Journal of Statistics Applications & Probability
An International Journal

http://dx.doi.org/10.18576/jsap/130112

Intrusion Detection System Employing Neural Network MLP and Detection Trees Using Different Techniques

M. H. Al-Mashagbeh[1,], W. Salameh [2], A. B. Alamareen[3], and S. Abu asal [4]*

[1]Cyber Security Department, Faculty of Information Technology, Zarqa University, Zarqa, Jordan
[2]Computer Science Department, Faculty of Computing Sciences, Princess Sumaya University for Technology, Amman, Jordan
[3]Computer Science Department, Faculty of Information Technology, Zarqa University, Zarqa, Jordan
[4]Software Engineering Department, Faculty of Information Technology, Zarqa University, Zarqa, Jordan

Received: 2 Mar. 2023 Revised: 25 Apr. 2023, Accepted: 27 Apr. 2023
Published online: 1 Jan. 2024.

Abstract: by addressing intruder attacks, network security experts work to maintain services available at all times. The Intrusion Detection System (IDS) is one of the available mechanisms for detecting and classifying any abnormal behavior. As a result, the IDS must always be up to date with the most recent intruder attack signatures to maintain the confidentiality, integrity, and availability of the services. This paper shows how the NSL-KDD dataset may be used to test and evaluate various Machine Learning techniques. It focuses mostly on the NLS-KDD pre-processing step to create an acceptable and balanced experimental data set to improve accuracy and minimize false positives. For this study, the approaches J48 and MLP were employed. The Decision Trees classifier has been demonstrated to have the highest accuracy rate for detecting and categorizing all NSL-KDD dataset attacks.
Keywords: IDS, machine learning, NSL-KDD, learning.

1 Introduction

The Internet, telephone, e-commerce, and PC-based connectivity have all become part of everyday life as a result of the world's rapid technological advancements and networking over the last two decades. While widespread use of these systems improves communication, data transmission, and information sharing while also improving life quality, they have many security vulnerabilities and are subject to numerous Viruses, worms, and Trojan horses are examples of cyber-attacks [1].

Hence, given the accelerating development in technology and the increase of web application usage which needs to be linked to the Internet, the web application security requirements are increasing. The network manager's job is to protect users' accounts, services, passwords, and personal information from any attack. There are four types of attacks that target web applications DOS and DDOS attacks, through which the attacker aims to disturb the service for various reasons that differ according to the hacker and his motivations, whether financial, political, etc. The R2L is the second, and the U2R is the third. Last but not least, the attacker checks the network for any Design flaws using (nmap, pin sweep, and others)[1].

Usually, Denial of service (DOS) attacks target two main layers of the network, which are the network layer and the application layer, through the consumption of resources and thus the inability of the server to respond to any new requests from legitimate users. The most famous types of this attack are User Datagram Protocol Flood, Structured Query Language injection Denial of service, Smurf attack, and HTTP Flood [2].

Intrusion detection systems are the most effective defense against these attacks. A network monitoring device that looks for harmful activities or policy infractions is known as an intrusion detection system. An administrator or (SIEM) device is typically notified of any intrusion activity or breach, which may monitor incoming and outgoing data on a network, perform analysis based on time, and alert when an intrusion is discovered [2]

IDS is an expression for "Intelligent The neural network" is a data mining method that has proven to be effective at resolving complex practical issues. Neural networks (artificial intelligence) have the potential to address several issues that the other current intrusion detection technique faces. There are three benefits of using a neural network to detect intrusions [3]:

- Since it can evaluate and check whether data is right or partially correct, the neural network offers flexibility in the intrusion detection process.
- A neural network can process data in a non-linear fashion from a variety of sources. This is particularly critical when multiple attackers launch a concerted attack against the network.
- The speed at which a neural network processes data distinguishes it.[3]

*Corresponding author e-mail: malmashagbeh@zu.edu.jo

The research paper used the NSL-KDD dataset, which was prepared in a scientific way, and its version of the KDD99cup, which was developed in 2009. Using multiple machine learning (ML) algorithms Neural Networks (Multi-Layer Perceptron -MLP) and Decision Trees And finally, examining the ability of the system to detect these attacks after it has been trained.

The KDD 99 dataset is a widely used classifier accuracy test. KDD-99, on the other hand, has many drawbacks, including data age, and completely imbalanced targets. There is non-stationarity between the training and test datasets is also a problem. Several countermeasures have been devised by other researchers to counteract these flaws. As a result, they launch the NSL-KDD data set, a more balanced resampling of the KDD-99 data set [4].

The following is how the study was structured: The first section was an introduction, In section 2, it was about background. In section 3, we present the related work. The goal and objectives are defined in section 4. The Materials and Methods is described in section 5, followed by the results in section 6, a performance comparison in Section 7, and finally the conclusions and future work of this study in section 8.

