Intrusion Detection System Using Machine Learning Approaches

Hany Mohamed, Hesham Hefny, Assem Alsawy

Computer Science and Information Dept. Institute of Statistical Studies and Research
Cairo University, Egypt
enghany230@gmail.com, hehefny@ieee.org, assem_issr@yahoo.com

Abstract

Network security is becoming an important issue in the field of information security. Hackers and Intruders can make many successful attempts to break down into networks or computer systems, and so overcome the need to create a powerful Intrusion Detection System (IDS) is a primary need.

IDS is the art of detecting attacks and any attempt to break down networks, also it's an effective tool to prevent unauthorized access to any network by analyzing its traffic.

The aim of this research is to build an Intrusion Detection Framework able to classify network activities, 'Normal' or 'Attack', using different Machine Learning algorithms, Random Forest (RF), Multi-Layer Perceptron (MLP), and Library for Support Vector Machine (LIBSVM). The proposed model had been tested by using a common dataset called NSL-KDD.

This paper investigates two techniques, the first technique is to apply the different Machine Learning algorithms over the NSL-KDD dataset, and the second technique used a Feature Selection algorithm called Correlation Feature Selection 'CFS', to drop some irrelevant attributes in the dataset, to cut the time taken in the training and testing phases,

Random Forest shows a superior response compared to the other algorithms, especially in terms of response time, detection rate and false positive rate.

Keywords: *Network Security, Intrusion Detection System, Machine Learning Approaches, WEKA, Random Forest, Library for Support Vector Machine (LIBSVM), Multilayer Perceptron (MLP).*

1. Introduction

The internet has become a "Real Life", and as in the life, there are vandalism and criminals. The big threat of vandalism and theft which has given users a need for security components protect themselves. In 1983 the ARPANET (Advanced Research Projects Agency Network) [1], and every network attached it, officially adopted the TCP/IP networking protocol. Which had been under development since 1973, and tested on an internet in the same year [2]. Which allows Internet sites and users to grow exponentially [3, 4]. When the Internet started to be widely used, the users were so excited about being connected to it and forgot the security.

The first Internet worm was unleashed on 2nd of November 1988 by Robert T. Morris [4, 5]. Since then, the number of incidents has grown rapidly each year. For example, in 2002, the number of incidents was about 83 thousand, only in USA [6]. The Enigma Software Group, the maker of Spy Hunter anti-malware software, analyzed more than 25 million

computer viruses picked up by its software in the U.S.A only. Lately, and according to Symantec [28] annual report in 2016, which has established a source of Internet threat data, using network sensors to detect and record the attack. This network monitors threat activity in over 150 countries, the sensors detected more than 430 million new unique pieces of malware in 2015, up 36% from the year before. The most common ways' computer systems are infected by malware is when the users are tricked into clicking on links that either download malware or take them to websites that have viruses. These links come in the form of e-mails or social media messages hijacked by hackers [7].

Today, computer systems have a variety of threats, such Integrity, Confidentiality, Denial of Service (DOS), and Authentication [8]. So, it is very important to keep up a high-level security to guarantee the safety of information, Intrusion Detection System (IDS) is a shield innovation technology for system security technology after classic technologies, such as firewall, message encryption, etc.

This work aims to design and build Intrusion Detection Framework for securing computer networks using Machine Learning approaches, which provide a promising alternative for detecting and classifying anomalies based on an initially large set of traffic dataset, we build the intrusion detection framework to improve the classification and prediction rate for 'Normal' and 'Attacks', with a low false alarm rate.

This paper is organized as follows: Section 2 introduces the Machine Learning techniques, Section 3 introduces intrusion detection system types, Section 4 shows a simple review of some related work in the frame time between 2009 and 2017, Section 5 illustrates briefly the NSL-KDD dataset used in the experiment, Section 6 explains the methodology and the proposed system architecture, Section 7 presents the experimental results and finally Section 8 concludes the paper and future work.

