Network Intrusion Detector using Multilayer Perceptron (MLP) Approach

Witcha Chimphlee [a], Siriporn Chimphlee [b]

[a, b]Assistant Professor of Data Science and Analytics, Suan Dusit University, Thailand.

Abstract: Currently, it is very important to maintain high-level security to ensure safe and trusted communication of information between various organizations. There has been much research conducted on intrusion detection in the past, especially anomaly based intrusion detection. In this paper, we use MLP for intrusion classification by using the CIC-IDS2018 dataset. Feature extraction is part of SelectKbest. These are used to test the attacks on binary and multiclass. The results found that the MLP with SelectKbest feature gives the performance with high performance. This method is capable of minimizing the number of features and maximizing the detection rates.

Keywords: Intrusion Detection System (IDS), MLP, CSE-CIC-IDS-2018, Classification algorithm, confusion matrix

1. Introduction

Recent growth in data volumes has been explosive due to the proliferation of applications that generate the data, which must be collected, stored, and processed (Ravikumar, 2021)**.** As daily life and communication are increasingly connected by networks such as the Internet, there is an increasing demand for protection and security. Intrusion detection systems have played an essential role in computer and network security. IDS monitoring and analyzing network traffic is used to classify different types of attacks (Dokas et al., 2002). Intrusion detection is one of the most critical network security problems in the technology world. The network traffic action consists of many features collected in the form of a dataset to detect different types of attacks (Hariyale et al., 2020). The increase in the massive amount of data being generated daily via the internet has caused the world of technology to face a large challenge (Samad et al., 2008).

There are three methods of intrusion detection (Koch, 2011): (Songma et al., 2013): (Yeo et al., 2017): known pattern recognition (signature-based), anomaly based detection, and a hybrid of the previous two. Anomaly based detection is currently mainly implemented as a support for the zero-day network perimeter defense of large infrastructures and network operators, while signature-based intrusion prevention remains the main mode of defense for most businesses and households. Pattern recognition or anomaly detection can be seen as classification problems. Classification problems refer to the problems in which the variable to be predicted is categorical. To enhance the performance of IDS, different classification algorithms are applied to detect various types of attacks **(S**alih & Abdulazeez, 2021)**.**

Detection-based IDS types: in this type of IDS, attacks are identified based on the signature and anomaly approaches used to detect the attacks. In signature-based (12), the IDS system monitored the network and identified the attacks with the predefined signature samples. It is also treated as a misuse detection method. The benefit of this approach is a low false alarm rate, and the efficiency increases with high signature data samples. The drawback is that only known attacks can be identified, which leads to a high missed alarm rate (1,7). Anomaly based IDS types: unusual behaviors by their degree of variation from the standard profile (26). The advantage of this approach is to identify unknown attacks. The pitfall of this method is the lower false alarm rate (1,7).

The classification model splits the dataset into stage training and testing (Sukhachandra, 2018). The massive number of features with high dimensions leads to complexity in the training phase and wastes time. Therefore, it needs to select some useful and relevant features from the whole range of features to improve the performance of the model in the testing phase (Shailesh et al., 2019).

Machine learning (ML) techniques widely used in computer security datasets have recently become a trend in security technology (Borisenko et al., 2021). Machine learning techniques are being implemented to improve the intrusion detection system (IDS). It contributes to analyses and handling the massive amount of data and extracts the essential features that are used in various techniques for feature selection (Khan, 2021). IDS is a commonly used machine learning classifier to distinguish between various attacks as a class.

This paper aims to present the results of evaluating different classification algorithms to build an IDS model in terms of confusion matrix, accuracy, recall, precision, f-score, specificity and sensitivity. Nevertheless, most researchers have focused on the confusion matrix and accuracy metric as measurements of classification

performance. It also provides a detailed comparison with the dataset, data preprocessing, number of features selected, feature selection technique, classification algorithms, and evaluation performance of algorithms described in the intrusion detection system.

The rest of the paper is laid out as follows. The study consists of the following sections: Section 2, significance of the study, Section 3, a review of related studies and objectives of the study in Section 4, hypotheses of the study are present in Section 5. Section 6 explores the dataset. Section 7 discusses the MLP model. Section 8 explores performance metrics. Section 9 discusses the experimental results. Section 10 proposes recommendations, and conclusions are presented in Section 11.

2. Significance Of The Study

A classification model-based IDS classifies all the network traffic into either normal or abnormal classifier algorithms. The obstacle to building the model is the massive amount of data (Al-Yaseen et al., 2017). Classification algorithms, facing many problems in building a model, need a data preprocessing stage, especially in high data dimensionality (Pattawaro & Polprasert, 2018). Choosing the best classification algorithm depends on the performance evaluation metrics in terms of the confusion matrix and accuracy (Abdulaziz et al., 2018). The data classification process in the dataset includes the training and testing stages. During the training and learning stage, a classifier is learned as a target, while during the second stage, the testing phase, the built model is used to predict the class labels for a given data. It is essential to analyze each classifier's required time for both stages of training and testing. Before applying the classifiers, preprocessing of the data helps the classification model decrease time and complexity by removing irrelevant data to improve the classifier algorithm efficiency (Sukhachandra, 2018).

Researchers want a solution to monitor network assets to detect anomalous behavior and misuse in networks. The significance of the study is as follows. I) An analysis of the best feature selection using SelectKbest. ii) Find the popularity of the ML approach used in the intrusion detection system. iii) a comparison of several selectKbest, which are being used high performance iv) an analysis of various performance metrics used to evaluate an intrusion detection system.

