Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 16 (2021) pp. 049-058

Machine learning for intrusion detection: Design and Implementation of an IDS based on Artificial Neural Network

Younes Wadiai[1], Yousef El mourabit[2], Mohammed Baslam[3], Youssef El Habouz[4]

[1,2,3] TIAD Laboratory; Sciences and Technology Faculty; Sultan Moulay Slimane University
Beni Mellal, Morocco
younes.wadiai@gmail.com, y.elmourabit@usms.ma, m.baslam@usms.ma

[4]Igdr Umr 6290 Cnrs- Rennes1 University
Rennes, France
youssef.elhabouz@univ-rennes1.fr

Abstract*: Securing the network from intrusions becomes a more challenging task to conduct for system administrators, and the need for a more powerful and efficient intrusion detection system emerges with the continuous development of cyber-attacks exploring various methods and techniques. A performed survey in [1] show the various emerging attacks in cyber security accompanied with the exponential growth of the internet interconnections, the attacks are affecting the confidentiality, availability, and the integrity of the data in the cyber world, as more data is now available in electronic format, and more access is provided to end users, the challenge is to secure the network from any intrusion. Rather than following the traditional way of detecting attacks by looking for signatures of known intrusion attempts, machine learning can help detect nonconformities over the network. We propose the usage of artificial intelligence to build a sophisticated Network Intrusion Detection System able to be trained/self-trained using models and algorithms found in machine learning/deep learning to detect malicious network traffic. The aim of this paper is to present a new IDS model based on machine learning approach to detect malicious traffic and protect the network from cyber-attacks. The usage of machine learning will allow better accuracy in detection and faster response time. This technique can also be used to continuously update the IDS knowledge base for instant response through malicious packets rejection. In order to implement and measure the performance of our model, we used NSL-KDD dataset which contains records of various mimicked attacks on a real IDS system, after the preprocessing phase which consist of data summarization, cleaning, and normalization, we used the most relevant attributes for the classification process based on CfsSubsetEval technique with BestFirst approach as an attribute selection algorithm to remove the redundant attributes and to allow the usage of the most pertinent attributes of the dataset. To build our prediction model we used a comparative evaluation of three algorithms (K-means, AdaBoost and Multilayer Perceptron), the experimental results show that the MLP algorithm provides a high detection rate and reduces false alarm rate. Finally, a set of principles is concluded, which will set path for future research for implementing an efficient and performant IDS. To help researchers in the selection of IDS, several recommendations are provided with future directions for this research.

***Keywords*:** Intrusion Detection System, Machine learning, Deep learning, Neural network, NSL-KDD, Weka platform, K-means, AdaBoost, Multi-Layer Perceptron, NIDS, ANIDS.

I. Introduction

As artificial intelligence is booming through various applications that mimic the human brain and bypass its computing capacities by performing complexed calculations in fraction of a second with high accuracy, and with the era of almost all electronic devices are capable of getting connected to the internet, securing the network from unauthorized access and all sort of intrusions becomes a necessity for all private and corporate parties. The presented paper is introducing a new aspect of network protection by using artificial intelligence in cyber security, as nowadays the security of the network is crucial, and benefiting from the promising outcomes of artificial intelligence can help identifying malicious attacks in networks. Since the Network Intrusion Detection Systems play a major role in monitoring and detecting abnormal and malicious network traffic, building a robust IDS model counting on the artificial intelligence capabilities is crucial for cyber security, where an IDS can be formed from a combination of hardware and software. The proposed approach helps IDS classify traffic packets transiting the network into benign/malicious packets using machine learning. The network packets are the sum of web visits performed by a client seeking access to a resource located in the web server, packets are either TCP or UDP and the produced heavy traffic

is collected as a set of attributes in a mega database called dataset. The dataset is the sum of traveling network packets where each packet consists of primarily two components: control information and user data also know as the payload, while the payload is the content of the transmitted packet such as video, audio, or image, the control information is responsible for the delivery, it has all required information to succeed the transmission such as: source and destination IP address, source and destination port number, internet protocol used, and other information. The data mining phase then comes to play to extract pertinent information from each packet by applying network sniffing tools techniques. Each connection made between the server and the client is classified into benign or malicious traffic. We used NSL-KDD [2] dataset form the Canadian Institute for Cyber Security [3] as our reference dataset to apply machine learning techniques. The feature selections and experiments are executed using the WEKA [4] platform. The analysis of the NSL-KDD dataset in this paper is made by applying different clustering algorithms offered in the WEKA platform for data mining. In order for the NSL-KDD dataset to be analyzed and categorized, we use the preprocessing techniques of CfsSubsetEval[5] which is a Weka attribute selection method that gage the value of a set of entries by applying the prediction technique on each attribute's feature and compare it with its redundancy, and BestFirst[6] which searches for the space between attributes to clean missing data and normalize it, this step allows the selection of only the most pertinent attributes of the dataset to be selected. The evaluation of the findings shows the outperformance of the multilayer perceptron MLP [7] algorithm compared to K-means [8] and AdaBoost [9]. The MLP algorithm has better detection rate with low false alarm rate. This indicates the outstanding performance of deep learning compared to machine learning and how neural network can play a major role in future IDS security solutions. The rest of the paper is presented as follows: Section II introduces some related work performed on using machine learning in intrusion detection systems with detailed analysis of the points of strength and weakness of each used method, Section III presents our approach with a description of the NSL-KDD components and attributes along with the applied method to get the results. Section IV summarizes the findings and the analysis of the experiments with a discussion of the results of each algorithm used. Section V is the conclusion of the article that evaluates the findings and suggests recommendations for future work.

