

Machine learning to improve the performance of anomaly-based network intrusion detection in big data

Siriporn Chimphee, Witcha Chimphee

Department of Data Science and Analytics, Faculty of Science and Technology, Suan Dusit University, Bangkok, Thailand

Article Info

Article history:

Received Nov 14, 2022

Revised Jan 5, 2023

Accepted Jan 10, 2023

Keywords:

Class imbalance

CSE-CIC-IDS-2018

Feature selection

Machine learning

Network intrusion detection

ABSTRACT

With the rapid growth of digital technology communications are overwhelmed by network data traffic. The demand for the internet is growing every day in today's cyber world, raising concerns about network security. Big Data are a term that describes a vast volume of complicated data that is critical for evaluating network patterns and determining what has occurred in the network. Therefore, detecting attacks in a large network is challenging. Intrusion detection system (IDS) is a promising cybersecurity research field. In this paper, we proposed an efficient classification scheme for IDS, which is divided into two procedures, on the CSE-CIC-IDS-2018 dataset, data pre-processing techniques including under-sampling, feature selection, and classifier algorithms were used to assess and decide the best performing model to classify invaders. We have implemented and compared seven classifier machine learning algorithms with various criteria. This work explored the application of the random forest (RF) for feature selection in conjunction with machine learning (ML) techniques including linear regression (LR), k-Nearest Neighbor (k-NN), classification and regression trees (CART), Bayes, RF, multi layer perceptron (MLP), and XGBoost in order to implement IDSS. The experimental results show that the MLP algorithm in the most successful with best performance with evaluation matrix.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Witcha Chimphee

Department of Data Science and Analytics, Faculty of Science and Technology, Suan Dusit University

295 Nakornratchasima Road, Dusit, Bangkok, Thailand

Email: witcha_chi@dusit.ac.th, witcha.chi@gmail.com

1. INTRODUCTION

Information and communication technology (ICT) now play a critical role in all aspects of business and people's lives. At the same time, in the big data era, cyber-attacks against ICT are growing more complicated and are rapidly expanding [1]. As a result, network attacks are the most pressing issue in modern society. Malicious threats, on the other hand, are always emerging and evolving, necessitating a sophisticated security solution for the network. The number of networked computers has been growing because of widespread Internet use. Data science techniques have been applied in recent years to construct effective intrusion detection systems (IDSs) that can discern between legitimate and hijacked communications. IDSs have been widely used to address for monitoring and detecting malicious actions in communications networks. IDSs can be divided into three categories: Systems for detecting intrusions that use signatures, anomaly-based systems, and hybrid systems. Anomaly-based can identify unknown harmful actions by identifying any deviations from a model built based on typical behavior, whereas signature-based can identify known assaults with corresponding collected signatures, however it has a high false alarm rate [2]–[5]. Current anomaly intrusion detection methods are suffering from accurate performance. Some of datasets suffer from providing diversity and volume of

network traffic, some do not contain different or latest attack patterns, while others lack feature set metadata information [6]. The hybrid IDS obtained combining anomaly-based and misuse-based IDSs and shows that the hybrid IDS is a more powerful system.

The idea of intrusion detection in a system is the recognition of an effort to enter a system and impact components like integrity, availability, confidentiality, or the standard of services in the system. One method for studying and monitoring diverse network operations to find signs of security concerns is to use an intrusion detection system. The network intrusion detection systems (NIDS) has a fundamental role in solving security challenges. NIDS monitors the traffic of network to detect any suspicious activity, it analyzes information from network traffic to detect breaches of security, that comprise intrusions, misuse, and anomaly. A NIDS is implemented in a network to analyze traffic flows to detect security threats and protect digital assets [7], [8]. NIDS should be dealing with issues like high dimensionality and enormous traffic volumes [9]. While machine learning techniques are useful in NIDS, they have limitations when dealing with large amounts of data on the network. Feature selection (FS) is a method for removing unnecessary and redundant features and choosing the most suitable feature subset that will result in a better classification of patterns which belong to various classes of attack [10]. As a result, three crucial variables for NIDS development are preprocessing, feature reduction, and classifier algorithms [11]. However, the challenges facing in network intrusion detection system are to handle huge amount of data, high false alarms, and imbalance data.

As an evaluation, use a more realistic IDS and a current security dataset with real network traffic CSE-CIC-IDS-2018 for a wide range of intrusions and normal behavior. The CSE-CIC-IDS-2018 dataset is big data, associated with specific properties, such as volume, variety, velocity, variability, value, and complexity [12], [13]. The term "volume" refers to the amount of data present. Velocity relates to high-speed data processing, while Variety refers to the data's complexity, such as data from multiple sources or data with different data structures [14]–[17]. Thus, further feature engineering should be done for the data set. Traditional methods may have difficulty handling the high data volume, the diversity of data formats [18]. Many studies concentrate on the machine learning (ML) technique and view the issue as a classification of benign and malicious traffic based on pertinent dedicated data sets in order to enhance IDS performance over huge data [2].

Machine learning techniques have been widely applied in the network security arena over the last two decades. As a result, previous researchers explored a variety of algorithms for intrusion detection based on traditional machine learning [1]. Various popular machine learning classification algorithms, namely Bayesian network, Naive bayes classifier, decision tree, random decision forest, and artificial neural network, to detect intrusions due to provide intelligent services in the domain of cyber-security [15]. Basnet et al. [19] addressed multip-layer perceptron (MLP) to binary classification. Decision tree (DT), random forest (RF), DT-based bagging, gradient boosting, extratree, Adaboost, XGBoost, k-Nearest Neighbor (k-NN), Ncentroid, linearSVC, RBFSVC, and Logistic Regression are the 12 supervised learning methods suggested by D'hooge *et al.* [20]. The tree-based classifiers outperformed the others, with XGBoost coming out on top. The highest levels of precision, accuracy, and recall were 96%, 99%, and 79%, respectively. Filho *et al.* [21]. there were 33 features obtained for each dataset. The precision and recall were both 100%. Ramos *et al.* [22] RF, DT, the RF and the accuracy of DT students was 99.99%, the precision was 100% and the recall was 99.99%. Kanimozhi and Jacob [23] only botnet examples were used to train the MLP classifier. The area under the curve (AUC) for this study was one, which is a perfect score. All of the related accuracy, precision, and recall ratings were perfect.

All above mentioned, the goal of this research was to find the best classifier among six options (MLP, RF, k-NN, SVM, Adaboost, and Nave Bayes). With an AUC of 1, the MLP model emerged as the clear winner. This perfect AUC score included accuracy, precision, and recall ratings of 99.97%, 99.96%, and 100%, respectively. Decision trees, random forests, and support vector machines are among the algorithms developed by Lypa *et al.* [24]. They discovered that the decision tree provided the best results. Zhou and Pezaros [25] compared six machine learning classifiers and they found that the decision tree the highest intrusion detection accuracy. Hua [3] implemented data pre-processing approach with under-sampling and embedded feature selection, then utilize LightGBM to classification attacks. Ashraf *et al.* [26] proposed comparing some of the most efficient machine learning algorithms-J48, Naive bayes, and random forest.

The main contributions of this work are summarized as follows:

- Exploration of the number of big data with a network malicious traffic.
- Presents feature dimensions that impact on the classification performance from a labeled dataset with benign and malicious traffic to the detection accuracy.
- Proposes and conducts the experiment while considering the impact of the data sample imbalance.
- Using cluster centroids in conjunction with sampling strategies.
- Using the CSE- CIC-IDS-2018 dataset for NIDS, evaluate the performance of seven different machine learning classifiers and scripts for detecting types of attacks.
- Reporting various evaluation metrics for comparison the IDS model with imbalance dataset.