3. Review Of Related Studies

The feature selection process requires dimensionality reduction to reduce redundant and irrelevant data. Moreover, the removal of useless features enhances the accuracy of the model. Simultaneously, it speeds up the training and testing time (Shailesh et al., 2019). Addressing big datasets is a difficult and time-consuming task, especially with different categorical data types. Reducing the high dimensionality of data improves the process of feature selection. In general, many datasets are used in the IDS. Each dataset covers various kinds of features to detect and prevent different malicious attacks (Ladha & Deepa, 2011). Hence, increasing the space of data, the computations need more complex calculations. Handling cases of the high number of features by reducing useless features by using dimension reduction techniques (Yu & Liu, 2003). Feature selection and feature extraction are two main techniques to overcome high dimensionality. Feature selection requires finding a subset of relevant features of the original dataset. Feature extraction reduces the data in the original high-dimensional dataset space to a lower dimension space (Talagala et al., 2019).

The quality of the building model in the classification task depends on the features selected in the data. The most crucial point in the process of feature selection is meant to overcome the curse of high dimensionality (Motoda & Liu, 2002). This operation removes unwanted features based on the feature importance top score and uses the feature ranking, leading to increased learning algorithm performance (Farhan et al., 2020). High-dimensional data, in terms of a number of features, are common datasets. To extract useful information from high volumes of data, we have to use SelectKbest to reduce the noise or redundant data. This is because we do not need to use every feature at our disposal to train a model.

Karatas et al. (2020) classified the CSE-CIC-IDS2018 dataset using KNN, RFT, GBC, ADA, DT (decision tree), and LDA (linear discriminant analysis with singular value decomposition solver) algorithms. Parameters that were selected for all the implemented algorithms are described in Karatas et al. (2020) Table 8. The number of classes was determined to be six (one for nonattack type and 5 for attack types), making the results directly incomparable with our multiclass approach. Cross-validation with 80%/20% split of training and test data was used. The results of the accuracy, precision, recall and F1 were obtained. The Precision, Recall and F1.

In their study, Kilincer et al. (2021) classified the CSE-CIC-IDS2018 dataset using KNN, DT, and SVM algorithms. Options of MATLAB for KNN with KNN Fine algorithm, DT with Fine tree and SVM Quadratic algorithm gave the best results in this research. The results on a limited number of records were used in this research for the CSE-CIC-IDS2018 dataset classes. The results of the accuracy, precision, recall, F1 and g-mean were obtained.

Kanimozhi and Jacob (2019a, 2019b) classified the CSE-CIC-IDS2018 dataset using ADA, RF, kNN, SVM, NB and ANN (artificial neural network) machine learning methods. For an ANN, the authors used MLP with two layers, the lbfgs solver, grid searched alpha parameter (for L2 regularization) and hidden layer sizes. In their research, authors used 0–1 classification. Either "Benign" or "Malicious" labels were used for training, making the results directly incomparable with our multiclass approach. The results of the accuracy, precision, recall, F1 and AUC were obtained.

4. Objectives Of The Study

- Using the CSE-CIC-IDS-2018 network intrusion detection dataset with MLP classifiers
- Remove any data that are useless or noisy.
- To implement feature dimensions that impact the classification performance
- To compare feature selection
- To report evaluation metrics for comparison with the IDS model
- Make the feature space less dimensional
- Reduce the detection time of the IDS model
- Improve the predictive accuracy of a classification algorithm
- Enhancement of performance to increase predictive accuracy
- To calculate the best accuracy, precision, and confusion matrix of classification algorithms.

5. Hypotheses Of The Study

IDSs have to be more accurate, with the capability to detect a varied ranging of intrusions for both SIDS and AIDS with fewer false alarms and new signature detection.

- This model in this paper is only tested on the CIC-IDS2018 dataset. It does not contain attacks that can be observed currently.
- The model objectively evaluates and compares approaches to anomaly based NIDS under practical aspects.
- The existing IDSs still face difficulties in improving the higher detection rate and reduced false alarm rate.
- In this study, the dataset, methods of dimensionality reduction, attack identification methods, various conditions, testing methods, and suitable evaluation metrics are compared.

6. Method Process

The model consists of different parts: dataset used for experiments, data cleaning, feature subset selection, MLP classifier, training and testing, and evaluation results. The block diagram of the model is shown in Figure 1.