2 Background

A. Machine Learning: An Overview

It is a subfield of Artificial Intelligence concerned with creating algorithms that enable computers to learn. Builds a data-driven account modem known as "training data," which is focused on making predictions to complete tasks.[5]

B. The Type of Machine Learning

Supervised Learning Method: In supervised learning, all prior data and previous outcomes must be used as feedback (the data must be labeled) and divided into classes. This data is used to train the model, which allows it to forecast more accurate outcomes. [5] As an example, if an algorithm needs to distinguish between cars in a set, the data must be labeled or categorized for each car. There are many algorithms like classification, regression, naïve Bayes theorem, SVM, KNN, decision tree, MLP, etc. [6].

The other type of machine learning Method is:

- Unsupervised Learning Method
- The Semi-supervised Learning Approach
- Reinforcement Learning Method.

C. Intrusion detection systems (IDS):

In 1986, Dorothy E. Denning and colleagues developed the first IDS .who worked for the SRI International Research organization. As a result, he's a software program that scans the network for malicious behavior, interference, or regulation, and then collects and reports the information using Security Information Management (SIEM) these crimes are divided into two categories:

1. Malicious conduct.
2. Alarms that aren't true.

It is the responsibility of the IDS to observe user behavior and, using pre-established rules or models, identify it as either normal or abnormal.[7]

The two most common types of intrusion detection systems are signature and anomaly intrusion detection systems.[8]

As a result, various forms of intrusion detection systems apply various methods:

- Intrusion detection system for networks (NIDS)
- Intrusion detection system for Hosts (HIDS)
- Intrusion detection method based on signatures (SIDS)
- Intrusion detection method focused on anomalies (AIDS)[8]

D. IN IDS, Data Extraction Methods

Because of the huge complexity and scale of datasets, the range of ML and data mining methods did not work well with

IDS. Because the network contains a huge number of features, these methods take a long time to distinguish threats, making deployment in real-time environments more difficult.

E. The Method that used

1. Neural Network (Multi-Layer Perceptron -MLP)

MLPs are not-linear statistics modeling tools. It is used for complexity construction relationships between the entry layer and outputs or to find patterns in data, The MLP is a neural network class feed forward that has three layers of nodes: input, hidden, and output [9].

2. Decision Tree (J48,C4.5)

A decision tree It is based on the division of space to enter data into exclusive and reciprocal areas which is a transparent mechanism where we can follow the structure of the tree easily to see how the decision-making process is done.[10].

3 Related Work

Many researchers used Ml in a field IDS and implemented many of the algorithms to improve results and accuracy, whereas scientists used the well-known and open database NSL-KDD, To achieve their scientific goal of improving intrusion detection.

For the NSL-KDD dataset [11], they propose using an IDS based on decision trees. For feature selection, the suggested research uses the Correlation Feature Selection (CFS) approach using Weka. The performance of IDS based on decision trees improves with feature selection. For five classes (normal and types of intrusions) and binary class (normal and attack) that give each one number to change a symbolic value to a numerical value, performance is tested before and after feature selection. The result obtained is contrasted and assessed against the other procedures that have been described. The suggested IDS based on decision trees has a high detection rate and accuracy when applied on MATLAB programming , according to the research, the result shows after following the selection of features for two classes Accuracy of detection is 90.3%.

On the other side, [12] they use deep learning to create an IDS model. They build an IDS model based on Convolution Neural Networks(CNN) instead of the typical ML employed in prior projects to perform better in extracting features of huge data considering the massive cyber traffic in real life, They compare the results with classic ML methods (Random Forest and Support Vector Machine) as well as DL approaches (Belief Network and Long Short Term Memory) they apply it using the Py Charm on NSL-KDD, and They increase IDS accuracy by 80.1321 % on KDD Test and open up a new research area for IDS.

Since this is an important and complex field with many algorithms and areas, researchers [13] for identifying key input properties in the development of a statistically efficient and effective IDS They test the effectiveness of common feature selection approaches (Correlation-based Feature Selection, Information Gain and Gain Ratio), As a result, they suggest a new feature selection approach based on the feature average of all classes. They evaluate feature reduction methods using the efficient classifier decision tree technique using Weka on NSL-KDD. After comparing the proposed approach to other methods, The results reveal that employing 22 characteristics, the highest accuracy is 99.794 %. As a result, the method's accuracy is higher than that of full data and comparable to that of other methods.

The researchers created a new IDS method based on a deep neural network (NDNN) model [14] they used the same Database (KDD99cup) and NSL KDD In order to increase network intrusion detection intelligence and accuracy while reducing false alarms. Experiments on the datasets KDD99 and NSL-KDD show that the NDNN-based strategy enhances the intrusion detection system's (IDS) efficiency, with an accuracy rate of up to 99.9%.

