Anomaly Detection for Network Security

Muhammad Fariz Azmi[1], Hezerul Abdul Karim[2*], Nouar AlDahoul[3]

[1,2]*Faculty of Engineering, Multimedia University, Cyberjaya, Selangor, MALAYSIA; E-mail: hezerul@mmu.edu.my*

[3]*Computer Science, New York University, Abu Dhabi, United Arab Emirates*

Abstracts: Today, network security is crucial due to the rapid development of network and internet technologies, as well as the continuous growth in network threats. Detecting network anomalies is one of the approaches that may be used to safeguard a network's security. Recent research has focused extensively on techniques for identifying abnormalities. Using the Autoencoder model together with data pre-processing techniques such as data resampling and feature selection, this research describes a novel approach for identifying network abnormalities. It has been shown that the suggested strategy is applicable to network intrusion data. A comparison of the reconstruction error to a threshold value determines whether the traffic data is normal or anomalous. CICIDS2017 dataset is selected to evaluate the implementation of the proposed Autoencoder model based on real-world, large-scale, current network traffic data The proposed model with data pre-processed achieved F1-Score of 76% which outperformed the baseline model without feature selection and data resampling in data pre-processing stages. This research project investigated the effect of data pre-processing techniques on the performance of the proposed Autoencoder. At the end of this research project, it is demonstrated that the proposed methodologies are applicable towards imbalanced network intrusion data.

Keywords: Anomaly Detection, Network Attack, Autoencoder, Feature Selection, Data Resampling, Under sampling, CICIDS 2017.

1. INTRODUCTION

Greater global connectivity is being facilitated by the Internet and other forms of networking technology. On the infrastructures of global networking, a vast amount of data from private companies, the military, commercial businesses, and government agencies are available. Considering the speed at which network and communication technologies are advancing, there has been an increased focus on the safety of networked computer systems. The ease with which intellectual property can be obtained using the internet has contributed to the rise in importance of network security. Networked computer systems are becoming a more desirable target for attackers, whether they are hackers, competitors, or criminals. Many of these assaults have a negative impact on the victim's financial situation.

If the issue of security is not effectively addressed, it might become a barrier. Because computer systems play such an important role and are becoming more essential in people's day-to-day lives, there is a significant increase in the amount of concern over their level of security. Over the last two decades, extensive research has been conducted on detecting intrusion in computer systems. There are several different preventative techniques now available to safeguard computer networks from assaults. Intrusion Detection Systems, often known as IDS, are one of the solutions that security professionals have implemented over the years in response to the rising number of assaults on networks. IDS are employed to recognise, classify, and, if necessary, react to actions that are invasive in computer networks.

The Internet and computer networking are becoming increasingly important, which has resulted in the IDS being an essential piece of infrastructure. Network Intrusion Detection System (NIDS) ought to perform effectively despite the growing number of threats and the increasing number of intrusion vectors. To get the best possible outcome, the network's IDS must operate in real time. IDS is going to have to overcome a lot of challenges. In a situation where connection capacity has increased to gigabits, the NIDS needs to operate at a faster rate. In addition to that, protocols have gotten more complicated, which requires an increase in processing power as well as an improvement in the algorithm used to effectively handle incoming packets. Given these challenges, the work that has been done to improve NIDS is the utmost importance.

2. RELATED WORK

The AdaBoost algorithms, which incorporate various classifiers such as multilayer perceptron, Support Vector Machine, K-Nearest Neighbour, decision tree, and naïve Bayes, have been used to show an ensemble technique for network anomaly detection. This approach uses multiple classifiers to identify anomalies in a network. Throughout the process of initialising data distribution, training machine learning models, assessing mistakes, and assigning weights to each machine learning model, the AdaBoost algorithms were utilised. After that, a technique called weighted voting was utilised to combine the predictions of the classifiers regarding the outliers.

To identify attacks, separate techniques based on payload categorization using recurrent neural networks (RNN) and convolutional neural networks (CNN) were used. Both CNN and RNN are responsible for recording both temporal and local characteristics. To obtain an accurate representation of the data for the purposes of data classification using the LSTM Model, a CNN was utilised to accomplish this goal.

Recently, an approach for deep learning known as Delayed Long Short-Term Memory, or DLSTM, has been applied to the problem of network anomaly detection involving time-series data. The usual training data was used to develop a predictive model, and the prediction error was then used to the observed data to detect outliers. The study recommended developing several LSTM predictive models with distinct prediction values. Following this, the model with the most precise prediction of the actual value was chosen. Their model may delay prediction until the matching measured value is acquired.

Deep neural networks may be trained to understand complicated patterns from uncommon abnormalities found in network traffic data by employing class weight optimization in the training process. This novel model fusion utilised a combination of two DNNs which were a binary normal and attack DNN for identifying the presence of any assault, and a multi-attack DNN for classifying the attacks.

Intrusion detection system contains numerous well-known datasets, such as DARPA1998 and KDDCUP99, although these datasets were not gathered from an IoT context. Several research, such as NSL-KDD and DS2OS, have begun to focus on IDS in IoT systems in recent years. However, in recent years, the number of IoT devices and unique attack approaches has increased. As a result, datasets must be updated to reflect the IoT environment and new assaults. Furthermore, the current IoT-based IDS datasets are deficient in a vast number of attributes. As a result, new datasets, such as IoTID20, have been introduced. These datasets are more focused on everyday household devices, whilst other datasets are more focused on academic network traffic. As a result, the study used these datasets to evaluate IoT IDS in IoT contexts.

2.1. Summary of Related Work

Paper	Method	Dataset	Result (%)	Remarks
Sornsuwit P.et al., 2015	Ensemble approach using AdaBoost	KDDCUP'99	Precision=76	Dataset does not reflect present day attacks
Liu H. et al., 2019	Delayed Long Short- Term Memory (dLSTM)	DARPA1998	F1-Score=90	Dataset does not reflect present day attacks
Maya S. et al., 2019	CNN and RNN	DARPA1998	Accuracy=99.36 Accuracy=99.98	Dataset does not reflect present day attacks. Method is relatively difficulty and requires some skill
Nouar A. et al., 2021	Model fusion of binary DNN and multiclass DNN	ZYELL	F1-Score=55.30	Low model performance
Ullah I. et al., 2020	SVM, Logistic Regression, Gaussian NB, LDA	IoTID20	F1-Score=70	Average model performance

3. MATERIALS AND METHODS

The network anomaly detection is built consist of several stages including pre-processing stage using a variety of methods, including resampling data and the selection of features. Comparison between several cases were made accordingly to research, analyse, and evaluate the performance of the classifier after implementing the suggested pre-processing methods. Fig. 1 shows the working scenario throughout this entire project.

