernel**	**10**	Default	$\gamma = 0.0909$	$coef = 0$		99.36%	99.72%	95.14%	47.63%	6.67%	98.88%
				Polynomial kernel	1	Default			$Degree = 3$	99.32%	99.70%	94.72%	46.92%	0.00%	98.80%
			Test 3	Gaussian RBF kernel	1	2^{nd}				97.30%	98.92%	84.08%	25.56%	1.33%	99.17%
				Polynomial kernel	500	2^{nd}			$Degree = 3$	97.07%	99.46%	64.53%	18.74%	0.52%	99.23%
		WELM	Test 5	Sigmoidal	1	Default	500			99.32%	99.71%	94.43%	39.35%	0.00%	98.94%
				Gaussian RB	1		500			99.10%	99.65%	93.18%	35.33%	0.00%	98.55%
			Test 6	Sigmoidal	10^4	3^{rd}	2000			96.29%	98.90%	63.83%	15.29%	1.84%	98.93%
				Gaussian RB	500		2000			97.65%	99.17%	81.16%	23.55%	1.22%	98.95%
	Secondary Experiments	WSVM Two-Fold CV	Test 4	Gaussian RBF kernel	**10^6**	Default	$\gamma = 0.0909$			99.22%	99.28%	99.31%	98.31%	99.05%	99.94%
						2^{nd}				99.00%	99.19%	98.81%	96.05%	98.98%	99.76%

Table 4.11: Comparison among the primary optimized WSVM and WELM models in this thesis and the optimized WSVM models in the previous thesis.

	The result of previous thesis [25].						The proposed methods		
	SVM-P			SVM-RBF			Gaussian RBF Kernel with WSVM		
	Accuracy	99.11%		Accuracy	0.94877		Accuracy	**99.36%**	
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
SSH	95.86%	100.00%	97.89%	84.18%	100.00%	91.41%	**98.92%**	**98.88%**	**98.88%**
Botnet	**18.18%**	**100.00%**	**30.77%**	0.00%	0.00%	0.00%	10.00%	5.00%	6.67%
DoS	**60.00%**	25.00%	35.29%	0.00%	0.00%	0.00%	52.62%	**44.17%**	**47.63%**
L2L	95.03%	**94.88%**	94.95%	58.65%	12.89%	21.14%	**96.27%**	94.07%	**95.14%**
Normal	99.56%	99.50%	99.53%	95.72%	99.02%	97.34%	**99.63%**	**99.81%**	**99.72%**
	The proposed methods								
	Polynomial Kernel with WSVM			WELM with Sigmoidal Function			WELM with Polynomial Function		
	Accuracy	99.32%		Accuracy	99.32%		Accuracy	99.10%	
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
SSH	98.76%	98.88%	98.80%	98.95%	98.94%	98.94%	98.61%	98.51%	98.55%
Botnet	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
DoS	53.90%	42.78%	46.92%	63.22%	31.39%	39.35%	36.50%	37.78%	35.33%
L2L	96.03%	93.45%	94.72%	94.98%	93.90%	94.43%	94.47%	91.98%	93.18%
Normal	99.59%	99.80%	99.70%	99.60%	99.82%	99.71%	99.63%	99.67%	99.65%

Chapter 5

Conclusion and Future Works

Conclusion and Future Works

The development of IDSs 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 thesis, a method was proposed for detecting the intrusions by ML tools that consolidated stratified sampling and different cost function schemes with both SVM and ELM methods 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 WSVM is more effective than WELM algorithm, although that the WELM is a good competitor. The experiments that performed using both algorithms were achieved competitive results of both overall accuracy and F-score per-class performance scale on both datasets. The best algorithm in this study that applied on the NSL-KDD dataset was WSVM with Gaussian kernel. Using the default weight scheme and with $C = 700$, the overall accuracy of this algorithm is 99.19 percent and the F-score for the DoS, Normal, Probing, R2L and U2R classes were 99.87, 99.30 98.79, 92.50 and 70.46 percent respectively. As well, the WSVM with Gaussian RBF kernel and default weight scheme was the best applied algorithm on the UNB ISCX2012 dataset, but $C = 10$ in this experiment. The overall accuracy in this experiment was 99.36 percent while the F-score

for the SSH, Botnet, DoS, L2L, Normal classes were 98.88, 6.67, 47.63, 95.14, 99.72 percentage respectively.

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 points through this effort. In the future work, we will start using 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.

Bibliography

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

[2] R. Walters, "Heritage," The Heritage Foundation, 27 October 2014. [Online]. Available: http://www.heritage.org/defense/report/cyber-attacks-us-companies-2014. [Accessed 17 2 2017].

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

[4] 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.

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

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

[7] 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.