The remainder of the paper is laid out as follows. In section 2, we explain the sequence of research, including research design and research procedure. Section 3 explains the results of research and the comprehensive discussion. Finally, section 4 concludes the article, as well as the strengths and weaknesses of the suggested model, as well as future work.

2. METHOD

A brief exploratory profile of the CSE-CIC-IDS-2018 [27] dataset has been discussed in this section. This dataset is proposed by the communications security establishment (CSE) and the Canadian institute for cybersecurity (CIC). The datasets contain different incursion states and show real-time network behavior. Furthermore, it is spread as a full network that encapsulates all the inner network traces to calculate data packet payloads. These dataset properties are relevant to our study. Six types of infiltration circumstances are included in the dataset: Denial-of-service (DoS) assault, Brute force attack, botnet attack, distributed denial-of-service (DDoS) attack, web attack, and infiltration, as well as 14 types of intrusions: brute force-web, botnet, SSH brute force, DDoS- high orbit ion cannon (HOIC) attacks, DDoS-low orbit ion cannon (LOIC)-UDP attacks, DDoS-LOIC-HTTP attacks, SQL injections, Brute Force-XSS, DoS GoldenEye attacks, DoS Hulk assaults, DoS slow HTTP test attacks, infiltration, and DoS Slowloris attacks are all examples of DDoS attacks [28].

We highlight the drawbacks and suggest an ML-based traffic classification technique for IDS in this study. We specifically suggest a data pre-processing method with embedded feature selection and under-sampling in order to correct the imbalance of traffic samples and extract dominating characteristics from input flows. The step of methodology framework is presented in Figure 1. In this section, it is explained the proposed framework as following.

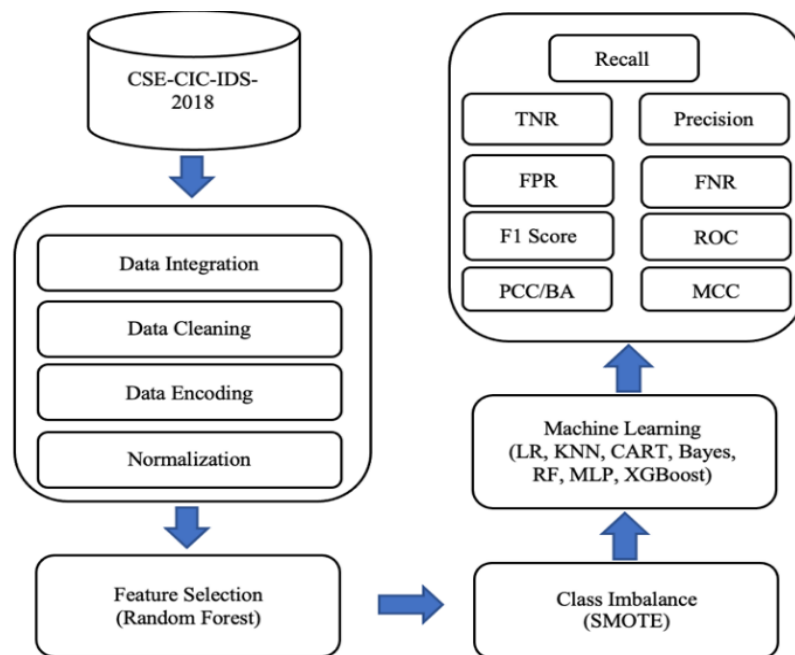


Figure 1. The proposed framework

2.1. Data preprocessing

Data processing plays a vital role in an IDS and essential first step in enhancing the training process for the machine learning models [29]. It can be used to derive data preprocessing, which has a direct impact on how well a model performs in terms of classification. By employing algorithms, it may overcome the technical issue of data pretreatment and achieve a higher level of performance. The process of data preparation, which includes data integration, data cleaning, data encoding, and data normalization utilizing feature selection, is described in detail in this section. The effectiveness of model training depends on data preparation techniques, which have not received enough attention while being essential given the large number of gathered samples [3].

2.1.1. Data integration

The CSE-CIC-IDS-2018 dataset containing roughly 16,233,002 instances divided on 10 files, each row having 80 features. The contents of these with about 17% of these instances representing attack traffic [30], [31]. This dataset includes 14 types of attacks in 6 different scenarios is quite large. The dataset comprised of ten raw-data files containing 16 million distinct network flows that covered various forms of attacks. Once the data was pre-processed, we then integrated the dataset into a single database. This was done by taking all of the data from the raw-data files and combining it into a single dataset. This dataset was then stored in a database, which allowed us to quickly access and analyze the data.

2.1.2. Data cleaning

The preparing of data helps to maintain quality and makes for more accurate analytics, which increases effective, intelligent decision-making. Data cleaning is critical for any data-driven company. Data cleaning is a subset of data pretreatment, which is a task that improves the usability of a dataset. It's possible that the presence of noisy data is due to a model's technological flaw. Missing values and worthless features were eliminated from the entire dataset during the data cleaning stage of this investigation. The samples with "Infinity", "NaN" and timestamps were removed. The missing values are filled by a mean value and various features value ranges are often scaled in standard format using StandardScaler.

2.1.3. Data encoding

To convert the labels into numerical values, data encoding may be utilized. The collection of potential values that the model might predict is equally binary because our dataset is a binary classification problem. In actuality, the dataset labels are "0" and "1," with "0" standing for "Benign" and "1" for "Attack". Thus, the model can predict either '0' or '1' for each observation. By encoding the labels, the model can better interpret the dataset. For example, if the labels were not encoded, the model would interpret a 'Benign' label to be a text string. By encoding the labels, the model can interpret them as numerical values, allowing it to better process and learn from the data.

2.1.4. Normalization

The transformation is required when there is a significant disparity between the maximum and least values of data. Data normalization is an essential part of data preprocessing, particularly for intrusion detection methods that rely on statistical attributes extracted from the data at hand [32]. Besides, machine learning-based techniques typically require the normalization of input data to avoid any undesirable bias due to existing differences in the magnitudes of the variables' values. To convert values into scaled values, normalization is required, which enhances model performance [33].

- *Standardization*: to transform continuous and quasi continuous features. The Standardization normalizes the data by removing the mean and scaling the data to unit variance [32]. Standardization can be denoted as $X_{scaler} = \frac{x-\mu}{\sigma}$, where X_{scaler} = generate value, μ =mean, and σ = standard deviation, $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$.
- *Min-Max Normalization*: Data is transformed into a range of 0-1 using the Min-Max normalization. On the data, Min-Max performs a linear transformation 9 [34]. Because the lowest and maximum values of features are unknown and the dataset is uneven, the technique is performed before splitting the dataset and after balancing to avoid bias induced by outliers in the unbalanced dataset [14]. The Min-Max normalization can be described as $X_{scaler} = \frac{x-\min(x)}{\max(x)-\min(x)}$ where X_{scaler} is transformed value [35].