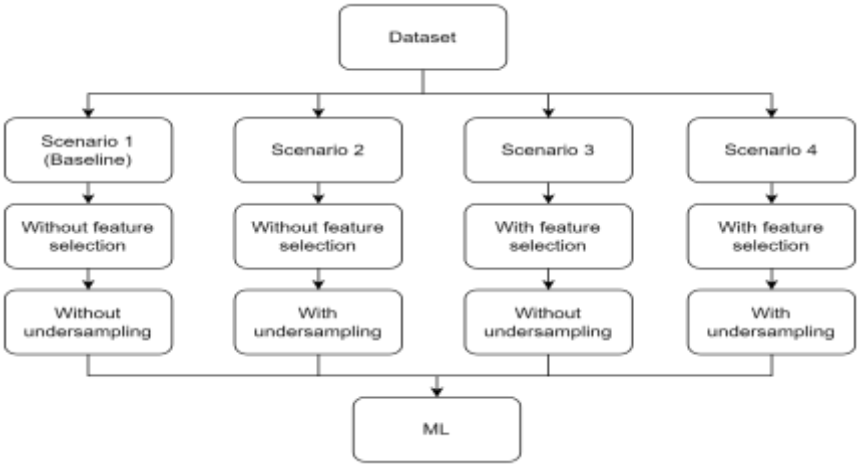


Fig. 1: Working Scenario

In the first scenario, the machine learning algorithm is tested and assessed using the raw data, without pre-processing techniques such as feature selection and data resampling. This scenario will act as the baseline model in this project. In the second scenario, the performance and assessment of the classifier are assessed using under sampled data without feature selection. For the third scenario, the classifier with feature selection technique is experimented and tested. Lastly, in the fourth scenario, the classifier together with data pre-processing techniques including feature selection and data under sampling is evaluated for the performance of the classifier. The recommended pre-processing techniques are used to evaluate classifiers, and the outputs of each case are presented and validated using statistical significance tests. Fig. 2 depicts the methodological procedure, which consists of data cleaning, data resampling, feature selection, data partitioning, data normalisation, ML modelling, classification, and evaluation of the ML model. These procedures will be detailed in the next subchapters.

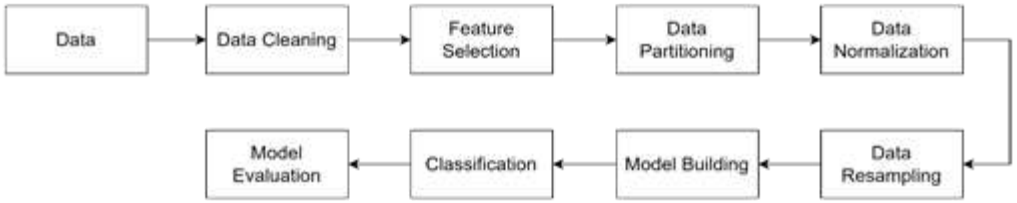


Fig .2: Methodological process of anomaly detection

3.1. Dataset Overview

A benign-profile technique was used to assemble CICIDS2017, a more recent product that was created to record users' abstract behaviour. The Canadian Institute for Cybersecurity located at the University of New Brunswick is responsible for the production of CICIDS2017. This data set was gathered over the period of five days (Monday through Friday), and it contains both data on assaults and data on routine occurrences. This dataset includes information about the network both with and without assaults, which were generated based on the user characteristics of a range of protocols. The information was collected both before and after the attacks. Because of this, the information is quite comparable to real-world network data.

The CICIDS dataset captures a total of 2,830,743 data samples, each of which comprises 79 attributes and the network instances that are linked with one of the 14 types of cyberattacks that are prominent in today's world. CICIDS2017 does not include training and testing data files that are kept separate, and the classes distribution is shown in Fig. 3. The CICIDS 2017 dataset is an imbalanced dataset since it contains a greater number of benign class samples than attack class samples, which constitute around 80% of the dataset. As can be seen in Fig. 3, the number of samples classified as attack is much lower when compared to the number of samples classified as benign, which constitute around only 20% of the dataset.

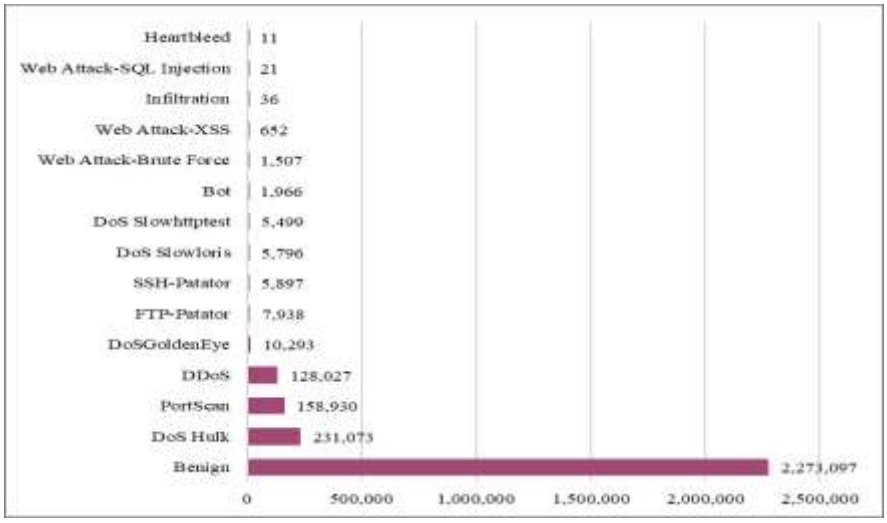


Fig. 3: Sample distribution of CICIDS 2017 according to data class

3.2. Data Cleaning

It is quite typical for a dataset to include some missing values. This research determined to see whether the dataset had any values that were missing or any values that were duplicated. Any data that is either missing or duplicated will be dropped. Fig. 4 depicts the amount of dataset samples before the data

cleaning and after the data cleaning stage.

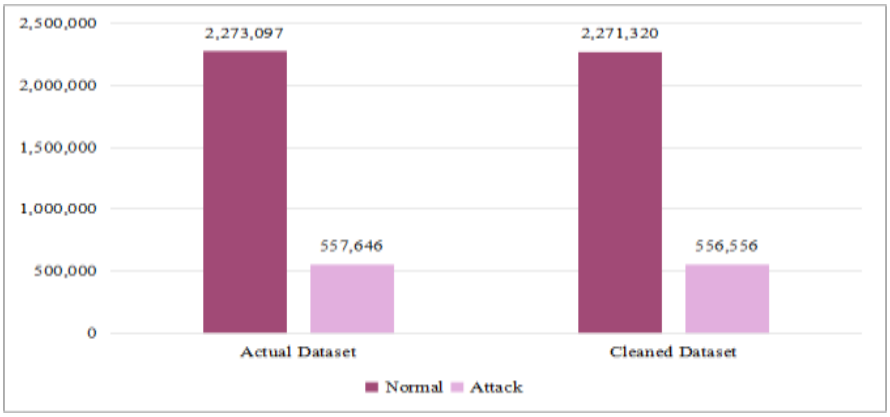


Fig. 4: Sample distribution of CICIDS 2017 before and after data cleaning

3.3. Feature Selection

In a wide variety of classification cases, the technique of feature selection is an essential component. The performance of a machine learning model may be improved by limiting the number of features it uses to just those that are absolutely required, as determined by their relevance scores with the consequence variable. The data is made simpler, data redundancy is removed, computational complexity is decreased, detection rates are increased, and the number of false alarms generated by machine learning models is decreased. Because the features that are chosen have such a big influence on how well the system operates, making those choices is an extremely important part of the process of developing any kind of intrusion detection system. Each feature offers unique attributes for dealing with various threat detection areas. To determine the effects of the technique on the performance of the classifier, many feature selection models are examined.