[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] 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.

[10] C. Thomas, "Improving intrusion detection for imbalanced network traffic," *Security and Communication Networks,* vol. 6, pp. 309-324, 2013.

[11] 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.

[12] 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.

[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] 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.

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

[18] R. Alejo, J. M. Sotoca and G. A. Casañ, "An empirical study for the multi-class imbalance problem with neural networks," in *Iberoamerican Congress on Pattern Recognition*, 2008.

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

[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] 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.

[23] 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.

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

[27] I. Homoliak, D. Breitenbacher and P. Hanacek, "Convergence Optimization of Backpropagation Artificial Neural Network Used for Dichotomous Classification of Intrusion Detection Dataset," *JCP*, vol. 12, no. 2, pp. 143--155, 2017.

[28] "Wikipedia," Wikipedia, the free encyclopedia, July 2012. [Online]. Available: https://en.wikipedia.org/wiki/Stratified_sampling. [Accessed 9 2 2017].

[29] 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.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning,* vol. 20, pp. 273-297, 1995.

[31] H. Daumé III, A course in Machine Learning, vol. 5, ciml.info, 2012, p. 69.

[32] S. L. Phung, A. Bouzerdoum and G. H. Nguyen, "Learning Pattern Classification Tasks with Imbalanced Data Sets," in *Pattern Recognition*, InTech, 2009.

[33] S. Wang, L. L. Minku and X. Yao, "Dealing with Multiple Classes in Online Class Imbalance Learning," in *Proc. 25th Int. Joint Conf. Artificial Intelligence, IJCAI/AAAI Press*, 2016.

Appendix

This part includes the result of all experiments that performed on NSL-KDD and UNB ISCX 2012 datasets to optimize some parameters of the WSVM and WELM methods.

Appendix A

This part includes the result of all experiments that performed on NSL-KDD dataset to optimize some parameters of the WSVM and WELM methods.

Table A.1:The complete result of Test 1 experiments.

WSVM, Sigmoidal Kernel, Default weight scheme.							
C	1	10	50	100	300	700	1000
Accuracy	82.95%	**83.35%**	82.79%	82.76%	82.73%	82.71%	82.73%
G-mean	0.00%	49.55%	**50.24%**	48.65%	44.30%	42.52%	42.05%
	F-score						
DoS	90.92%	90.04%	90.76%	**90.80%**	90.74%	90.74%	90.76%
Normal	86.67%	86.30%	**86.87%**	86.72%	86.69%	86.67%	86.67%
Probing	**45.74%**	49.30%	44.26%	44.22%	44.29%	44.28%	44.31%
R2L	17.79%	18.71%	18.35%	18.05%	18.79%	18.71%	**18.86%**
U2R	0.00%	53.54%	**56.53%**	49.83%	37.69%	32.88%	30.91%
WSVM, Sigmoidal Kernel, First weight scheme.							
C	1	10	50	100	300	700	1000
Accuracy	**78.38%**	78.19%	77.82%	77.61%	77.81%	77.81%	77.81%
G-mean	**74.86%**	73.59%	73.04%	72.48%	72.94%	72.94%	72.94%
	F-score						
DoS	**90.47%**	90.29%	88.79%	88.73%	88.78%	88.78%	88.78%
Normal	82.46%	82.36%	82.85%	82.56%	**82.86%**	82.85%	82.85%
Probing	**57.49%**	57.37%	55.92%	55.88%	55.92%	55.92%	55.92%
R2L	**25.36%**	24.06%	25.21%	24.77%	25.19%	25.19%	25.19%
U2R	**29.10%**	28.66%	26.50%	27.64%	26.37%	26.37%	26.37%

Table A.2: The complete result of Test 2 experiments.

WSVM, Gaussian RBF Kernel, Default weight scheme.						
C	1	100	300	500	700	1000
Accuracy	98.25%	99.16%	99.17%	99.18%	**99.19%**	**99.19%**
G-mean	52.60%	93.15%	93.56%	93.64%	**93.59%**	**93.56%**
	F-score					
DoS	99.59%	99.86%	99.85%	99.86%	99.86%	**99.87%**
Normal	98.43%	99.28%	99.29%	**99.30%**	**99.30%**	**99.30%**
Probing	97.10%	98.67%	98.74%	98.74%	98.78%	**98.79%**
R2L	83.99%	92.23%	92.37%	92.45%	92.48%	**92.50%**
U2R	10.00%	68.50%	69.31%	69.50%	69.84%	**70.46%**
WSVM, Gaussian RBF Kernel, First weight scheme.						
C	1	100	300	500	700	1000
Accuracy	98.15%	98.92%	98.95%	98.98%	98.98%	**99.00%**
G-mean	94.42%	94.38%	**94.40%**	94.19%	94.03%	94.01%
	F-score					
DoS	99.75%	99.86%	**99.87%**	**99.87%**	**99.87%**	99.86%
Normal	98.38%	99.05%	99.07%	99.09%	99.09%	**99.10%**
Probing	97.07%	**98.48%**	**98.48%**	**98.48%**	98.47%	98.47%
R2L	82.99%	89.48%	89.62%	89.76%	89.76%	**89.86%**
U2R	57.73%	66.17%	68.71%	70.72%	72.54%	**73.75%**

Table A.3: The complete result of Test 3 experiments.