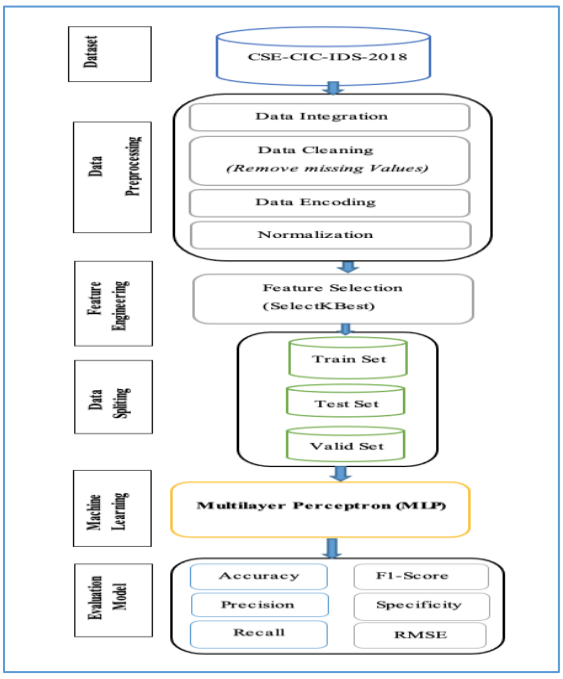

Figure 1. Proposed model

a. Load dataset

Various benchmark datasets have been used to evaluate the intrusion detection model. The work done on the various datasets is to exhibit better classification accuracy and detection rate (J.P. Anderson, computer security, 1998). CIC-IDS2018 contains 15 different classes, 14 of which are attack types and one of which is benign. Out of a total of 16,233,002 (approx. 16 million). The CSE-CIC-IDS2018 dataset (Sharafaldin et al., 2018) is made available by the Canadian Institute for Cyber Security Research at the University of New Brunswick.4 Data were emulated in the CIC test environment within an environment of 50 attacking machines, 420 victim PCs and 30 victim servers during the period from February 14 to March 2, 2018. The dataset contains records from 14 distinct attacks and is labeled and presented together with anonymised PCAP5 files. Eighty network traffic features were extracted and calculated using the CICFlowMeter tool. Ten CSV files are made available for machine learning, containing 16,232,943 records. Table 1 presents a summary of the class representation of this dataset.

Table 1. CSE-CIC-IDS2018 Data Distribution

File/Day	Normal Instances	Attack Instances
Wednesday-14-02-2018	667,626	FTP-BruteForce (193,360), SSH-Bruteforce (187,589)
Thursday-15-02-2018	996,077	DoS attacks-GoldenEye (41,508), DoS attacks-Slowloris (10,990)
Friday-16-02-2018	446,772	DoS attacks-SlowHTTPTest (139,890), DoS attacks-Hulk (461,912)
Thursday-20-02-2018	7,372,557	DDoS attacks-LOIC-HTTP (576,191)
Wednesday-21-02-2018	360,833	DDOS attacks-LOIC-UDP (1730), DDOS attack-HOIC (686,012)
Thursday-22-02-2018	1,048,213	Brute Force -XSS (79), Brute Force-Web (249), SQL Injection (34)
Friday-23-02-2018	1,048,009	Brute Force -XSS (151), Brute Force -Web (362), SQL Injection (53)
Wednesday-28-02-2018	544,200	Infiltration (68,871)
Thursday-01-03-2018	238,037	Infiltration (93,063)
Friday-02-03-2018	762,384	Bot (286,191)

Figure 2 shows the class distribution in the dataset. For data capturing and feature selection, CSE-CIC-IDS-2018 is used, which contains 16,232,943 instances. The Normal are 13,484,708 and 2,748,235 are attack instances.

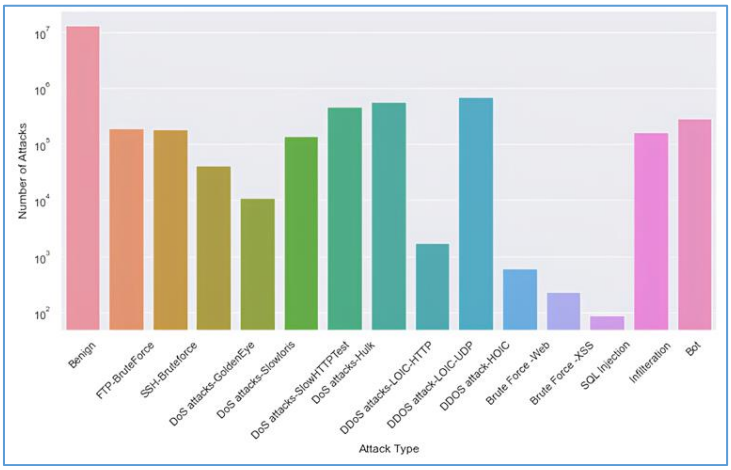

Figure. 2. Bar charts showing the class distribution in the dataset

b. Data preprocessing

Preprocessing and normalization of the data are performed before training. This phase is used for the removal of noise or redundant information from data and preserves only meaningful and important information. In the proposed model, preprocessing involves the following main tasks:

- Remove features containing NaN values.
- The timestamp column and related record duplicates were removed, as no time series-dependent machine learning methods were chosen in this research. Afterwards, 8 features, 'Bwd URG Flags', 'Bwd Pkts/b Avg', 'Bwd PSH Flags', 'Bwd Blk Rate Avg', 'Fwd Byts/b Avg', 'Fwd Pkts/b Avg', 'Fwd Blk Rate Avg', and 'Bwd Byts/b Avg', containing no information were removed.

- Normalization of the data was performed to make the data values comparable. Thus, minmax normalization was used to ensure that all attribute values lie in between [0,1].

After data cleaning, the dataset was normalized with standardization, and we obtained 16137183 instanc es with 70 attributes.

c. Encoding categorical attributes

The transformation of an anomaly into its type at the end of each instance is performed by assigning a numeric value to each attack type, i.e., 0 to Normal data and 1 to Attack. Class encoding for binary and multiclassification is shown in Table 2 and Table 3, respectively.