3.3.1. Correlation

Some of the features are deemed redundant features due to the small range of values and probable linear correlation between all variables. Consequently, statistically independent features were screened using the Pearson's correlation coefficient.

The correlation between the characteristics of the traffic samples was analysed, along with the degree of correlation that existed between them. A heatmap that uses color-coding provides a visual representation of the matrix of correlation that exists between each pair of parameters. The strength of the link that may be found between the variables is measured by the correlation coefficients, which have values that range from 1.0 to 1.0. A correlation of 1.0 indicates a perfect negative correlation, while a correlation of 1.0 shows a perfect positive correlation. A correlation of -1.0 implies a perfect positive correlation (the pair of features are highly correlated). On the other hand, correlation ratings that are either 0 or very near to zero imply that this combination of attributes is very weakly connected. The selection of features, which is the most crucial phase that comes before classification, is directly related to the significance of the correlation matrix. When there is a substantial connection between two traits, it is possible to delete one of them.

In this research, we defined some threshold level of the correlation value for the features to be dropped. The correlated features that show correlation value of 1.0 and the features that exceed threshold level above 0.9 will be dropped as to observe the performance of the classifier depending on the number of

selected features.

3.3.2. SelectKBest

Since there are several methods to implement feature selection, SelectKBest is chosen to be tested in this research. The scikit-learn machine library serves a universal purpose SelectKBest, which may choose the k best feature based on any measure; the ANOVA f-test is utilised as SelectKBest's score function in this research. Analysis of Variance, or ANOVA, is a sort of F-statistic, where an F-statistic, or F-test, is a set of statistical tests that calculate the ratio of variance values.

Calculating the specified metric between the target and each feature, sorting them, and then selecting the K best features is the main concept here. This method's outcomes may be utilised for feature selection, in which independent features can be eliminated from the dataset. In the current investigation, the SelectKBest class is used to choose the top 30 and 40 most relevant characteristics in order to investigate the effect that reducing the total number of features has on the overall performance of the classifier.

3.3.3. XGBoost

XGBoost is a gradient-boosting decision tree ensemble machine learning system that employs boosting decision trees. To enhance the efficiency of producing new decision trees throughout the feature selection process, XGBoost assigns a significance score to each feature in each iteration, indicating the value of each feature to model training and providing a foundation for constructing a new decision tree with gradient direction in the following iteration.

The weighted mean of each tree's gains is the feature's significance. The greater the feature importance score, the more significant and efficient a feature is to the XGBoost classifier model. In this research, we first categorise XGBoost based on all features, then calculate the importance of feature variables (FI) and arrange them in decreasing order based on information from the created model process. Lastly, the filtered features are entered into the classifier to build the prediction model.

3.4. Data Partitioning

During the process of data partitioning, the dataset is divided into the training, validation, and testing datasets respectively. The models were trained using the training dataset, which allowed them to understand the patterns derived from the input and fine-tune their weights. At the same time, the issue of overfitting was addressed by using the validation dataset while the model was being trained. On the other hand, a testing dataset was employed so that the models could be evaluated, and performance metrics could be calculated. For this research, the rule of 80/20 is used to split the dataset into a training set and a test dataset. In this splitting, 80% of the dataset was allocated into the training dataset, while 20% of the dataset was selected into the testing dataset. Following that, the training dataset was split once again using the same 80/20 approach to produce the final training dataset and the validation dataset. Fig. 5 shows the number of samples of actual dataset that has been split into training, validation, and testing dataset.

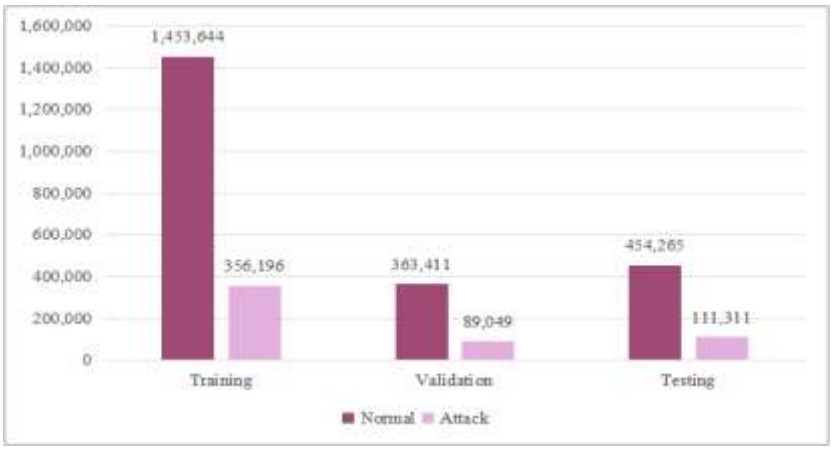


Fig. 5: Samples distribution of CICIDS 2017 according to the datasets

3.5. Data Resampling

Most machine learning models operate based on the assumption that each sample has the same number of classes. However, when there is an imbalance in the distribution of classes, this leads to the machine learning models having poor performance [21]. This is especially true for the classes that are underrepresented, while the performance of the classes that are overrepresented may be deceptive. The data from the majority class in the dataset are removed when using under sampling, but the data from the minority class are duplicated when using oversampling. In this paper, we have used under sampling technique to balance the CICIDS2017 dataset. This under sampling technique was implemented on the initial training dataset before it undergoes second data split into training and validation dataset. Fig. 6 shows the amount of dataset samples for training, validation and testing dataset after the under sampling approach.

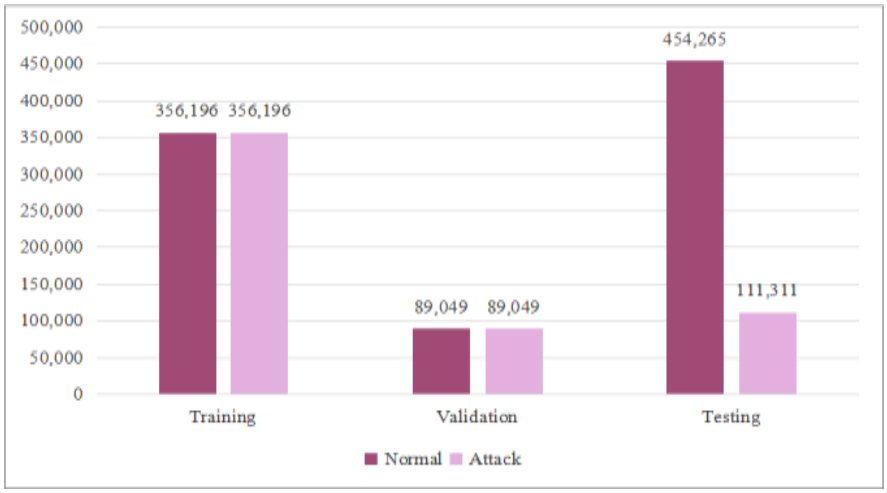


Fig. 6: Samples distribution of CICIDS 2017 after under sampling according to the datasets

3.6. Data Normalization

To handle greatly variable size or values, feature scaling normalises the individual features within a specified range. To scale features, various methods can be used. The Min-Max Normalization is one of them. The MinMax scaler was used to scale the feature vector X by individually rescaling each feature between zero and one.