II. Related Work

Set your page as A4, width 210, height 297 and margins as follows: This section introduces different research made on the NSL-KDD dataset using artificial intelligence. In [10] the J48 decision tree is used as a classifier to train the model with tenfold cross validation applied on the testing dataset where only 22 attributes are used instead of the full 41 attributes. A similar work is evaluated in [11] where a combined classifier model is used based on the random tree and NBTree[12] algorithms, this allowed to achieve an accuracy of 89% bypassing the single random tree algorithm. Another model is proposed for validation through cross-validation method in [13], and it outperformed metered values of other existing machine learning models such as Support Vector Machine SVM [14], naive Bayes [15], and logistic regression [16]. In [17] the particle swarm optimization algorithm was used on selected attributes of the dataset which reduces the false positives rate and enhances the true positives, this increased the accuracy rate compared to other traditional classifiers. While a new improved method for anomaly detection was proposed in [18], by using the gradient boosted machine GBM [19] through applying grid search, the results are then compared with other models including support vector machine, random forest [20], and deep neural network. The GBM technique outperformed most other models used in the IDS. In [21] an enhanced J48 algorithm was used to improve the detection accuracy where the dataset was split in two subsets: a training and a testing dataset. The implementation of this algorithm guaranteed a high classification rate of 76% accuracy using all the dataset features which remains moderate results compared to the aimed goal. While on [22] the dataset was used to analyze the bond relationship between the network protocols and the attacks through the classification algorithms. The analysis came out with the result that most network attacks on IDS are performed using the existing cons of the TCP/IP suite, furthermore the CFS method reduced the detection time and enhanced the accuracy rate after getting filtered attributes to be used in training and testing environment. Another approach was tested on the article [23] by applying a Hybrid FilterWrapper Feature Selection HFWFS to detect DDoS attacks, the performance was evaluated on the NSL-KDD dataset through a Random Tree classifier where features were reduced from 40 to 9 attributes while maintaining a high detection accuracy. Whereas in [24] a distributed IDS system was proposed named Cooperative IDS to protect wireless nodes, the Cooperative Fuzzy Qlearning(Co-FQL)[25] optimization algorithmic technique was used to training and testing on the dataset and results showed the suggested IDS model has 90% accuracy rate than applying individual algorithms including Fuzzy Logic Controller or Q-learning algorithm. A proposition in [26] aiming to create a multi-objective optimization process to distinguish false IDS alarms (false negatives and false positives) using a clustering system between different IDSs and a binary multi-objective optimization algorithm for detection. Experiments on NSL-KDD dataset show the outperformance of the applied method over concurrent solutions. In [27] a new convergence methodology was applied on fuzzy systems to optimize the performance over the genetic algorithms by using the fuzzy logic. The same fuzzy logic is introduced in the modular neural network in [28] where granular computing is used for particle swarm optimization (PSO). In article [29] a comparison between the supervised machine learning technique and the unsupervised deep learning using neural network while the proposed method is the unsupervised feature learning to integrate it in the NIDS network Intrusion Detection System over the ANIDS Anomaly-based NIDS [30]. A similar work was performed in [31] where the research consists of building an anomaly-based network intrusion system using deep learning, the study shows the capability of the model to build good classifiers and add signatures to the IDS system. A new approach was adopted in [32] where an improved version of NIDS was introduced by using Word Embedding-based Deep Learning WEDL-NIDS to get more accurate results and better true/false alarm rate.

A detailed analysis of the previous applied method on the NSL-KDD dataset can be combined can be found in the following table 1 to compare performance between algorithms:

Classification Algorithm	Class Name	Test Accuracy %
Random Forest [33]	Normal	99.1
	DOS	98.7
	Probe	97.6
	U2R	97.5
	R2L	96.8
J48 [34]	Normal	78.9
	DOS	82.4
	Probe	80.2
	U2R	73.9
	R2L	87.6
SVM [35]	Normal	98.1
	DOS	97.8
	Probe	90.7
	U2R	93.7
	R2L	91.8
CART [36]	Normal	88.9
	DOS	82.7
	Probe	82.1
	U2R	73.1
	R2L	80.8
Naive Bayes [37]	Normal	70.3
	DOS	72.7
	Probe	70.9
	U2R	70.7
	R2L	69.8

Table 1. Accuracy of used algorithms

III. Proposed Approach

In this section, a definition of the various used tools is provided in order to deploy and implement the NSL-KDD dataset, as well as an explanation of the various used algorithms to come out with used attributes, the mathematical formulas are also provided to show the logic behind each operation. The architecture of our System is presented in the following figure:

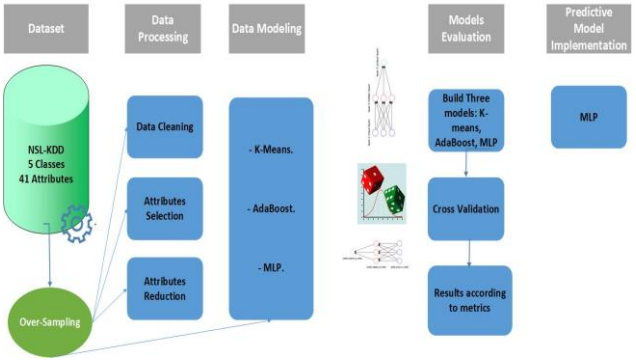

Figure. 1 Architecture of our approach

Firstly, we used NSL-KDD dataset, which is a collection of TCP/IP connection logs along the period of nine weeks in a local area network environment of the United States Air Force. The NSL-KDD dataset was chosen in our experiment for the following reasons: Absence of redundant entries in the training subset which guarantee production of good prediction results and less frequent registrations, also there are no repeated records in the testing subset to ensure excellent rate of reduction, and finally the selected number of entries from each category is balanced to perpetually match the number of records originally in the dataset. In each TCP/IP connection there are 41 extracted attributes and five classes in the multiclass with four types of attacks. In the binary case we have two results: either normal or abnormal. NSL-KDD dataset has three types of attributes: numeric, nominal, and binary. The attributes 2,3, and 4 are nominal, the attributes 7,12,14,15,21, and 22 are binary, and the rest of the attributes are numeric. The following table 2 shows the types of attributes in detail:

Type	**Attribute**
Nominal	Protocol_type(2), Service(3), Flag(4)
Binary	Land(7), logged_in(12), root_shell(14), su_attempted(15), is_host_login(21), is_guest_login(22)
Numeric	Duration(1), src_bytes(5), dst_bytes(6), wrong_fragment(8), urgent(9), hot(10), num_failed_logins(11), num_compromised(13), num_root(16), num_file_creations(17), num_shells(18), num_access_files(19), num_outbound_cmds(20), count(23) srv_count(24), serror_rate(25), srv_serror_rate(26), rerror_rate(27), srv_rerror_rate(28), same_srv_rate(29) diff_srv_rate(30), srv_diff_host_rate(31), dst_host_count(32), dst_host_srv_count(33), dst_host_same_srv_rate(34), dst_host_diff_srv_rate(35), dst_host_same_src_port_rate(36), dst_host_srv_diff_host_rate(37), dst_host_serror_rate(38), dst_host_srv_serror_rate(39), dst_host_rerror_rate(40), dst_host_srv_rerror_rate(41)

Table 2: Attributes of the dataset

The dataset is a collection of 37 various attack that is summarized under four major categories as indicated in the following table 3:

Attack Name	Category
Back,Land,Neptune,Pod,Smurf, Teardrop,Mailbomb,Processtable,Udpstor m,Apache2,Worm	DOS
Satan,IPsweep,Nmap,Portsweep,Mscan,Sa int	Probe
Guess_password,Ftp_write,Imap,Phf,Multi hop,Warezmaster,Xlock,Xsnoop,Snmpgue ss,Snmpgetattack,Httptunnel,Sendmail, Named	R2L
Buffer_overflow,Loadmodule,Rootkit,Perl ,Sqlattack,Xterm,Ps	U2R

Table3: Categories of attack in the dataset

Secondly, for dataset preprocessing step, we used the following operations: Data cleaning and Selection/Reduction of Attributes. Selection is important to improve the efficiency of algorithms, and to minimize the dimension of the explored data. This process allows deletion of non-pertinent, redundant, and noisy attributes which speeds up the learning and increases the readability of the models. In order to apply the filtering approach, a score of pertinence is calculated for each entity based on an evaluation of entities where low scores are ejected. The filtering technique is one of the methods used to clean the dataset, this mechanism allows the evaluation of entities according to heuristics based on data characteristics. In general attributes relevance scores are calculated and the attributes that have the lowest scores are removed. The filtering approach goes through three main steps:

1) Calculation of pertinent scores:

For each attribute, this means to calculate the importance of each attribute to determine its pertinence by giving a score to each attribute based on its importance.

2) Elimination of Attributes:

Where their scores don't meet prerequisite criteria: The attributes with low scores are removed from the dataset for better performance.

3) Use of the sub category:

Use sub category of selected attributes as an entry to the classification system: This means using the outcome from the filtered phase to feed the classifier. The Figure. 2 shows the preprocessing phase where data first filtered using clusters then classified.