WSVM, Polynomial Kernel, Default weight scheme.						
C	1	100	300	500	700	1000
Accuracy	97.65%	99.04%	99.10%	99.11%	**99.12%**	**99.12%**
G-mean	54.55%	93.00%	93.31%	93.65%	93.75%	**93.87%**
	F-score					
DoS	99.34%	99.80%	99.83%	**99.84%**	99.83%	99.83%
Normal	97.81%	99.20%	99.25%	99.26%	99.26%	**99.27%**
Probing	97.06%	98.52%	98.58%	98.59%	98.64%	**98.65%**
R2L	72.95%	90.90%	91.53%	91.65%	91.64%	**91.76%**
U2R	14.79%	67.97%	68.48%	68.71%	69.09%	**69.27%**

WSVM, Polynomial Kernel, First weight scheme.						
C	1	100	300	500	700	1000
Accuracy	97.66%	98.78%	98.87%	**98.89%**	**98.89%**	**98.89%**
G-mean	94.38%	94.84%	**94.94%**	94.88%	94.93%	94.78%
	F-score					
DoS	99.45%	99.81%	99.82%	**99.82%**	**99.82%**	99.81%
Normal	98.01%	98.96%	99.03%	**99.04%**	99.03%	99.02%
Probing	96.96%	98.23%	98.32%	**98.38%**	98.36%	98.35%
R2L	79.24%	88.17%	89.27%	**89.43%**	89.33%	89.41%
U2R	50.94%	63.02%	65.99%	67.50%	68.22%	**68.24%**

Table A.4: The complete result of Test 4 experiments.

Ten-fold WSVM, Gaussian RBF Kernel,$C = 100$, First weight scheme									
Accuracy	99.08%	G-mean	97.17%	Round 1					
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5329	8	1	0	1	99.94%	99.81%	99.88%	0.04%
Normal	2	7609	33	46	16	99.72%	98.74%	99.23%	
Probing	0	6	1400	1	0	97.63%	99.50%	98.56%	2.30%
R2L	1	5	0	336	13	87.50%	94.65%	90.93%	11.98%
U2R	0	2	0	1	42	58.33%	93.33%	71.79%	22.22%
Accuracy	98.95%	G-mean	93.85%	Round 2					
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5331	4	4	0	0	99.85%	99.85%	99.85%	0.13%
Normal	7	7603	22	52	21	99.58%	98.68%	99.13%	
Probing	1	12	1395	0	0	98.17%	99.08%	98.62%	1.55%
R2L	0	10	0	331	14	85.75%	93.24%	89.34%	13.47%
U2R	0	6	0	3	36	50.70%	80.00%	62.07%	29.58%
Accuracy	99.02%	G-mean	96.10%	Round 3					
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5335	4	0	0	0	99.91%	99.93%	99.92%	0.07%
Normal	4	7605	27	42	27	99.54%	98.70%	99.12%	
Probing	0	16	1391	1	0	98.03%	98.79%	98.41%	1.90%
R2L	1	12	0	335	6	88.39%	94.63%	91.41%	11.08%
U2R	0	3	1	1	40	54.79%	88.89%	67.80%	36.99%