2. Machine Learning

Machine Learning is a branch of Artificial Intelligence (AI) that acquires knowledge from a given data based on known facts. We can define it as a study which allows computers to learn knowledge without programming those computers [11], using specified algorithms and methods. Machine Learning algorithms are a way to help you to make better decisions and predictions [26], with finding a natural pattern in the given data. Those algorithms used every day to make critical decisions in medical diagnosis [27], stock trading, energy load forecasting, and more. Machine learning mainly uses two types of techniques:

2.1 Supervised Learning

The goal of supervised machine learning is to create a model that makes predictions based on evidence in the presence of uncertainty [12]. A supervised learning algorithm takes a known set of input data and known responses to the data and trains a model to generate reasonable predictions for the response to new data. There are several supervised learning algorithms as an Artificial Neural Network, Nearest Neighbor algorithm, Support Vector Machine (SVM), Decision Trees (Random Forrest), C4.5 classifier [27], K-nearest neighbor and Quadratic classifiers.

2.2 Unsupervised Learning

In unsupervised learning data instances are unlabeled. The algorithm goal is to find hidden patterns or intrinsic structures in data. It is used to draw inferences from datasets

consisting of input data without labeled responses. Some common unsupervised learners are K-means clustering, Fuzzy clustering, Self-organizing map, Apriori algorithm [12].

2.3 Machine Learning Algorithms

As mentioned before, there are two types of machine learning techniques, each technique has various algorithms used to learn the dataset instances, in this section, and we are going to discuss only three algorithms, as follows:

2.3.1 Multilayer Perceptron

Multi-Layer Perceptron (MLP) is a class of feed-forward artificial neural network directed graph [3], consists of at least three layers of nodes. Each node is apart from the input nodes, has a nonlinear activation function. It uses backpropagation as a supervised learning technique. Since there are multiple layers of neurons, as shown in figure (1).

MLP is widely used for solving problems that require supervised learning as well as research into computational neuroscience and parallel distributed processing. Applications include Speech Recognition, Text Mining [26], Image Recognition and Machine Translation [3].

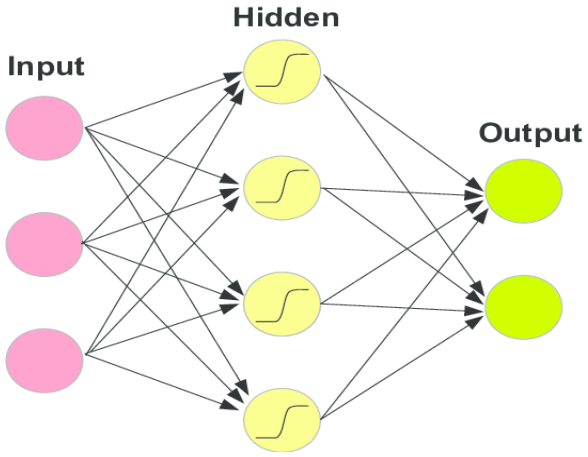

Figure 1: Multi-layer perceptron [3]

2.3.2 Random Forest (RF)

Random Forest is an ensemble learning method for classification and regression [21], that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or means prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set [21], as shown in figure (2).

The first algorithm for random decision forests was created by Tin Kam using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to the classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, and "Random Forests" is their trademark. The extension combines Breiman's "bagging" idea and random selection of features introduced first by Ho and later independently by Amit and Geman in order to construct a collection of decision trees with controlled variance [21].

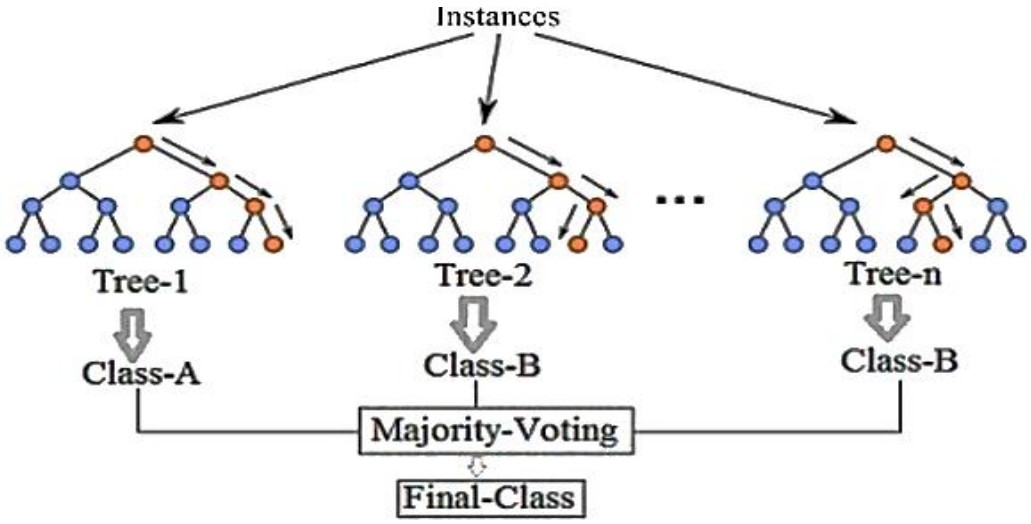

Figure 2: Random Forest Classifier [21]