In [15] They used a variety of algorithms on KDD and NSL-KDD, including (Logistic Regression, Decision Tree, KNN, SVM, Random Forest, Multi-Layer Perceptron, Ada Boost, Naive Bayes) and Feature Extraction, which was based on other research and involved assigning a number to each label based on a specific methodology. The KNN has a high false prediction accuracy, according to the results, and a high false value, and the Ada Boost algorithm has a low false-positive rate of less than 1% and a detection rate of more than 80%.

They attempt to build a model for IDS using a random forest classifier in this paper [16]. Random Forest is an ensemble classifier that uses numerous decision trees to boost accuracy and has a low classification error when compared to other standard classification techniques that apply using NSL-KDD. They execute a feature selection for a pre-processing step that reduces and removes extraneous information to improve accuracy using Weka and compare the result with j48 which shows accuracy is better at 99.67%.

In [17] they offer an intelligent NIDS that uses the AODE algorithm to detect various forms of attacks using NSL-KDD. AODE is a recent modification to the naïve Bayes algorithm that overcomes the independence problem by averaging all models generated by the classic one-dependency estimator and is well suited for incremental learning. They used Weka to perform the experimental research and they achieved 96.64% accuracy better than naïve Bayes with only 90.28%.

Assist the user in achieving high accuracy in the classification process, the goals are:

1. A range of attacks, known and unknown, were identified as quickly and briefly as possible.
2. Providing a simple and quick translation for human understanding (readable).
3. Reduce the number of false positives and false negatives by achieving precision.
4. Reduce the time for the attack to be checked and searched.
5. Evaluate our model using MLP, Decision tree.

4 Materials and Methods

Step 1: Select the algorithms

We taught machine learning algorithms first, then read a series of studies to determine which algorithms are the best, and last tested these algorithms on the WEKA to see which algorithms work together, thus applying normalization first and then utilizing algorithms:

- MLP algorithm.
- Decision tree (J48, C4.5).

Step 2: Process of gathering data

We will use NSL-KDD in this experience. It is a developed version of the KDD99cup, which was developed in 2009. Respect to It is greater than the KDD99cup in terms of:

- Does not contain duplicate train test data and documents where duplicate values would not be unbiased during classification.
- Does not include duplicate data in the test set because the output is not unbiased and detection rates are higher.
- The percentage of records commonly used for rating processes is inversely proportional to the number of records chosen.
- Most experiences can be conducted in various and detailed ways.
- Because research can be done by the entire community rather than just a subset of it, the number of records in the train test and test set is fair and proportionate.[18]
-

Step 3: Overview of the dataset

The dataset was made up of server files:

- KDD-Train+.ARFF: all NSL-KDDs found in the ARFF format (binary labels)
- KDD-Train+.TXT: contains all NSL-KDD including the CSV format (attack-type labels and others)
- KDD-Train+ 20 %.ARFF: A KDD-Train+.arff file subset of 20 %
- KDD-Train+ 20%.TXT is a 20% subset of the KDDTrain+.txt file.
- KDD-Test+.ARFF: all NSL-KDDs found in the ARFF format (binary labels)
- KDD-Test+.TXT: contains all NSL-KDD including the CSV format (attack-type labels and others)
- KDD-Test-21.ARFF is a subset of the KDD-Test+.arff file that excludes records from 21 of the 21 difficulty levels.
- KDDTest-21.TXT: a subset of the KDDTest+.txt file that does not contain 21 out of 21 difficulty-level records[19]

We mentioned that the number of duplicate records and data is one of the most significant problems and drawbacks in KDD 99 cup, so NSL-KDD was developed and the researchers carried out the experiments randomly at 3 subgroups to achieve this result, Each containing 50,000 A record of data where 21 machine learning have been used (7) for learning and

each is trained 3 times to get rid of Repeat and give a mark to each record [20].

Step 4: Experimental

In this study, we apply machine learning frameworks and execute the experiment on WEKA. In the NSL-KDD dataset, we designed an experiment to study the training and testing of five-class classification using MLP and decision trees. To see how it stacks up against other machine learning algorithms that have been available for a while.

Weka is a Machine learning toolset. It was created at New Zealand's University of Waikato. It implements a wide range of state-of-the-art machine-learning techniques that are developed in Java. WEKA includes tools for regression, classification, clustering, and association rules, among other data mining techniques, as well as many visualization tools, in this interface, which includes The Experimenter, the Explorer, the Simple Command Line Interface, and the Knowledge Flow [21].