2.1.5. Feature selection

Feature engineering is the process of transforming raw data into useful features that help us to understand our model better and increase its predictive power. Dimensionality reduction strategies are discussed, then feature selection methods are classified to show how good they are for training and detection [36]. Feature choosing the right characteristics and converting them is only one aspect of machine learning engineering. The models perform better because of the dataset preparation since it makes it compatible with the algorithm [13]. Feature selection is a technique for extracting useful characteristics that accurately reflect the original dataset. It improves model performance by deleting inessential or noisy characteristics without distorting the original data pattern or removing essential features [37], [38]. Selection of the most pertinent features in data has been shown to boost the efficiency of detection in terms of accuracy and computing efficiency, therefore playing a crucial role in the design of an anomaly-based IDS [36]. The selection of features is aimed at finding a subcategory of features within a given set that are sufficiently complete to reflect the information, and the elements in the subcategory are extremely important for prediction [39], [40].

2.2. Class imbalance dataset

For cybersecurity and machine learning, class imbalance is a critical consideration. The most crucial goal for researchers is to improve detection accuracy. However, if the dataset is unbalanced and a single category makes up most of the data, using accuracy as a single metric is not recommended. This uneven structure must be formulated, as evidenced by the system's efficiency. To address the problem of class-imbalanced data, which frequently leads to a low rate of anomaly detection, random oversampling and synthetic minority oversampling technique (SMOTE) are used [41], which typically results in a low anomaly detection rate can be used to generate additional data in minority classes where there is a scarcity of data [12]. *Imbalance Ratio* = $\frac{\max_i\{x_i\}}{\min_i\{x_i\}}$ may be used to compute the imbalanced ratio, which can then be utilized as the matrices [42]. Where X_i shows the data size in the class i . In other words, the imbalance ratio is the ratio of the maximum and minimum numbers of instances of each class. As a result, this imbalance rate should be reduced in order to improve the system's efficiency. Class imbalance occurs when one class label is disproportionately represented as compared to another class label. As shown in Table 1, the number ratio of benign traffic to the total is up to 83.07%, causing an imbalanced classification problem.

Table 1. cse-cic-ids-2018 data distribution [42]

Class Label	Number	Volume (%)
Benign	13484708	83.0700
DDoS attack-HOIC	686012	4.2260
DDoS attacks-LOIC-HTTP	576191	3.5495
DoS attacks-Hulk	461912	2.8455
Bot	286191	1.7630
FTP-BruteForce	193360	1.1912
SSH-Bruteforce	187589	1.1556
Infiltration	161934	0.9976
DoS attacks-SlowHTTPTest	139890	0.8618
DoS attacks-GoldenEye	41508	0.2557
DoS attacks-Slowloris	10990	0.0677
DDoS attack-LOIC-UDP	1730	0.0107
Brute Force -Web	611	0.0038
Brute Force -XSS	230	0.0014
SQL Injection	87	0.0005
Total	16,232,943	100

2.3. Classification model

The training model is created and is made to fit with different classifiers to check with delivers the best performance by using a `model.fit()` method. The efficiency of an IDS is directly related to the learning model and dataset quality [43]. Classification is the way toward predicting the class of given data. The IDS classify binary and multiclass attacks in terms of detecting whether the traffic has been considered as benign or an attack. The binary classification consisted of two clusters while the multiclass dataset consisted of n clusters. Thus, multiclass classification is considered more complex than binary classification due to the nature of the classification being more than two classes; this in turn causes strains on algorithms through computational power and time and can therefore lead to less effective result from the algorithm [44]. Classification involves analyzing and assigning each dataset to either a normal or an abnormal class. As new instances, the existing structures are maintained. Classification can be used for both misuse and detection of anomalies but used for misuse more predominantly. In the present study, seven machine learning algorithms and feature selection with class imbalance handle as following.

2.3.1. Linear regression (LR)

One supervised machine learning algorithm is linear regression. This approach is referred to as a mix of input variable (x) and output variable prediction (y). Simple linear regression is the linear model's representation when there is just one input value (x). The model is referred to as multiple linear regression if the input values (x) are multiple. This model uses numeric data for both the input and output variables. The correlation between the independent and dependent variables is the definition of this method. Kumar and Raaza [45] the independent variable is thought of as input values, and the dependent variable is thought of as output values.

2.3.2. Random forest classifier (RF)

Random forest (RF) is a machine learning classifier that averages the outcomes of many decision trees applied to distinct subsets of a dataset to enhance prediction accuracy. It is comparable to the

bootstrapping algorithm with the CART decision tree model. It makes an effort to construct many CART models using various samples and initial variables. RF consists of large number of decision trees working individually to predict an outcome of a class where the final prediction is based on a class that received majority votes [46]. The error rate is low in random forest as compared to other models.

2.3.3. K-nearest neighbors (KNN)

The K-nearest neighbor (KNN) algorithm is a simple machine learning algorithm that takes into account the closest neighbors. It is a nonparametric technique for classification and is a simple and straightforward machine learning algorithm. The idea of this algorithm is that a sample is most similar to S samples in the dataset. If most of these S samples is most similar in the dataset. If most of these S samples belongs to a specific category, then the sample also belongs to that category [47]. The principle of KNN is that when a new value x is predicted, the category to which x belongs is determined by the category of the nearest K points. K is chosen.

2.3.4. Classification and regression trees (CART)

A decision tree is a set of decision nodes that start at the root. the benefits of utilizing a decision tree include easy interpretation, efficient handling of outliers, no need for the linear separation of classes, dependent features. CART is a simple nonlinear supervised ML algorithm used for classification and regression. In CART, the target variable should be categorical, whereas in regression tree the target variable should be continuous. In CART, Gini index is a metric used for classification as $Gini\ index = 1 - \sum_{i=1}^c P_i$ [48]. Where, c is the number of classes and P_i is the probability of each class in the dataset.

2.3.5. Multip-layer perceptron (MLP)

An artificial neural network (ANN) called a multilayer perceptron (MLP) contains three layers: an input layer that receives input data, an output layer that generates predictions, and one or more hidden layers in between the input and output layers. The layers cover up the computing engine. MLP is a well-known algorithm that works in the same way as feedforward in that data flows forward. Neural networks are a type of artificial intelligence. A backpropagation technique is used to train MLP neurons. MLP applications include pattern categorization, recognition, prediction, and approximation [6].

2.3.6. Naïve Bayes (Bayes)

Among the most popular is the Bayesian algorithm. It is an easy approach of developing classifiers that assign class labels to problematic cases identified as values of feature vectors, where class tags are chosen from a restricted collection [49]. The Bayesian formula is $p(c|x) = \frac{p(x|c)p(c)}{p(x)}$ [46]. Where p(c) is the class "prior" probability, p(x|c) is the class conditional probability of sample x concerning class token c, and p(c|x) is the posterior probability, which reflects the confidence that hypothesis c holds after seeing the training sample data x.

2.3.7. XGBoost

XGBoost was mainly designed for speed and performance using gradient-boosted decision trees. XGBoost or eXtreme Gradient Boosting helps in exploiting every bit of memory and hardware resources for tree boosting algorithms. It gives the benefit of algorithm enhancement, tuning the model, and can also be deployed in computing environments. XGBoost can perform the three major gradient boosting techniques, that is gradient boosting, regularized boosting, and Stochastic Boosting. It also allows for the addition and tuning of regularization parameters, making it stand out from other libraries. The algorithm is highly effective in reducing the computing time and provides optimal use of memory resources [34].