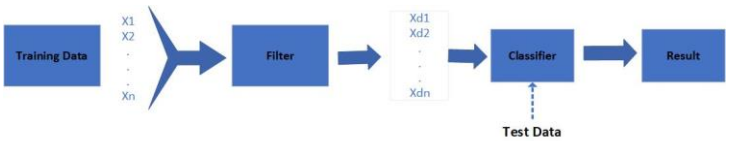

Figure 2. Diagram of data filtering [38]

The following figures 3 and 4 shows graphs of the analysis of both training and testing subset with the number of entries for each type of attack:

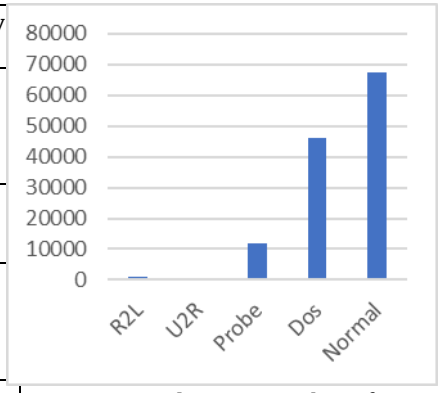

Figure 3. Training dataset: Number of Instances

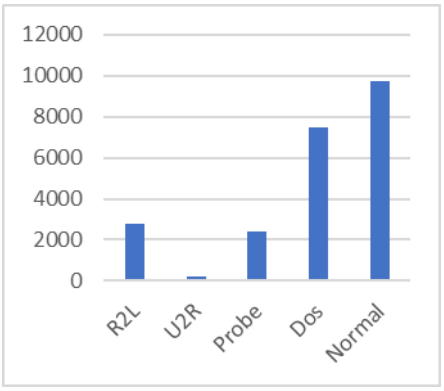

Figure. 4 Testing dataset: Number of Instances

To identify the pertinent attributes in our NSL-KDD dataset we apply the information gain technique which is calculated by reducing the entropy while the dataset is being transformed. The gain is computed through the comparison of the entropy before and after transforming the dataset. We used the entropy metric [39] which is a tool used for measurement of the system unpredictability. The following formula is used to represent the entropy of the variable Y.

$$H(Y) = -\Sigma y \epsilon Y\, P(y) log2\,(P(Y))$$

where p(y) is the density function of the marginal probability for the random variable Y.If the observed values of Y in the assembly S of the data are partitioned based on the second entity X and the entropy Y is smaller than the value of Y before the partition, we conclude a relationship between the characteristics of Y and X. The entropy of Y after observing X is now under the following formula:

$$H(Y|X) = -\sum y \epsilon y P(y|x) log2(p(y|x))$$

Where p(y|x) is the conditional probability of y given x. The final formula to represent the information gain for an attribute Y is then:

$$GI = H(Y) - H(Y|X)$$

In the Data Modelling phase, we used three of the most used and efficient machine learning algorithms in this field, to build the most performant IDS.

K-means: This algorithm allows us to find the groupings of a set of data by finding similarities and the number of groupings is represented by K variables. This algorithm is characterized by its simplicity and it can be used for any type of large size dataset.

The mathematical function for K-means clustering algorithm is based on calculation the squared error function as follows:

$$J = \sum i = 1k \sum j = 1n (|| xi - vj ||)2 = 1$$

Where $|| x_i - v_j ||$ is the Euclidean distance between the two points xi and vj.

AdaBoost: AdaBoost (Adaptive boosting or adaptive stimulation) is an algorithm of heuristic supervised classification based on training data. The role of the algorithm is to group iteratively a set of weak algorithms, in our case we worked with the algorithm decision stump [21], to have at the end a strong algorithm which allows us to predict the class where it belongs as an example. The classifier is weak in the sense that its performance must be at less slightly better than chance, this amounts to defining an error are mathematically represented as:

$$f(x) = X T t = 1 \alpha t h t(x)$$

Where $ht(x)$ is a simple classifier and the output for the strong classifier is:

$$H(x) = sign X T t = 1 \alpha t h t(x)$$

MLP: The Multilayer Perceptron is a deep learning algorithm formed of connected input and output layers, there are also hidden layers in the middle of the structure as the below figure 5 illustrates:

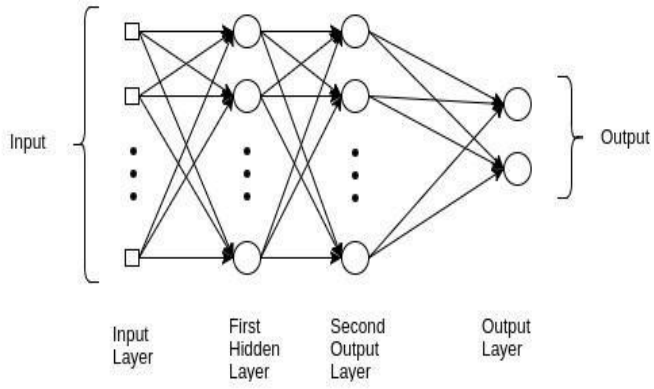

Figure. 5: MLP algorithm mechanism

For Models evaluation phase, we compare three IDS models according to the following metrics: **True positive:** An example belongs to a class, and the classifier got it right. **False negative:** An example belongs to a class, but the classifier commits an error. **False positive**: An example doesn't belong to a class, but the classier attributes it to the class. **True negative:** An example doesn't belong to a class, and the classifier doesn't attribute it to that class. We can summarize these four cases in the table 4 below:

	Predicted Class	
	Predicted	Not Predicted
Example belong to the class	True Positive	False Negative
Example not belonging to the class	False Positive	True Negative

Table 4: The four main cases

Performance measurement:

To start with the classification phase, the Weka program handles the NSL-KDD dataset in order to apply the selected algorithm in four different options: use training set, supplied test, cross-validation [19], and percentage test. We first focus on the training set where the previously test files are loaded to train the machine and create the classifier. Then the supplied test set helps us to enter an extern file to evaluate the created model. The cross-validation is the most used since it permits to divide the training set into ten parties, nine of them are for training and the tenth party is for testing. This process is repeated ten times and each time a different party out of the ten is taken for testing. The last option is the percentage split

where it allows the division of the training data into percentage values. The training set is used as a first step to show results in the classifier, then the percentage split will next be used as it is the most suitable to build the model. In other words, the training set is to train and test the algorithm using the same base with the same distributions to insure the validity of our model. The results are shown in four major categories: general measures, accuracy measures by class, confusion matrix, and the ROC curve. In the general measure, the following values are calculated:

Correctly classified instances: Is the number of examples correctly classified in absolute value over the total number of examples.

incorrectly classified instances: Is the number of examples incorrectly classified in absolute value over the total number of examples.

Kappa statistic (K): Helps to measure the degree of agreement between two or more factors. In Weka we have just two components the classifier and the real class of the example as well as judgment is the class of an example. To get the ratio of concordance between two or more factors. K is calculated like:

$$K = P_0 - P_E / 1 - P_E$$

Thus, the proportion of the sample on both factors is the probability of agreeing at random with:

$$Pe = \sum iPi.Pi /n2$$

Where: $-1 \leq K \leq 1$
And: P_i is the sum of the elements of line i and n is the sample size.
As mentioned before the Kappa coefficient takes values between -1 and 1. If the value is maximum so the two judgments are the same but if the value is 0 or -1 therefore the two judgments are independent or in total disagreement respectively. The following Table 5 includes the degree of agreement according to the value of the coefficient:

Level of agreement	Kappa Coefficient
Excellent	>0.81
Good	0.80-0.61
Moderate	0.6-0.41
Poor	0.4-0.21
Bad	0.2-0.0
Very bad	<0

Table 5: Level of agreement based on Kappa coefficient

Next, we will measure the accuracy by measuring the five following variables:

TP Rate = Number of true positives / number of examples in the class
FP Rate = Number of false positives/ number of examples in the class
Precision = Number of true positives / numbers of true positives + number of false positives
Recall = Number of true positives / numbers of true positives + number of false negatives
F-measure = 2 * Recall * Precision / Recall + Precision

For the confusion matrix, it is known as an error matrix, it is represented as a table where each line represents the true class and each column indicates the output of the classifier. Finally, we will trace the ROC (Receiver Operating Characteristics) curve, which indicates the efficiency function of the recipient. This curve allows us to measure the classier performance.

Graphically the curve has two axes, one axis for the true positive rate and the other for the false positive rate. The values of the curve range from 0 to 1, if the curve follows the left border and then the upper limit of the ROC range, the more accurate the test. the closer the curve is to the 45-degree diagonal of the ROC area, the less the test is accurate as shown in figure 6.

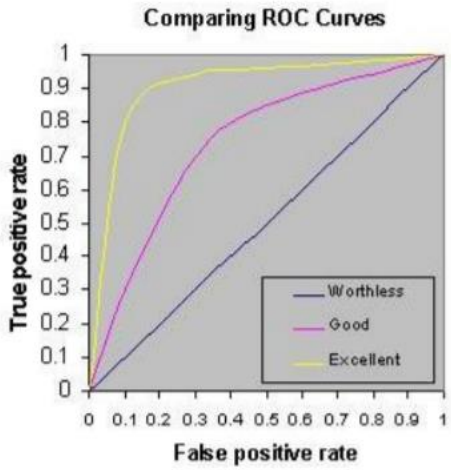

Figure. 6: ROC curve

IV. Results and Discussion:

Various experiments are conducted to evaluate our system, according to the two categories: Binary classification: In the binary classification, we divided the dataset in two classes: normal and abnormal, the model is built with the cross-validation option using the AdaBoost machine learning algorithm divided the dataset and 20% of the NSL-KDD dataset. The following table 6 show results:

Measure	Result
Examples correctly classified	92.02%
Kappa Statistic	83.97%
True Positive Rate	92%
False Positive Rate	8.1%
Accuracy	92%
Recall	92%
F-Measure	92%
ROC Zone	98.5%

Table. 6 AdaBoost Classification Results

Based on results shown in the table, AdaBoost algorithm has a good percentage rate (92.02%) in differentiating between normal and abnormal network traffic as well as an excellent value of Kappa coefficient (83.97%). The true positive rate is higher (92%) compared to false positive (8.1%) and the accuracy and the recall have nearly the same percentage which means predicted normal packets are actually normal. Finally, the ROC zone has a high value (98.5%) thus the model has high performance.
Another model was built as well using the neural networks which is called the Multilayer Perceptron and by using 20% of the NSL-KDD dataset and following are the outputs:

Measure	**Result**
Examples correctly classified	97.08%
Kappa Statistic	94.13%
True Positive Rate	97.1%
False Positive Rate	3.1%
Accuracy	97.1%
Recall	97.1%
F-Measure	97.1%
ROC Zone	99.1%

Table. 7 Multilayer Perceptron Classification Results

The table shows that the Multilayer Perceptron algorithm has better percentage (97.08%) in classifying normal and abnormal network packets and an excellence Kappa coefficient (94.13%) which means the classifier can determine whether network traffic is harmful or benign. The other measurements are far better than the AdaBoost results where rate of true positives is (97.1%), rate of false positives is (3.1%), and the accuracy and recall both have the same value of 97.1%. Finally, the ROC zone is 99.1% which reflects the high performance of this classifier. Based on both tables, it is obvious that there is a shift between the results of the multilayer perceptron and the AdaBoost algorithms, by comparing the percentage rates, ROC zone, and the F-Measure, we note that multilayer perceptron have better success rate compared to AdaBoost algorithm which means it can widely benefit the scanning process of normal and abnormal network packets. Next, we will analyze the application of k-means algorithm using 20% of

NSL-KDD dataset and group results into Classes to clusters evaluation.

Measure	Correctly classified examples	Incorrectly classified examples
Result	69.29%	30.70%

Table. 8 K-Means Classification Results

We note the low rate of incorrectly classified examples compared to the correctly classified ones, which means most examples are classified either normal or abnormal traffic packets, however k-means algorithm performed moderately in filtering multiclass classification. To conclude, based on the three applied algorithms, in a multiclass classification Multilayer Perceptron has best performance compared to K-means and AdaBoost, this means MLP will assure better learning in our model.

Multiclass classification:

In this part, we compare between MLP, AdaBoost, and K-Mean supervised algorithms, the graphs below show instances of correctly and incorrectly classified examples in the binary and multiclass cases:

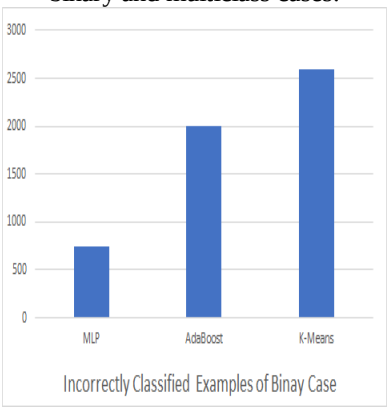

Figure. 7 Incorrectly Classified of Binary

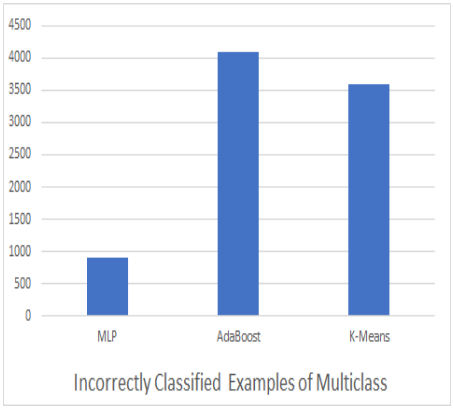

Figure. 8 Incorrectly Classified of Multiclass

The graphs show that MLP algorithm has the biggest number in finding correctly classified examples, whereas we observe the opposite for AdaBoost and K-Means Thus, MLP allows better classification model for instances. Also, we note as it's shown in the accuracy graph below that MLP algorithm has the highest rate in detecting true positives, unlike the other two algorithms AdaBoost and K-Means that have lower detecting rates. On the other hand, MLP has the lowest rate in false positives bypassing the other two compared algorithms.