Step 5: Feature extraction

To improve feature classification, quantity, and efficiency, if the number of features chosen is small, classification quality will suffer, and if they are in excess of what is expected, generalization will suffer. Experiments show that using feature extraction techniques improves accuracy and reduces computational cost so the Feature Extraction: is Samples are represented by rows, whereas characteristics are represented by columns, with features resulting from both quantitative and subjective outcomes in a dataset. Feature extraction is a technique for reducing a data set's dimensionality by reducing the number of features while preserving attack detection precision and reducing discovery time.[22]

Step 5: Numericalization

There are 38 numeric features and three non-numeric features in the NSL-KDD dataset. Because the input value must be a numeric matrix for greater precision, we must convert certain non-numeric qualities, such as "protocol type," "operation," and "flag," into numeric form attributes. The 'protocol type' function, for example, has three attribute types: 'TCP,' 'UDP,' and 'ICMP,' and its numerical values are encoded as binary vectors (1,0,0), (0,1,0), and (0,1,0), respectively (0,0,1). After transformation, the 'service' function has 70 attribute kinds, and the 'flag' feature has 11 attribute types. We filter the data using Weka, increasing the number of characteristics from 42 to 123 to improve accuracy [23][24] Fig 1 Shows the normalization process using Weka.

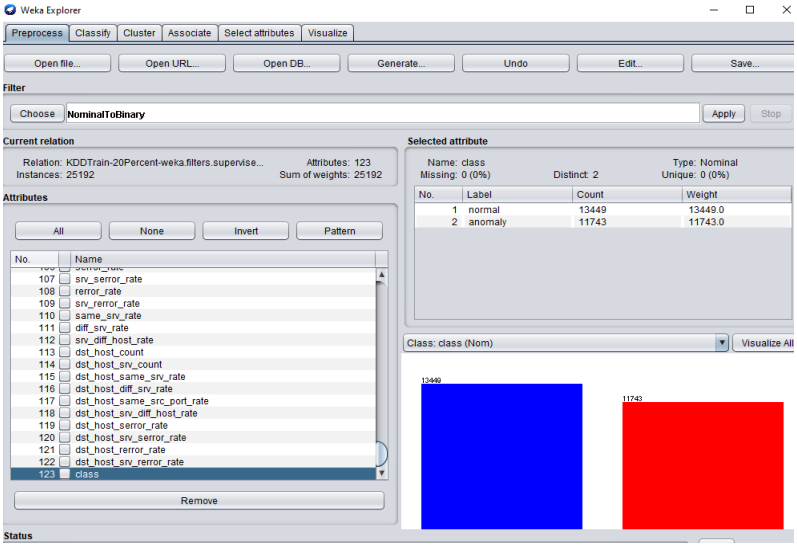

Fig. 1: Show normalization process using weka

Step 6: Evaluation Metric

The accuracy of intrusion detection is the most essential performance indicator, and it is used to evaluate the NSL-KDD model's performance. In addition to accuracy, we considered the detection rate and false positive rate. In the field of intrusion detection, the following notation is used:

In Equation (1), The percentage of correctly classified records in relation to the total number of records is known as accuracy.[25]

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

Weka can also be used to establish sensitivity (positive value ratio) and specificity (negative value ratio). [25]:

$$\text{Sensitivity:}\frac{t-pos}{pos} \quad (2)$$

$$\text{Specificity:}\frac{t-neg}{neg} \quad (3)$$

So the Accuracy is:

$$\text{Sensitivity:}\frac{t-pos}{pos}+\text{Specificity:}\frac{t-neg}{neg} \quad (4)$$

So:

T-pos: the correct number of classifications (True Positives) Pos: the right number of classifications (positive)

T-neg: the number of properly categorized (genuine negatives) Neg: the number of correctly categorized negatives

When compared to the total number of anomaly records in Equation (5), the True Positive Rate (TPR), also known as the Detection Rate (DR), is the proportion of anomaly records correctly detected as anomaly .[25]

$$DR = TPR = \frac{TP}{TP + FN} \quad (5)$$

For kappa statistics (Cohen kappa coefficient) , a kappa of 1 denotes complete agreement, whereas a kappa of 0 denotes agreement that is random. One drawback of kappa is that it is influenced by the prevalence of the observational finding, sTo compute it, we use the equation below:

$$k = \frac{p_{o-Pe}}{1-p_e} = 1 - \frac{1-p_o}{1-p_e} \quad (6)$$

Po is the raters' observed relative agreement.

Pe denotes the raters' expected relative agreement..[25]

A False Positive Rate (FPR)is calculated by dividing the percentage of normal records that are wrongly identified as anomalies by the total number of normal records in Equation (7) [26]:

$$FPR = \frac{FP}{FP + TN} \quad (7)$$

The Recall (Rc) or Sensitivity, as well as the Precision (Pr) or Predictive Positive Value, are determined as a consequence. The proportion of correctly identified instances (TP) in the front of c (TP+FP) classified instances is shown by the accuracy. Assume that z is at the forefront of all z occurrences (TP+FN), and that recall is the proportion of correctly classified instances [27].