2.4. Evaluation model

A crucial aspect of any project is evaluating the machine learning algorithms. When measured against accuracy score, a model might produce results that are satisfactory, but when measured against other metrics, it might produce unsatisfactory results. The suggested IDS was assessed using a number of criteria, including accuracy (ACC), false alarm rate (FAR), and detection rate (DR) [37]. The primary instance for evaluation includes connections that are classified as true positive (TP: Number of connections successfully classified as anomalies by the classifier), true negative (TN: Number of normal connections successfully classified as normal by the classifier), false negative (FN: Number of anomalies connections that are misclassified as normal by the classifier), and false positive (FP: Number of normal connections that are misclassified as anomalies by the classifier). The CM is a 2 x 2 matrix, where the rows represent actual classes and the columns represent expected classes.

This paper uses nine key criteria to evaluate a particular intrusion detection algorithm: Sensitivity/Recall/TPR, Specificity/TNR, Precision, FPR, FNR, F1-score, ROC, PCC/BA, and MCC.

- Sensitivity also known as the true positive rate (TPR) or Recall is calculated as $Sensitivity = \frac{TP}{TP+FN}$. Since the formula doesn't contain FP and TN, sensitivity may give a biased result, especially for imbalanced classes.
- Specificity also known as true negative rate (TNR) is calculated as $Specificity = \frac{TN}{TN+FP}$. By dividing the total number of true positives (TP) by the total number of true positives and false positives, precision is determined (FP). In an imbalanced dataset, FN and TN are usually much higher than TP and FP, so ignoring FN and TN in the precision calculation can lead to an overestimation of the precision score.
- False-positive rate (FPR) is a measure of the probability that normal traffic is classified as malicious traffic by the detection model. FPR is calculated as $FPR = \frac{FP}{FP+TN}$.
- False-negative rate (FNR) or miss rate is the probability that a true positive will be missed by the test. FNR is calculated as $FNR = \frac{FN}{TP+FN}$.
- F1-score incorporates both Recall and Precision and is calculated as $F1\ Score = 2 * (Precision * Recall) / (Precision + Recall)$. The F1-score gives more weight to the lower value of the two and is the harmonic mean of precision and recall. This means that if either precision or recall is low, then the F1-score will be significantly lower as well. However, if both precision and recall are high, then the F1-score will be close to 1. This can lead to a biased result if one of the metrics is much higher than the other.
- The Matthews correlation coefficient (MCC), instead, is a more reliable statistical rate which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset. MCC takes all the cells of the Confusion Matrix into consideration in its formula. The MCC is used in ML as a measure of quality of binary (2-class) classification. MCC is a correlation coefficient between the exact and predicted binary classification, usually return value of 0 or 1. MCC is calculated as $MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}}$.
- Most metrics are affected by the imbalance of classes in the datasets. Therefore, a single metric cannot be used to differentiate between models [29]. Thus, The ROC curves plotting both the DR and FAR for distinguishing between attack and benign on the x- and y-axes respectively.
- Probability of correct classification (PCC) tell us how good the classifier in detecting either of the class, and it is a probability value, [0,1]. Note that using total accuracy over both positive and negative cases is misleading, even if our training data is balanced in production, batches we measure the performance may not be balanced, so accuracy alone is not a good measure.
- Balanced accuracy (BA) is calculated as the average of sensitivity and specificity, the average of the proportion corrects of each individually. It involves classifying the data into two groups: BA and PCC is calculated as $BA/PCC = 0.5 * (\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$. When all classes are balanced, so there are the same number of samples in each class, $TP + FN \approx TN + FP$ and binary classifier's "regular" Accuracy is approximately equal to balanced accuracy.
- ROC-AUC score handled the case of few negative labels in the same way as it handled the case of few positive labels. An interesting thing to note here is that F1 score is pretty much same for model because positive labels are large in number, and it cares only for the misclassification of positive labels. The probabilistic interpretation of ROC-AUC score is that if randomly choose a positive case and a negative case, the probability that the positive case outranks the negative case according to the classifier is given by the AUC. Here, rank is determined according to order by predicted values.

The rationale for this is that the quantity of true positives and true negatives in the data has a significant impact on accuracy and F1 score. Since class imbalance skews the data towards one class, the other class is often underrepresented, resulting in a decrease in both accuracy and F1 score. In contrast, AUC and MCC are more robust to class imbalance because they are less sensitive to the number of true positives and true negatives. AUC and MCC are better at evaluating the relative performance of different classifiers, rather than absolute performance. Thus, they are better suited for situations where the data is imbalanced.

3. RESULT AND DISCUSSION

The computer utilized had a 64-bit version of macOS called Big Sur, and it had the following specs: 2.6 GHz Dual-Core Intel Core i5, with 8 GB of 1600 MHz DDR3 memory. For implementation and evaluation of the suggested model, the Python (version 3.9) environment was used with the numpy, pandas, and sklearn tools for data processing.

We did preprocessing, exploratory data analysis, and feature selection. We double-checked for duplicates after selecting features. The dataset is divided into three sections: training, testing, and validation. To begin, the sample data is divided into two parts: 80 percent train data and 20 percent test data. The test data is then separated into two portions with a 50 percent ratio: test set and validation set. These datasets are not balanced, as is evident. It is necessary to formulate this unbalanced structure for the improvement of the system's effectiveness. We commonly use the under-sampling strategy to overcome the problem of class imbalance. As a result of the under-sampling strategy, we lost a significant amount of data [2], [37]. After remove "NaN" we get 16137183. 95760 rows removed. While running the model on test dataset, to prevent such noises by makes training less sensitive to the scale of features. Most of the numerous data was replaced with its standard deviation and then rescaled from 0 to 1 by using sklearn. preprocessing. MinMaxScaler. thus, each feature with numerical values is set to the range of 0.0 to 1.0, and each value after normalization is indicated as equation above.

- We removed feature contain "NaN" value such as "Bwd PSH Flags", "Fwd URG Flags", "Bwd URG Flags", "CWE Flag Count", "Fwd Byts/b Avg", "Fwd Pkts/b Avg", "Fwd Pkts/b Avg", "Fwd Blk Rate Avg", "Bwd byts/b Avg", "Bwd Pkts/b Avg", "Bwd Blk Rate Avg"
- We removed 8 fields contained constants of zero for every instance are "Bwd PSH flags", "Bwd URG flags", "Fwd Avg Bytes Bulk", "Fwd Avg Packets Bulk", "Fwd Avg Bulk Rate", "Bwd Avg Bytes Bulk", "Bwd Avg Packets Bulk", and "Bwd Avg Bulk Rate".
- We excluded negative value are "Init Win bytes forward" and "Init Win bytes backward".
- We dropped the Protocol field because it is somewhat redundant, since the Dst Port (Destination_Port) field mostly contains equivalent Protocol values for each Destination_Port value
- The largest value in two columns, "Flow Bytes" and "Flow Pkts," both have values of infinity.
- Remove columns ('Timestamp') as we wanted to learners not to discriminate between attack predictions based on time, especially with more stealthy attacks in mind.
- Columns ('Label') contains the identified attacks names, changed to numerical values.

We utilize RF to select dominant features for anomaly detection, due to its advantages in high performance and robustness. When applied to feature selection, all samples divided into 2 parts. The dataset contains 79 features, one of which is the target feature, label. This paper explores the effects of applying feature selection. We can improve our model by feeding in only selected features that are correlated and non-redundant. This is where feature selection plays an important role.