2.3.3 Library for Support Vector Machine (LIBSVM)

Support Vector Machine (SVM) is a popular machine learning method for classification, regression, and other learning tasks. It's a supervised learning model with associated learning algorithms that analyze data used for classification and regression.

LIBSVM is an integrated software for SVMs, written in C++ though with a "C API". It also supports classification and regression. And multi-class classifications too. A typical use of LIBSVM involves two steps: first, training a dataset to obtain a model and second, using the model to predict information of a testing dataset [23], as shown in Figures (3) and (4).

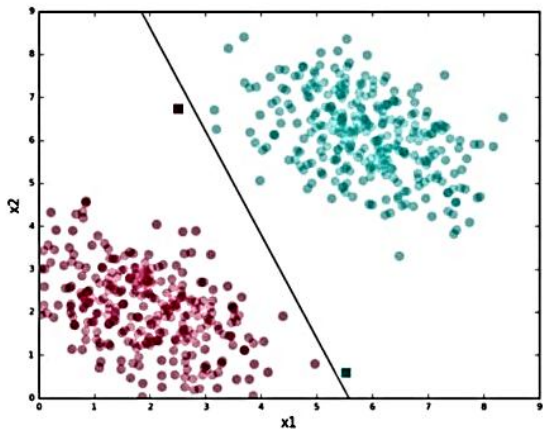

Figure 3: Training data with LIBSVM classifier [23]

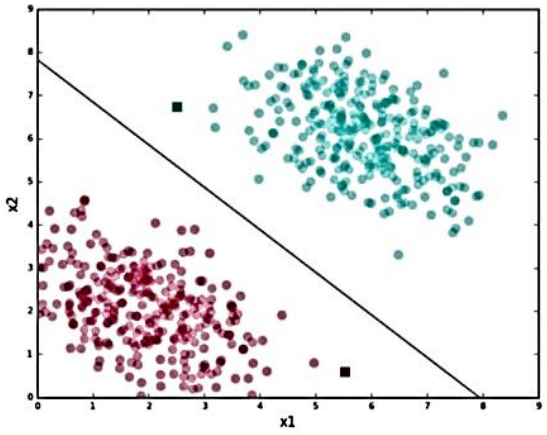

Figure 4: Applying LIBSVM with test data [23]

3. Intrusion Detection System (IDS)

Intrusion Detection System (IDS) is designed to identify - in real time - unauthorized access, misuse, and attacks on any computer systems. It maintains a set of historical profiles of users, update the profiles whenever necessary, and reports any anomalies detected. IDS

main function is determining attacks and to alert the system administrators, that there is a possible security violation.

A few numbers of researchers of IDS proposed by different researchers [5]. Next section illustrates the main categories of the IDS.

3.1 Host-Based Intrusion Detection (HIDS)

HIDS could be described as a piece of software uploaded on a system to monitor it, it was the first Intrusion Detection software designed, it is capable of monitoring and analyzing the network packets coming into a computer system.

HIDS main disadvantage is it cannot detect attacks coming out of this host, but it can analyze the file system of a host, users' login activities and its running processes [5].

HIDS also Able to verify if an attack was successful or not, whereas a network-based IDS only give an alert of the attack, can monitor all users' activities which is not possible in a network-based system, and do not require any extra hardware since they can be installed in the existing host servers, so it's cost-effective for a small-scale network.

3.2 Network-Based Intrusion Detection (NIDS)

NIDS is designed to monitor and analyze the traffic on its network segment to detect intrusion attempts. It can be made of many sensors, each sensor is being in charge of monitoring the traffic passing through its own segment.

The sensors cannot monitor anything outside their own segment or switch. NIDS could be described as an ID system that monitors the traffic on its network segment as a data source [10]. NIDS can easily detect attacks. In Real-time with a quick response, it's also a cost-effective tool.