5 Results

We used one of the Weka (preprocess) characteristics to display the user data acquired in the previous stage after performing normalization. To determine the program's accuracy and inaccuracy, as well as the time it took to accomplish the assignment, the MLP and J48 algorithms were employed.

So we apply the algorithm (MLP and J48) in (KDD Test +, KDD Test 21, KDD Train +, and KDD Train 21) To determine the Training Set Results as well as the Training and Simulation Error, we calculate the correctly classified, wrongly categorized, and Kappa statistics for each algorithm. The correctly categorized is the genuine correct classification, and success is measured by the test community's ranking for each case, where real classifications are compared to determine accuracy and the training set is split to 66 for training and the remainder for testing.

We compute data error measures (KDD Test and KDD Train) in each algorithm to determine how long the actual value is expected to last. To calculate training and simulation error, use (squared error, absolute error, relative squared error, and relative absolute error) with MLP and J48 using the following equation :

$$\text{Absolute error: } |y_i - \grave{y_i}| \quad (8)$$

$$\text{Squared error: } (y_i - \grave{y_i})^2 \quad (9)$$

Were:

yi: the error measures' lost functions.

yi: anticipated outcome .[27]

$$\text{Relative absolute error : } \frac{\sum_{i=1}^{d} |y_i - \grave{y_i}|}{\sum_{i=1}^{d} |y_i - \bar{y}|} \quad (10)$$

$$\text{Relative squared error : } \frac{\sum_{i=1}^{d} (y_i - \grave{y_i})^2}{\sum_{i=1}^{d} (y_i - \bar{y})^2} \quad (11)$$

Where:

yi: is the training data of yi`

yi`: is the mean value of yi.

$\grave{y}: \frac{\sum_{i=1}^{t} y_i}{d}$ [28]

Table 1: Show Result of the Training Set for MLP and J48 in KDD-Test +

ALG	Correctly classified	Incorrectly classified	Kappa statistic
MLP	96.347	3.653	0.9254
J48	98.304	1.696	0.9654

Table 1 shows the Training Set Results for MLP and J48 in KDD-Test +. It can be seen that J48 has given more accurate values than MLP since the Decision tree C4.5 method has attributes that affect the total attribute is correctly identified, which was the case with MLP (98.304), but the most effective attributes were the number of failure times, which was (1.696), The number of failure times (1.696), which was incorrectly classified, was the most effective attribute. It's worth noting that the Kappa statistic was (0.9654), which could be interpreted as a reasonable indication of the extent to which the data collected in the study are accurate representations of the variables examined. For the record, the tree had 117 leaves

with a size tree 233.

Table 2: Show Result of the Training Set for MLP and J48 in KDD-Test -21

ALG	Correctly classified	Incorrectly classified	Kappa statistic
MLP	93.0504	6.9496	0.7437
J48	97.6669	2.3331	0.9221

Table 2 shows the Training Set Results for MLP and J48 in KDD-Test -21. It can be seen that J48 has given more accurate values than MLP since the Decision tree C4.5 method has attributes that affect the total attribute is correctly identified, which was the case with MLP (97.669), but the most effective attributes were the number of failure times, which was (2.3331), The number of failure times (2.3331), which was incorrectly classified, was the most effective attribute. It's worth noting that the Kappa statistic was (0.9221), which could be interpreted as a reasonable indication of the extent to which the data collected in the study are accurate representations of the variables examined. For the record, the tree had 117 leaves with a size tree 233.

Table 3: Show Errors in training and simulation for MLP and J48 in KDD-Test +

AlG	Mean absolute error	Root mean squared error	Relative Absolute error (%)	Root relative squared error (%)
MLP	0.0472	0.1777	9.6353	35.8863
J48	0.0197	0.1183	4.0245	23.8942

In Table 3, the Errors in training and simulation for MLP and J48 in KDD-Test +, which refers to the results of measuring the difference between two continuous variables, are shown (absolute values and Prediction error), The Mean Absolute Error, or MAE, is one of the best metrics for summarizing and assessing the quality of a machine learning mode, with J48 having the best result (0.0197). The Root Mean Squared Error, which is the Average Squared Difference between the Estimated Values and the Actual Value, on the other hand, was (0.1183), and the Relative Absolute Error was (4.0245) Root relative squared error (23.8942) was the most important component, which "normalizes the total squared error by dividing it by the default predictor's total squared error." This matches Weka's approach.