Accuracy	98.98%	G-mean	93.78%		Round 4				
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5334	5	0	0	0	99.89%	99.91%	99.90%	0.11%
Normal	6	7597	36	42	24	99.63%	98.60%	99.11%	
Probing	0	11	1396	1	0	97.42%	99.15%	98.28%	2.51%
R2L	0	5	0	339	11	88.28%	95.49%	91.75%	10.94%
U2R	0	7	1	2	35	50.00%	77.78%	60.87%	34.29%
Accuracy	98.89%	G-mean	92.77%		Round 5				
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5334	3	0	1	0	99.93%	99.93%	99.93%	0.02%
Normal	1	7592	34	56	22	99.53%	98.53%	99.03%	
Probing	0	11	1395	1	1	97.62%	99.08%	98.34%	2.38%
R2L	3	14	0	331	7	84.44%	93.24%	88.62%	14.29%
U2R	0	8	0	3	34	53.13%	75.56%	62.39%	34.38%
Accuracy	98.88%	G-mean	96.92%		Round 6				
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5327	9	2	0	0	99.92%	99.79%	99.86%	0.04%
Normal	2	7578	28	68	30	99.63%	98.34%	98.98%	
Probing	1	9	1398	0	0	97.90%	99.29%	98.59%	1.96%
R2L	1	7	0	342	5	83.21%	96.34%	89.30%	16.55%
U2R	0	3	0	1	41	53.95%	91.11%	67.77%	39.47%
Accuracy	99.02%	G-mean	95.70%		Round 7				
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5333	3	2	1	0	99.83%	99.89%	99.86%	0.13%
Normal	7	7599	17	60	22	99.69%	98.62%	99.15%	
Probing	0	9	1398	0	0	98.66%	99.36%	99.01%	1.20%
R2L	1	9	0	336	9	84.21%	94.65%	89.12%	15.04%
U2R	1	3	0	2	39	55.71%	86.67%	67.83%	31.43%
Accuracy	98.98%	G-mean	96.80%		Round 8				
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5333	4	0	0	1	99.93%	99.91%	99.92%	0.06%
Normal	3	7588	29	60	25	99.72%	98.48%	99.10%	
Probing	1	7	1397	1	1	97.97%	99.29%	98.62%	2.03%
R2L	0	6	0	339	10	84.75%	95.49%	89.80%	15.00%
U2R	0	4	0	0	41	52.56%	91.11%	66.67%	32.05%

Accuracy	99.08%	G-mean	96.93%	Round 9				
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5330	8	0	0	0	99.85%	99.85%	99.85%	0.15%
Normal	8	7601	28	49	19	99.69%	98.65%	99.17%	
Probing	0	3	1403	0	2	98.04%	99.64%	98.84%	1.96%
R2L	0	9	0	339	7	87.37%	95.49%	91.25%	12.63%
U2R	0	4	0	0	42	60.00%	91.30%	72.41%	27.14%
Accuracy	98.92%	G-mean	95.39%	Round 10				
	Confusion matrix					Precision	Recall	F-score	FP
DoS	5329	9	0	0	0	99.89%	99.83%	99.86%	0.07%
Normal	4	7589	36	56	21	99.57%	98.48%	99.02%	
Probing	1	9	1397	1	0	97.49%	99.22%	98.35%	2.51%
R2L	1	8	0	339	7	85.61%	95.49%	90.28%	14.14%
U2R	0	7	0	0	39	58.21%	84.78%	69.03%	31.34%

Table A.5: The complete result of Test 5 experiments.

WELM, Sigmoidal Activation Function, Default weight scheme.										
L	500		700		1000		1500		2000	
C	1	10	1	10	1	10	1	10	1	10
Accuracy	98.04%	98.06%	98.34%	98.35%	98.56%	98.59%	98.78%	98.78%	98.89%	98.89%
G-mean	69.24%	69.74%	89.57%	89.79%	91.04%	91.15%	92.12%	92.22%	92.26%	92.54%
	F-score									
DoS	99.49%	99.49%	99.57%	99.57%	99.67%	99.68%	99.75%	99.74%	99.75%	99.74%
Normal	98.31%	98.32%	98.61%	98.62%	98.79%	98.81%	98.98%	98.98%	99.09%	99.10%
Probing	96.45%	96.47%	96.87%	96.86%	97.35%	97.42%	97.98%	97.93%	98.04%	98.01%
R2L	80.63%	80.88%	83.89%	84.20%	86.05%	86.20%	87.73%	87.85%	89.59%	89.78%
U2R	38.12%	38.99%	65.96%	66.28%	67.04%	66.98%	67.42%	67.42%	67.17%	67.36%
WELM, Gaussian RB Activation Function, Default weight scheme.										
L	500		700		1000		1500		2000	
C	1	10	1	10	1	10	1	10	1	10
Accuracy	97.60%	97.61%	97.91%	97.97%	98.34%	98.41%	98.64%	98.67%	98.72%	98.78%
G-mean	54.00%	53.75%	68.10%	68.49%	89.14%	89.80%	91.43%	91.68%	91.89%	92.37%
	F-score									
DoS	99.39%	99.40%	99.54%	99.56%	99.63%	99.63%	99.71%	99.69%	99.71%	99.69%
Normal	97.81%	97.82%	98.16%	98.22%	98.60%	98.69%	98.86%	98.90%	98.95%	99.02%
Probing	96.59%	96.60%	96.86%	96.83%	97.09%	97.10%	97.58%	97.63%	97.63%	97.86%
R2L	71.97%	72.00%	74.04%	75.69%	81.36%	83.28%	86.13%	87.20%	87.89%	88.49%
U2R	14.38%	14.04%	39.55%	39.08%	67.37%	66.73%	67.49%	66.47%	66.67%	66.54%

Table A.6: The complete result of Test 6 experiments.