3.7. Model Building

A deep neural network technique called autoencoder is trained to discover the best way to represent the input data to ensure certain desired qualities. Autoencoder in anomaly detection generally consists of two parts which are encoder and decoder. An encoder takes the input data and transforms it into a latent space which the dimension has been reduced, while a decoder takes the latent space and returns an output that is almost identical to the input. The success of feature extraction may be seen by comparing the input with the recovered output. The difference between the input and output is very minimal if the feature extraction is effective. Or else, there is a significant difference.

The edge and bias of the encoder and decoder are computed and tailored to the training set of data during the training phase. As a result, the autoencoder fails to successfully reconstruct the input if the input is different from the training input. The reconstruction error, defined as the difference between the input and reconstructed output. This attribute is used to find anomalies. The overview of architecture of an autoencoder is shown in Fig. 6. The architecture and the hyperparameters of the autoencoder are shown as in Fig. 7.

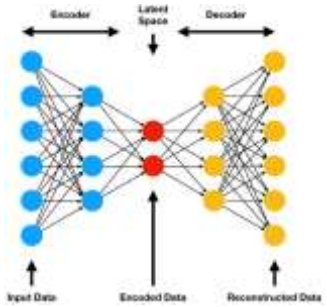


Fig. 6: Illustration of autoencoder architecture

InputLayer(shape)	Encoder
Dense(512)	
ReLU activation Function	
Dropout(0.1)	
Dense(64)	
ReLU activation Function	
Dropout(0.1)	
Dense(32)	
ReLU activation Function	
Dropout(0.1)	
Dense(8)	
ReLU activation Function	
Dense(32)	Bottleneck
ReLU activation Function	
Dropout(0.1)	Decoder
Dense(64)	
ReLU activation Function	
Dropout(0.1)	
Dense(512)	
ReLU activation Function	
Dropout(0.1)	
Dense(shape)	
Sigmoid activation function	

Batch size=256
Epochs=50
Optimizer=Adam
Learning rate=0.001
Loss function= Mean Square Logarithmic Error (MSLE)

Fig. 7: Architecture and Hyperparameters of Autoencoder

3.8. Performance Metrics

To evaluate the model, confusion matrix components were used in the performance metrics computations. The confusion matrix displays the amount of output that was properly or incorrectly classified in relation to the actual outcomes. In the confusion matrix, the data that belong to each predicted class are denoted by a column, while the data that belong to each actual class are denoted by a row, as can be seen in Fig. 8. When determining the performance metrics, the main components to consider are:

- True Positive (TP)

The attack samples were correctly predicted.

- True Negative (TN)

The non-attack samples were correctly predicted.

- False Positive (FP)

The attack samples were incorrectly predicted.

- False Negative (FN)

The non-attack samples were incorrectly predicted.

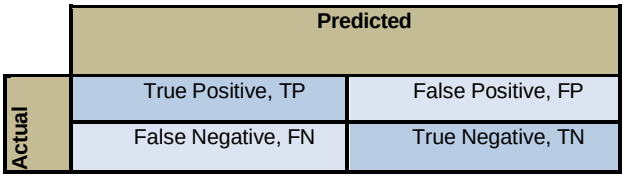


Fig. 8: Confusion Matrix

This study evaluates the effectiveness of a classifier using a variety of performance indicators, including precision, recall, and F1-Score. The performance metrics are briefly discussed in the subsection as below:

- Precision

A metric that determines the proportion of positive classification that were properly classified. Precision is useful for measuring whether a False Positive will result in a significant cost loss. Increases in precision value reflect a better performance of the model.

- Recall

A metric that determines the proportion of actual positives that were accurately detected. It evaluates the model's sensitivity. The better the model performs, the greater the recall value. Given the significant cost of False Negative, it is an effective metric of measurement.

- F1-Score

A metric that summarizes recall and precision in a single term. It is often employed when the model

yields an unequal class distribution, such as high recall and poor precision or low recall and high precision.

4. RESULTS AND DISCUSSION

Since there are several methods of data pre-processing in this research, we have conducted a variety of observations to comprehend the performance implications of the autoencoder to detect the anomalous or attack network data. Based on data resampling and feature selection technique, the findings describe the performance of autoencoder to detect the anomalies.

4.1. Data Resampling as Data Pre-processing

This data pre-processing technique aims to determine the impacts of dataset size to detect the intrusion attack samples based on two scenarios shown in Fig. 9. The results of the two scenarios, with and without data resampling, are shown in Table 1.

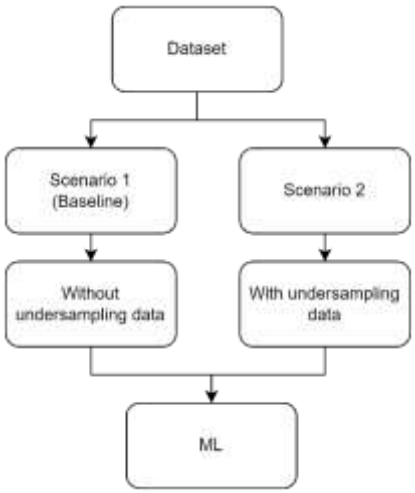


Fig. 9: Working scenarios.

The analyses in Table 1 below demonstrate that the outcomes of the suggested data resampling approach are quite encouraging. In terms of attack detection, we can observe that using the under sampled dataset with autoencoder method produces noticeably better results than using autoencoder on actual datasets. It was found that using under sampling techniques on the dataset produced better performance in detecting anomalies by about 83% precision and 67% recall. This produced a 71% F1-Score. While by using the actual dataset as input for the autoencoder, it is only able to produce maximum of 46% F1-Score due to 52% of precision and 50% recall in detecting anomalies.

Table 1: Performance metrics of Autoencoder using actual dataset and under sampled dataset

	Precision	Recall	F1-Score
Actual (Baseline)	0.52	0.50	0.46
Undersampled	0.83	0.67	0.71

The generation of a confusion matrix for the categorization of the network data shown in Fig. 10 and Fig. 11 is done so that the influence of the data resampling approach on the CICIDS2017 dataset can be evaluated. As illustrated in Figure 4.2 and Figure 4.3, the Autoencoder with data under sampling technique able to detect 39,817 anomalies traffic compared to 2,644 anomalies traffic detected without under sampling the dataset. The traffic detected by the autoencoder without under sampling constitutes around

37,000 network traffic were misclassified. This is because the baseline model without data resampling has an uneven distribution of classes which leads the model to be biased towards the majority class or the normal traffic. While in detecting the normal traffic accurately, there is no significant difference between both scenarios. In Scenario 1, the normal network traffic that able to be detected by the autoencoder is 445,781 without data resampling pre-processing technique while using the under sampled data as the input to the autoencoder detected 444,317 normal network traffic.