Table 4: Show Errors in training and simulation for MLP and J48 in KDD-Test -21

ALG	Mean absolute error	Root mean squared error	Relative Absolute error (%)	Root relative squared error (%)
MLP	0.0752	0.2518	25.2782	65.2634
J48	0.0312	0.139	10.4904	36.0193

In Table 4 When comparing J48 to MLP to show Errors in training and simulation for MLP and J48 in KDD-Test -21, which refers to the outcomes of measuring the difference between two continuous variables(absolute values and Prediction error), the Mean Absolute Error, commonly known as MAE, is one of the best many metrics for summarizing and assessing the quality of a machine learning model that shows in J48 has the best result (0.0312), The Root Mean Squared Error which the Average Squared Difference between the Estimated Values and the Actual Value, on the other hand, was (0.139), the Relative Absolute error was (10.4904) The most important component was Root relative squared error (36.0193), which

takes the total squared error and normalizes it by dividing it by the default predictor's total squared error." This corresponds to Weka's implementation.

Table 5: Show Result of the Training Set for MLP and J48 in KDD-Train +

ALG	Correctly classified	Incorrectly classified	Kappa statistic
MLP	98.7602	1.2398	0.9751
J48	99.8366	0.1634	0.9967

Table 5 represents the Training Set Result for MLP and J48 in KDD-Train + it can be noticed that using J48 has given more accurate values than MLP since the Decision tree C4.5 algorithm have the qualities that affect the overall attribute is Correctly classified, which (99.8366), but the most effective attributes were the number of failure times, which was (0.1634), which was incorrectly categorized, It's worth noting that the Kappa statistic was (0.9967), which might be seen as a reasonable indication of the extent to which the data collected in the study are accurate representations of the variables examined. And for the record the number of leaves was 113 with a size tree of 255.

Table 6: Show Errors in training and simulation for MLP and J48 in KDD-Train+

ALG	Mean absolute error	Root mean squared error	Relative Absolute error (%)	Root relative squared error (%)
MLP	0.0149	0.1007	2.998	20.1812
J48	0.0025	0.0395	0.5114	7.9093

In Table 6 , the Errors in training and simulation for MLP and J48 in KDD-Test +, which refers to the results of measuring the difference between two continuous variables, are compared (absolute values and Prediction error), The Mean Absolute Error, or MAE, is one of the best metrics for summarizing and assessing the quality of a machine learning mode, with J48 having the best result (0.0025). The Root Mean Squared Error, or the Average Squared Difference between the Estimated Values and the Actual Value, on the other hand, had the worst result (0.0395), The most crucial component was the Root relative squared error (7.9093), which takes the total squared error and normalizes it by dividing it by the total squared error of the default predictor." This conforms to Weka's implementation.

Table 7: Show Result of the Training Set for MLP and J48 in KDD-Train -21

ALG	Correctly classified	Incorrectly classified	Kappa statistic
MLP	98.4822	1.5178	0.9695
J48	99.603	0.397	0.992

Table 7 shows the Training Set Result for MLP and J48 in KDD-Train-21. It can be seen that J48 has given more accurate values than MLP because the Decision tree C4.5 method contains qualities that affect the overall classification of the attribute, which was correctly classified (99.603), The number of failure instances (0.397), which was mistakenly classified, was the most effective attribute. Worth It's noting that the Kappa statistic was (0.992), This could be interpreted as a reasonable indicator of the degree to which the data acquired in the study are accurate representations of the variables under investigation. For the record, the tree had 71 leaves with a size of tree is 141.

Table 8: Show Errors in training and simulation for MLP and J48 in KDD-Train -21

ALG	Mean absolute error	Root mean squared error	Relative Absolute error (%)	Root relative squared error (%)
MLP	0.0199	0.1118	4.009	22.4304
J48	0.0054	0.0625	1.0914	12.5383

When comparing J48 to MLP in KDD-Train -21, which refers to the results of evaluating the difference between two continuous variables, Errors in training and simulation for MLP and J48 (absolute values and Prediction error), The Mean Absolute Error, also known as MAE, is one of the best metrics for summarizing and assessing the quality of a machine learning mode, with J48 having the best result (0.0054). On the other hand, the Root Mean Squared Error, which is the Average Squared Difference between the Estimated Values and the Actual Value, had the worst result (0.0625), The most essential component was the Root relative squared error (12.5383), which takes the total squared error and normalizes it by dividing it by the total squared error of the default predictor." This conforms to Weka's implementation.

6 Performance Comparisons

To objectively test the effect of our strategy, the test results are compared to data from other studies. Table 9 shows our experience with the J48 machine learning algorithm in Weka, which uses an effective intrusion detection strategy, and our experience delivers the best accuracy on the (KDD Test +, KDD Test 21, KDD Train +, and KDD Train 21 datasets).