3.3 Misuse based detection

Misuse IDS contains a database of known vulnerabilities. It monitors traffic and seeks a pattern or a signature match. It operates in much the same way as a virus scanner, by searching for a known identity or signature for each specific intrusion event. It can be placed on a network to watch the network vulnerabilities or can be placed on a host [9].

This is why the number of false positive alarms is comparatively less in signature-based IDS. If the system is not entirely new, i.e. when it has an up to date database of signatures of known attacks, this technique works extremely well. Some advantages of this type of IDS are Flexibility, Detect Multiple Attacks and Fast [10].

3.4 Anomaly-based detection

Anomaly detection systems are also known as behavior-based systems. They rely on the fact that intrusions can be detected by observing deviations from the expected behaviors of the system monitored. These "normal" behaviors can either correspond to some observations made in the past or to some forecasts made by various techniques.

Everything that does not correspond to this 'normal' pattern will be flagged as an attack. The core process of anomaly detection is not to learn what attack is but to learn what is normal or expected.

The main advantage of anomaly detection systems is that they can detect previously unknown attacks. By defining what's 'normal' and what's 'attack' [5].

4. Related work

Lately, there have been so many researchers in the field of Intrusion Detection System, most of them used the KDD Cup99, and NSL-KDD dataset for training and testing the created model. A simple review made on many research papers in the time frame between 2009 and 2017.

Table (1): Intrusion Detection System Previous Works.

Year	Paper title	Feature Selection	Algorithm used	Accuracy %
2009	Data Mining based intrusion detectors [13]	No	1-C4.5 2-SVM	C4.5: 65 SVM: 62.7
2012	An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection [16]	NO	SVM	SVM: 88.64
2013	Advanced probabilistic approach for network intrusion forecasting and detection [17]	NO	1-Markov chain. 2-K-means clustering 3-APAN	K-mean: 90
2014	Malicious web content detection by machine learning[14]	1-URL Lexical features	1-NB 2- DT 3- SVM 4-KNN 5-ANN	NB: 88.47 DT: 95.12 SVM:93.75 KNN: 92.90 ANN: 96.01
2014	An Ensemble Model for Classification of Attacks with Feature Selection based on KDD99 and NSL-KDD Data Set [18]	Gain Ratio	1-CART 2-ANN 3-Ensemble	1-96.56 2-95.98 3- 97.76
2014	An enhanced hybrid anomaly-based detection approach[15]	Swarm algorithm	Detector Generation algorithm	EHAD: 96.3
2016	Feature Selection for Intrusion Detection using Random Forest[23]	Random Forest	RF	RF: 91.9
2017	A Network Intrusion Detection Framework based on Bayesian Network using Wrapper Approach[19]	Wrapper approach	1-NB 2- C4.5 3-SMO 4-WBNAD	NB: 84.86 C.45: 97.44 SMO: 97.88 WBNAD: 98.3

5. NSL-KDD Dataset

In this research, the NSL-KDD [20] dataset is used, which is the enhancing dataset of the KDD Cup 1999, it was the dataset used for The Third International Knowledge Discovery and Data Mining Tools Competition. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between “bad” connections, called intrusions or attacks, and “good” normal connections.