WELM, Sigmoidal Activation Function, Third weight scheme.												
L	**500**			700		1000		1500		2000		
C	1	10^5	10^6	10^5	10^6	10^5	10^6	10^5	10^6	1	10^5	10^6
Accuracy	83.81%	95.66%	95.66%	96.37%	96.37%	97.07%	97.07%	97.63%	97.63%	87.59%	97.96%	97.95%
G-mean	87.90%	94.40%	94.40%	94.83%	94.84%	94.67%	94.71%	94.62%	94.54%	90.56%	94.59%	94.51%
	F-score											
DoS	95.68%	99.23%	99.23%	99.42%	99.42%	99.53%	99.54%	99.70%	99.70%	96.88%	99.72%	99.71%
Normal	84.35%	96.14%	96.14%	96.78%	96.78%	97.44%	97.44%	97.94%	97.94%	88.26%	98.25%	98.26%
Probing	77.11%	94.76%	94.76%	95.28%	95.31%	95.95%	95.95%	96.70%	96.70%	82.90%	97.26%	97.26%
R2L	35.65%	65.27%	65.28%	70.55%	70.58%	74.42%	74.38%	78.28%	78.24%	40.40%	80.99%	80.84%
U2R	21.68%	39.29%	39.33%	40.54%	40.58%	47.05%	46.97%	49.41%	49.00%	30.21%	50.88%	49.92%
WELM, Gaussian RB Activation Function, Third weight scheme.												
L	**500**			700		1000		1500		2000		
C	1	10^5	10^6	10^5	10^6	10^5	10^6	10^5	10^6	1	10^5	10^6
Accuracy	82.69%	95.67%	95.68%	96.43%	96.42%	96.87%	96.88%	97.39%	97.40%	86.90%	97.70%	97.67%
G-mean	86.23%	93.91%	93.86%	94.39%	94.37%	93.90%	93.91%	93.87%	93.90%	90.21%	94.49%	94.34%
	F-score											
DoS	94.84%	99.26%	99.26%	99.40%	99.40%	99.51%	99.52%	99.58%	99.58%	97.12%	99.64%	99.65%
Normal	83.08%	96.12%	96.13%	96.89%	96.89%	97.30%	97.30%	97.77%	97.78%	87.29%	98.07%	98.05%
Probing	82.57%	94.47%	94.48%	95.28%	95.29%	95.69%	95.73%	96.62%	96.74%	85.61%	96.90%	96.93%
R2L	31.57%	65.66%	65.75%	70.66%	70.62%	73.03%	73.06%	75.66%	75.67%	36.81%	78.28%	78.38%
U2R	16.17%	39.35%	39.20%	40.00%	39.98%	43.25%	43.28%	47.83%	46.64%	26.75%	49.63%	46.65%

Appendix B

This part includes the result of all experiments that performed on UNB ISCX 2012 dataset to optimize some parameters of the WSVM and WELM methods.

Table B.1: The complete results of Test 2 experiments.

WSVM, Polynomial Kernel, Default weight scheme.										
Accuracy	**99.32%**	99.28%	99.22%	99.21%	98.99%	98.83%	98.65%	98.21%	97.60%	96.92%
G-mean	0.00%	9.02%	9.03%	17.67%	17.65%	26.00%	25.84%	**34.78%**	25.82%	25.72%
	F-score									
C values	1	10	50	100	300	500	1000	10000	100000	1000000
SSH	98.80%	99.02%	99.02%	99.02%	99.02%	99.02%	**99.03%**	**99.03%**	**99.03%**	99.02%
Botnet	0.00%	3.33%	3.33%	**4.31%**	3.86%	4.26%	4.12%	2.97%	1.74%	1.22%
DoS	**46.92%**	44.91%	44.28%	44.40%	41.87%	40.77%	37.99%	33.29%	29.60%	29.60%
L2L	**94.72%**	94.27%	93.96%	93.68%	92.65%	90.72%	88.70%	82.79%	72.16%	67.52%
Normal	**99.70%**	**99.70%**	99.67%	99.67%	99.57%	99.53%	99.51%	99.40%	99.07%	98.68%
WSVM, Gaussian RBF Kernel, Default weight scheme.										
Accuracy	99.21%	**99.36%**	99.35%	99.30%	99.25%	99.16%	99.11%	98.56%	98.05%	97.68%
G-mean	0.00%	7.85%	9.11%	17.77%	17.77%	17.77%	26.05%	**31.49%**	22.16%	22.14%
	F-score									
C values	1	10	50	100	300	500	1000	10000	100000	1000000
SSH	98.62%	**98.88%**	**98.88%**	**98.88%**	**98.88%**	**98.88%**	**98.88%**	**98.88%**	**98.88%**	**98.88%**
Botnet	0.00%	6.67%	10.00%	**10.67%**	10.48%	10.39%	10.72%	4.52%	1.66%	0.72%
DoS	22.00%	**47.63%**	45.76%	44.85%	44.63%	44.55%	43.56%	36.97%	35.11%	35.11%
L2L	93.81%	**95.14%**	**95.11%**	94.98%	94.60%	94.56%	94.42%	86.70%	77.16%	72.58%
Normal	99.63%	99.72%	**99.73%**	99.71%	99.69%	99.65%	99.62%	99.56%	99.28%	99.07%