But almost all of these datasets have imbalance ratios that range from 648 to 112,287. Datasets that are unbalanced have a tendency to favor the dominant class, which might be problematic in particular circumstances. Minority groups are largely disregarded. Furthermore, these minority groups are overwhelmingly positive. Therefore, the imbalance ratio should be decreased to increase system effectiveness while decreasing average accuracy. Figure 2 depicts the distribution of network traffic Imbalance and after under-sampler. A data sampling model was employed to reduce the imbalance-ratio by reducing the data size of majority groups. In datasets selected for the research, the benign class takes from 82.98% and malignant from 17.02%. (see Figure 2(a)). The following figure is a summary of the data set imbalance. It is important to ensure that there is the same percentage of attacks in the training and test datasets, respectively. To achieve this, the dataset is divided in such a way as to ensure that the distribution of benign and malicious traffic in both the training and test datasets is balanced. Under sampler is used during training with result in Figure 2(b).

We used dataset with all its original features and compare with feature selected data set. Malicious and non-malicious data were mixed in with network traffic, which was categorized. This research provides a data preparation strategy that reduces the dimension of feature vectors by using Rainforest feature selection. The results with various criteria such as importance score more than 0.01, 0.02, and 0.03 respectively. From Table 2, we can reduce number of features from 80 columns to 33, 31, 15, 14, 3, 12, and 10, respectively. After selecting best features, the model is fitted, and the predictions are done. Classifiers are then designed for a suitably chosen number of the highest importance score.

When compared to others, the RF feature selection with an imbalance value of 0.03 and a significance score of greater than or equal to 0.02 achieves the greatest results (see Figure 3). Based on the criteria, RF feature selection with under sampling Imbalance by importance score more than and equal to 0.02 in Table 3, we get 16 importance features.

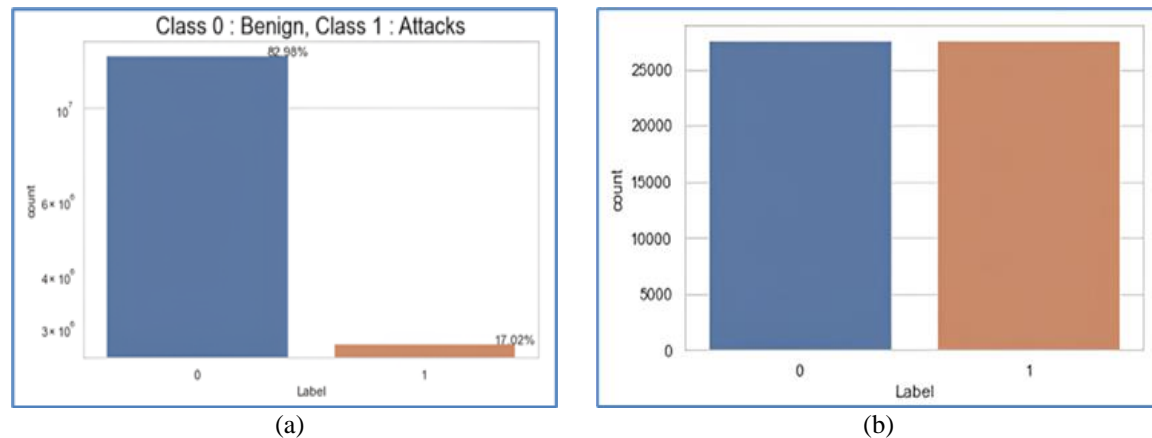


Figure 2. Network traffic distribution (a) imbalanc and (b) after under-sampling

Table 2. RF feature selection with under-sampling

Imbalance Frac	importance	No. of Feature
0.02	≥ 0.01	31
0.02	≥ 0.02	13
0.02	≥ 0.03	10
0.03	≥ 0.01	30
0.03	≥ 0.02	14
0.03	≥ 0.03	10
0.05	≥ 0.01	33
0.05	≥ 0.02	15
0.05	≥ 0.03	12

From the table, we can observe that each. We propose using nine indicators to assess the success of the proposed machine learning method. Recall, TNR, Precision, FPR, FNR, F1 score, ROC, BA, and MCC are chosen as the key evaluation indicators for a comparison of seven state-of-the-art classifier algorithms. The model's performance on the test set in Table 4 is the experimental outcome. Each performance's evaluation metrics are listed in the columns, while the classifiers are shown in the rows. The better the classifier works on this type of data, the greater the value of Recall, TNR, Precision, F1 score, ROC, BA, and MCC. The lower the FPR and FNR levels, however, the better the results. Scores for statistical metrics vary from 0 to 1. For the greatest model, give it a 1 and for the worst model, give it a 0. The MLP method, which has an MCC score of 0.98151, is the most successful, as can be seen in the table. With an MCC score of 0.94703, the XGBoost method comes in second place among the algorithms used to analyze the sampled dataset.

However, making comparisons solely based on MCC scores is not accurate. Due to the fact that categorization IDs need take into account additional indexes as BA, ROC, F1-score, FNR, FPR, Precision, TNR, and TPR. After considering the comprehensive indexes of recall score shown in Figure 4. The performance evaluation parameters for validating the performance of the ML algorithms on the dataset. The plot of the number of accuracies reached by the MLP achieved within the best performance of 0.99496.

Figure 5 gives an overview of the trade-off between classifying all threats (high recall) and reducing the false positives (high precision) when picking a threshold. The ROC curve of each algorithm and focuses on the left-top corner. To compare the performance of different classification algorithms, the LR, KNN, Cart, Bayes, RF, MLP, and XGBoost methods are also used for intrusion detection classification. The AUC of Cart is 0.89960, LR is 0.90996, KNN is 0.98825, Bayes is 0.91386, RF is 0.98942, MLP is 0.99428, and XGB is 0.99209. It is worth nothing that the AUC has a significant impact on the classifier's performance, with 1.0 representing ideal performance. The ROC curves of machine learning (KNN, RF, MLP and XGB) are slightly more approximated to the left and upper axes than the other models. Their AUC values were above 0.98, slightly higher than those of Cart, LR, and Bayes. If also shows that machine learning is more suitable for intrusion detection classification. The MLP model AUC score is 0.99428, the high scores indicate that the classifier provided accurate results (high precision). In addition, a majority of the results were positive (high recall).

Overall, this paper demonstrated that the MLP approach is an effective and efficient way to tackle class imbalance problems. The results of the experiments showed that the MLP approach outperformed or

was comparable to the existing state-of-the-art techniques. The MLP approach achieved better performance in terms of F1 score, ROC, BA, and MCC. These results suggest that the MLP approach is an effective and efficient way to handle class imbalance problems. Furthermore, the results of the experiments suggest that the MLP approach can be used to improve the accuracy of classifiers in a variety of applications.