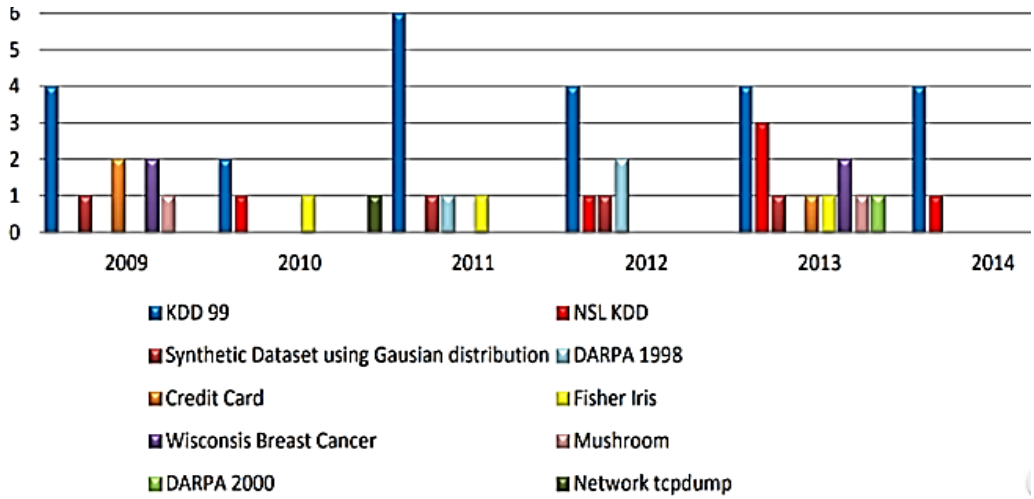

Figure 5: Number of usage of different Datasets for IDS [20]

NSL-KDD is about 4 GB of squashed raw (binary) TCP abandon data of 7 weeks of network traffic, it can be developed into 5 million association records each with about 100 bytes. For each TCP/IP connection, this dataset represented in 41 features, as described in table (2), 34 features (numeric) and 7 features (symbolic), each data target indicates the attack names, the dataset covers over 20 different attack types as outputs [19], as described in table (2).

Many researchers in the last decade had been used the KDD Cup99 dataset, and its enhanced version called NSL-KDD dataset [20], as shown in figure (5).

Table 2: NSL-KDD dataset attribute features [19]

Feature name	Description	Feature name	Description
Duration	Duration of the connection.	Is_guest_login	1 if the login is a "guest" login; 0 otherwise
Protocol type	Connection protocol (e.g. TCP, UDP)	Count	Count of connections to the same host as the current connection in the past two seconds
service	Destination service (e.g. Telnet, FTP)	Srv_count	number of connections to the same service as the current connection in the past two seconds
Flag	Status flag of the connection	Serror_rate	percentage of connections that have “SYN" error
Src_bytes	Bytes sent from source to destination	Srv_serror_rate	percentage of connections that have “SYN" errors
Dst_bytes	Bytes sent from destination to source	Rerror_rate	percentage of connections that have “REJ" errors
Land	1 if the connection is from/to the same host/port; 0 otherwise	Srv_rerror_rate	percentage of connections that have “REJ" errors
Wrong fragment	Count of wrong fragments	Same_srv_rate	percentage of connections to the same service
Urgent	Count of urgent packets	Diff_srv_rate	percentage of connections to different services
Hot	Count of "hot" indicators	Srv_diff_host_rate	percentage of connections to different hosts

Table 2 (continued) : NSL-KDD dataset attribute features [19]

Feature name	Description	Feature name	Description
Num_failed _logins	Count of failed logins	Dst_host_count	Number of connections having the same destination host
logged_in	1 if successfully logged in; 0 otherwise	Dst_host_srv _count	count of connections having the same destination host and using the same service
Num_compro mised	Count of "compromised" conditions	Dst_host_same_ srv_rate	percentage of connections having the same destination host and using the same service
Root_shell	1 if root shell is obtained; 0 otherwise	Dst_host_diff _srv_ rate	percentage of different services on the current host
Su_attempted	1 if "su_root" command attempted; 0 otherwise	Dst_host_same_ src_port_rate	percentage of connections to the current host having the same src port
Num_root	Count of "root" accesses	Dst_host_srv_diff _ host rate	percentage of connections to the same service coming from different hosts
Num_file _creations	Count of file creation operations	Dst_host_serror _rate	percentage of connections to the current host that has an S0 error
Num_shells	Count of shell prompts	Dst_host_srv_ serror_rate	percentage of connections to the current host and specified service that have an S0 error
Num_access_f iles	Count of operations on access control files	Dst_host error _rate	percentage of connections to the current host that has an RST error
Num_outboun d_ cmd	Count of outbound commands in an FTP session	Dst_host_srv_rerr or_rate	percentage of connections to the current host and specified service that have an RST error
Is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise		

All the attacks in the NSL-KDD dataset are divided into five categories [19], Normal is a category, and the other four categories are, Denial of Service attacks (DOS), Probing attacks (Probe), Remote to Local attacks (R2L) and User to Root attacks (U2R).