Table 2. Class Encoding for Binary Classification

Code	Class
0	Benign
1	DDOS attack-HOIC
1	DDoS attacks-LOIC-HTTP
1	DoS attacks-Hulk
1	Bot
1	FTP-BruteForce
1	SSH-Bruteforce
1	Infilteration
1	DoS attacks-SlowHTTPTest
1	DoS attacks-GoldenEye
1	DoS attacks-Slowloris
1	DDOS attack-LOIC-UDP
1	Brute Force -Web
1	Brute Force -XSS
1	SQL Injection

Table 3. Class Encoding for Multi-Classification

Code	Class
0	Benign
1	DDOS attack-HOIC
2	DDoS attacks-LOIC-HTTP
3	DoS attacks-Hulk
4	Bot
5	FTP-BruteForce
6	SSH-Bruteforce
7	Infilteration
8	DoS attacks-SlowHTTPTest
9	DoS attacks-GoldenEye
10	DoS attacks-Slowloris
11	DDOS attack-LOIC-UDP
12	Brute Force -Web
13	Brute Force -XSS
14	SQL Injection

d. Feature extraction

To reduce the total cost of computing and increase the performance of the model, the number of variables should be minimized. Thus, the performance can maintain accuracy even if a significant amount of data is absent (Yogesh, Suresh, 2022). Based on the ideas of research and practical implementation recommendations made by Sharafaldin et al. (2018). In this research, features were selected with SelectKBest from the Scikit-learn library (Pedregosa et al., 2011). The SelectKBest method takes as a parameter a score function, such as $\chi 2$ or ANOVA Fvalue, or information gain function and retains the first k features with the highest scores. When performing feature selection, SelectKBest focuses on the largest classes; therefore, a possible improvement would be to perform feature selection in a pipeline by first selecting the most important features for the rarest class and then adding features needed for every class. Generating additional synthetic features was not attempted in this

research, as all chosen datasets contain a significant number of such features (Bulavas, 2021). Feature selection will lead the classifier to work in a competent way and enhance the overall performance of intrusion detection. The redundant and irrelevant features increase the overheads and confuse the classifier.

e. Splitting the Dataset

In this paper, we partition the sample into a training set and a test set. because it increases the efficiency of the classification model. The model's performance will deteriorate if we train it adequately and its training accuracy is good, but then give it a new dataset (Yogesh, Suresh, 2022). The training set is a part of the sample used to build the classification model, and the result is now known for these samples. The test set is a part of the sample that is used to test the model, and it uses the test set to forecast output. For the experiment, the dataset is divided into 2 parts, i.e., 80% for training purposes and 20% for testing purposes. The training dataset consists of 16,232,943 labeled connections. We divide the training dataset into.

7. Multiple Layer Perceptron (MLP) Model

Multiple different types of methods were used in this research to improve the performance of ML methods. The methods employed could be grouped into preprocessing and machine learning methods. Data record sampling methods. Record oversampling, feature selection, scaling and frequency transformation and preprocessing activities. Machine learning methods capable of cost-sensitive learning were chosen for performance comparison in this paper. The multiple layer perceptron (MLP) was proposed by Rosenblatt (1962) as an extension to a linear perceptron model (Rosenblatt, 1957). It is a supervised learning artificial neural network implementation, utilizing backpropagation for training, that can have multiple layers and a chosen, non necessarily linear, activation function. A multilayer perceptron (MLP) is a feedforward neural network that maps sets of input data onto a set of appropriate outputs. Here, we used an MLP architecture consisting of three layers: input, hidden and output. In this architecture, the hidden layer and output layer consist of neurons (processing elements), and each neuron has a nonlinear activation function. The layers are fully connected from one layer to the next. MLP is an amendment of the standard linear perceptron, which can discriminate data that are not linearly separable. The architecture we used here is shown in Figure 3.

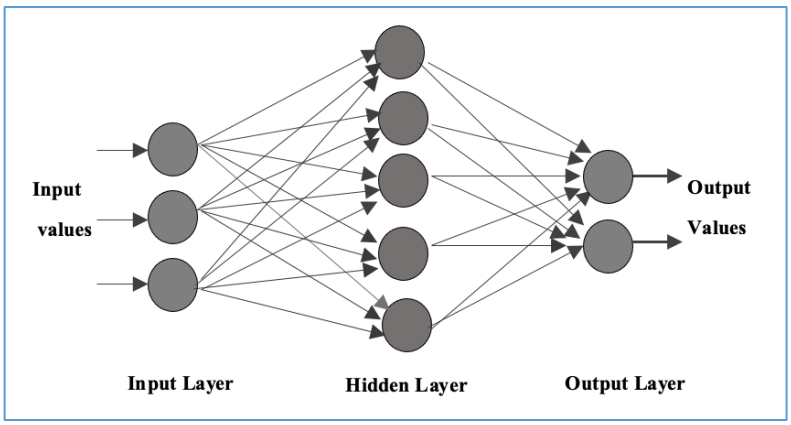

Figure 3. MLP Architecture

This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications, the units of these networks apply a sigmoid function as an activation function[12]. Multilayer networks use a variety of learning techniques, the most popular being backpropagation. Here, the output values are compared with the correct answer to compute the value of some predefined error function. By various techniques, the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection to reduce the value of the error function by a small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the error of the calculations is small. In this case, the network has learned a certain target function. To adjust weights properly, one applies a general method for nonlinear optimization that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated, and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). For this reason, backpropagation can only be applied on networks with differentiable activation functions.