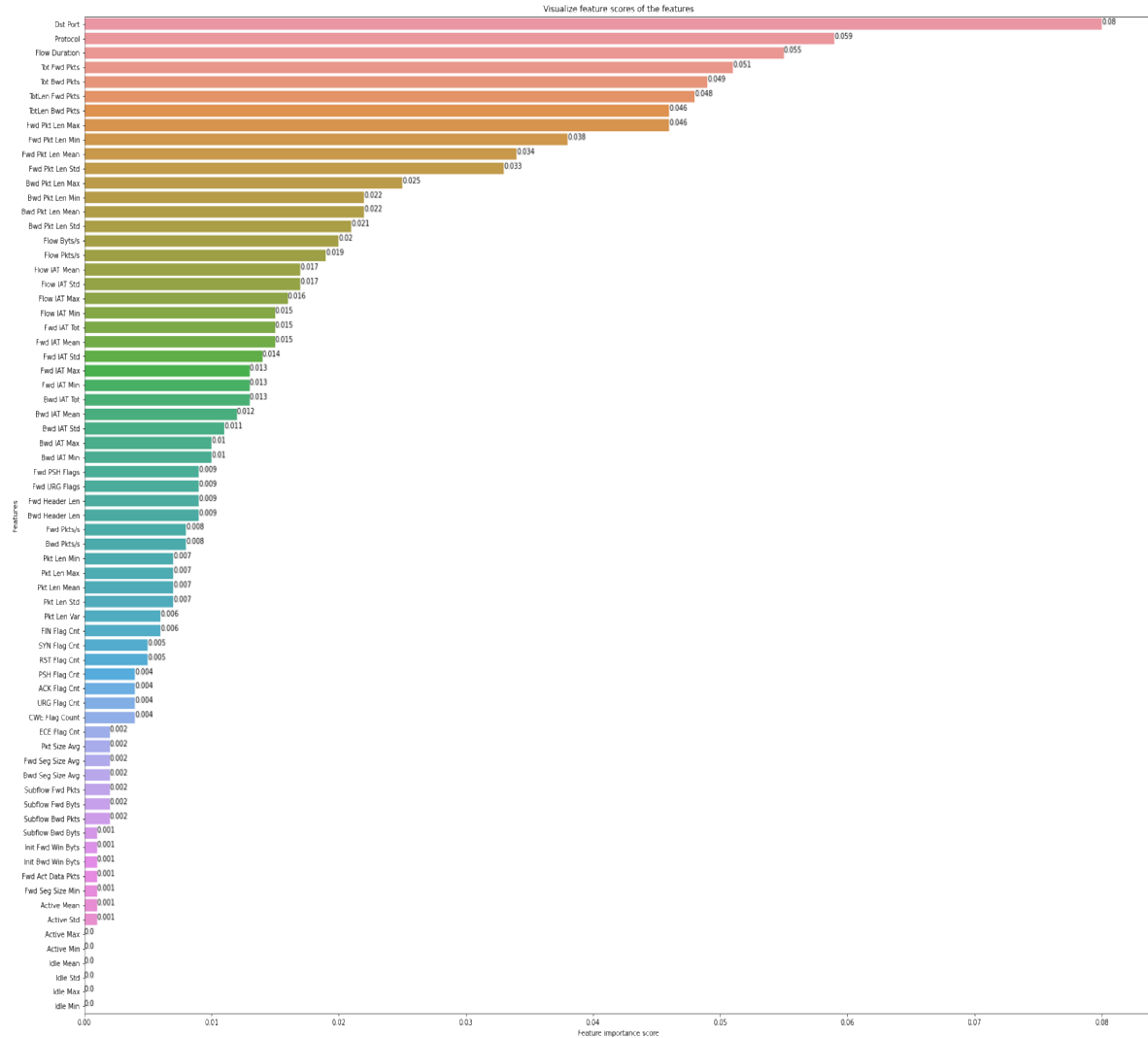


Figure 3. Feature selection by using RF with under sampling imbalance

Table 3. List Feature and importance score

Features	Importance score	Features	Importance score
Dst Port	0.080	Fwd Pkt Len Mean	0.034
Protocol	0.059	Fwd Pkt Len Std	0.033
Flow Duration	0.055	Bwd Pkt Len Max	0.025
Tot Fwd Pkts	0.051	Bwd Pkt Len Min	0.022
Tot Bwd Pkts	0.049	Bwd Pkt Len Mean	0.022
TotLen Fwd Pkts	0.048	Bwd Pkt Len Std	0.021
TotLen Bwd Pkts	0.046	Flow Byts/s	0.020
Fwd Pkt Len Max	0.046		
Fwd Pkt Len Min	0.038		

Table 4. Summary of classification performance with each evaluation matrix

Classifiers	Recall/TPR	TNR	Precision	FPR	FNR	F1 score	ROC	PCC/BA	MCC
LR	0.91126	0.84675	0.91215	0.15325	0.18045	0.91140	0.91037	0.87901	0.66661
KNN	0.98898	0.97572	0.98819	0.02428	0.04107	0.98954	0.98926	0.98235	0.93484
Cart	0.89847	0.90257	0.90039	0.09743	0.10338	0.89931	0.89588	0.90052	0.79923
Bayes	0.91330	0.83596	0.91569	0.16404	0.16205	0.91449	0.91256	0.87463	0.67391
RF	0.99026	0.96001	0.98798	0.03999	0.03835	0.98972	0.99045	0.97513	0.92165
MLP	0.99496	0.99173	0.99321	0.00827	0.01022	0.99462	0.99530	0.99334	0.98151
XGBoost	0.99279	0.97770	0.99181	0.02230	0.03073	0.99227	0.99311	0.98524	0.94703