Table B.2: The complete result of Test 3 experiments.

WSVM, Polynomial Kernel, Second weight scheme.										
Accuracy	96.96%	96.80%	96.87%	96.93%	97.06%	**97.07%**	96.93%	96.68%	96.22%	95.99%
G-mean	**37.36%**	20.79%	14.24%	14.23%	0.00%	0.00%	0.00%	6.14%	17.55%	12.30%
	F-score									
C values	1	10	50	100	300	500	1000	10000	100000	1000000
SSH	99.26%	**99.29%**	**99.29%**	**99.29%**	**99.29%**	**99.29%**	**99.29%**	99.27%	99.27%	99.27%
Botnet	0.99%	0.89%	0.62%	0.66%	0.28%	0.29%	0.36%	0.51%	**1.24%**	0.24%
DoS	**22.47%**	16.63%	17.71%	18.03%	18.00%	18.01%	17.38%	18.38%	18.33%	18.36%
L2L	**70.57%**	56.47%	55.66%	55.87%	54.61%	54.50%	52.13%	50.70%	50.03%	49.11%
Normal	99.11%	99.33%	99.38%	99.41%	**99.49%**	**99.49%**	99.42%	99.28%	99.02%	98.89%
WSVM, Gaussian RBF Kernel, Second weight scheme.										
Accuracy	97.30%	96.86%	97.00%	97.21%	**97.32%**	97.24%	97.27%	97.00%	96.60%	96.63%
G-mean	**45.15%**	27.43%	21.04%	6.56%	6.32%	12.38%	12.56%	33.13%	34.41%	21.22%
	F-score									
C values	1	10	50	100	300	500	1000	10000	100000	1000000
SSH	99.17%	99.22%	**99.23%**	**99.23%**	**99.23%**	**99.23%**	**99.23%**	**99.23%**	**99.23%**	**99.23%**
Botnet	**1.33%**	0.92%	0.69%	0.28%	0.28%	0.52%	0.58%	1.14%	1.62%	0.41%
DoS	**25.56%**	17.32%	18.69%	19.78%	19.35%	18.74%	18.71%	19.20%	19.20%	19.20%
L2L	**84.08%**	61.01%	60.94%	64.22%	65.33%	64.53%	64.59%	63.20%	58.55%	57.93%
Normal	98.92%	99.37%	99.41%	99.45%	**99.48%**	99.46%	99.47%	99.31%	99.09%	99.10%

Table B.3: The complete result of Test 4 experiments.

	WSVM, Gaussian RBF Kernel, Default weight scheme.									
Accuracy	97.84%	97.94%	97.99%	98.00%	98.06%	98.91%	99.06%	99.15%	99.15%	**99.22%**
G-mean	88.60%	89.51%	89.52%	89.53%	90.07%	97.43%	98.58%	99.03%	99.06%	**99.09%**
	F-score									
C values	1	10	50	100	300	500	1000	10000	100000	1000000
SSH	99.69%	99.72%	99.69%	99.71%	99.77%	99.78%	99.79%	99.79%	99.93%	**99.94%**
Botnet	98.77%	98.79%	98.82%	98.84%	98.85%	98.86%	98.86%	98.88%	98.90%	**99.05%**
DoS	71.17%	73.29%	73.48%	73.59%	74.80%	94.03%	96.95%	98.25%	**98.31%**	**98.31%**
L2L	95.84%	95.95%	96.00%	96.01%	96.13%	98.72%	99.21%	99.42%	**99.39%**	99.31%
Normal	99.01%	99.12%	99.15%	99.17%	99.19%	99.20%	99.20%	99.22%	99.21%	**99.28%**
	WSVM, Gaussian RBF Kernel, Second weight scheme.									
Accuracy	97.44%	97.17%	98.65%	98.74%	98.97%	98.98%	98.98%	98.99%	98.99%	**99.00%**
G-mean	90.52%	96.67%	98.72%	98.83%	99.11%	**99.12%**	**99.12%**	99.07%	99.08%	99.08%
	F-score									
C values	1	10	50	100	300	500	1000	10000	100000	1000000
SSH	99.73%	99.73%	99.75%	99.77%	99.75%	99.73%	**99.80%**	99.77%	99.79%	99.76%
Botnet	98.79%	98.83%	98.85%	98.85%	98.86%	98.86%	98.87%	98.86%	98.89%	**98.98%**
DoS	68.34%	73.54%	90.91%	92.15%	95.60%	95.75%	95.79%	96.01%	95.98%	**96.05%**
L2L	94.32%	92.77%	97.90%	98.16%	98.90%	98.94%	98.93%	**98.95%**	**98.95%**	98.81%
Normal	99.08%	99.14%	99.17%	99.18%	**99.20%**	**99.20%**	**99.20%**	99.19%	99.18%	99.19%