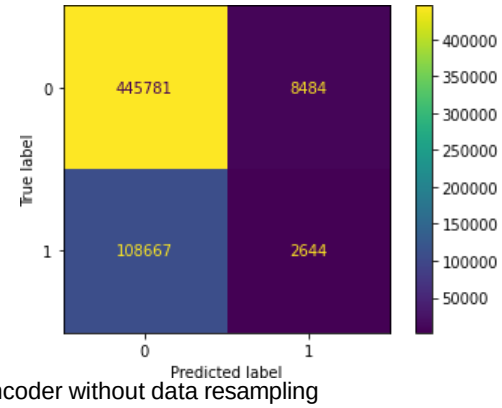


Fig. 10: Confusion matrix for Autoencoder without data resampling

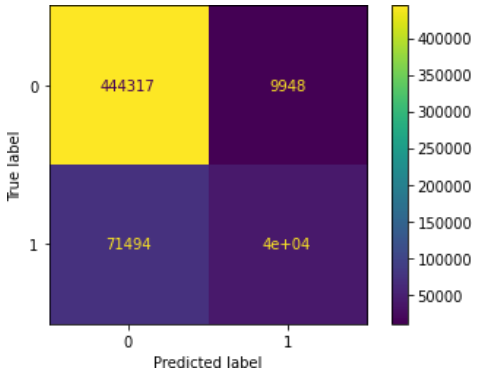


Fig. 11: Confusion matrix for Autoencoder with data resampling

4.2. Feature Selection as Data Pre-processing

The results of autoencoder which is represented as Scenario 3 that has been trained and evaluated with the dataset pre-processed with feature selection are discussed in this subsection. There are several feature selection models and parameters tested as shown in Fig. 12 to examine the performance of the machine learning algorithm.

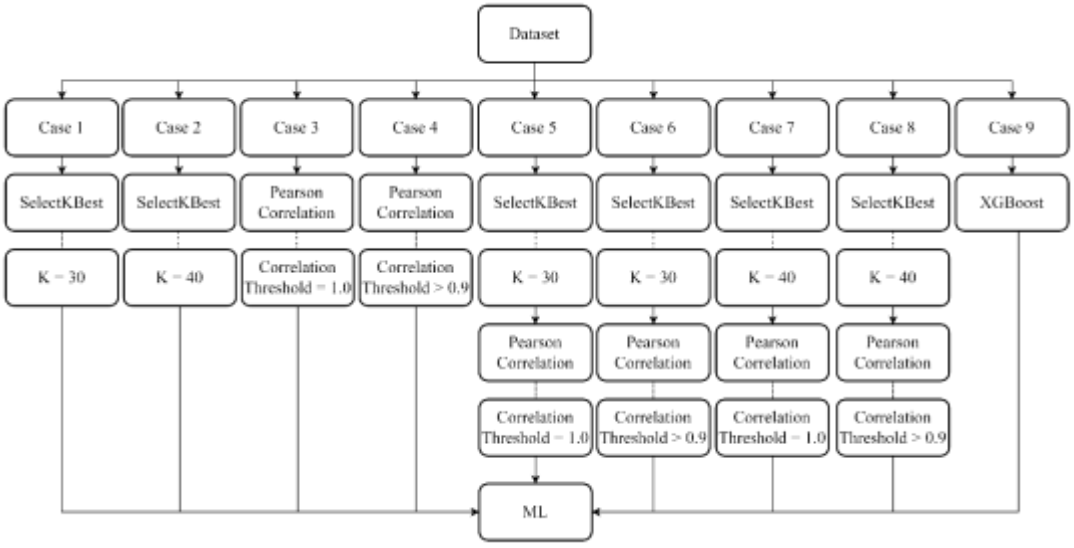


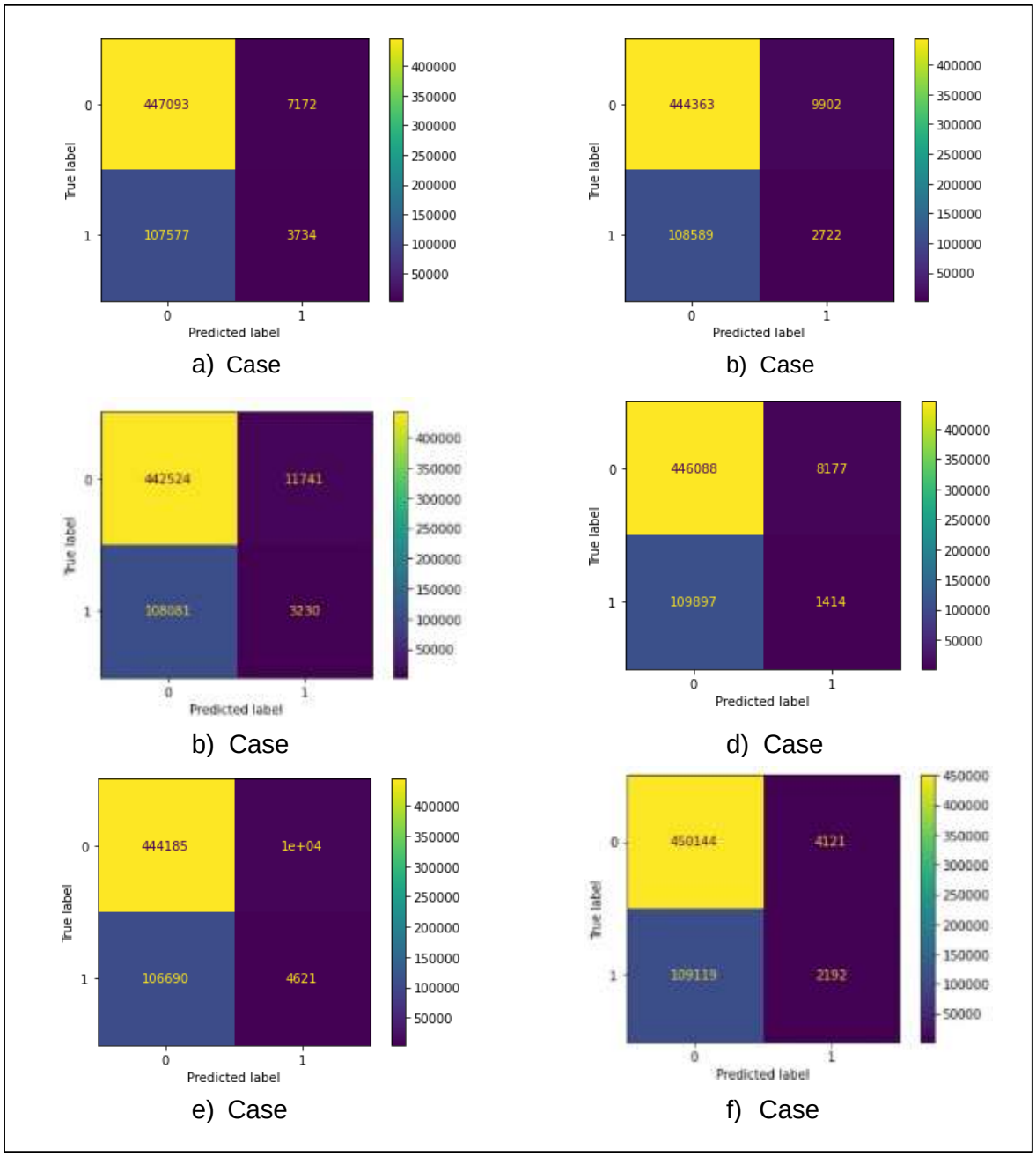
Fig. 12: Working cases for the feature selection

In Case 1 and 2, the top 30 and 40 of the best features are selected using SelectKBest algorithm while for Case 3 and 4, the absolute correlated features and features with correlation value that exceeds 0.9 are removed as calculated by Pearson correlation. Furthermore, a combination of SelectKBest and Pearson correlation model is examined with different parameters in Case 5, 6, 7, and 8 as shown in Fig. 12. Case 5 and 6, both cases are defined to have the same K value of 30 which SelectKBest are instructed to select 30 features that have the best feature importance. After that, in Case 5, Pearson correlation model is assigned to remove all the absolute correlated features with 1.0 correlation value while in Case 6, the Pearson correlation model eliminates the correlated features that exceeds the correlation threshold level of 0.9. Lastly, XGBoost is computed to determine the selection of the dataset features in Case 9. The result of each proposed model of feature selection by cases are summarized in Table 2.