8. Performance Metrics

Evaluation metrics describe the performance of the classification model shown in Table 4. The critical point behind the classification is an evaluation metric used to understand the performance and efficiency of an algorithm.ello Nazifi & Maheyzah, 2020). The best fit classification techniques for IDS application from tested methods. The need to check the performance measure among all methods can be achieved with performance metrics. After training, the model is provided with test data for evaluation to find performance measures such as accuracy, false positive rate, and true positive rate. IDS are typically evaluated based on the following standard performance measures:

Table 4. Metrics for classification algorithms

Formula	Description Evaluation Metrics
Accuracy = (TP+TN)/(TP+TN+FP+FN)	Total correct classified over the total number of records
Precision = TP/(TP+FP)	True positive that are correctly predicted from the total predicted patterns in a positive class.
Recall = TP/(TP+FN)	Positive patterns that are correctly classified.
F1-Score = 2*(Recall*Precision)/(Recall+Precision)	This metric represents relation between recall and precision values
Specificity = TN/(TN+FP)	Negative patterns that are correctly classified

In this model, the most common intrusion detection evaluation metric (confusion matrix) is used to evaluate the MLP model performance. This experiment is executed on 64-bit macOS Big Sur with the following specifications: 2.6 GHz Dual-Core Intel Core i5, and 8 GB 1600 MHz DDR3. The dataset was split into 80% training (12909746 samplings) and 20% testing (3227437 samplings) datasets. We used the following metrics to evaluate our models:

8.1. Confusion Matrix:

A confusion matrix is a statistical measurement used in machine learning classification algorithm performance for finding the accuracy of the model. The confusion matrix includes four measures: true positive (TP), false positive, true negative (TN), and false negative (FN). A good model result would be the one that contains zero false positives and negatives. The impact of splitting the dataset ratio into training and testing phases affects the result of a confusion matrix (Sukhachandra, 2018). This is a useful table that presents both the class distribution in the data and the classifier-predicted class distribution with a breakdown of error types.

8.2. Precision:

Precision is the number of true positives divided by the sum of true positives and false positives. It is the number of positive predictions divided by the total number of positive class values predicted. A low precision can also indicate a large number of false positives.

8.3. Recall:

Recall is the number of true positives divided by the sum of true positives and false negatives. It is the number of positive predictions divided by the number of positive class values in the test data. A low recall indicates many false negatives.

8.4. F1 Score:

The F1 Score is the harmonic mean of precision and recall. It conveys the balance between the precision and recall. Some features can be extracted and supplemented, which might be used in future research; however, extraction requires a high degree of previous network traffic logging, whereas authors are aware that organizations lack resources to collect data on such a level of detail.

9. Experimental Results

The implementation of MLP for IDS to classify different types of attacks. Machine learning techniques have been applied to the field of network security to improve intrusion detection systems. Previous sections reviewed some studies about classification algorithms applied to build the IDS model and evaluated the performance by different metrics in terms of accuracy, recall, precision, f-score, specificity, sensitivity, and dependable tool confusion matrix. The dimension reduction and feature selection had a good effect on the classification model performance because it reduced training and testing time by removing the irrelevant features, making the classification process more accurate and less complicated.

This study aims to show MLP classification algorithms' performance by using different measurements to select a suitable classifier best model to gain speed and accuracy. We performed four different experiments. Our aim is to select features that produce optimal results in terms of accuracy. The features are reduced to 40, 30, 20, and 10. The results of the first classification (Benign and attack) are shown in Table 5. The 40 best features from SelectKBest were passed through the variance inflation factor procedure with a threshold of 40, which was selected to eliminate collinearity of features. The above experiments show that optimal features increased accuracy, precision, recall, f1-score, and specificity with the highest accuracy, highest recall, highest f1-score, and highest value.

Table 5. Binary-Classification report

K	Accuracy	Precision	Recall	F1-Score	Specificity
10	0.98785507	0.987302476	0.998205553	0.992724078	0.937315218
20	0.98862224	0.987293249	0.999228191	0.993224868	0.935095093
30	0.98827057	**0.988430634**	0.997544440	0.992966625	**0.942987653**
40	**0.98877190**	0.987391832	**0.999231751**	**0.993276510**	0.937697998

More details for binary classification with SelectKBest (K=40) are shown in Table 6. Macro avg and Weighted avg are received with support value.

Table 6. Binary Classification with SelectKBest (K=40)

Code	Precision	Recall	F1-score	Support
0	0.99	1.00	0.99	2678819
1	1.00	0.94	0.97	548618
Accuracy			0.99	3227437
Macro avg	0.99	0.97	0.98	3227437
Weighted avg	0.99	0.99	0.99	3227437

However, the results in this table are derived from the data in Table 7. It shows a confusion matrix that contains benign and attack data. This table shows that 2676761 normal packets are detected as normal from 2678819 normal packets, and the error is obtained on 2058 packets. They are detected as an attack. In the same case in attacks, 548618 packets in the testing data were an attack, 514438 was detected as an attack, and the error was obtained on 34180 packets.

Table 7. Confusion matrix for binary classification

		Normal	Attacks	Total
Actual	**Normal**	2676761	2058	2678819
	Attacks	34180	514438	548618
Total		2710941	56496	3227437

Figure 4 shows the binary classification of data using a confusion matrix, which is a two-by-two matrix consisting of outcomes produced by the binary classifier as overall accuracy, sensitivity, precision, and specificity. The binary classifier produces results with labels of 0/1 and Yes/No. The instances of all the test data are predicted using the classifier as true positive, true negative, false positive and false negative. The matrix derives the error rate and accuracy as the primary measure. Here, the confusion matrix computes the accuracy as 0.99, and the matrix is built between the true label and predicted label with labels of 0/1 and a data scale.