Table 9: Compares the results of the supplied method to the findings of other existing data performance studies.

scholars	Algorithm	Data Set	Accuracy
Our model	J48	KDD Test +	98.304%
Our model	J48	KDD Test 21	97.6669%
Our model	J48	KDD Train +	99.8366%
Our model	J48	KDD Train 21	99.603%
Ding et al	Cnn	41 features NSL-Kdd	80.13%
Yin et al	RNN	41 features mapping in to 122 NSL-Kdd	83.28%
XIANWEI GAO ,at al,2019	Ensemble Voting	KDDTest+	85.2%
Kehewu ,et al,2018	CNN	KDDTest+	79.48%
Wu et al	CNN,RNN	41 features NSL-Kdd	79.48%
Aloqaily et al	Deep Belief network and J48	41 features mapping in to 122 NSL-Kdd	99.43%
Al-qatf M,et al,2018	SAE SVM	KDDTest+	84.96%

7 Conclusions and Future work

To improve performance, we used MLP and J48 to identify incursions in this paper. In terms of accuracy Kappa statistic and low Relative Absolute error, we used the NSL-KDD dataset, which contains intrusion, benign, and other attributes, and the results were correctly classified (98.304 %) in KDD-Test and (97.6669 %) in KDD-Test-21, (99.8366 %) in KDD-Train+, and (99.603 %) in KDD-Train-21. where the outcomes were better when J48 was used, Tables were created as part of the project to show the algorithms we used and the results we acquired so that other researchers can duplicate the experiment using the same information. For future projects, we intend to re-experiment with different algorithms. and we'll focus more on lowering the rate of error and improving accuracy, as well as applying it to better applications and filtering the data, and removing those aspects in order to improve performance.

Acknowledgments

We would like to show our gratitude to everyone who helped us complete this project in this manner. They really aided in our learning about this area and in shaping the current state of our research. We are very appreciative of all of them.

Conflicts of Interest Statement

The authors certify that they are NOT associated with or actively involved in any group or entity that has an interest in the topics or resources covered in this paper.

References

[1] I. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin, and M. I. AlZubi, “Intensive Pre-Processing of KDD Cup 99 for Network Intrusion Classification Using Machine Learning Techniques,” International Journal of Interactive Mobile Technologies (iJIM), vol. 13, no. 01, p. 70, Jan. 2019, doi: 10.3991/ijim.v13i01.9679.

[2] A. Verma and V. Ranga, “Statistical analysis of CIDDS-001 dataset for Network Intrusion Detection Systems using Distance-based Machine Learning,” Procedia Computer Science, vol. 125, pp. 709–716, 2018, doi: 10.1016/j.procs.2017.12.091.

[3] S. M. Mehibs and S. H. Hashim, “Proposed Network Intrusion Detection System In Cloud Environment Based on Back Propagation Neural Network,” JOURNAL OF UNIVERSITY OF BABYLON for Pure and Applied Sciences, vol. 26, no. 1, pp. 29–40, Dec. 2017, doi: 10.29196/jub.v26i1.351.

[4] C. A. M. and R. K., “Performance evaluation of data clustering techniques using KDD Cup-99 Intrusion detection data set,” International Journal of Information and Network Security (IJINS), vol. 1, no. 4, Sep. 2012, doi: 10.11591/ijins.v1i4.821.

[5] H. Liu and B. Lang, “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,” Applied Sciences, vol. 9, no. 20, p. 4396, Oct. 2019, doi: 10.3390/app9204396.

[6] A. C. M. Fong and G. Hong, “Boosted Supervised Intensional Learning Supported by Unsupervised Learning,” International Journal of Machine Learning and Computing, vol. 11, no. 2, pp. 98–102, Mar. 2021, doi: 10.18178/ijmlc.2021.11.2.1020.

[7] F. Mokbal, W. Dan, M. Osman, Y. Ping, and S. Alsamhi, “An Efficient Intrusion Detection Framework Based on Embedding Feature Selection and Ensemble Learning Technique,” The International Arab Journal of Information Technology, vol. 19, no. 2, 2022, doi: 10.34028/iajit/19/2/11.

[8] S. Shakya, “MACHINE LEARNING BASED NONLINEARITY DETERMINATION FOR OPTICAL FIBER COMMUNICATION-REVIEW,” Journal of Ubiquitous Computing and Communication Technologies, vol. 2019, no. 02, pp. 121–127, Dec. 2019, doi: 10.36548/juckt.2019.2.006.

[9] I. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin, and M. I. AlZubi, “Intensive Pre-Processing of KDD Cup 99 for Network Intrusion Classification Using Machine Learning Techniques,” International Journal of Interactive Mobile Technologies (iJIM), vol. 13, no. 01, p. 70, Jan. 2019, doi: 10.3991/ijim.v13i01.9679.