6. Methodology

This research presents a complete framework to select the best set of NSL-KDD dataset features that efficiently characterize normal traffic and distinguish it from abnormal traffic using three different machine learning approaches. This research uses an approach for feature selection called Correlation Feature Selection ‘CFS’, and experimenting the dataset with three different classifiers to learn the dataset and designing a prediction model. The framework of the proposed model consists of the following components and as shown in figure (6):

- NSL-KDD dataset Pre-processing.
- Feature Selection using Recursive Feature Elimination.
- Modeling of the detection model using Multi-Layer Perceptron, LIBSVM, and Random Forest.

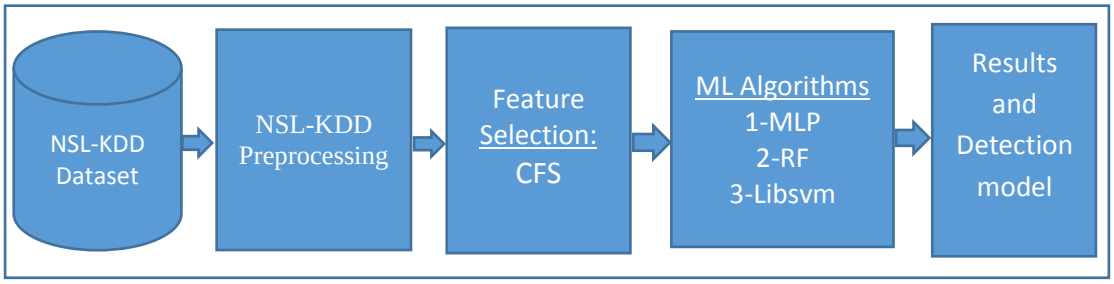

Figure 6: IDS Proposed Model.

6.1 Data Pre-processing

In this Experiment, WEKA, which is a Machine Learning tool, is used. Weka deals with Attribute-Relation File Format (ARFF) formatted files, and spreadsheet file (CSV), the first step is converting the dataset from text (TXT) format into (ARFF) format.

While Machine Learning doesn't concern with "Text" and only deals with "Numeric" data, so it is necessary to convert the dataset from “text” to “numeric” to deal with, using “String to Vector” function in the WEKA.

It’s also necessary to implement scaling since the data have significantly varying resolution and ranges. The attribute data are scaled to fall within the range [-1, 1]. Initial data are collected and pretreated as network connection data, including particular attributes.

6.2 Cross-validation

It is a very useful technique for assessing the performance of the proposed model, it also helps in avoiding the over-fitting, and even though the proposed model is generalized to independent data, by dividing the dataset into a number of folds, using K-folds method, therefore, all the entries in the original training dataset are used for both training as well as testing. This proposed model divided the dataset into 10 folds.

6.3 Feature Selection

In this step, the simple correlation-based feature selection “CFS” is used, CFS is a filter method that selects the best feature subset according to some evaluation function, where features are assumed to be conditionally independent. Based on the former assumption, CFS is not guaranteed to select all relevant features, when there are strong features dependencies [24]. It also needed to select a search method, for the feature selection algorithm, “Best First” search method is used, which Searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility.

CFS is used to minimize the number of features in the NSL-KDD dataset. It evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. And Subsets of features that are highly correlated with the class while having low inter-correlation are preferred.

6.4 Applying of Machine learning algorithm

After eliminating the unnecessary features, the classification algorithm is applied over the NSL-KDD dataset, to carry out rule mining to further distinguish normal behavior and attack behaviors.

Three different algorithms will be experimented, with the various number of features, and then check the accuracy with all of the dataset features, and after elimination of some of the features.

7. Results

The experiment runs on an Intel® Core™ i5-3210M CPU @ 2.2 GHz (4 CPUs), ~2.5GHz with 8G memory running on Windows 10. The experiment conducted with the help of WEKA 3.8 machine learning tools and Weka Library functions for feature selection techniques.

	TRUE	FALSE
TRUE	True Positive	False Positive
FALSE	False Negative	True Negative

Figure 7: Confusion Matrix [25]

7.1 Performance measurement

Any classifier predicts all data instances of a test dataset as either positive (P) or negative (N). This classification or prediction produces four outcome values [25], which are true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

- True positive (TP): correct positive prediction.
- False positive (FP): incorrect positive prediction.
- True negative (TN): correct negative prediction.
- False negative (FN): incorrect negative prediction.

Those four values can produce the Confusion Matrix, as shown in figure (7), some measures could be derived out from the confusion matrix, such Accuracy, Precision, Recall, and F-measure [25].