Table 2: Performance metrics of Autoencoder with feature selection

Case	Model	No. of Features	Precision	Recall	F1-Score
1	SelectKBest	30	0.57	0.51	0.47
2	SelectKBest	40	0.51	0.50	0.46
3	Correlation	63	0.51	0.50	0.47
4	Correlation	39	0.47	0.50	0.45
5	SelectKBest + Correlation	29	0.49	0.50	0.46
6	SelectKBest + Correlation	14	0.58	0.51	0.46
7	SelectKBest + Correlation	37	0.52	0.50	0.47
8	SelectKBest + Correlation	22	0.51	0.50	0.46
9	XGBoost	38	0.56	0.51	0.48

Table 2 summarise the outcomes of each model of feature selection by cases that has been developed. Table 2 shows the precision, recall, and F_1-Score for the autoencoder's performance using different feature selection models and parameters. The confusion matrix is produced and displayed for each feature selection strategy by cases as shown in Fig. 13.


447093
7172
107577
3734
True label
Predicted label
a) Case
444363
9902
108589
2722
b) Case
442524
11741
108081
3230
b) Case
446088
8177
109897
1414
d) Case
444185
1e+04
106690
4621
e) Case
450144
4121
109119
2192
f) Case

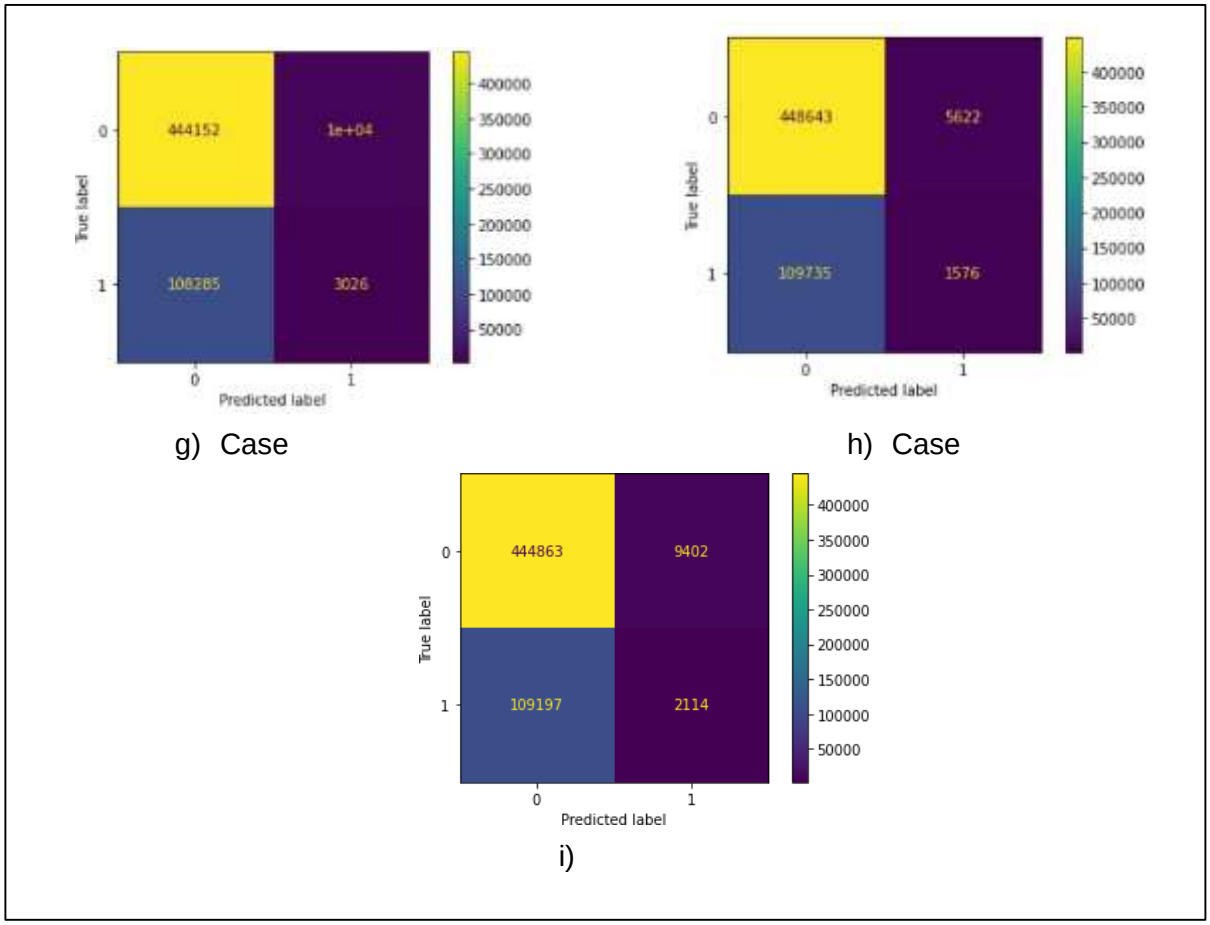


Fig. 13: Confusion matrix for each feature selection models by cases

Because F1-Score is a measurement that combines precision and recall, it is used to evaluate the feature selection procedures discussed in this part. F1-Score is mainly observed. When detecting the network traffic data into normal and attack class data, there is not much of a difference in F1-score between the various feature selection models and parameters, as demonstrated in Table 2. Most of the approaches achieved F1-score of between 45 and 48 percent. The machine learning model Autoencoder achieves the maximum F1-Score result of 48% by employing the XGBoost model, which selects 38 features in Case 9. By employing only correlation algorithm to remove correlated features that exceed the threshold level of 0.9, the approach to detect the anomalies with the lowest F1-score of 45%. In terms of precision score, Case 4 was only able to achieve 47% which is the lowest score among the other models while Case 6 is the highest score with 58%. In terms of recall score, there is only a slight difference between the cases with only 1% difference.