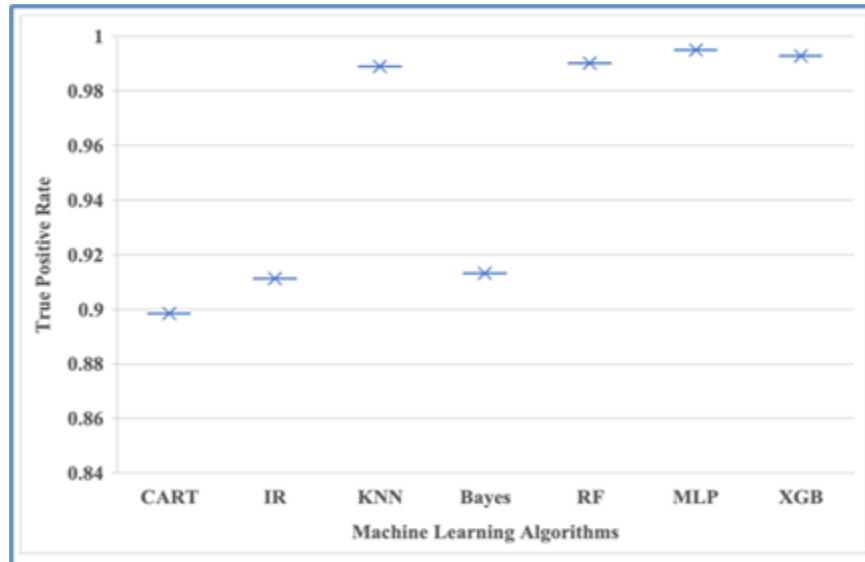


Figure 4. Plots of classification accuracy for standalone models

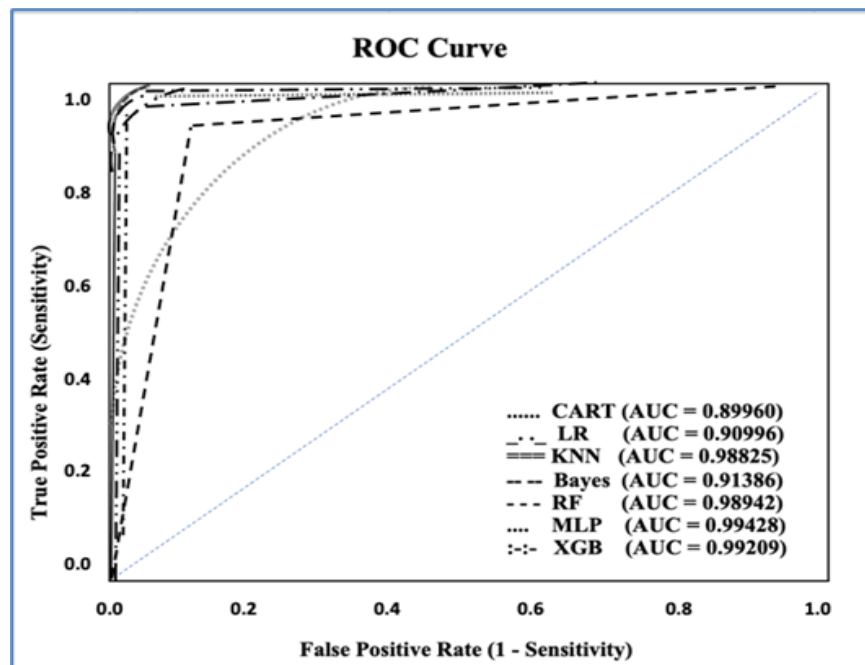


Figure 5. ROC curve of the classifier algorithms

4. CONCLUSION

The proposed used seven classifiers model outperforms the dataset in a comparative experimental analysis with different class imbalance and feature selection with score. ML classification algorithm is used as RF, KNN, CART, MLP, Bayes and XGBoost. The experimental study has recommended for detecting the attacks in intrusion detection system depend on criteria that meet by user such as TPR, TNR, Precision. On the other hand, depend on FPR, FNR. Moreover, F1 score, ROC, BA and MCC indexes were able to achieve excellent in class imbalance. Important components such as feature selection, class imbalance handling, and detection technique must be considered while constructing the model for an effective intrusion detection procedure. Because data imbalance is such a widespread problem in machine learning, this study uses the data under sampling approach to solve the sample imbalance problem. Imbalanced classes are the result of categorization issues in which the classes are not equally represented. As the proposed framework deals with a very specific aspect of the ML pre-processing chain, it can also be used to improve and achieved higher performance methods. Based on the validation results the proposed model was able to identify the most suitable intruders. Also, with methods combine pre-processing and imbalance handle methods to improve ML-based IDS. We conducted a large-scale experiment employing feature selection, which also dealt with class imbalance. The study employed an experimental investigation to develop an anomaly-based intrusion detection system that is suited for network security. The following conclusions are offered because of the analyses: Machine learning ROC curves (KNN, RF, MLP, and XGB) are slightly closer to the left and upper axes than the other models. Their AUCs were more than 0.98. When a single model of classification is used to evaluate each indication (9 indexes), the MLP classifier meets the best evaluation indexes in TNR, Precision, FPR, and ROC on feature selection and class balance. When trained on benign flows, MLP classifier is a classification model that may be used to detect anomalies. Machine learning classification techniques, as we already know, can be utilized to achieve this goal, and will be used to assess and anticipate the infiltration. After applying the preprocessing strategy and a class imbalance handle, the algorithm performed admirably. Although this proposal performed the best, it may be useful in some instances. We believe that trained models are far from ready for application in real-world settings, such as improving created models and testing new algorithms to better fit the unbalanced dataset, and dynamically testing on more forms of incursion. In our upcoming work, we intend to consider the impact of time complexity on the employed time of network intrusion detection in real-time settings with DoS or DDoS assault types.

ACKNOWLEDGEMENTS

We would like to express my sincere gratitude to the anonymous reviewers for their constructive feedback, opinions, findings, recommendations which helped improve the quality of this paper. This work was supported by Suan Dusit University, Thailand.

REFERENCES




- [1] M. A. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system," *Processes*, vol. 9, no. 5, p. 834, May 2021, doi: 10.3390/pr9050834.
- [2] M. Al-Imran and S. H. Ripon, "Network intrusion detection: an analytical assessment using deep learning and state-of-the-art machine learning models," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, p. 200, Dec. 2021, doi: 10.1007/s44196-021-00047-4.
- [3] W. Chimphlee, A. H. Abdullah, M. N. M. Sap, S. Chimphlee, and S. Srinoy, "To detect misuse and anomaly attacks through rule induction analysis and fuzzy methods," *WSEAS Transactions on Computers*, vol. 5, no. 1, pp. 49–54, 2006.
- [4] Y. Hua, "An efficient traffic classification scheme using embedded feature selection and LightGBM," in *2020 Information Communication Technologies Conference, ICTC 2020*, May 2020, pp. 125–130, doi: 10.1109/ICTC49638.2020.9123302.
- [5] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.
- [6] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296–303, Jan. 2017, doi: 10.1016/j.eswa.2016.09.041.
- [7] S. Ali, "Intrusion detection using the WEKA machine learning tool by intrusion detection using the WEKA machine learning tool," Thesis, Dawood University of Engineering and Technology, 2021.
- [8] R. T. Sataloff, M. M. Johns, and K. M. Kost, "The InfSec Handbook: an introduction of information security," in *Apress*, 2014.
- [9] N. Kaja, A. Shaout, and D. Ma, "An intelligent intrusion detection system," *Applied Intelligence*, vol. 49, no. 9, pp. 3235–3247, Sep. 2019, doi: 10.1007/s10489-019-01436-1.
- [10] W. Chimphlee, A. H. Abdullah, M. N. M. Sap, S. Srinoy, and S. Chimphlee, "Anomaly-based intrusion detection using fuzzy rough clustering," in *Proceedings - 2006 International Conference on Hybrid Information Technology, ICHIT 2006*, Nov. 2006, vol. 1, pp. 329–334, doi: 10.1109/ICHIT.2006.253508.
- [11] H. Motoda and H. Liu, "Feature selection, extraction and construction," *Communication of IICM*, vol. 5, pp. 67–72, 2002.
- [12] K. V. Krishna, K. Swathi, and B. B. Rao, "A novel framework for NIDS through fast Knn classifier on CICIDS 2017 dataset," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 5, pp. 3669–3675, Jan. 2020, doi: 10.35940/ijrte.E6580.018520.