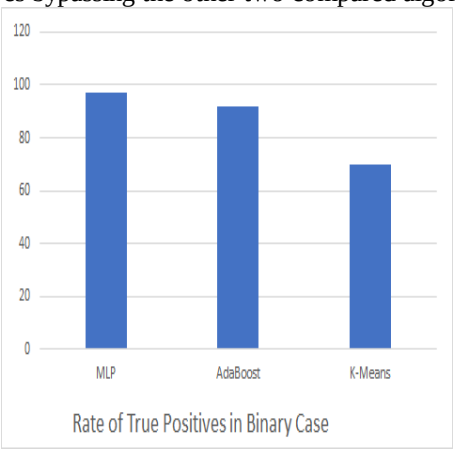

Figure. 9 Rate of True Positives in Binary

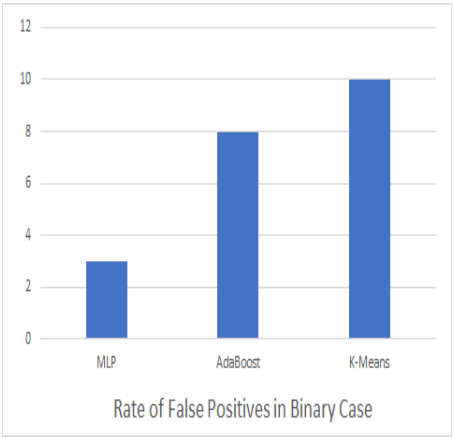

Figure. 10 Rate of False Positives in Binary

Figure. 11 Rate of True Positives in Multiclass

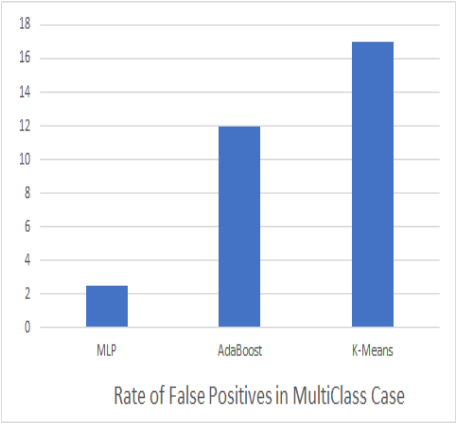

Figure. 12 Rate of False Positives in Multiclass

V. Conclusion:

As security of network becomes crucial to protect sensitive information over the internet, there is a little focus of research made to utilize artificial intelligence in this field especially deep learning. This article proposed a new way to create a network model in detecting intrusions based on machine learning, this technique can help in avoiding network threats caused by cyber-attacks if used with the right algorithm. The experiment showed the promising results in classifying inbound network traffic by an IDS using multilayer perceptron neural network. While the proposed model still needs improvement in terms of accuracy, the founded results in classification for various attacks can be expanded for future work. In our case, by using the Weka platform we prove the efficiency of the multilayer perceptron compared to the other algorithms AdaBoost and K-means with high accuracy rate in detecting true positives and low false positives in both binary and multiclass case. The MLP is chosen over the other algorithms for its outstanding performance as a class feedforward artificial neural network (ANN) algorithm, thus its capabilities to improve the accuracy and recall are applied to solve complex prediction problems such intrusion detection in our case. This shows how how machine learning can play a major role in the field of cyber security by implementing it in the intrusion detection systems to build an automated mechanism to protect the network. The used dataset in this research was manually explored to simulate real network attacks, a future research will focus on using a newer version of the dataset which contains newer attack methods and it will take advantage of cloud computing performance rather than using local computer resources for better optimization. Overall, this research can be used as a starting point for further studies to build a robust real time intrusion detection system using cloud-based machine learning solutions such us azure machine learning by Microsoft or amazon web services from amazon instead of using a trained model of a dataset in a local machine.

References

[1] Jang-Jaccard, Julian, and Surya Nepal. "A survey of emerging threats in cybersecurity." Journal of Computer and System Sciences 80.5 (2014): 973-993.

[2] Liu, Guojie, and Jianbiao Zhang. "CNID: Research of Network Intrusion Detection Based on Convolutional Neural Network." Discrete Dynamics in Nature and Society 2020 (2020).

[3] Panigrahi, Ranjit, and Samarjeet Borah. "A Statistical Analysis of Lazy Classifiers Using Canadian Institute of Cybersecurity Datasets." *Advances in Data Science and Management*. Springer, Singapore, 2020. 215-222.

[4] Jaber, Firas Kh, Faiq MS Al-Zwainy, and Saba W. Hachem. "Optimizing of predictive performance for construction projects utilizing support vector machine technique." *Cogent Engineering* 6.1 (2019): 1685860.

[5] Jain, Rachna, et al. "Assessing risk in life insurance using ensemble learning." Journal of Intelligent & Fuzzy Systems 37.2 (2019): 2969-2980.

[6] Toneva, Diana H., et al. "Data mining for sex estimation based on cranial measurements." Forensic Science International 315 (2020): 110441.

[7] Baglaeva, Elena, et al. "Recognition of chromium distribution features in different urban soils by multilayer perceptron." *AIP Conference Proceedings*. Vol. 2040. No. 1. AIP Publishing LLC, 2018.

[8] Jaffuel, Dany, et al. "Patterns of adaptive servo-ventilation settings in a real-life multicenter study: pay attention to volume!." *Respiratory Research* 21.1 (2020): 1-13.

[9] Hastie, Trevor, et al. "Multi-class adaboost." *Statistics and its Interface* 2.3 (2009): 349-360.

[10] Liu, Guojie, and Jianbiao Zhang. "CNID: Research of Network Intrusion Detection Based on Convolutional Neural Network." *Discrete Dynamics in Nature and Society* 2020 (2020).

[11] Kevric, Jasmin, Samed Jukic, and Abdulhamit Subasi. "An effective combining classifier approach using tree algorithms for network intrusion detection." *Neural Computing and Applications* 28.1 (2017): 1051-1058.

[12] Kevric, Jasmin, Samed Jukic, and Abdulhamit Subasi. "An effective combining classifier approach using tree algorithms for network intrusion detection." *Neural Computing and Applications* 28.1 (2017): 1051-1058.

[13] Devan, Preethi, and Neelu Khare. "An efficient XGBoost–DNN-based classification model for network intrusion detection system." *Neural Computing and Applications* (2020): 1-16.