4.3. Feature Selection and Under sampling as Data Pre-processing

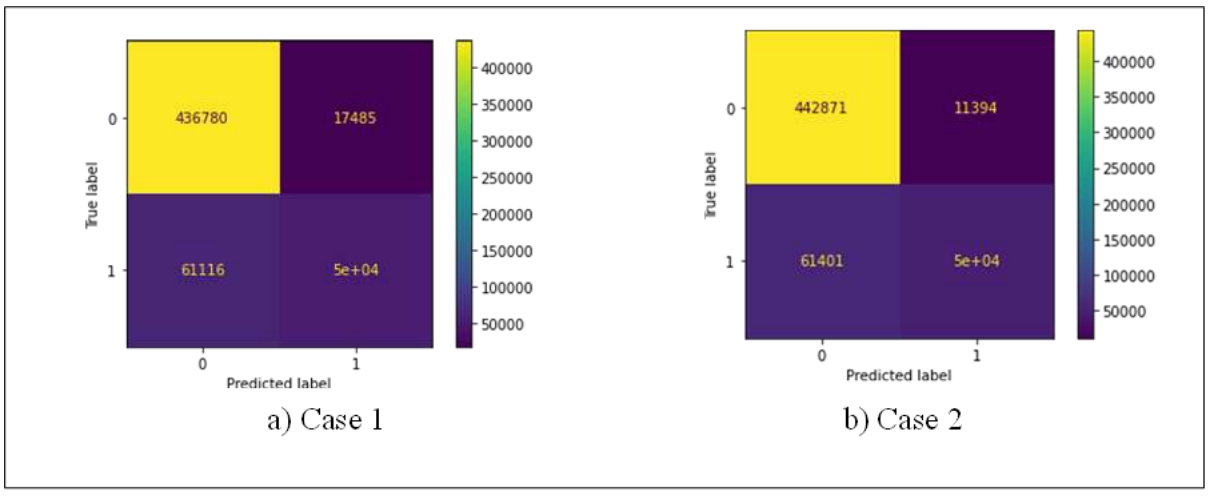
Referred to Fig. 1, the results of Scenario 4 which the autoencoder that was trained and tested using a dataset that had been pre-processed with feature selection and data under sampling are briefly discussed. The impact of feature selection together with data under sampling technique towards the performance of the Autoencoder are observed in this section. Table 3 summarise the outcomes of each model of feature selection by cases that has been developed. Table 3 shows the precision, recall, and F1-Score for the autoencoder's performance using under sampled data as the input with different models and parameters

as feature selection.

Table 3: Performance metrics of Autoencoder with feature selection and under sampling

Case	Model	No. of Features	Precision	Recall	F1-Score
1	SelectKBest	30	0.81	0.71	0.74
2	SelectKBest	40	0.85	0.71	0.75
3	Correlation	63	0.84	0.71	0.75
4	Correlation	39	0.83	0.69	0.73
5	SelectKBest + Correlation	29	0.85	0.68	0.73
6	SelectKBest + Correlation	14	0.88	0.70	0.75
7	SelectKBest + Correlation	37	0.84	0.70	0.74
8	SelectKBest + Correlation	22	0.85	0.68	0.72
9	XGBoost	38	0.85	0.72	0.76

Since F1-Score is a trade-off metric of precision and recall, it is also used to evaluate the feature selection procedures discussed in this part. Thus, F1-Score is mainly analysed. Referring to Table 3, there is an improvement when utilizing feature selection and under sampling approach as machine learning pre-processing to classify the network traffic into normal and attack traffic. This observation tallies with the imbalance classes found in the dataset. Most of the approaches in Table 3 scored between 72% and 76% on F1-Score. However, by employing XGBoost as the feature selection model and using under sampled data as the input to the classifier in Case 9, the autoencoder outperformed the other approaches in Table 3 by 76% of F1-Score and 72% of recall. Case 8 achieved 72% which is the lowest F1-Score in this analysis. The autoencoder in Case 5 and Case 8 struggle to make detection of anomalies where the recall is low with only 68%. The precision for Case 4 is the lowest with 83% whereas Case 6 managed to achieve 88% precision. The confusion matrix is produced and displayed for each feature selection strategy by cases as shown in Fig. 14.



a) Case 1 b) Case 2

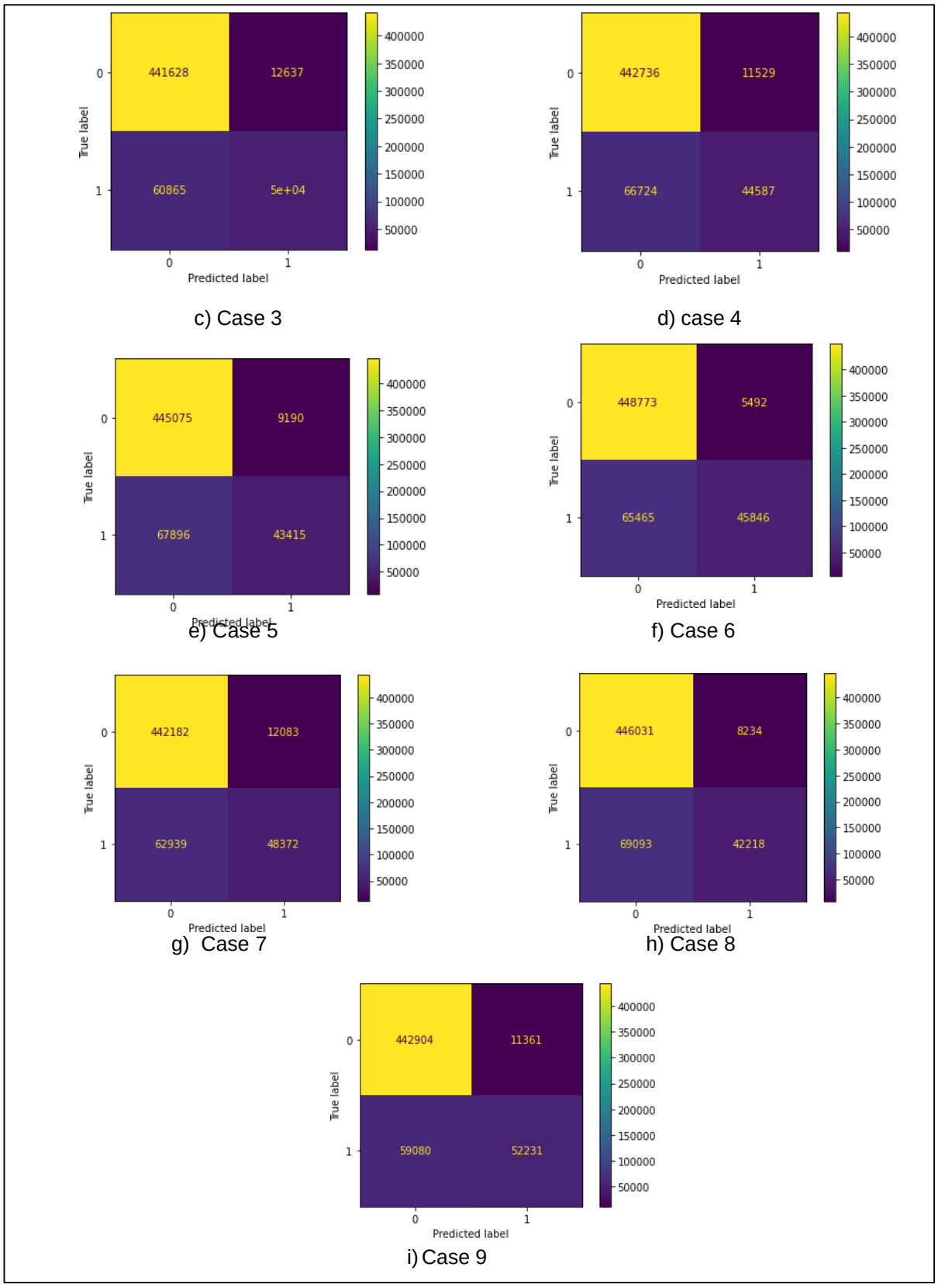


Fig. 14: Confusion matrix for each feature selection models with under sampling pre-processing by cases

4.4. Comparison with Previous Work

Ullah in [40], proposed a new dataset that was generated in IoT environment as shown in Figure 4.16 to create a dataset that able to reflect the current novel attack. This dataset named IoTID20 is publicly published. The proposed dataset is trained and tested with several machine learning models such as Support Vector Machine (SVM), Logistic Regression, Gaussian NB and LDA [40]. The models undergo several data pre-processing techniques including feature selection. Correlation algorithm is used as the feature selection model to reduce the number of features which correlated with each other. The correlated features with correlation coefficient of 0.7 and above are removed in the project. The remaining features are then ranked according to the importance score using Shapiro-Wilk algorithm to select the significant features that influenced the performance of experiment ML models.