1) **Accuracy (ACC):** it can be calculated by dividing the number of all correct predictions over the total number of the dataset. Best accuracy is 1.0.

$$\textbf{Accuracy Rate (ACC)} = \frac{TP+TN}{P+N} \qquad (1)$$

2) **Precision (PREC):** it can be calculated by dividing the number of a number of correct positive predictions over the total number of positive predictions. It is also called positive predictive value (PPV). The best precision is 1.0.

$$\textbf{Precision (PREC)} = \frac{TP}{TP+FP} \qquad (2)$$

3) **Recall or The Sensitivity (SN):** it can be calculated by dividing the number of correct positive predictions over the total number of positives. The best recall is 1.0.

$$\textbf{Recall} = \frac{TP}{TP+FN} \qquad (3)$$

4) **False positive rate (FPR)**: it can be calculated by dividing the number of incorrect positive predictions over the total number of negatives. The best false positive rate is 0.0 whereas the worst is 1.0.

$$\textbf{False Positive Rate (FPR)} = \frac{FP}{TP+FP} \qquad (4)$$

7.2 Model Evaluation

The first classifier was Multilayer perceptron (MLP), which showed an accuracy rate of 97.1 %, with a false positive rate 0.017%, while dealing with the 41 features, as shown in the

table (2). With using the Feature Selection "CFS", MLP showed a lower accuracy rate 93.8%, than with using all features, as shown in the table (3).

Table 3: Multilayer perceptron classifier Confusion Matrix

Algorithm	Precision	Recall	F-Measure
MLP (41 attributes)	95.7%	97.1%	96.3%
MLP (17 attributes)	91.7%	93.8%	92.5%

The second classifier was Random Forest (RF), which showed an accuracy rate of 99.6% along with a very low false positive rate while using 41 attributes as shown in figure (4). And with using the Feature Selection 'CFS', it gave the same accuracy rate.

Table 4: Random Forest Classifier Confusion Matrix

Algorithm	Precision	Recall	F-Measure
RF (41 attributes)	99.6%	99.6%	99.6%
RF (17 attributes)	99.6%	99.7%	99.6%

The third classifier was LIBSVM which showed an accuracy rate of 95.1%, with low false positive rate, as shown in the table (5). And with using the Feature Selection "CFS", LIBSVM showed an accuracy with 97.2%, with low false positive rate, as shown in the table (5).

Table 5: LIBSVM Classifier Confusion Matrix

Algorithm	Precision	Recall	F-Measure
LIBSVM (41 attributes)	94.8%	95%	94.5%
LIBSVM (17 attributes)	97.2%	97.2%	97.2%

8. Conclusion and Future work

The objective of the proposed framework was to build an independent Intrusion Detection capable of predicting attacks, and differentiating between "normal" activates, and "attacks". The issue was about the pre-processing phase, which we can define it as "Main Phase", which is divided in different steps, converting the dataset, dividing it into multiple equal folds, normalizing the data, scaling the data, and feature elimination.

In this experiment, three different approaches had been used to detect attacks, applied over the NSL-KDD dataset. Multilayer perceptron, Random Forest, and LIBSVM. Multilayer perceptron classifier accuracy rate was 95.7%, while Random Forest accuracy rate was 99.6%, and LIBSVM classifier accuracy rate was 94.8%. Feature Selection "CFS", which used to eliminate the number of features to only 17 features, applied over the same classifiers, the MLP showed a low accuracy 91.7%, and the same accuracy rate while using the Random Forest, but with the LIBSVM, the accuracy rate raised to 97.2%.

These results show that the proposed model with Ensemble Random Forest classifier gives a high accuracy rate than using the MLP and LIBSVM, whether using the 41 features or after eliminating features to 17 attributes, but the classification phase with only 17 features, took less time than dealing with 42 features, achieving a "Time Reduction", which is one of the IDS main goals.

This experimental study showed a way to enhance the intrusion detection overall performance by a normalizing and scaling data, splitting it into multiple equal folds, eliminating some irrelevant features, which led towards the time and complexity reduction in training and testing phases.

Our most encouraging direction for future research in the field of IDS concerns the combination of multiple intrusion detection classifications at the runtime, which may provide more accurate detections of Malware and intruders, another promising area, includes the using of feature elimination algorithms abilities, to examine data features, in a way to enhance the performance of the intrusion detection and reduce the time taking in detecting the attacks.