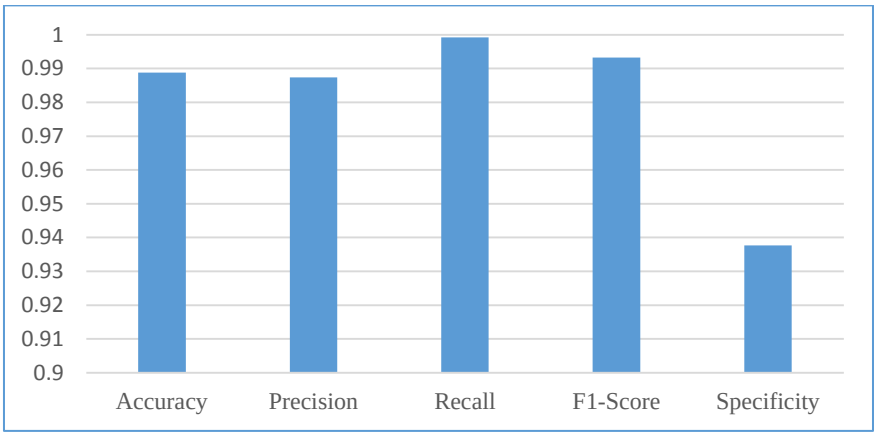

Figure 4. Evaluation results for binary classification

In the case of multiclass classification, a comparison of performance measures is given in Table 8. The classifications of each attack type are compared in Table yy. This table shows an improvement in multiclass detection for auto ML in terms of accuracy in comparison with various conditions. The table mentions different SelectKbest and MLP classifiers. The experiments show that optimal features with SelectKBest (K=20) increased accuracy, precision, recall, f1-score, and specificity with the highest accuracy, highest precision, highest recall, highest f1-score, and highest specificity for multiclass classification.

Table 8. Multiclass Classification

K	Accuracy	Precision	Recall	F1-Score	Specificity
10	0.972968334	0.984527515	0.98325697	0.983891835	0.91910975
20	**0.982071532**	**0.987242884**	**0.99139055**	**0.989312368**	**0.93422213**
30	0.981349786	0.987194434	0.99057380	0.988881229	0.93389838
40	0.981696622	0.987117374	0.99106639	0.989087943	0.93358223

The attack is divided into 14 categories: 15 categories with a benign (normal) class. In the testing part, the accuracy was 0.98, and the results of the confusion matrix for the testing data are shown in table 10. The table describes the confusion matrix for multi class classification that is dependent on testing data samples.

Table 9. Multiclass Classification with SelectKBest (K=20)

Code	Precision	Recall	F1-score	Support
0	0.99	1.00	0.99	2678819
1	1.00	1.00	1.00	137023
2	0.99	0.97	0.98	115286
3	0.96	1.00	0.98	92053
4	1.00	1.00	1.00	57125
5	0.71	0.88	0.79	38496
6	0.99	1.00	1.00	37612
7	0.44	0.01	0.02	31852
8	0.75	0.51	0.61	28032

9	0.99	0.55	0.71	8387
10	0.95	0.94	0.94	2228
11	0.72	0.99	0.84	339
12	0.00	0.00	0.00	131
13	0.00	0.00	0.00	38
14	0.00	0.00	0.00	16
Accuracy			0.98	3227437
Macro avg	0.70	0.66	0.66	3227437
Weighted avg	0.98	0.98	0.98	3227437

The table below suggests that there are 2678819 normal packets detected true as normal from 2677240 normal packets, so according to this result, the accuracy of detection is very high, nearly 1. In addition, so in the rest of the results of the attacks.

Table 10. Confusion matrix for multiclass classification

2677240	68	651	90	218	1	59	358	7	15	112	0	0	0	0
1	137022	0	0	0	0	0	0	0	0	0	0	0	0	0
3142	0	112017	0	0	0	0	0	0	0	0	127	0	0	0
7	0	0	92003	0	0	43	0	0	0	0	0	0	0	0
70	0	0	0	57055	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	33858	0	0	4638	0	0	0	0	0	0
0	0	0	3	0	8	37599	0	2	0	0	0	0	0	0
31556	0	13	2	2	1	0	278	0	0	0	0	0	0	0
0	0	0	0	0	13804	0	0	14228	0	0	0	0	0	0
0	0	0	3654	0	0	94	0	0	4629	10	0	0	0	0
3	0	0	74	0	0	37	0	0	13	2101	0	0	0	0
3	0	2	0	0	0	0	0	0	0	0	334	0	0	0
113	0	18	0	0	0	0	0	0	0	0	0	0	0	0
29	0	9	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11 shows more details on each class with accuracy, precision, recall, F1-score, and specificity. This metric is very important to guarantee for an intrusion detection system that tries to identify as many intrusions as possible, considering all of them as critical. From the table it can be seen that results are very similar, with values close to 1 except for Infilteration, DoS attacks-SlowHTTPTest and DDOS attack-LOIC-UDP, with values of Specificity being 0.437106918, 0.753881206 and 0.724511931 respectively.