[10] M. Faisal Elrawy, T. K. Abdelhamid, and A. M. Mohamed, “IDS in Telecommunication Network using PCA,” International journal of Computer Networks & Communications, vol. 5, no. 4, pp. 147–157, Jun. 2013, doi: 10.5121/ijcnc.2013.5412.

[11] S. Sonawane, “Rule Based Learning Intrusion Detection System Using KDD and NSL KDD Dataset,” Prestige

International Journal of Management & IT - Sanchayan, vol. 04, no. 02, pp. 135–145, Dec. 2015, doi: 10.37922/pijmit.2015.v04i02.009.

[12] Ding, Y. and Zhai, Y.“Enhanced Network Intrusion Detection using Deep Convolutional Neural Networks,” KSII Transactions on Internet and Information Systems, vol. 12, no. 10, Oct. 2018, doi: 10.3837/tiis.2018.10.028.

[13] Chae, H.S., Jo, B.O., Choi, S.H. and Park, T.K., “Feature selection for intrusion detection using nsl-kdd”. Recent advances in computer science, 20132, pp.184-187.2013

[14] Y. Jia, M. Wang, and Y. Wang, “Network intrusion detection algorithm based on deep neural network,” IET Information Security, vol. 13, no. 1, pp. 48–53, Jan. 2019, doi: 10.1049/iet-ifs.2018.5258.

[15] R. Rama Devi and M. Abualkibash, “Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper,” International Journal of Computer Science and Information Technology, vol. 11, no. 03, pp. 65–80, Jun. 2019, doi: 10.5121/ijcsit.2019.11306.

[16] N. Farnaaz and M. A. Jabbar, “Random Forest Modeling for Network Intrusion Detection System,” Procedia Computer Science, vol. 89, pp. 213–217, 2016, doi: 10.1016/j.procs.2016.06.047.

[17] Sultana, Amreen and M. A. Jabbar. “Intelligent network intrusion detection system using data mining techniques.” 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)329-333. (2016).

[18] Tavallaee, Mahbod, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani. “A detailed analysis of the KDD CUP 99 data set.” 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications 1-6 (2009).

[19] S. Gurung, M. Kanti Ghose, and A. Subedi, “Deep Learning Approach on Network Intrusion Detection System using NSL-KDD Dataset,” International Journal of Computer Network and Information Security, vol. 11, no. 3, pp. 8–14, Mar. 2019, doi: 10.5815/ijcnis.2019.03.02.

[20] Y. S. Alsalman, N. Khamees Abu Halemah, E. S. AlNagi and W. Salameh, "Using Decision Tree and Artificial Neural Network to Predict Students Academic Performance," 2019 10th International Conference on Information and Communication Systems (ICICS),pp. 104-109, (2019) doi: 10.1109/IACS.2019.8809106.

[21] M. S. Hammoodi, H. A. A. Essa, and W. A. Hanon, “The Waikato Open Source Frameworks (WEKA and MOA) for Machine Learning Techniques,” Journal of Physics: Conference Series, vol. 1804, no. 1, p. 012133, Feb. 2021, doi: 10.1088/1742-6596/1804/1/012133.

[22] Kunal and M. Dua, "Machine Learning Approach to IDS: A Comprehensive Review," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 117-121, doi: 10.1109/ICECA.2019.8822120.

[23] C. Yin, Y. Zhu, J. Fei, and X. He, “A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/access.2017.2762418.

[24] M. Alowaidi, “Modified Intrusion Detection Tree with Hybrid Deep Learning Framework based Cyber Security Intrusion Detection Model,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 10, 2022, doi: 10.14569/ijacsa.2022.0131038.

[25] [25]C. Nasa and S. Suman, “Evaluation of Different Classification Techniques for WEB Data,” International Journal of Computer Applications, vol. 52, no. 9, pp. 34–40, Aug. 2012, doi: 10.5120/8233-1389.

[26] Guezzaz, Azidine, Mourade Azrour, Said Benkirane, Mouaad Mohy-eddine, Hanaa Attou and Maryam Douiba. “A Lightweight Hybrid Intrusion Detection Framework using Machine Learning for Edge-Based IIoT Security.” Int. Arab J. Inf. Technol.822-830. (2022).

[27] Habibi Lashkari, Arash, Gerard Draper-Gil, Mohammad Saiful Islam Mamun and Ali A. Ghorbani. “Characterization of Tor Traffic using Time based Features.” International Conference on Information Systems Security and Privacy (2017).

[28] Al-Mashagbeh, Malak Hamad and Mohammad Ababneh. “Tor Detection using a Machine Learning Approach Using Correlation based Feature Selection with Best First and Random Forest.” 2021 International Conference on Information Technology (ICIT 893-898.)(2021)