This section will compare the results of precision, recall and F1-Score in previous work [40] with our precision, recall and F1-Score of Autoencoder with XGBoost as feature selection model and undersampling technique during data pre-processing stages. The comparison is tabulated in Table 4.

Table 4: Performance metrics previous work and current work

Model	Precision	Recall	F_1-Score
SVM (%)	55	37	16
Logistic Regression (%)	25	39	30
Gaussian NB (%)	70	66	62
LDA (%)	71	71	70
Autoencoder (Our Approach) (%)	94	63	72

Based on Table 4, the current work has managed to achieve slightly higher detection performance than previous findings for the highest model which utilises LDA model. The current work which utilizes Autoencoder using under sampled data as input and using XGBoost as feature selection model during data pre-processing stages, has managed to improve by 56% compared to the lowest model performance in the previous work which is SVM. However, only a minor improvement of 2% if compared with the highest model performance in the previous work which is LDA. Comparing between the three models, the current work makes Autoencoder the top performing model at 72% F1-Score compared to previous research which is 70%.

5. DISCUSSION OF FINDINGS

In this chapter, the proposed dataset as well as the performance evaluation metric used in this thesis is introduced. Four main procedures in data pre-processing techniques which includes under sampling and feature selection are performed extensively on the dataset with three main feature selection algorithms. The promising classification results show the influence of the data pre-processing techniques towards the performance of the Autoencoder classifier.

CONCLUSIONS

This thesis focuses on network anomaly detection that is used to detect intrusion activities in computer networks. Machine learning has been an exciting field in the network security area. In this research, we are encouraged to employ network data that accurately replicates the modern-day actual attack scenario. Due to that, the dataset that was made available as CICIDS 2017 was utilised for the detection of anomalies.

A necessary yet crucial component of efficient network security is the ability to recognise malicious traffic and to recover the system once it has been attacked. This ability is necessary but not sufficient on its own. Therefore, a machine learning algorithm called Autoencoder with proposed architecture is used with the intention of identifying malicious traffic patterns. The dataset was cleaned to remove the unnecessary samples. Three new datasets were then created from the original dataset for the training, validating, and testing process of the autoencoder. The ML

classifier was trained and tested with those datasets. The performance of the classifier was then evaluated based on the usual performance metrics.

To study the potential effects that the pre-processing of the data may have on the performance of the machine learning model, a variety of different data pre-processing procedures are carried out. This is because, during the research, the CICIDS 2017 dataset was found to be imbalanced for its distribution of normal and attack class samples. One of the data pre-processing techniques that were examined towards the effectiveness of classifier is data resampling. Based on the findings and accompanying discussions, we are able to reach the conclusion that the autoencoder obtained greater results in terms of F1-Score when it came to the identification of anomalous traffic when it utilised the under sampling method. The F1-score difference between the autoencoder detecting the anomalies using the normal and under sampled dataset is 25%. The second method of data pre-processing experimented in this project is feature selection. It was found that the Autoencoder's performance can be improved by employing the XGBoost algorithm as the feature selection model. This leads to an approximately 5% boost in the F1-Score for spotting network anomalies. Thus, this research examined the potential impact of data pre-processing, such as data resampling and feature selection, that have influenced on the capability of an Autoencoder to recognise anomalies which referred as network attacks occurring within a network system towards an imbalanced dataset.

A comparison is made between the previous work and current work. The current work which employing XGBoost algorithms as feature selection model and under sampled data as the input to the Autoencoder, has managed to achieve 72% of F1-Score, which an improvement for the detection of anomalies compared to previous work that utilized LDA model as the top model performance in the previous work.

REFERENCES

[1] M. v. Pawar and J. Anuradha, “Network security and types of attacks in network,” in Procedia Computer Science, 2015, vol. 48, no. C. doi: 10.1016/j.procs.2015.04.126.

[2] S. T. Sarasamma, Q. A. Zhu, and J. Huff, “Hierarchical Kohonenen Net for anomaly detection in network security,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 35, no. 2, 2005, doi: 10.1109/TSMCB.2005.843274.

[3] Lilien, G.L., A. Rangaswamy. 1998. Marketing Engineering: Computer-Assisted Marketing Analysis and Planning, Addison-Wesley, 67-84.

[4] B. B. Gupta, G. M. Perez, D. P. Agrawal, and D. Gupta, Handbook of computer networks and cyber security: Principles and paradigms. 2019. doi: 10.1007/978-3-030-22277-2.

[5] P. Sornsuwit and S. Jaiyen, “Intrusion detection model based on ensemble learning for U2R and R2L attacks,” in Proceedings - 2015 7th International Conference on Information Technology and Electrical Engineering: Envisioning the Trend of Computer, Information and Engineering, ICITEE 2015, 2015. doi: 10.1109/ICITEED.2015.7408971.

[6] Jam, F. A., Sheikh, R. A., Iqbal, H., Zaidi, B. H., Anis, Y., & Muzaffar, M. (2011). Combined effects of perception of politics and political skill on employee job outcomes. African Journal of Business Management, 5(23), 9896-9904.

[7] H. Liu, B. Lang, M. Liu, and H. Yan, “CNN and RNN based payload classification methods for attack detection,” Knowl Based Syst, vol. 163, 2019, doi: 10.1016/j.knosys.2018.08.036.

[8] S. Maya, K. Ueno, and T. Nishikawa, “dLSTM: a new approach for anomaly detection using deep learning with delayed prediction,” Int J Data Sci Anal, vol. 8, no. 2, 2019, doi: 10.1007/s41060-019- 00186-0.

[9] N. AlDahoul, H. Abdul Karim, and A. S. Ba Wazir, “Model fusion of deep neural networks for anomaly detection,” J Big Data, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00496-w.

[10] Doan, T.-N. (2023). A Novel LoRa-Based Platform for Remote Monitoring of Large-Scale Rice Fields. International Journal of Membrane Science and Technology, 10(2), 1301-1322. https://doi.org/10.15379/ijmst.vi.1307

[11] I. Ullah and Q. H. Mahmoud, “A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020, vol. 12109 LNAI. doi: 10.1007/978-3-030-47358-7_52.

[12] M. Klassen and N. Yang, “Anomaly based intrusion detection in wireless networks using Bayesian classifier,” in 2012 IEEE 5th International Conference on Advanced Computational Intelligence, ICACI 2012, 2012. doi: 10.1109/ICACI.2012.6463163.

DOI: https://doi.org/10.15379/ijmst.v10i1.1808