Table 11. Multi class classification report by class

Class	Accuracy	Precision	Recall	F-Score	Specificity
Benign	0.988304960	0.999202260	0.986871610	0.992998658	0.995851670
DDOS attack-HOIC	0.999975483	0.999974601	0.999999626	0.999987114	0.999503975
DDoS attacks-LOIC-HTTP	0.998641986	0.999756898	0.998827779	0.999292123	0.994221962
DoS attacks-Hulk	0.999964970	0.999966384	0.999997385	0.999981885	0.999022727
Bot	0.999894682	0.999918579	0.999973854	0.999946216	0.996193669
FTP-BruteForce	0.999999631	0.999999626	1.000000000	0.999999813	0.999970466
SSH-Bruteforce	0.999978268	0.999977963	1.000000000	0.999988981	0.998433268
Infilteration	0.988221147	0.999866298	0.988350544	0.994075071	0.437106918
DoS attacks-SlowHTTPTest	0.993192042	0.998268009	0.994870392	0.996566305	0.753881206
DoS attacks-GoldenEye	0.998598840	0.999994397	0.998602008	0.999297718	0.996770026

DoS attacks-Slowloris	0.999907075	0.999954433	0.999952565	0.999953499	0.945119208
DDOS attack-LOIC-UDP	0.999950700	0.999952565	0.999998132	0.999975348	0.724511931

From Table 10, bring it to form visualization as shown in Figure 5. it displays the precision, recall, f1-score, and specificity per class dependent on the testing data.

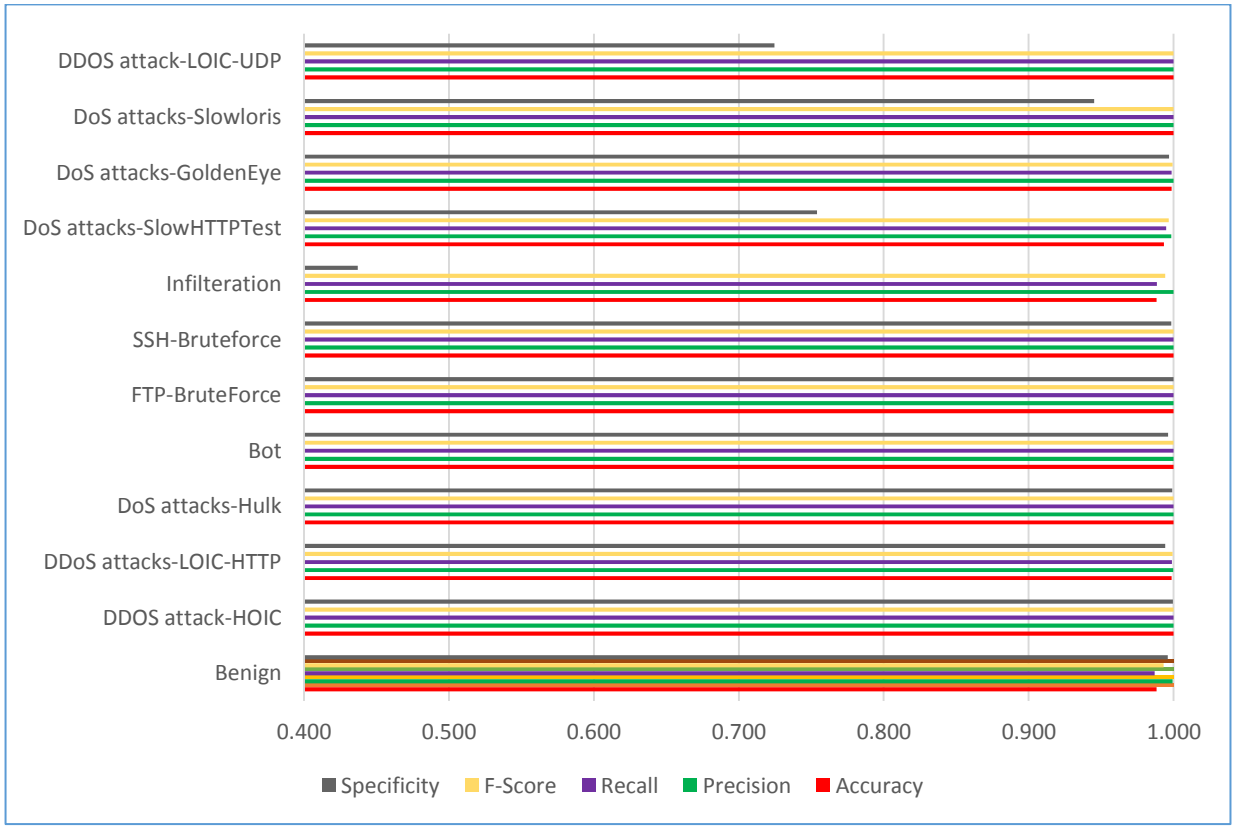

Figure 5. Evaluation results for multiclass classification

10. Recommendations

Intrusion Detection Systems (IDS) are automated systems that monitor and analyze network traffic and generate alerts in response to activity either match known patterns of malicious activities or is unusual. In some cases, an unknown or a novel pattern should be to determine. This research has important research significance.