[14] Cauwenberghs, Gert, and Tomaso Poggio. "Incremental and decremental support vector machine learning." *Advances in neural information processing systems*. 2001.

[15] Rish, Irina. "An empirical study of the naive Bayes classifier." *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. No. 22. 2001.

[16] Menard, Scott. *Applied logistic regression analysis*. Vol. 106. Sage, 2002.

[17] Kunhare, Nilesh, Ritu Tiwari, and Joydip Dhar. "Particle swarm optimization and feature selection for intrusion detection system." *Sadhana* 45.1 (2020).

[18] Tama, Bayu Adhi, and Kyung-Hyune Rhee. "An in-depth experimental study of anomaly detection using gradient boosted machine." *Neural Computing and Applications* 31.4 (2019): 955-965.

[19] Zhou, Jian, Xiuzhi Shi, and Xibing Li. "Utilizing gradient boosted machine for the prediction of damage to residential structures owing to blasting vibrations of open pit mining." *Journal of Vibration and Control* 22.19 (2016): 3986-3997.

[20] Svetnik, Vladimir, et al. "Random forest: a classification and regression tool for compound classification and QSAR modeling." *Journal of chemical information and computer sciences* 43.6 (2003): 1947-1958.

[21] Aljawarneh, Shadi, Muneer Bani Yassein, and Mohammed Aljundi. "An enhanced J48 classification algorithm for the anomaly intrusion detection systems." *Cluster Computing* 22.5 (2019): 10549-10565.

[22] Dhanabal, L., and S. P. Shantharajah. "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms." *International Journal of Advanced Research in Computer and Communication Engineering* 4.6 (2015): 446-452.

[23] Belouch, Mustapha, Salah Elhadaj, and Mohamed Idhammad. "A hybrid filter-wrapper feature selection method for DDoS detection in cloud computing." *Intelligent Data Analysis* 22.6 (2018): 1209-1226.

[24] Shamshirband, Shahaboddin, et al. "Co-FQL: Anomaly detection using cooperative fuzzy Q-learning in network." *Journal of Intelligent & Fuzzy Systems* 28.3 (2015): 1345-1357.

[25] Yi, Zeren, et al. "A navigation method for mobile robots using interval type-2 fuzzy neural network fitting Q-learning in unknown environments." *Journal of Intelligent & Fuzzy Systems* 37.1 (2019): 1113-1121.

[26] Hachmi, Fatma, Khadouja Boujenfa, and Mohamed Limam. "Enhancing the accuracy of intrusion detection systems by reducing the rates of false positives and false negatives through multi-objective optimization." *Journal of Network and Systems Management* 27.1 (2019): 93-120.

[27] Castillo, Oscar, et al. "Fuzzy parameter adaptation in genetic algorithms for the optimization of fuzzy integrators in modular neural networks for multimodal biometry." Computación y Sistemas 24.3 (2020).

[28] Sánchez, Daniela, Patricia Melin, and Oscar Castillo. "Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular granular neural networks for human recognition." J. Intell. Fuzzy Syst. 38.3 (2020): 3229-3252.

[29] Rawat, Shisrut, and Aishwarya Srinivasan. "Intrusion detection systems using classical machine learning techniques versus integrated unsupervised feature learning and deep neural network." arXiv preprint arXiv:1910.01114 (2019).

[30] Chouhan, Naveed, and Asifullah Khan. "Network anomaly detection using channel boosted and residual learning based deep convolutional neural network." Applied Soft Computing 83 (2019): 105612.

[31] Van, Nguyen Thanh, and Tran Ngoc Thinh. "An anomaly-based network intrusion detection system using deep learning." 2017 international conference on system science and engineering (ICSSE). IEEE, 2017.

[32] Cui, Jianjing, et al. "Wedl-nids: improving network intrusion detection using word embedding-based deep learning method." International Conference on Modeling Decisions for Artificial Intelligence. Springer, Cham, 2018.

[33] Akar, Özlem, and Oguz Güngör. "Classification of multispectral images using Random Forest algorithm." Journal of Geodesy and Geoinformation 1.2 (2012): 105-112.

[34] Aljawarneh, Shadi, Muneer Bani Yassein, and Mohammed Aljundi. "An enhanced J48 classification algorithm for the anomaly intrusion detection systems." Cluster Computing 22.5 (2019): 10549-10565.

[35] Yao, Yukai, et al. "K-SVM: An Effective SVM Algorithm Based on K-means Clustering." JCP 8.10 (2013): 2632-2639.

[36] Bel, Liliane, et al. "CART algorithm for spatial data: Application to environmental and ecological data." Computational Statistics & Data Analysis 53.8 (2009): 3082-3093.

[37] Chen, Shenglei, et al. "A novel selective naïve Bayes algorithm." Knowledge-Based Systems 192 (2020): 105361.

[38] Cui, Ting, et al. "Data filtering-based parameter and state estimation algorithms for state-space systems disturbed by coloured noises." International Journal of Systems Science (2020): 1-16.

[39] Hu, Bo, Lvqing Bi, and Songsong Dai. "Information distances versus entropy metric." Entropy 19.6 (2017): 260.