References

[1]. John Naughton, "The evolution of the Internet: from military experiment to General Purpose Technology", Journal of Cyber Policy, ISSN: 2373-8871, 2016.

[2]. B. A. Forouzan, "TCP/IP Protocol Suite", 1st Edition, McGraw-Hill Companies, 2000.

[3]. E. G. Britten, J, Tavs, R. Bournas, "TCP/IP: The Next Generation", IBM Systems Journal, Vol. 34, No. 3, pp. 452-472, 1995.

[4]. C. J. P. Moschovitis, H. Poole, T. Schuyler, and T. M. Senft, "History of the Internet: A Chronology, 1843 to the Present", Santa Barbara, CA: ABC-CLIO, 1999.

[5]. B.Carlson, and C.Miller, "Timeline of Computing History", http://www.computer.org /computer/timeline/timeline.pdf, accessed 16.June 2003.

[6]. CERT Coordination Center,"CERT Statistics for 1988-2002", http://www.cert.org/stats/, accessed 16.June 2003.

[7]. http://america.aljazeera.com/articles/2016/1/21/computer-viruses-attack-in-some-cities-more-than-others.html.

[8]. Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, "Security in Computing ", 2015.

[9]. Srinivas Mukkamala, "Intrusion Detection using Neural Networks and Support Vector Machine", IEEE International Honolulu, 2002.

[10]. James Cannady, "Artificial Neural Networks for Misuse Detection", National Information Systems Security Conference (NISSC'98), Arlington, VA, 1998.

[11]. Zhenwei Yu, Jeffrey J P Tsai, "Intrusion Detection system, A Machine Learning Approach", (University of Illinois, Chicago, USA & Asia University, Taiwan), 2011.

[12]. https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms.

[13]. Su-Yun Wua, E. Y. "Data mining-based intrusion detectors", ELSEVIER, 2009.

[14]. Yung-Tsung Hou, Y. C.-S.-M, "Malicious web content detection by machine learning", ELSEVIER, 2010.

[15]. Tamer F. Ghanem, Wail S. Elkilani, "An Enhanced Hybrid Anomaly-Based Detection Approach", International Conference of Artificial Intelligence & Manufacturing Engineering, pp. 169-177, December 2014.

[16]. Carlos A. Catania, F. B, "An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection", ELSEVIER, 2012.

[17]. Seongjun Shin, S. L., "Advanced probabilistic approach for network intrusion forecasting and detection", ELSEVIER, 2013.

[18]. Akhilesh Kumar Shrivas, Amit Kumar Dewangan. An Ensemble Model for Classification of Attacks with Feature Selection based on KDD99 and NSL-KDD Data Set. International Journal of computer applications. Volume 99, No.15, 2014.

[19]. Md Reazul Kabir, "A Network Intrusion Detection Framework based on Bayesian Network using Wrapper Approach", International Journal of Computer Applications, Volume 166 – No.4, May 2017.1

[20]. L. Dhanabal," A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015.

[21]. https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

[22]. Francis R. Bach & Gert R. G. Lanckriet, "Fast Kernel Learning using Sequential Minimal Optimization", Division of Computer Science, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 2004.

[23]. Chih-Chung Chang and Chih-Jen Lin, "A Library for Support Vector Machines", https://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf, Last update Marc, 2013.

[24]. M. a Hall, "Correlation-based Feature Selection for Machine Learning," Methodology, vol. 21i195-i20, no. April, pp. 1-5, 1999.

[25]. Sanyam Kapoor, "Visualizing the Confusion Matrix", https://www.sanyamkapoor.com/machine-learning/confusion-matrix-visualization/, 2017.

[26]. Mona Gamal, Ahmed Abo El-Fatoh, Elsayed Radwan, Aziza Asem, "Hybrid Intelligent Model of Genetic Algorithms and Association Rules in Text Mining", Egyptian Computer Science Journal, 2010, http://ecsjournal.org/JournalArticle.aspx?articleID=26.

[27]. Nesma Ibrahim, Taher Hamza, Elsayd Radwan, "An Evolutionary Machine Learning Algorithm for Classifying Thyroid diseases Diagnoses", Egyptian Computer Science Journal, 2011, http://ecsjournal.org/JournalArticle.aspx?articleID=283.

[28]. https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf.