Machine Learning Approaches to Intrusion Detection System Using BO-TPE

Yoon-Teck Bau and Tey Yee Yang Brandon(✉)

Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Malaysia
ytbau@mmu.edu.my, teyyeeyang16@gmail.com

Abstract. Intrusion detection system (IDS) has been intensively studied in the research community. The cyber threats that are evolving rapidly have caused a major challenge for IDS to achieve a reliable detection rate. Despite the application of various machine learning approaches to improve the efficiency of IDSs, present intrusion detection approaches still struggle to reach good performance. In this paper, the Canadian Institute for Cybersecurity on Intrusion Detection Systems 2017 (CICIDS-2017) dataset was selected. To solve the multi-class imbalanced classification problem, multiple imputation by chained equations (MICE) was implemented on the dataset to deal with missing data existing in the dataset. Recursive feature elimination (RFE) method with an estimator of decision tree classifier was also implemented to reduce the number of features through computation of feature importance. The training data was resampled using synthetic minority oversampling technique with combination of the edited nearest neighbor (SMOTE-ENN) to improve the detection of minority classes. Four machine learning approaches were implemented in this research which are K-nearest neighbor, random forest, XGBoost, and LightGBM were trained and tested. The hyperparameter importance of each of the models was also analyzed using Bayesian Optimization with Tree-structured Parzen Estimator (BO-TPE) to enable more experimentation on the tuning of the hyperparameters. All four machine learning approaches achieved at least 98% for all three performance metrics which are accuracy, Matthews correlation coefficient (MCC) and area under the receiver operating characteristic curve (AUROC).

Keywords: Intrusion Detection System · IDS · CICIDS-2017 · Multi-class Imbalanced Classification · Hyperparameters optimization · Bayesian Optimization with Tree-structured Parzen Estimator · Machine Learning Approaches

1 Introduction

Cyber threats and cyber security have arisen as major challenges since the inception of the Internet and they appear to be growing in prominence and significance throughout history. Cyber threats might be described as any illegal conducts that take place on the internet. However, because of the behaviors of cybercriminals that are currently maturing, its more exact definition is still ambiguous. Today, the phrase cyber threats

S.-C. Haw and K. Sonai Muthu (Eds.): CITIC 2022, 10, pp. 104–119, 2022.
https://doi.org/10.2991/978-94-6463-094-7_9

are primarily used to refer to issues of information security. With an increasing number of data intrusions each year, the global cyber threat is evolving at a rapid rate. The most prevalent cyber threats consist of 53% hacking a device, 38% debit/credit card fraud, 34% compromised account passwords, 34% hacking email or social media accounts, 33% fraudulent online transactions, and 32% phishing scams [10]. One of the ways to defend ourselves from cyber threats is by using the Intrusion Detection System (IDS) which helps to provide a more secure and reliable network.

Many studies investigate the use of machine learning approaches to meet the criteria of a modernized IDS. Machine learning falls under the artificial intelligence (AI) domain and strives to extract meaningful information from large amounts of data [18]. Machine learning approaches are used in IDS because they are very good at extracting relevant information from the network and predicting normal and abnormal actions constructed on the patterns learnt, making intrusion detection more efficient. However, there is no obvious answer to the issue of which machine learning approach and IDS dataset is clearly superior over the others. To keep up with emerging threats that are more sophisticated and do more harm, IDS detection accuracy must be significantly enhanced in order to increase the number of true alarms while decreasing the number of false alarms.

The objectives of this research are to transform raw data of dataset into the format that is ready for machine learning, implement four machine learning approaches for intrusion detection systems, evaluate the performance of the approaches using three evaluation metrics and study the hyperparameter importance of each approach for more performance improvements.

2 Literature Review

2.1 Datasets for Intrusion Detection System (IDS)

Datasets are crucial for machine learning approaches to evaluate naturalistic evaluation. Well-known available datasets, along with related analysis approaches, results from previous research, and their characteristics are shown below [14].

The first dataset is DARPA 98. An observation is made where the signature-intrusion detection system (SIDS) is applied on it without anomaly-intrusion detection system (AIDS). SVM on DARPA 98's subset also achieves the detection rate as high as 99.6% with a more outstanding performance on binary class problems. It is given the observation that the ability of SVM in dealing with multidimensional information allows it to separate information into different classes by hyperplane(s). DARPA 98 has the characteristics of realistic traffic, label data, and full packet capture but it does not fulfill the criteria of IoT traces and zero-day attacks.

The second dataset is KDDCUP 99. A 90% detection rate is attained when the multivariate approach which is used for decreasing false alarm rates is applied, while the C4.5 approach has achieved a 95% true positive rate as its generated decision trees can be utilized for classification. Another example given is sequential minimal optimization (SMO) implementation with an SVM-based classifier which attained a 97% detection rate. Its recorded accuracy is less than that with SVM on DARPA 98 due to the KDDCUP 99 dataset being more complex and extensive. According to the given result, the Hidden Naïve Bayes (HNB) model is concluded as the best model with a given observation

that HNB works well in the IDS area which encounters high associated attributes, high network speed, and dimensionality. KDDCUP 99 has the characteristics of realistic traffic, label data, and full packet capture but it does not fulfill the criteria of Internet of Things (IoT) traces and zero-day attacks as well.

The third dataset is the NSL-KDD of 2009. The application of K-Nearest Neighbor (KNN) approach has accomplished a 94% detection rate with observations where it uses all labeled training instances as target function model and it computes a locally optimal hypothesis function during the classification phase using a similarity-based search approach. For this dataset, Naïve Bayes approach offers mediocre accuracy at 89% detection rate due to it focuses more on classifying classes for instances instead of exact probabilities. While the C4.5 approach achieves the highest detection accuracy at 99% as it increases the accuracy by choosing the data characteristics that divide its collection of samples into subsets effectively. The SVM-based SMO approaches achieve similar accuracy with that of DARPA 98 at 97% detection rate. Lastly with the expectation maximization (EM) clustering approaches, it attains 78% accuracy. NSL-KDD has the characteristics of realistic traffic, label data and full packet capture, but it does not fulfill the criteria of IoT traces and zero-day attacks.

The fourth dataset is ADFA-WD, which is a dataset that consists of new attack types. Among the approaches applied to this dataset, which are Hidden Markov Model (HMM), Extreme Learning Machine (ELM) and SVM. SVM accomplished the highest accuracy at 99.64%. ADFA-WD dataset of 2014 has the characteristics of realistic traffic, label data, zero-day attacks, and full packet capture, except for IoT traces. As for the fifth dataset which is ADFA-LD dataset, a surprising 100% detection accuracy is achieved with the ELM approach. This is due to the application of new semantic features, resulting ELM to be able to use them easily by including the semantic phrases. ADFA-LD dataset has the characteristics of realistic traffic, label data, zero-day attacks, and full packet capture, except for IoT traces as well.

The sixth and seventh datasets, which are CAIDA and ISCX 2012, are not given any related results or observations. CAIDA only has the characteristic of realistic traffic but no other characteristics that are label data, IoT traces, zero-day attacks, and full packet capture. While ISCX 2012 has the characteristics of realistic traffic, label data and full packet capture, it does not fulfill the criteria of IoT traces and zero-day attacks.

The Canadian Institute for Cybersecurity on Instruction Detection Systems 2017 (CICIDS-2017) comes in next as eighth dataset. The application of only multilayer perceptron (MLP) approaches attains a 94.5% detection rate, but it can further accomplish a 95.2% detection accuracy when it is applied together with a payload classifier. The observation for this result is given that the Fisher Score approach is implemented to do feature selection. CICIDS-2017 has the characteristics of realistic traffic, label data, zero-day attacks, and full packet capture, except for IoT traces. The last known dataset or ninth dataset is the Bot-IoT dataset. The result is given where the SVM model achieved the highest detection accuracy at 98%. The Bot-IoT (Year 2018) dataset fulfills the criteria for the characteristics of realistic traffic, label data, IoT traces, zero-day attacks, and full packet capture.

The CICIDS-2017 dataset creation procedure is depicted in [19]. According to this research, two wholly distinct networks, which are Victim-Network and Attack-Network

have been implemented into the testbed infrastructure in order to develop a comprehensive testbed. In the Victim-Network, every common and essential equipment is included, which comprises routers, firewalls, switches, and several versions of the three major operating systems which consist of Windows, Linux, and Macintosh. The architecture of Victim-Network is made up of three servers, a firewall, two switches, and ten personal computers (PC) that are interconnected by a domain controller and active directory. Furthermore, one port of the main switch in this network is configured as the mirror port, which comprehensively captures all the network sent and received traffic. While the attack network consists only of a router, a switch, and four PCs that are running the Kali and Windows 8.1 operating systems. For the benign profile agent, a proposed B-Profile system is used to create a realistic benign background traffic after analyzing the abstract behaviors of human interactions based on 25 users via the HTTP, HTTPS, FTP, SSH, and email protocols. Initially, the benign profile agent applied machine learning and statistical analysis approaches when trying to encapsulate network events generated by users. After extracting B-Profiles from users, a Java-developed agent is then utilized to generate naturalistic benign events while concurrently performing B-Profile on the victim network for the five specified protocols.

In the CICIDS-2017 dataset, there are six attack profiles developed based on the most recent list of regular attack families and executed with associated tools and codes. The six attack profiles are Brute Force Attack, Heartbleed Attack, Botnet, DoS Attack, DDoS Attack, Web Attack, and Infiltration Attack. To generate a new intrusion detection dataset, the capturing period began at 9 a.m. on Monday and lasted exactly 5 days, finishing at 5 p.m. on Friday of the same week, where attacks were subsequently carried out throughout the period. In this phase, Monday is recognized as the normal day and it only consists of benign traffic. Later, BruteForce Attack via SFTP and SSH, DoS Attack, Heartbleed Attack, Web Attack, Infiltration Attack, Botnet Attack, and DDoS Attack along with PortScan are conducted in the morning and afternoon from Tuesday to Friday respectively. Afterwards, the proposed dataset is analyzed in a fourfold manner where CICFlowMeter is used to extract the 80 traffic features from the dataset. The constructed dataset's quality is assessed based on the 11 criteria from the recently suggested dataset review framework by Canadian Institute for Cybersecurity (CIC). The evaluation is conducted by looking for common errors and critiques in other synthetically generated datasets.

The disadvantages of CICIDS-2017 are also discussed in [17]. There have also been suggestions about how to deal with the problems. The labels in the dataset were renamed using the Canadian Institute of Cybersecurity's labeling information [7]. Furthermore, it is shown that such class relabeling has decreased a key issue of imbalanced classes.

2.2 Machine Learning Approaches to Intrusion Detection System (IDS)

According to [11] machine learning approaches are to be used to detect and classify intrusions to improve the efficiency of intrusion detection and decrease false alarms. The machine learning approaches used in their research are Naïve Bayes and Support Vector Machine (SVM). The two main machine learning approaches are then combined with another two approaches each. Its main purpose in this research is to achieve different

outcomes and to increase the dataset performance through the feature reduction. It computes the value of attributes by taking into consideration each feature's individual predicting estimation including the redundancy degree between them. To further diversify the results and to enhance the dataset performance, normalization methodology is also used in this research. Consequently, there are six methodologies applied in this research which are SVM, Naïve Bayes, SVM-CfsSubsetEval combination, Naïve Bayes-Cfs SubsetEval combination, SVM-Normalization combination, and Naïve Bayes Normalization combination. The dataset applied in this research is NSL-KDD. The dataset contains symbolic features that the classifiers cannot comprehend. As a result, pre-processing is required. Every non-numeric or symbolic feature is eliminated or replaced during this phase. Protocol, service, and flag are all symbolic properties that can be altered or removed. Finally, the instances are categorized into four groups, which are Normal, DoS, Probe, and R2L. Firstly, the raw dataset is analyzed and the class attribute consists of 24 different attack types which are categorized under the four groups mentioned above. Afterwards, labeling pre-processing comes next to transform nominal attribute to binary attribute. Non-numeric features are discarded to enhance the performance of the intrusion detection system. Following that, the dataset will go through randomization and it will be processed in the WEKA tool using the randomize filter. The first 19,000 instances are collected for comparative analysis, which is done between SVM and Naïve Bayes classification approaches along with the combination of methodologies to assess their accuracy and misclassification rate. This research applied the confusion matrix to calculate the performance for each involved methodology. When comparing SVM and Naïve Bayes approaches, it can be concluded that SVM achieves 97.29% accuracy while Naïve Bayes achieves 67.26% accuracy. For 19,000 instances, Naïve Bayes shows a higher rate of misclassification than the SVM approach. According to the following comparison between SVM and Naïve Bayes after CfsSubsetEval, a feature reduction approach, it can be seen that SVM accomplishes 93.95% accuracy while Naïve Bayes accomplishes 56.54% accuracy. For the same 19,000 instances, Naïve Bayes still shows a higher misclassification rate than SVM. As for the comparison between SVM and Naïve Bayes after normalization, SVM has 93.95% accuracy while Naïve Bayes has 71.00% accuracy. In this case, Naïve Bayes remains a higher misclassification rate than SVM approach. From the given performance analysis, it is concluded that SVM surpasses the Naïve Bayes approach with or without the involvement of CfsSubsetEval and normalization methodologies.

In the research from [9], NSL-KDD dataset was also being used. The redundant data on KDD-99 will influence accuracy. Hence, it is not included within NSL-KDD. To increase the accuracy of data, repetitive and non-correlated data are removed. The data processed also needs to be merged. Cross validation was used to prove the accuracy of a replica constructed on a certain dataset. There are several machine learning approaches used for this study. One of them is the k-means clustering machine learning approach. The other approach used is Naive Bayes approach for classifications. Naive Bayes only need a little bit of training data to confirm the estimated range that is needed in the classification process, hence the advantage of using Naive Bayes. The testing phase will be held on 2 different phases, the first phase is done without the k-means clustering method. During the testing phase, three different variants of clusters will be formed,

which is 3-cluster, 5-cluster and 8-cluster. Weka application is used during the testing phase to obtain the value of the confusion matrix. Optimum accuracy and optimum f-measure value are obtained from the result of the confusion matrix.

The approach used in [1] is using Weka data mining tools, MySQL database, and CICIDS 2017 dataset to implement the model. Three different classifier approaches are also combined. Then, all rows that have the feature "Flow Packets/s" that have instances of 'Infinity' or 'NaN' were removed. The training and test subsets are then extracted. For the training subset, the starting rows of each class are chosen. Moreover, for the other subset, the rows after the suppression of the training subset rows have been selected randomly. Finally, normalization is performed on each data of the feature. To evaluate the proposed model, it has been compared with some well-known classifiers. The evaluation was performed by using a real traffic data set CICIDS-2017 showed that the stratified model outperformed common machine learning models.

In the research from [13], the K-fold approach is used for carrying out training and tests with sampled data to achieve a genuine result by eliminating the effect of randomness. This operation made use of the previously added ID column to the dataset. This was applied to test the accuracy of the detection by removing the influence of the sample data as test samples. The other approach used in this research is the approach for IDS using the NSL-KDD dataset. This approach included the joint of SMOTE, cluster centers, and nearest neighbors. This approach also uses the leave-one-out method to select important features. According to this research, machine learning approaches are to be used to train the system to determine the usual behavior of the network flow. The machine learning approaches used are Adaptive Boosting (AdaBoost), Decision Tree, Random Forest, KNN, Gradient Boosting, and Linear Discriminant Analysis. While handling the machine learning models, missing values were set to zero in order to remove value errors. The infinity values that are present in their dataset are being set to the one number higher than the highest value available in the column. Timestamp columns were split into time and date columns in order to remove text values from datasets. A new column will be created with the same name and adding "Neg" behind it if the original column contains negative values, where negative values are set to 1 and non-negative values are set to 0 in the new column. New column named Label will be added to store identified attack names. The identified attack names will be converted to numbers. Datasets are being randomly shuffled to increase the randomness of the datasets. In the end, the features of the dataset after pre-processing are 83. After that, to generate an artificial sample data, SMOTE method was used. This way, the imbalance ratio was reduced to a bearable rate from 53,887 to 9.98. The final dataset size was enlarged to about 17%. Performance metrics that obtained from the original dataset and extended dataset were sampled data on attack types. The AdaBoost approach was the most successful approach out of all the approaches with an accuracy of 0.9969. These approaches and the other approach were being used for both original dataset and sampled dataset. The machine learning approach calculates all the attack types accurately. Brute force, infiltration, and SQL injection are the three types of attack that have quite low accuracy rates. The exact%age of these attack types is around 3%. In order to expand the rates, new data is generated artificially, and the total quantity of these classes has gone up to 0.162. Benign, Bot, and DoS received small enhancements with the use of sampled

data results. Six machine learning approaches and three data types were being measured by comparing them together. Original datasets held the best result for five of them while the six of them yielded the same result. The sampled data brings the finest accuracy rates for the seven of them. Nevertheless, within the minority classes, substantial growth can be found, which was 72.35% accuracy increase for an average.

In the research from [4], several approaches to building intelligent hybrid systems solutions are implemented to ANN. In network-based IDS (NIDS), data mining approaches have been applied. This approach involves clustering, regression, classification and association rule analysis. The machine learning approaches applied are split into two groups, supervised and unsupervised. In this study, supervised approaches are focused as the dataset involves pre-labeled classes that are defined. Since each approach has specific disadvantages, this research entails hybridization and optimization. Only the advantages of previous approaches that could function in the current area and for the specific challenge are examined in the hybridized method. Thus, SVM, Apriori approach, Decision Forest approach, and Naive Bayes are known approaches that produce accurate results using KDD99 dataset. Based on the conducted a thorough investigation of the KDD99 dataset, there are two major flaws discovered leading to the accuracy of anomaly detection being extremely low, and thus affecting the given overall optimality system. As a result, NSL-KDD was developed as a new dataset which contains the chosen records from the previous dataset without its inefficiencies. There are several advantages. At first, there is less redundancy in the train test sets, thus leading to fair evaluation which provides more accuracy. Secondly, there are no duplicate values, resulting in higher reduction rates. Known attack exists, but the extra and undefined classes do not exist in the training sets. Therefore, each record is composed of 41 different properties that are classified as normal or abnormal. The classification approaches on the NSL-KDD dataset are analyzed and the results of the rates of accurate measurement are recorded.

3 Theoretical Framework

3.1 Machine Learning Approaches

Machine learning approaches were used significantly. Several approaches have been used to extract knowledge from intrusion datasets [21]. Machine learning approaches may be trained in a variety of approaches, each with its own set of benefits and drawbacks. There are three primary forms of machine learning at the moment.

To fully utilize the labeled data in CICIDS-2017 dataset. This research will focus more on supervised learning for the multi-class imbalanced classification problem. Two of the most popular machine learning approaches for this problem will be discussed in detail in the section below.

3.1.1 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) approach believes that related items are close together. The usefulness of this approach is contingent on this belief being accurate. By computing the distance between points on a graph, KNN encapsulates the concept of closeness. The Euclidean distance is a well-known and often used approach of determining distance.

KNN is started by initializing a value which is K to the number of neighbors you choose. The value of K is highly reliant on the training data, altering the position of only a few training data can result in a considerable drop in performance. The approach will be run numerous times with different values of K to find the K that decreases the number of mistakes encountered while retaining the approach's capacity to generate correct detections when it is given data unknown to it. From the data, make a distance calculation between the query example and the current example. The distance and the example's index should be added to an ordered collection. In ascending order, the ordered collection and indices are sorted by distances. The first value of K must be chosen from the sorted collection. The labels for the selected K values should be obtained. The mean of the K labels will be returned if it is regression while the mode of the K labels will be returned if it is classification. The advantages of choosing the right value for K are that the approach is direct and easy to accomplish, with no models built, adjust multiple hyperparameters, or make other assumptions and the approach is extremely adaptable. It has classification, regression, and search capabilities. The disadvantage is as the number of examples and predictors or independent variables increases, the approach becomes significantly slower [12].

3.1.2 Decision Tree

The problems of classification can be solved by implementing the decision tree approach. The purpose of employing a decision tree is to develop a training model to predict the category or value of the target variable by observing and studying the training data [8]. There are two types of decision trees, which are classified according to the type of target variable. A decision tree with categorical target variables is called categorical variable decision tree and a decision tree with continuous target variables is called continuous variable decision tree.

The following are the assumptions when implementing a decision tree:

- The entire training set is regarded as the root at first.
- The feature value is preferably classified. If these values are continuous, discretize them before building the model.
- Records are dispersed recursively based on attribute values.
- Using a statistical approach, the order in which characteristics are placed as the tree's root or internal node is determined.

Each branch from the tree's root to a leaf node with the same class is a result of values, while separate branches terminating in that class constitute a sum. In addition, the main challenge in decision tree implementation is to determine which attributes should be considered for the root node of each level. Handling this is the so-called selection of attributes. At each level, different selection of attribute measures is used to determine the attribute that can be called the root. Numerous approaches in the decision tree are used to evaluate whether to break a node into two or more sub-nodes. The development of sub-nodes improves the homogeneity of the sub-nodes that result. This means that the purity of the node improves as the target variable grows. Based on all available variables,

the decision tree separates the nodes into sub-nodes, then the split that yields the most homogeneous sub-nodes is chosen.

3.2 Multiple Imputation by Chained Equations (MICE)

All datasets will likely include null values, therefore dealing with them is a crucial step. Multiple imputation means repeatedly replacing in the null values, resulting in numerous "filled" datasets. If the measured variables are included in the imputation kernel, the values are imputed based on the measured values for a specific individual and the patterns found in the data for other instances. Because multiple imputation entails making numerous guesses for each missing value, studies of multiply imputed data account for the imputations' uncertainty and produce appropriate standard errors. There are four general steps in the chained equations approach. For each unknown value in the dataset, a basic imputation is done, which includes imputing the mean. Imputations involving a single variable are reset to missing using the mean imputation. The values obtained of the variable are regressed on the other values in the imputation kernel, which might or might not include all of the dataset's values. In other words, in a regression model, that one variable is the dependent variable, whereas all the other variables are independent variables. The regression model's detections are used to fill in the null values for that single value. Both the measured and imputed values will be utilized when that single value is used as an individual entity in regression models for other variables. Aside from the first step, all missing values are repeated with the same steps as above. Each repetition is considered as a cycle. At the end of a cycle, all missing values will be predicted from the regression models that are based on the measured relationship in the dataset. While MICE approach has a number of benefits over other imputation systems in terms of flexibility. It has the disadvantage of not having any theoretical foundation [2].

3.3 Recursive Feature Selection (RFE)

Feature selection selects the features to be retained or deleted from the dataset while dimensionality reduction generates totally new input features by projecting the data [6]. The number of input variables should be reduced in order to lower the calculation cost of modeling and increase the model's performance. The category of feature selection approaches can be classified into two which are unsupervised and supervised, whereas supervised approaches are divided into wrapper, filter and intrinsic. The difference is related to whether the features are selected based on the target variable. Unsupervised feature selection approaches implement the correlation approach to remove the redundant variables while supervised feature selection approaches implement the target variables approach to remove irrelevant variables.

In this research, RFE is selected as the way to reduce the features in the dataset. This method starts by creating a model based on the whole list of predictors and assigning a significance value to each one. The model is then rebuilt, and significance scores are computed again after the least important predictor(s) are eliminated. By deleting features, RFE may be an effective and reasonably efficient strategy for lowering model complexity. Despite the fact that it is a greedy approach, it is one of the most extensively

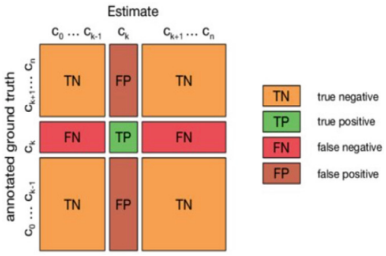

Fig. 1. "N × N" matrix [15].

used feature selection methods. One of the research projects also suggests using RFE together with a decision tree estimator for intrusion detection approaches [16].

3.4 Z-score Normalization

Before analyzing the data on machine learning algorithms, data pre-processing is a must to clean out the dataset and increase the final performance. Data normalization is a data pre-processing technique in which the data is scaled or altered to ensure that each Z-Score of the features in the dataset is set uniformly. The quality of the data used to create a generalized predictive model of the classification issue is critical to the success of machine learning techniques [20]. Normalization approaches will greatly affect the classification results and Z-score normalization is one the approaches that has performed well in various cases. Equation of Z-score normalization is stated in Eq. (1):

$$\text{Z-score} = \frac{value - \mu}{\sigma} \tag{1}$$

So, μ represents the mean of feature while σ represents the standard deviation of the column in the dataset. The Z-score will be set to 0 if it is the same as the mean of all values in the column of the dataset. A positive Z-score is set if it is higher than the mean and vice versa. The magnitude of these negative and positive Z-scores is identified by the standard deviation of the original features. The final Z-Scores would be near to 0 if the original data have a considerable large σ.

3.5 Evaluation Metrics

Confusion matrix is an N × N matrix that is used to evaluate the classification model's performance with N denoting the number of target classes [3]. It not only offers visibility into the mistake produced by the classifier, but also into the sorts of errors made, allowing researchers to get over the constraint of utilizing classification accuracy alone. The actual target value is compared to the value predicted by the machine learning model in this matrix. This shows a comprehensive view of the multi-class classification model and the types of errors made (Fig. 1).

True Positive (TP) is the actual value of the detected intrusion which turns out to be true. True Negative (TN) is the actual value of the detections of non-intrusion which

turns out to be true. False positive (FP) is the actual value of the detections of detected intrusion which turns out to be incorrect. False negative (FN) is the actual value of the detections of non-intrusion which turns out to be incorrect.

Confusion Matrix also allows us to assess the model's performance using measures like precision, recall, F1-score, accuracy, Matthews Correlation Coefficient (MCC) and area under receiver operating characteristic curve (AUROC).

3.6 Bayesian Optimization with Tree-Structured Parzen Estimators (BO-TPE)

Most machine learning approaches have a collection of hyperparameters whose values must be carefully selected and which frequently have a significant influence on performance [5]. There are a lot of approaches involving hyperparameters optimization, Bayesian Optimization (BO) stood out the most among all of the approaches. Bayesian Optimization varies from previous systems in that it improves search time by analyzing prior results, making it similar to manual search. The Bayes theorem is a method for estimating an event's likelihood function. Using the equation:

$$P(A|B) = P(B|A) * P(A), \tag{2}$$

where A is the hyperparameter, and B is the score. We are able to obtain the likelihood function that was estimated which is also known as the posterior probability. Every observation about the objective function is presented by the posterior. It takes consideration of the objective function and may be used to estimate the cost of various candidate samples that we may like to analyze. This stage of Bayesian optimization can alternatively be seen as estimating the objective function with a surrogate function. The surrogate function estimates the objective function, which may be used to guide future sampling. Sampling is accomplished by the precise application of the posterior in an acquisition function. The acquisition will maximize the conditional probability of areas in the search to create the next sample by utilizing the assumption about the objective function to sample the region of the search space that would most likely pay off. After collecting more samples and evaluating them using the objective function, they are added to the data and the posterior is updated. This method is continued until the goal function's extrema is found, a good enough outcome is found or resources are depleted.

Tree-structure Parzen Estimators (TPE) is one of the choices of surrogate models that can be implemented in BO. This approach uses a tree-structured approach to handle categorical hyperparameters. While the concept of Parzen estimators is similar to Bayesian optimization, it is theoretically opposed. To initiate the approach, the first step in TPE is to begin sampling the response surface using random search. The observations are then divided into two groups: the best performing one and the others, with score B* establishing the splitting value for the two groups. The equation is shown below.

$$P(A|B) = \begin{cases} l(A) \text{ if } B < B* \\ g(B) \text{ if } B \geq B* \end{cases} \tag{3}$$

The model's projected improvement criteria allow it to balance exploration and exploitation. Because l(A) is a distribution rather than a single value, the hyperparameters chosen are likely to be close but not exactly at the maximum of the projected

improvement. Furthermore, because the surrogate is only an approximation of the objective function, the chosen hyperparameters may not result in an improvement when tested, and the surrogate model will need to be modified. The present surrogate model and the history of objective function evaluations are used to update this model.

4 Methodology

The implementation was done in a hardware experimental environment on a desktop with a processor of 11th Gen Intel(R) Core(TM) i5–11400 of 2.60 GHz, 16 GB installed RAM and Windows 10 Pro operating system. The machine learning approaches were developed using the CICIDS-2017 dataset in a web-based Python IDE which is JupyterLab.

In the beginning, the Python modules that will be used during the implementation will be imported such as sklearn, pandas, matplotlib, etc. Since the CICIDS-2017 dataset scatters across eight files, which will make the development very complex. All the files were combined as a single pandas dataframe. It was shown that the dataset contains 2,830,743 number of instances with 79 features. After the data have been imported into the dataframe, data analysis was done to learn about the information of the dataset. The classes were relabeled where all the same classes category. Then, the labels were encoded into numerical values as most machine learning approaches only accept numeric labels as input.

To reduce some of the instances in class 0 since it has more instances compared to other classes. The duplicated values are removed which there exist a number of 176613 instances in class 0. The inf values of the dataset will be converted into missing values. To apply MICE imputation to deal with the missing values in the dataset, the miceforest module was imported. In the kernel specified, 5 datasets that were imputed were created as multiple imputation is an approach for examining the uncertainty and other consequences caused by missing values by producing multiple imputed datasets.

Later, some of the features that have zero variance in the data were dropped from the dataframe. These features consist of 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. The feature "Fwd Header Length.1" was also dropped as it has the exact same value as "Fwd Header Length". For the feature selection, RFE with cross validation was used to plot a graph of accuracy against the number of features used to choose the optimum number of features to be selected for the machine learning approaches. After analyzing the results from the graph, the number of features selected to be used will be reduced to fifteen not including the label.

In order to apply the Z-score normalization on the dataset, the StandardScaler module was used as it can set the data into the same scale while saving the values needed to scale each feature, in case there are new instances to be added. The dataset is then split into training and testing data using stratified train test split of a 3:1 ratio.

To solve the imbalanced dataset problem, the SMOTEENN module was used to resample the dataset. Only the training set will be resampled to reserve the test set in its most original form to ensure an accurate performance evaluation.

After the data pre-processing phase is completed, the baseline model will be created with default hyperparameters. The models that were used in this implementation are

KNN, Random Forest, XGBoost and LightGBM. After developing the models, hyperparameters optimization will be applied using the optuna module. The search space for the hyperparameters were defined before starting the optimization. The optuna module uses Bayesian Optimization with Tree-structured Parzen Estimator to find the optimal value within the search space provided. The direction was set to maximize as the study was done with a cross validation of three stratified folds that return an average metric score using MCC. Using the results given by the optimization, the approaches were developed using a new set of hyperparameters.

5 Evaluation of Findings

All of the approaches implemented will be evaluated using the several evaluation metrics. The training time and testing time of each approach is also recorded in Table 1. For the baseline models with default parameters, Table 1 is the results that are obtained. In all of the approaches, XGBoost obtained the highest performance in detecting intrusions while achieving the shortest detection time. LightGBM however was the fastest in training the model among the ensemble models. Table 2 shows the final results of the approaches after using hyperparameters optimization. All of the approaches have achieved higher performance except for the AUROC score while significantly lower the time for detection. There is also a decrease in the training time of random forest and LightGBM after hyperparameters optimization.

The importance of the hyperparameters defined in the search space for each approach is also shown in the Figs. 2, 3, 4 and 5.

Table 1. Results of baseline models.

Model	Accuracy	MCC	AUROC	Train time(s)	Test time(s)
KNN	0.994018	0.983043	0.992509	77.0311	267.4930
RF	0.998654	0.996190	0.999722	1402.7659	5.9481
XGBoost	0.998377	0.995411	0.999764	2963.7682	0.9189
LightGBM	0.998065	0.994534	0.999726	109.1747	2.7048

Table 2. Results of optimized models.

Model	Accuracy	MCC	AUROC	Train time(s)	Test time(s)
KNN	0.994685	0.984907	0.992375	77.6135	206.6033
RF	0.998719	0.996373	0.999689	730.3860	1.7652
XGBoost	0.998626	0.996110	0.999723	8092.8688	0.6782
LightGBM	0.998283	0.995148	0.999824	56.7344	1.3225

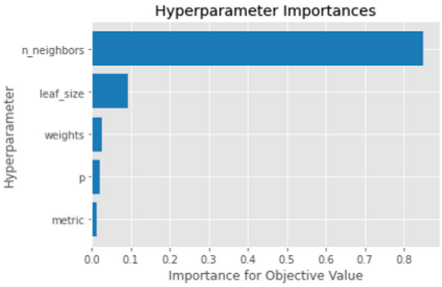

Fig. 2. Hyperparameter importance of KNN.

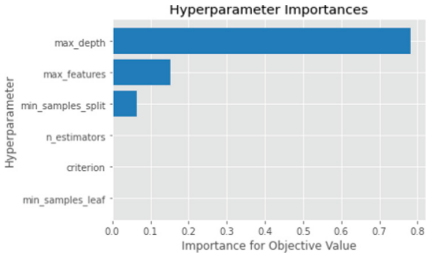

Fig. 3. Hyperparameter importance of RF.

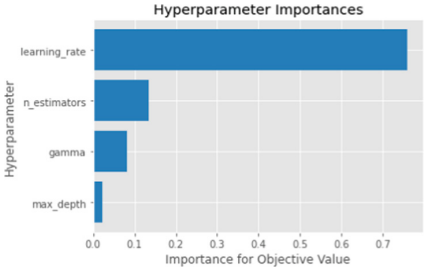

Fig. 4. Hyperparameter importance of XGBoost.

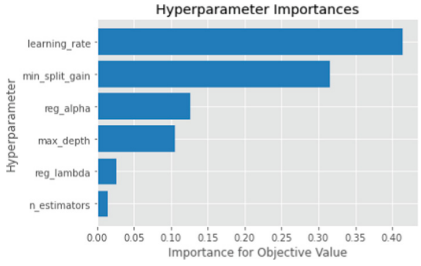

Fig. 5. Hyperparameter importance of LightGBM.

6 Conclusion

Machine learning approaches such as KNN, Random Forest, XGBoost and LightGBM were implemented to address the multi-class imbalanced classification problem in intrusion detection systems. The approaches show excellent results in the evaluation metrics tested and did show some improvements after applying the hyperparameters optimization approach using BO-TPE approach. The importance of the hyperparameters was displayed, which were acquired during the optimization process. By revealing the importance of the hyperparameters, it is possible to expand the search space for the more important hyperparameters to explore the most optimum tuning while discarding the less important ones during the optimization process as this enables higher performance.

There are still other improvements that can still be done to improve the performance of the approaches implemented. One of the ways is to apply stacking to all the ensemble learning models that combine the classification detections of all the models chosen. Other than that, deep learning approaches can also be implemented in substitute of the proposed machine learning approaches. For the hyperparameters optimization of the models, the numbers of trials can also be increased to ensure that the importance analyzed for each hyperparameter.

References

1. Ahmim, A., Maglaras, L., Ferrag, M. A., Derdour, M., & Janicke, H. (2019). A novel hierarchical intrusion detection system based on decision tree and rules-based models. In 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 228–233). IEEE.
2. Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work?. International journal of methods in psychiatric research, 20(1), 40–49. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/.
3. Bhandari, A. (2020). Confusion Matrix for Machine Learning. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/.
4. Bhumgara, A., & Pitale, A. (2019). Detection of Network Intrusions using Hybrid Intelligent Systems. In 2019 1st International Conference on Advances in Information Technology (ICAIT) (pp. 500–506). IEEE.
5. Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., ... & Lindauer, M. (2021). Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. arXiv preprint arXiv:2107.05847.
6. Brownlee, J. (2020). How to Choose a Feature Selection Method for Machine Learning. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/.
7. Canadian Institute for Cybersecurity. (April 13, 2018). Intrusion Detection Evaluation Dataset (CIC-IDS2017). Retrieved from https://www.unb.ca/cic/datasets/ids-2017.html.
8. Chauhan, N. S. (2020). Decision Tree Algorithm, Explained. KDnuggets. Retrieved from https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html.
9. Effendy, D. A., Kusrini, K., & Sudarmawan, S. (2017). Classification of intrusion detection system (IDS) based on computer network. In 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE) (pp. 90–94). IEEE.

10. Farahbod, K., Shayo, C., & Varzandeh, J. (2020). Cybersecurity indices and cybercrime annual loss and economic impacts. Journal of Business and Behavioral Sciences, 63.
11. Halimaa, A., & Sundarakantham, K. (2019). Machine learning based intrusion detection system. In 2019 3rd International conference on trends in electronics and informatics (ICOEI) (pp. 916–920). IEEE.
12. Harrison, O. (2019). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Medium. Retrieved from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.
13. Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. IEEE Access, 8, 32150–32162.
14. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1), 1–22.
15. Krüger, F. (2016). Activity, context, and plan recognition with computational causal behavior models (Doctoral dissertation, University).
16. Lian, W., Nie, G., Jia, B., Shi, D., Fan, Q., & Liang, Y. (2020). An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning. Mathematical Problems in Engineering, 2020.
17. Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS-2017 dataset for designing Intrusion Detection Systems. International Journal of Engineering & Technology, 7(3.24), 479–482.
18. Prasad, R., & Rohokale, V. (2020). Artificial intelligence and machine learning in cyber security. In Cyber Security: The Lifeline of Information and Communication Technology (pp. 231–247). Springer, Cham.
19. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSP, 1, 108–116.
20. Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. Applied Soft Computing, 97, 105524.
21. Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?. IEEE Signal Processing Magazine, 35(5), 41–49.