11. Conclusion

In this paper, we propose a framework to study and analyze machine learning by using MLP classification. We reviewed existing research based on the proposed framework, including the research objective construction, preprocessing, feature extraction and selection, and performance evaluation. IDS improvement performance depends on different machine learning techniques. Classification algorithms have a significant role in helping IDS to distinguish different types of attacks. This paper aims to test classifier algorithms and find the evaluation performance by using different metrics. The metric is very important to guarantee for an intrusion detection system that tries to identify as many intrusions as possible, considering all of them as critical. The study, applying various metric measurements to evaluate classifiers' performance, noticed that the MLP achieved sufficient results and the highest accuracy to classify different types of attacks. Obtaining high performance of the model, most researchers used the MLP for building intrusion detection systems rather than individual classification. The effectiveness of dimension reduction in reducing the complexity of big datasetssets leads to the selection of optimal features to obtain better classification performance in terms of accuracy and speed. MLP and SelectKBest used in this study obtained good intrusion detection results.

As future work, we plan to investigate the application to detect novel pattern.

References

Abdulaziz, A., Mohamed, A. Z., Debatosh, D., & George, C. (2018). Classification Approach for Intrusion Detection in Vehicle Systems. *Wireless Engineering and Technology*, *9*(4), 79–94.

Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017). Multilevel hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications*, *67*, 296–303. https://doi.org/10.1016/j.eswa.2016.09.041

Bello Nazifi, K., & Maheyzah, M. S. (2020). A Review on Network Intrusion Detection System Using Machine Learning. *International Journal of Innovative Computing*, 27–34. https://doi.org/http://doi.org/10.11113/ijic.v10n1.252

Borisenko, B. B., Erokhin, S. D., Fadeev, A. S., & Martishin, I. D. (2021). *Intrusion Detection Using Multilayer Perceptron and Neural Networks with Long Short-Term Memory; Intrusion Detection Using Multilayer Perceptron and Neural Networks with Long Short-Term Memory*. https://doi.org/10.1109/SYNCHROINFO51390.2021.9488416

Bulavas, V. (2021). *Improving machine learning model performance on detection of network infiltration; Improving machine learning model performance on detection of network infiltration*. https://doi.org/10.1109/HSI52170.2021.9538669

Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., & Tan, P.-N. (2002). Data mining for network intrusion detection. *National Science Foundation Workshop on Next Generation Data Mining*, *38*(7), 21–30.

Farhan, R. I., Maolood, A. T., & Hassan, N. F. (2020). Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning. *Indonesian Journal of Electrical Engineering and Computer Science*, *20*(3), 1413–1418. https://doi.org/10.11591/ijeecs.v20.i3.pp1413-1418

Hariyale, N., Manjari, S. R., Ritu, P., & Praneet, S. (2020). A hybrid approach for intrusion detection system. In *Soft Computing for problem solving*. Springer US.

Khan, M. A. (2021). *HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System*. https://doi.org/10.3390/pr9050834

Ladha, L., & Deepa, T. (2011). Feature Selection Methods And Algorithms. *International Journal on Computer Science and Engineering*, *3*(5), 1787–1797. http://journals.indexcopernicus.com/abstract.php?icid=945099

Motoda, H., & Liu, H. (2002). Feature selection, extraction and construction. *Communication of IICM*, *5*, 67–72.

Pattawaro, A., & Polprasert, C. (2018). Anomaly Based Network Intrusion Detection System through Feature Selection and Hybrid Machine Learning Technique. *2018 16th International Conference on ICT and Knowledge Engineering (ICT KE)*, 1–6. https://doi.org/10.1109/ICTKE.2018.8612331

Ravikumar, D. (2021). *Toward Enhancement of Machine Learning Techniques Using CSE-CIC-IDS2018 Cybersecurity Dataset*. https://scholarworks.rit.edu/theses

Salih, A. A., & Abdulazeez, A. M. (2021). Evaluation of Classification Algorithms for Intrusion Detection System: A Review. *Journal of Soft Computing and Data Mining*, *02*(01). https://doi.org/10.30880/jscdm.2021.02.01.004

Samad, A., Ismail, H., Abdullah, A. H., Bin, K., Bak, A., bin Ngadi, A., Dahlan, D., & Chimphlee, W. (2008). A Novel Method for Unsupervised Anomaly Detection Using Unlabeled Data; A Novel Method for Unsupervised Anomaly Detection Using Unlabeled Data. *2008 International Conference on Computational Sciences and Its Applications*. https://doi.org/10.1109/ICCSA.2008.70

Shailesh, S. P., Y.P., R., & Lokesh, S. P. (2019). Evaluation of network intrusion detection with features selection and machine learning algorithms on CICIDS-2017 dataset. *International Conference on Advances in Engineering Science Management & Technology (ICAESMT)- 2019*.

Songma, S., Chimphlee, W., Kiattisak, M., & Sanguansat, P. (2013). *Intrusion Detection throught Rule Induction Analysis*.

Sukhachandra, B. (2018). Intrusion detection using machine learning: A comparison study. *International Journal of Pure and Applied Mathematics*, *19*, 101–114.

Talagala, P. D., Statistics, B., Frontiers, S., Hyndman, R. J., Statistics, B., Frontiers, S., & Smith-miles, K. (2019). *Anomaly Detection in High Dimensional Data Anomaly Detection in High Dimensional Data*. *August*, 1–30.

Yeo, L. H., Che, X., & Lakkaraju, S. (2017). *Understanding Modern Intrusion Detection Systems: A Survey*. http://arxiv.org/abs/1708.07174

Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. *Proceedings, Twentieth International Conference on Machine Learning*, *2*, 856–863.