- [13] L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-based intelligent intrusion detection system in internet of vehicles," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6, doi: 10.1109/GLOBECOM38437.2019.9013892.
- [14] M. D. Mauro, G. Galatro, G. Fortino, and A. Liotta, "Supervised feature selection techniques in network intrusion detection: A critical review," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104216, May 2021, doi: 10.1016/j.engappai.2021.104216.
- [15] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Analytics*, vol. 1, no. 1, p. 9, Dec. 2016, doi: 10.1186/s41044-016-0014-0.
- [16] L. F. Dias and M. Correia, "Big data analytics for intrusion detection," *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*, 2020, pp. 292–316, doi: 10.4018/978-1-5225-9611-0.ch014.
- [17] M. Umer, H. Xiaoli, and S. Abdul, "Big data security analysis in network intrusion detection system," *International Journal of Computer Applications*, vol. 177, no. 30, pp. 12–18, Jan. 2020, doi: 10.5120/ijca2020919759.
- [18] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data," *Journal of Big Data*, vol. 7, no. 1, p. 104, Dec. 2020, doi: 10.1186/s40537-020-00382-x.
- [19] R. Alshamy and M. Ghurab, "A review of big data in network intrusion detection system: challenges, approaches, datasets, and tools," *International Journal of Computer Sciences and Engineering*, vol. 8, no. 7, pp. 62–75, 2020, doi: 10.26438/ijcse/v8i6.115.
- [20] L. D'Hooge, T. Wauters, B. Volckaert, and F. De Turck, "Classification hardness for supervised learners on 20 years of intrusion detection data," *IEEE Access*, vol. 7, pp. 167455–167469, 2019, doi: 10.1109/ACCESS.2019.2953451.
- [21] F. S. D. L. Filho, F. A. F. Silveira, A. D. M. B. Junior, G. Vargas-Solar, and L. F. Silveira, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," *Security and Communication Networks*, vol. 2019, pp. 1–15, Oct. 2019, doi: 10.1155/2019/1574749.
- [22] K. S. H. Ramos, M. A. S. Monge, and J. M. Vidal, "Benchmark-based reference model for evaluating botnet detection tools driven by traffic-flow analytics," *Sensors (Switzerland)*, vol. 20, no. 16, pp. 1–31, Aug. 2020, doi: 10.3390/s20164501.
- [23] V. Kanimozhi and T. P. Jacob, "Artificial intelligence outflanks all other machine learning classifiers in network intrusion detection system on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," *ICT Express*, vol. 7, no. 3, pp. 366–370, Sep. 2021, doi: 10.1016/j.icte.2020.12.004.
- [24] B. Lypa, O. Iver, V. Kifer, and N. Zagorodna, "Application of machine learning methods for network intrusion detection system," *Engineerxxi*, p. 8, 2018.
- [25] Q. Zhou and D. Pezaros, "Evaluation of machine learning classifiers for zero-day intrusion detection -- an analysis on CIC-AWS-2018 dataset," *arxiv preprints*, May 2019, [Online]. Available: <http://arxiv.org/abs/1905.03685>.
- [26] N. Ashraf, W. Ahmad, and R. Ashraf, "Annals of emerging technologies in computing (AETiC)," *IAER*, 2018. [Online]. Available: www.aetic.theiaer.org.
- [27] Canadian Institute for Cybersecurity, "CSE-CIC-IDS2018 Dataset," 2018. [Online]. Available: <https://www.unb.ca/cic/datasets/ids-2018.html>.
- [28] P. Verma *et al.*, "A novel intrusion detection approach using machine learning ensemble for IoT environments," *Applied Sciences*, vol. 11, no. 21, p. 10268, Nov. 2021, doi: 10.3390/app112110268.
- [29] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, and M. Portmann, "Feature extraction for machine learning-based intrusion detection in IoT networks," *Digital Communications and Networks*, Sep. 2022, doi: 10.1016/j.dcan.2022.08.012.
- [30] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, Feb. 2020, doi: 10.1016/j.jisa.2019.102419.
- [31] R. Zuech, J. Hancock, and T. M. Khoshgoftaar, "Detecting web attacks in severely imbalanced network traffic data," in *Proceedings - 2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science, IRI 2021*, Aug. 2021, pp. 267–273, doi: 10.1109/IRI51335.2021.00043.
- [32] M. A. Siddiqi and W. Pak, "An agile approach to identify single and hybrid normalization for enhancing machine learning-based network intrusion detection," *IEEE Access*, vol. 9, pp. 137494–137513, 2021, doi: 10.1109/ACCESS.2021.3118361.
- [33] M. A. Umar and C. Zhanfang, "Effects of feature selection and normalization on network intrusion detection," *IEEE Access*, pp. 1–25, 2020, doi: 10.36227/techrxiv.12480425.
- [34] S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information*, vol. 9, no. 7, p. 149, Jun. 2018, doi: 10.3390/info9070149.
- [35] M. Al-Imran and S. H. Ripon, "Network Intrusion Detection: An Analytical Assessment Using Deep Learning and State-of-the-Art Machine Learning Models," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, p. 3, 2021, doi: 10.1007/s44196-021-00047-4.
- [36] M. Torabi, N. I. Udzir, M. T. Abdullah, and R. Yaakob, "A review on feature selection and ensemble techniques for intrusion detection system," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 538–553, 2021, doi: 10.14569/IJACSA.2021.0120566.
- [37] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers and Security*, vol. 86, pp. 147–167, Mar. 2019, doi: 10.1016/j.cose.2019.06.005.
- [38] C.-F. Tsai, "Feature selection in bankruptcy prediction," *Knowledge-Based Systems*, vol. 22, no. 2, pp. 120–127, Mar. 2009, doi: 10.1016/j.knosys.2008.08.002.
- [39] L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation-based filter solution," in *Proceedings, Twentieth International Conference on Machine Learning*, 2003, vol. 2, pp. 856–863.
- [40] B. N. Kagara and M. Md Siraj, "A review on network intrusion detection system using machine learning," *International Journal of Innovative Computing*, vol. 10, no. 1, May 2020, doi: 10.11113/ijic.v10n1.252.
- [41] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002, doi: 10.1613/jair.953.
- [42] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020, doi: 10.1109/ACCESS.2020.2973219.
- [43] Q. R. S. Fitni and K. Ramli, "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems," in *Proceedings - 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2020*, Jul. 2020, pp. 118–124, doi: 10.1109/IAICT50021.2020.9172014.
- [44] K.-A. Tait *et al.*, "Intrusion detection using machine learning techniques: An experimental comparison," in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, Jul. 2021, pp. 1–10, doi: 10.1109/ICOTEN52080.2021.9493543.




- [45] S. P. Kumar and A. Raaza, "Study and analysis of intrusion detection system using random forest and linear regression," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 6, no. 1, p. 197, Jun. 2018, doi: 10.21533/pen.v6i1.289.
- [46] G. C. Amaizu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Investigating network intrusion detection datasets using machine learning," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2020, vol. 2020-October, pp. 1325–1328, doi: 10.1109/ICTC49870.2020.9289329.
- [47] B. Silva, R. Silveira, M. S. Neto, P. Cortez, and D. Gomes, "A comparative analysis of undersampling techniques for network intrusion detection systems design," *Journal of Communication and Information Systems*, vol. 36, no. 1, pp. 31–43, 2021, doi: 10.14209/jcis.2021.3.
- [48] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: a review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020, doi: 10.1016/j.procs.2020.04.133.
- [49] R. Chitrakar and H. Chuanhe, "Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and naïve bayes classification," in *2012 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2012*, 2012, doi: 10.1109/WiCOM.2012.6478433.

BIOGRAPHIES OF AUTHORS



Siriporn Chiphlee    received the Ph.D. degree in Computer Science from Universiti Teknologi Malaysia (UTM), Malaysia. She is currently an Assistant Professor at Faculty of Science and Technology, Suan Dusit University, Thailand. She has supervised and co-supervised masters and Ph.D. students. Her research areas are data mining, soft computing and big data analytics. She can be contacted at email: siriporn_chi@dusit.ac.th and siriporn.chi@gmail.com.



Witcha Chiphlee    received the Ph.D. degree in Computer Science from Universiti Teknologi Malaysia (UTM), Malaysia. He used to hold administrative posts with the Faculty of Science and Technology, Suan Dusit University, BKK, from 2012-2022, including the Deputy Dean for Administration, and the Dean of Faculty of Science and Technology. He is currently an Assistant Professor at Faculty of Science and Technology, Suan Dusit University, Thailand. He has supervised and co-supervised masters and Ph.D. students. His research areas are network security, soft computing, data mining and big data analytics. He can be contacted at email: witcha_chi@dusit.ac.th and witcha.chi@gmail.com.