Table B.4: The complete result of Test 5 experiments.

WELM, Sigmoidal Activation Function, Default weight scheme.												
L	**500**			700		1000		1500		2000		
C	1	10	100	1	10	1	10	1	10	1	10	100
Accuracy	99.32%	99.27%	99.10%	99.28%	99.23%	99.25%	99.18%	99.28%	99.19%	99.27%	99.21%	98.74%
G-mean	0.00%	0.00%	8.36%	0.00%	0.00%	0.00%	8.39%	0.00%	8.39%	0.00%	8.43%	23.63%
	F-score											
SSH	98.94%	98.87%	98.84%	98.91%	98.81%	98.89%	98.92%	98.89%	98.91%	98.91%	99.15%	99.15%
Botnet	0.00%	0.00%	0.26%	0.00%	0.00%	0.00%	0.36%	0.00%	0.39%	0.00%	0.34%	0.69%
DoS	39.35%	43.59%	39.96%	43.15%	42.88%	43.92%	42.80%	43.46%	40.05%	43.39%	41.46%	32.37%
L2L	94.43%	94.24%	93.66%	94.15%	93.88%	94.11%	94.06%	94.38%	94.15%	94.34%	94.15%	92.08%
Normal	99.71%	99.71%	99.64%	99.68%	99.68%	99.68%	99.66%	99.70%	99.67%	99.70%	99.69%	99.51%
WELM, Gaussian RBF Activation Function, Default weight scheme.												
L	**500**			700		1000		1500		2000		
C	1	10	100	1	10	1	10	1	10	1	10	100
Accuracy	99.10%	98.56%	97.22%	99.07%	98.30%	99.05%	98.34%	98.96%	98.16%	98.93%	98.22%	96.91%
G-mean	0.00%	15.37%	22.44%	0.00%	14.81%	0.00%	14.95%	6.73%	14.94%	6.73%	15.05%	15.13%
	F-score											
SSH	98.55%	99.09%	98.99%	99.08%	99.16%	98.71%	98.85%	98.75%	98.78%	98.81%	99.06%	99.15%
Botnet	0.00%	0.51%	0.99%	0.00%	0.44%	0.00%	0.34%	0.21%	0.37%	0.20%	0.36%	0.31%
DoS	35.33%	28.50%	19.93%	37.12%	26.72%	37.32%	29.62%	37.55%	30.43%	35.08%	27.32%	19.44%
L2L	93.18%	91.49%	83.77%	92.44%	87.87%	92.83%	88.14%	92.75%	86.93%	92.43%	87.60%	78.10%
Normal	99.65%	99.40%	98.83%	99.64%	99.36%	99.63%	99.39%	99.57%	99.30%	99.56%	99.32%	98.87%

Table B.5: The complete result of Test 6 experiments.

WELM, Sigmoidal Activation Function, Third weight scheme.															
L	**500**		700			1000			1500				2000		
C	10000	100000	100	300	500	10	100	300	1000	10000	100000	1000000	1000	10000	100000
Accuracy	95.97%	95.36%	95.93%	95.99%	95.80%	94.32%	96.14%	95.87%	95.90%	96.26%	95.81%	94.22%	96.03%	96.29%	96.24%
G-mean	47.00%	48.50%	61.30%	56.65%	56.47%	77.76%	56.90%	56.59%	58.11%	49.79%	42.84%	31.04%	54.37%	49.87%	44.84%
	F-score														
SSH	98.92%	98.56%	98.72%	98.84%	98.73%	95.24%	98.82%	98.98%	99.01%	99.03%	98.36%	98.35%	98.97%	98.93%	99.00%
Botnet	1.56%	2.07%	1.62%	1.71%	1.62%	1.68%	1.62%	1.62%	1.63%	1.68%	1.39%	0.97%	1.76%	1.84%	1.42%
DoS	15.58%	11.54%	22.16%	17.01%	15.15%	21.26%	22.27%	15.81%	17.36%	14.99%	13.05%	13.13%	17.13%	15.29%	15.54%
L2L	66.03%	61.79%	80.94%	79.30%	75.79%	77.64%	82.42%	76.03%	68.94%	65.13%	67.43%	59.24%	69.33%	63.83%	70.31%
Normal	98.65%	98.37%	98.06%	98.23%	98.27%	97.09%	98.19%	98.31%	98.54%	98.83%	98.52%	97.64%	98.62%	98.90%	98.72%
WELM, Gaussian RBF Activation Function, Third weight scheme.															
L	**500**		700			1000			1500				2000		
C	100	300	10	100	300	500	1000	10000	100	300	500	1000	300	500	1000
Accuracy	97.18%	97.17%	96.56%	97.24%	97.01%	97.25%	97.47%	96.58%	97.11%	97.38%	97.61%	97.55%	97.60%	97.65%	97.54%
G-mean	49.08%	48.21%	69.67%	49.37%	49.45%	48.16%	33.71%	29.14%	48.83%	40.89%	34.44%	33.18%	41.06%	32.55%	31.40%
	F-score														
SSH	98.75%	98.67%	98.83%	99.14%	99.14%	98.96%	98.87%	99.01%	98.98%	98.97%	98.97%	98.90%	98.93%	98.95%	98.97%
Botnet	1.42%	1.85%	1.84%	1.36%	1.48%	1.63%	1.19%	0.80%	1.45%	1.37%	1.22%	1.05%	1.58%	1.22%	0.85%
DoS	21.10%	19.10%	20.20%	23.81%	22.61%	21.43%	21.77%	16.81%	23.76%	23.47%	24.59%	24.98%	23.32%	23.55%	23.42%
L2L	80.49%	78.22%	83.94%	79.04%	76.32%	77.74%	79.75%	75.41%	74.39%	76.75%	80.22%	81.31%	80.37%	81.16%	80.99%
Normal	98.93%	98.99%	98.48%	98.99%	98.92%	99.04%	99.11%	98.74%	99.06%	99.15%	99.18%	99.11%	99.16%	99.17%	99.12%

الملخص

إن من أهم القضايا المتعلقة بأنظمة كشف التسلل IDS هي حساسية هذه الأنظمة تجاه وقوع الأخطاء وكذلك القصور في عملية التقيم في النماذج السابقة. إن معظم الجهود السابقة إهتمت بتحسين الدقة الشاملة لهذه الأنظمة من خلال زيادة قدرة هذه النماذج على كشف التسلل بالتزامن مع تقليل الإنذارات الكاذبة، وهو أمر جدير بالإهتمام. وبالرغم من نجاح هذه الجهود في تحسين الدقة الكلية للنظام؛ إلا أنها سقطت في مفارقة الدقة Accuracy paradox. إن خوارزميات التعلم الألي Machine Learning غالبا ما تصنف معظم أو حتى جميع سجلات الفئات الأقل عددا إلى واحدة من الفئات الأساسية الكبيرة دون ترك أثر يذكر في الدقة الكلية للخوارزميات. الجدير بالذكر، هو أن الهجمات الممثلة بهذه الفئات الصغيرة خطيرة على أمان أنظمة الحاسوب. إن عجز غالب المحاولات السابقة عن حل مفارقة الدقة هذه وبقاء الحاجة الملحة لتحسين أداء هذه الأنظمة هو الدافع لهذه الرسالة. في هذه الأطروحة، لقد قمنا بدمج أسلوب أخذ العينات الطبقي Stratified Sampling وأسلوب الأوزان المختلفة للفئات المختلفة Cost-Function مع خوارزميتي التعلم الألي WSVM و WELM لبناء حلول لتحسن أداء أنظمة كشف التسلل والتقليل من حدوث مشكلة التناقض في الدقة مع مراعات القيام بالتجارب بشكل عادل ومتسق مع غيراها من الأعمال. لقد تم تنفيذ التجارب الأساسية على قاعدة البيانات NSL-KDD، في حين تم إجراء التجارب مرة أخرى على قاعدة البيانات UNB ISCX2012، من أجل إثبات تحقق الهدف من الأطروحة. أظهرت التجارب التي أجريت أن الخوارزمية WSVM هي أكثر فعالية من الخوارزمية WELM، على الرغم من أن الخوارزمية WELM هي منافس جيد. كما أظهرت التجارب على كلتا قاعدتي البتانات وباستخذام كلا الخوارزميتين نتائج منافسة من حيث الدقة الكلية ومقياس F-score لكل فئة على حدة بالمقارنة مع أفضل الأعمال السابقة الحديثة.