**Indonesian Journal of Electrical Engineering and Informatics (IJEEI)**
Vol. 9, No. 2, June 2021, pp. 498~511
ISSN: 2089-3272, DOI: 10.52549/ijeei.v9i2.3028 

# Important Features of CICIDS-2017 Dataset For Anomaly Detection in High Dimension and Imbalanced Class Dataset

**Kurniabudi Kurniabudi[1], Deris Stiawan[2], Darmawijoyo[3], Mohd Yazid bin Idris[4], Bedine Kerim[5], Rahmat Budiarto[6]**

[1]Faculty of Engineering, Universitas Sriwijaya, Palembang & Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia
[2] Faculty of Computer Science, Universitas Sriwijaya, Palembang, Indonesia
[3]Faculty of Mathematics and Natural Sciences, Universitas Sriwijaya, Palembang, Indonesia
[4]Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
[5,6]College of Computer Science and IT, Albaha University, Al Aqiq, Saudi Arabia

**Article Info**

***Article history:***

Received Feb 20, 2021
Revised May 7, 2021
Accepted May 25, 2021

***Keywords:***

Feature Selection
Information Gain
Random Forest
High Class Imbalance
CICIDS-2017

**ABSTRACT**

The growth in internet traffic volume presents a new issue in anomaly detection, one of which is the high data dimension. The feature selection technique has been proven to be able to solve the problem of high data dimension by producing relevant features. On the other hand, high-class imbalance is a problem in feature selection. In this study, two feature selection approaches are proposed that are able to produce the most ideal features in the high-class imbalanced dataset. CICIDS-2017 is a reliable dataset that has a problem in high-class imbalance, therefore it is used in this study. Furthermore, this study performs experiments in Information Gain feature selection technique on the imbalance class datasaet. For validation, the Random Forest classification algorithm is used, because of its ability to handle multi-class data. The experimental results show that the proposed approaches have a very surprising performance, and surpass the state-of-the-art methods.



***Corresponding Author:***

Deris Stiawan,
Faculty of Computer Science,
Universitas Sriwijaya,
Palembang, Indonesia.
Email: deris@unsri.ac.id

## 1. INTRODUCTION

It has been stated by many researchers that feature selection is able to reduce dimensional data by removing redundant features and selecting the most optimal features [1],[2],[3]. Pervez and Farid [4] have applied a feature selection algorithm to reduce input features of the classification engine. Tama and Rhee [5] have used particle swarm optimization (PSO)-based feature selection to select attributes. Aghdam and Kabiri [6] have implemented Ant-Colony-based feature selection technique to produce optimal features. Meanwhile, Kushwaha et al.[7] have applied a filter-based feature selection technique to remove unnecessary features.

In intrusion detection system (IDS) research, an effective feature selection technique can be used to produce relevant features that help in improving system's capability in term of attack detection with minimum false alarms rate and low computation time [6]. Various techniques have been proposed to produce an ideal feature selection technique that can improve IDS performance. Chen et al. [8], have proposed a combination of K-nearest neighbor (KNN) and tree seed algorithm (TSA) and the proposed method is able to increase the accuracy and efficiency of network intrusion detection. Gottwalt et al. [9] have introduced CorrCorr as a feature selection technique, and resulted in good detection capabilities with low false alarm rates. Meanwhile Zhou et al. [10] have developed effective IDS with feature selection techniques and ensemble classifier and experimental results show superior performance.

Having done surveying previous research works, it was found that various feature selection techniques have been developed to produce the most optimal features that can be used to detect accurately various types of attacks including new types of attacks [1], [11]. Each proposed technique results in a different optimal number of features with different performance [12]. Authors in [13] have proposed information gain feature selection technique to eliminate irrelevant features. The implementation of this technique on CICIDS-2017 dataset with 7 traffic class labels (1 benign, and 6 attack), produces 22 important features. Then, the selected features are combined with Random Forest classification algorithm detect attacks. The best accuracy of 99.86% is achieved. The proposed work in [13] did not consider wider variety of attacks at high-class of imbalance of data distribution in huge dataset. Thus, this work attempts to address the issue of high-class imbalanced dataset and also use the CICIDS-2017 dataset, but considering 15 traffic class labels (consists of 1 normal and 14 attack). As mentioned in [14], feature selection techniques must be able to work on extreme datasets and recognize minority classes when working with imbalanced data. This work also makes comparisons with previous studies, to evaluate the reliability of the proposed approaches. Lastly, this study provides recommendation for the most optimal features to be used for detecting attacks on unbalanced datasets, especially for the CICIDS-2017 dataset.

## 2. RELATED WORK

In IDS research, especially on feature selection techniques, high class imbalance is an important issue. In this case, the data communication traffic on real networks produces high dimensional data and tends to be imbalanced then furhter has an impact on the performance of the classification engine.

A study by Rodda and Erothi [15] described the problem of imbalance class in NSL-KDD. The researchers has experimented 4 (four) classification techniques, i.e.: Naïve Bayes, Bayes Network, J48 and Random Forest and the results show that the technique used was not able to classify properly a class whose distribution is small. Meanwhile, Reza et al. [16] solved the problem of imbalanced class by proposing a combination of synthetic minority oversampling technique (SMOTE) and cluster center and nearest neighbor (CANN). The experimental results show that the proposed method can improve the detection of minor attacks such as remote to local (R2L) and user to root (U2R). Furthermore, in Yan et al.'s research work [17], Region Adaptive Synthetic Minority Oversampling Technique (RA-SMOTE) is proposed to recognize an attack and normal traffic on imbalanced data. The experimental results show an increase in detection performance for attacks with low frequency.

Research work carried out by Seo and Kim [18], used SMOTE technique, through optimization of SMOTE ratio to overcome the imbalanced class. The experimental results using Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) show an increase in the detection performance of the minor class. The SMOTE method was also used in research work by Yulianto et al. [19], which was applied to the CICIDS-2017 dataset. Principal Component Analysis (PCA) and Ensemble Feature Selection (EFS) are used for feature selection. The experimental results using the Adaboost classification technique, show good classification performance with the Area Under the Receiver Operating Characteristic (AUROC) value reaching 92%.

Meanwhile [20] propose Auto Encoder and PCA to reduce dataset dimensions and Random Forest to detect attacks and the imbalance dataset problem was solved by Uniform Distribution Based Balancing (UDDB). The experimental results show that the proposed method is able to reduce feature spaces of CICIDS-2017 dataset while maintaining a detection accuracy of 99.6%.

Research work by Abdulhammed et al. [21] proposed a method called Imbalance Generative Adversarial Network (IGAN) and combined it with IDS. The aim is to produce a representative sample by generating a sample from the minor class. The experimental results show that the proposed method is superior to the state-of-the-art methods.

From several previous studies, the researchers used a data-level approach to solve the imbalance data problem. The technique that is widely used is SMOTE. As stated by Bedi et al. [22], previous researchers addressed the class imbalance problem by using a data-level approach. Although this data-level approach may improve Network-based Intrusion Detection System (NIDS) performance, it does not solve the underlying problem with NIDS. Inspired by this work, researchers used the same approach to solve this imbalance data problem by optimizing feature selection, resulting in features that are capable of detecting various types of attacks, even on a small scale.

## 3. METHODOLOGY

This section describes systematically and in detail the datasets used in this study, the experimental framework, classification algorithms, and performance measurement matrices.

### 3.1. The Dataset

The CICIDS-2017 dataset was developed to meet the scarcity of realtime network traffic datasets [23]. The CICIDS-2017 dataset has the most recent and relevant data for testing security systems [24]. Nevertheless, the main reason of the use of this dataset is because it contains high-class imbalance data as stated in the study by Panigrahi and Borah [25], and Injadat et al. [26]. Other IDS Datasets such as NSL-KDD or UNSW-NB15 have a limited number of features, i.e.: NSL-KDD has 42 features; UNSW-NB15 has 49 features [27], while CICIDS-2017 dataset has a total of more than 80 features [24]. Thus, we consider CICIDS-2017 dataset is superior in terms of data dimensionality.

In the experiment only 30% of the MachineLearningCSV version of the CICIDS-2017 dataset were used. The data profile used is presented in Table 1. The MachineLearningCSV version of the CICIDS-2017 dataset contains 15 traffic classes consisting of normal and attack traffic. The data in the table also shows an unbalanced data distribution among the 15 classes. The imbalance of this data can also be seen from the percentage of data distribution against the main class and the distribution for each class. In the dataset, there are also classes with a small number of traffic attacks such as Web Attack-SQl Injection, Infiltration and Heartbleed. Regarding the imbalance class in the CICIDS-2017 dataset, it is also stated in the study by Abdulhammed et al. [20], Pelletier and Abualkibash [24] and Panigrahi and Borah [25].

Table 1. 30% Profile of the CICIDS-2017 Dataset

| No. | Class Lable | Number of Instance | % Number of instances against majority class | % Number of instaces to total instances |
|---|---|---|---|---|
| 1 | Benign | 681,995 | 100 | 80.3081 |
| 2 | DDoS | 38,427 | 5.6345 | 4.5250 |
| 3 | PortScan | 47,487 | 6.9630 | 5.5918 |
| 4 | Bot | 574 | 0.0842 | 0.0676 |
| 5 | Web Attack–Brute Force | 455 | 0.0667 | 0.0536 |
| 6 | Web Attack–XSS | 202 | 0.0296 | 0.0238 |
| 7 | Web Attack–Sql Injection | 8 | 0.0012 | 0.0009 |
| 8 | Infiltration | 8 | 0.0012 | 0.0009 |
| 9 | DoS slowloris | 1,739 | 0.2550 | 0.2048 |
| 10 | DoS Slowhttptest | 1,605 | 0.2353 | 0.1890 |
| 11 | DoS Hulk | 69,259 | 10.1554 | 8.1556 |
| 12 | DoS GoldenEye | 3,206 | 0.4701 | 0.3775 |
| 13 | Heartbleed | 5 | 0.0007 | 0.0006 |
| 14 | FTP- Patator | 2,422 | 0.3551 | 0.2852 |
| 15 | SSH-Patator | 1,831 | 0.2685 | 0.2156 |
| | **Total** | **849,223** | | |

For experimental purposes, 30% of the dataset is separated with a portion of 70% for training data and 30% for testing data. The training data profile in the experiment is presented in Table 2, while the testing data profile is presented in Table 3. Referring to this data profile, the two data portions both have 15 traffic classes (normal and attack) and both contain high class imbalance. This condition means that the characteristics and completeness of the data, both training and testing data, have met the needs of the experiment. The amounts of data used in the experiment are the training data that consists of 594,456 records of analyzed data and testing data that consists of 254,767 records.

Table 2. Profile of Training Data (70%)

| No. | Class Lable | Number of Instance | % Number of instances against majority class | % Number of instaces to total instances |
|---|---|---|---|---|
| 1 | Benign | 477,172 | 69.9671 | 56.1892 |
| 2 | DDoS | 26,974 | 3.9552 | 3.1763 |
| 3 | PortScan | 33,202 | 4.8684 | 3.9097 |
| 4 | Bot | 392 | 0.0575 | 0.0462 |
| 5 | Web Attack–Brute Force | 311 | 0.0456 | 0.0366 |
| 6 | Web Attack–XSS | 145 | 0.0213 | 0.0171 |
| 7 | Web Attack–Sql Injection | 4 | 0.0006 | 0.0005 |
| 8 | Infiltration | 7 | 0.0010 | 0.0008 |
| 9 | DoS slowloris | 1,208 | 0.1771 | 0.1422 |
| 10 | DoS Slowhttptest | 1,133 | 0.1661 | 0.1334 |
| 11 | DoS Hulk | 48,653 | 7.1339 | 5.7291 |
| 12 | DoS GoldenEye | 2,278 | 0.3340 | 0.2682 |
| 13 | Heartbleed | 4 | 0.0006 | 0.0005 |
| 14 | FTP- Patator | 1,707 | 0.2503 | 0.2010 |
| 15 | SSH-Patator | 1,266 | 0.1856 | 0.1491 |
| | **Total** | **594,456** | | |

Table 3. Profile of Testing Data (30%)

| No. | Class Label | Number of Instance | % Number of instances against majority class | % Number of instaces to total instances |
|---|---|---|---|---|
| 1 | Benign | 204,823 | 30.0329 | 24.1189 |
| 2 | DDoS | 11,453 | 1.6793 | 1.3486 |
| 3 | PortScan | 14,285 | 2.0946 | 1.6821 |
| 4 | Bot | 182 | 0.0267 | 0.0214 |
| 5 | Web Attack–Brute Force | 144 | 0.0211 | 0.0170 |
| 6 | Web Attack–XSS | 57 | 0.0084 | 0.0067 |
| 7 | Web Attack–Sql Injection | 4 | 0.0006 | 0.0005 |
| 8 | Infiltration | 1 | 0.0001 | 0.0001 |
| 9 | DoS slowloris | 531 | 0.0779 | 0.0625 |
| 10 | DoS Slowhttptest | 472 | 0.0692 | 0.0556 |
| 11 | DoS Hulk | 20,606 | 3.0214 | 2.4265 |
| 12 | DoS GoldenEye | 928 | 0.1361 | 0.1093 |
| 13 | Heartbleed | 1 | 0.0001 | 0.0001 |
| 14 | FTP- Patator | 715 | 0.1048 | 0.0842 |
| 15 | SSH-Patator | 565 | 0.0828 | 0.0665 |
| | **Total** | **254,767** | | |

### 3.2. Experimental Framework

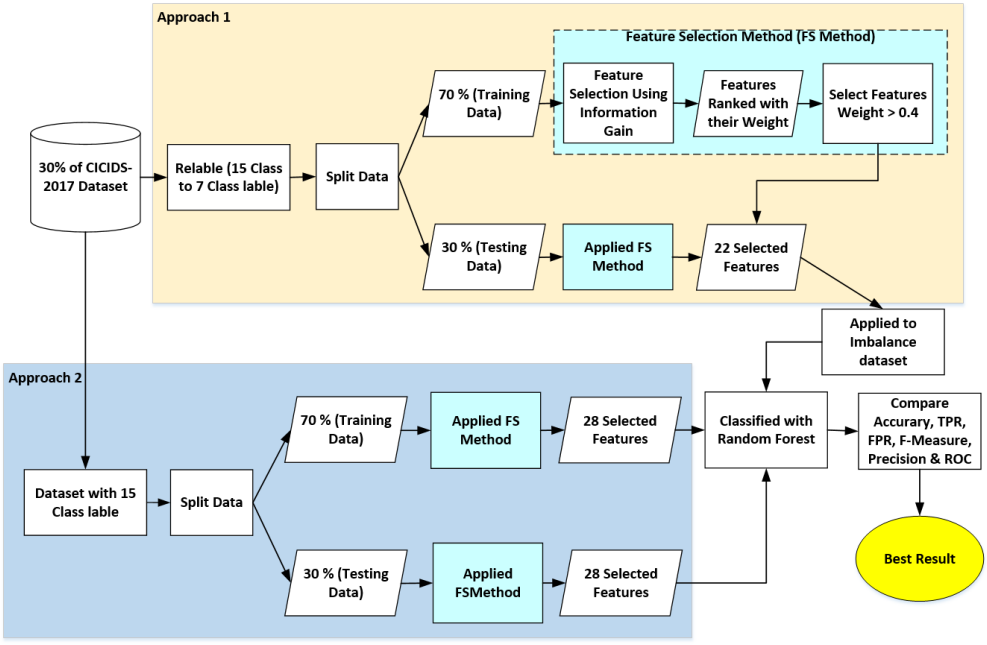



Figure 1. Experimental Framework

In this study, the selected features generated from previous study [13] (presented in Table 4) will be validated using large-dimensional data that contain high class imbalance data, i.e.: CICIDS-2017 dataset with 15 traffic class labels. In addition, this study examines the Information Gain feature selection technique for high class imbalance dataset. The research experiment framework is illustrated in detail in Figure 1. Two feature selection approaches are proposed, named as Approach-1 and Approach-2:

- *Approach*-1, the researchers use the approach introduced by Panigrahi and Borah [25], by grouping similar attack traffics and given a new label. For the experiment, the dataset that has been re-labeled (new label) with 7 class labels is divided into 70% as training data and 30% as testing data. Furthermore, feature selection is carried out using Information Gain. Based on previous research, this approach produces 22 features that result in ideal detection performance. Furthermore, the features of this selection are used to identify attacks on imbalanced datasets.
- *Approach-2*, the researchers apply the Information Gain feature selection technique to input dataset with 15 traffic class labels. By applying the same feature weights as approach-1, which is a minimum of 0.4

on the Information Gain output, 28 features are obtained. Furthermore, the 28 selected features are used to classify attack traffic using the Random Forest algorithm.

- *In the final stage*, the classification results of feature selection with Approach-1 and Approach-2 are compared, to see the most optimal results.

**3.3. Experiment Configuration**

In this study, the authors use a Core i7 Notebook with 8GB RAM and 500 GB HDD and running Windows 10 operating system. Meanwhile, for analysis purposes, authors use Waikato Environment for Knowledge Analysis (WEKA) versuib 3.8. It is a machine learning software [28] and is widely used in data mining and machine learning researches including IDS research [28-30]. In normal and attack traffic classification experiments, several test options, which available at WEKA tool are used, such as:

- *Use training set:* classification performance test using all input data
- *Cross Validation:* classification performance test using k-fold cross-validation. In the experiment, 10-fold and 5-fold cross-validation were used.
- *Presentage Split:* classification performance test using split data. The experiment use 10 to 90 splits.

**3.3. Random Forest (RF)**

Random forest is one method in the decision tree. Random forest combination of each tree collected in a model. There are three important aspects in the random forest process, namely: (1) conducting bootstrap sampling with the aim of building a prediction tree; (2) every tree predicting decisions using random predictors; (3) then perform random forest prediction with combines the results from each decision tree by means of a majority vote for classification [30]. That is why Random Forest is known as the ensemble classifier method. If a classifier in an ensemble is a decision tree classifier, the classifier set is "forest". Each individual decision tree is created through a random selection of attributes at each node for separation [31]. The Random Forest algorithm was proposed by Breich in 2001[32]. Some anomaly detection studies using Random Forest include researches conducted by Belavagi and Muniyal [33], Jiang et al. [34], and Abd and Hadi [35].

**3.3. Measurement**

In this experiment, the detection performance measurement was carried out for the Random Forest classification algorithm by measuring Accuracy, True Positive Rate (TPR), False Positive Rate (FPR), Precision, F-Measure and (Receiving Operating Curve) ROC metrics.

- *Accuracy:* is defined as the level of closeness between the categorization value and the actual value. Often used to measure the effectiveness of classification algorithms. Also known as Classification Rate (CR)

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

- *TPR:* is defined as actual positive are correctly categorized as the positive class. Also known as Recall or Detection Rate (DR) or Sensitivity.

$$TPR = \frac{TP}{TP+FN} \quad (2)$$

- *FPR:* is defined as actual negative are categorized as the positive class. Normal traffic is considered an attack. Also known as the False Acceptance Rate (FAR) or fall-out.

$$False\ Positive\ Rate = \frac{FP}{FP+TN} \quad (3)$$

- *Precission:* is defined as a measure of the estimated probability of a correct positive prediction. Also known as Positive Predictive Value (PPV).

$$Precission = \frac{TP}{TP+FP} \quad (4)$$

- *F-Measure* atau *F1-Score*: This is the mean harmonic weight of recall and precision. Used as a comparison of weighted recall and precision rates.

$$F - Measure = \frac{2TP}{2TP+FP+FN} \quad (5)$$

- *ROC* **:** This curve is used to evaluate the performance of the classification algorithm [36]. The X-axis represents the FAR value and the y-axis represents the Sensitivity value.

## 4. RESULTS

This section describes the results of the experiments that have been carried out in this study. The explanation includes the results of selecting features from each approach (Approach-1 and Approach-2), and testing results of the attack detection performances using the classification algorithms.

### 4.1. Selected Features

As have being described in the methodology section, in this study the approach used in feature selection testing is Information Gain. For Approach-1, 22 selected features are presented in Table 4. These features are the most relevant features based on Approach-1. Furthermore, these features will be used to detect normal and attack traffics.

Table 4. Selected Feature from the dataset with 7 Class label (Approach-1)

| No. | Feat. ID | Feature Names |
|---|---|---|
| 1 | 41 | Packet Length Std |
| 2 | 13 | Total Length of Bwd Packets |
| 3 | 65 | Subflow Bwd Bytes |
| 4 | 8 | Destination Port |
| 5 | 42 | Packet Length Variance |
| 6 | 20 | Bwd Packet Length Mean |
| 7 | 54 | Avg Bwd Segment Size |
| 8 | 18 | Bwd Packet Length Max |
| 9 | 67 | Init_Win_bytes_backward |
| 10 | 12 | Total Length of Fwd Packets |
| 11 | 63 | Subflow Fwd Bytes |
| 12 | 66 | Init_Win_bytes_forward |
| 13 | 52 | Average Packet Size |
| 14 | 40 | Packet Length Mean |
| 15 | 39 | Max Packet Length |
| 16 | 14 | Fwd Packet Length Max |
| 17 | 22 | Flow IAT Max |
| 18 | 36 | Bwd Header Length |
| 19 | 9 | Flow Duration |
| 20 | 26 | Fwd IAT Max |
| 21 | 55 | Fwd Header Length |
| 22 | 24 | Fwd IAT Total |

In Approach-2, the Information Gain selection technique is applied to the dataset with 15 high class imbalanced data. A list of sorted ranking features based on the weight generated through Approach-2 is presented in Table 5. These features were subsequently eliminated. By applying the same minimum weight as Approach-1, i.e.: 0.4, then 28 selected features are produced, as displayed in Table 6. Through this process, Approach-2 reduces 63.64% of features number. The features produced by Approach-2 will also be validated using the Random Forest classification algorithm. The validation results from Approach-1 and Approach-2 will then be compared, to see which approach is the most ideal.

Table 5. Feature List from the dataset with 15 Class label

| No. | Weigth | Feat. ID | Feat. Names | No. | Weigth | Feat. ID | Feat. Names |
|---|---|---|---|---|---|---|---|
| 1 | 0.7521 | 41 | Packet Length Std | 40 | 0.337 | 29 | Bwd IAT Mean |
| 2 | 0.7197 | 13 | Total Length of Bwd Packets | 41 | 0.3242 | 7 | Bwd Packets/s |
| 3 | 0.7197 | 65 | Subflow Bwd Bytes | 42 | 0.3237 | 19 | Bwd Packet Length Min |
| 4 | 0.6937 | 66 | Init_Win_bytes_forward | 43 | 0.2874 | 69 | min_seg_size_forward |
| 5 | 0.6916 | 63 | Subflow Fwd Bytes | 44 | 0.2843 | 76 | Idle Max |
| 6 | 0.6916 | 12 | Total Length of Fwd Packets | 45 | 0.2781 | 74 | Idle Mean |
| 7 | 0.6823 | 42 | Packet Length Variance | 46 | 0.2773 | 27 | Fwd IAT Min |
| 8 | 0.6694 | 40 | Packet Length Mean | 47 | 0.2757 | 77 | Idle Min |
| 9 | 0.6571 | 18 | Bwd Packet Length Max | 48 | 0.2752 | 70 | Active Mean |
| 10 | 0.6511 | 39 | Max Packet Length | 49 | 0.2717 | 31 | Bwd IAT Min |
| 11 | 0.6472 | 67 | Init_Win_bytes_backward | 50 | 0.2711 | 72 | Active Max |
| 12 | 0.6401 | 52 | Average Packet Size | 51 | 0.2705 | 73 | Active Min |
| 13 | 0.64 | 20 | Bwd Packet Length Mean | 52 | 0.2596 | 38 | Min Packet Length |
| 14 | 0.64 | 54 | Avg Bwd Segment Size | 53 | 0.2573 | 15 | Fwd Packet Length Min |
| 15 | 0.6313 | 14 | Fwd Packet Length Max | 54 | 0.2489 | 68 | act_data_pkt_fwd |
| 16 | 0.6096 | 8 | Destination Port | 55 | 0.2463 | 23 | Flow IAT Min |
| 17 | 0.6089 | 22 | Flow IAT Max | 56 | 0.2174 | 6 | Bwd IAT Std |
| 18 | 0.5835 | 9 | Flow Duration | 57 | 0.1319 | 46 | PSH Flag Count |
| 19 | 0.5769 | 55 | Fwd Header Length | 58 | 0.0955 | 51 | Down/Up Ratio |
| 20 | 0.5707 | 26 | Fwd IAT Max | 59 | 0.0743 | 47 | ACK Flag Count |
| 21 | 0.5485 | 36 | Bwd Header Length | 60 | 0.0545 | 75 | Idle Std |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 22 | 0.5438 | 24 | Fwd IAT Total | 61 | 0.0477 | 43 | FIN Flag Count |
| 23 | 0.5051 | 25 | Fwd IAT Mean | 62 | 0.0308 | 48 | URG Flag Count |
| 24 | 0.4752 | 21 | Flow IAT Mean | 63 | 0.0294 | 71 | Active Std |
| 25 | 0.4718 | 53 | Avg Fwd Segment Size | 64 | 0.0186 | 32 | Fwd PSH Flags |
| 26 | 0.4718 | 16 | Fwd Packet Length Mean | 65 | 0.0186 | 44 | SYN Flag Count |
| 27 | 0.4673 | 1 | Bwd Packet Length Std | 66 | 0 | 50 | ECE Flag Count |
| 28 | 0.4604 | 2 | Flow Bytes/s | 67 | 0 | 61 | Bwd Avg Bulk Rate |
| 29 | 0.3891 | 64 | Subflow Bwd Packets | 68 | 0 | 56 | Fwd Avg Bytes/Bulk |
| 30 | 0.3891 | 11 | Total Backward Packets | 69 | 0 | 45 | RST Flag Count |
| 31 | 0.3835 | 30 | Bwd IAT Max | 70 | 0 | 58 | Fwd Avg Bulk Rate |
| 32 | 0.375 | 4 | Flow IAT Std | 71 | 0 | 57 | Fwd Avg Packets/Bulk |
| 33 | 0.3695 | 5 | Fwd IAT Std | 72 | 0 | 35 | Bwd URG Flags |
| 34 | 0.3693 | 17 | Fwd Packet Length Std | 73 | 0 | 49 | CWE Flag Count |
| 35 | 0.3625 | 28 | Bwd IAT Total | 74 | 0 | 59 | Bwd Avg Bytes/Bulk |
| 36 | 0.3543 | 3 | Flow Packets/s | 75 | 0 | 33 | Bwd PSH Flags |
| 37 | 0.354 | 62 | Subflow Fwd Packets | 76 | 0 | 34 | Fwd URG Flags |
| 38 | 0.354 | 10 | Total Fwd Packets | 77 | 0 | 60 | Bwd Avg Packets/Bulk |
| 39 | 0.3501 | 37 | Fwd Packets/s | | | | |

Table 6. Selected Feature from 15 traffic class labels dataset (Approach-2)

| No. | Weigth | Feat. ID | Feat. Names |
|---|---|---|---|
| 1 | 0.7521 | 41 | Packet Length Std |
| 2 | 0.7197 | 13 | Total Length of Bwd Packets |
| 3 | 0.7197 | 65 | Subflow Bwd Bytes |
| 4 | 0.6937 | 66 | Init_Win_bytes_forward |
| 5 | 0.6916 | 63 | Subflow Fwd Bytes |
| 6 | 0.6916 | 12 | Total Length of Fwd Packets |
| 7 | 0.6823 | 42 | Packet Length Variance |
| 8 | 0.6694 | 40 | Packet Length Mean |
| 9 | 0.6571 | 18 | Bwd Packet Length Max |
| 10 | 0.6511 | 39 | Max Packet Length |
| 11 | 0.6472 | 67 | Init_Win_bytes_backward |
| 12 | 0.6401 | 52 | Average Packet Size |
| 13 | 0.64 | 20 | Bwd Packet Length Mean |
| 14 | 0.64 | 54 | Avg Bwd Segment Size |
| 15 | 0.6313 | 14 | Fwd Packet Length Max |
| 16 | 0.6096 | 8 | Destination Port |
| 17 | 0.6089 | 22 | Flow IAT Max |
| 18 | 0.5835 | 9 | Flow Duration |
| 19 | 0.5769 | 55 | Fwd Header Length |
| 20 | 0.5707 | 26 | Fwd IAT Max |
| 21 | 0.5485 | 36 | Bwd Header Length |
| 22 | 0.5438 | 24 | Fwd IAT Total |
| 23 | 0.5051 | 25 | Fwd IAT Mean |
| 24 | 0.4752 | 21 | Flow IAT Mean |
| 25 | 0.4718 | 53 | Avg Fwd Segment Size |
| 26 | 0.4718 | 16 | Fwd Packet Length Mean |
| 27 | 0.4673 | 1 | Bwd Packet Length Std |
| 28 | 0.4604 | 2 | Flow Bytes/s |

### 4.2. Detection Performances

To test whether the features generated by the proposed method can be used to detect normal or attacks traffics on high-dimensional and imbalance data, validation is carried out through detection performances testing using features selected through Approach-1 and Approach -2. This detection test uses the Random Forest classification algorithm. Experiment results show very high accuracy as selected features are more and they are relevant and important in characterizing the attacks patterns, thus the classification algorithms are able to identify very well the attacks. In addition, to maintain the reliability of the test results, several testing modes were used, i.e.: the use of full train, 5-fold cross-validation, 10-fold cross-validation, and 10-90% data splitting which were applied to training data and testing data.

### 4.3 Measuring TPR, FPR, Precision, F-Measure, and ROC for Approach-1

In this experiment, the features generated by Approach-1 were used to detect attacks using the Random Forest classification algorithm. Table 7 presents the results of detection testing using the features selected in Approach-1. Based on the results of TPR, FPR, Precision, F-Measure, and ROC, it shows that using the features selected by Approach-1 on training dataset, the Random Forest algorithm, has an excellent performance for identifying normal and attack traffics.

Table 7. Detection Performances for Approach-1 on Training Dataset

| Class | TPR | FPR | Precision | F-Measure | ROC |
|---|---|---|---|---|---|
| BENIGN | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DDoS | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| PortScan | 1.000 | 0.000 | 0.999 | 0.999 | 1.000 |
| Bot | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack Brute Force | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack XSS | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack Sql Injection | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Infiltration | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS slowloris | 0.999 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS Slowhttpte | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS Hulk | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS GoldenEye | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Heartbleed | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| FTP-Patator | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| SSH-Patator | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |

The Random Forest algorithm performance also looks excellent when tested using testing dataset as presented in Table 8. The measurement results show that the TPR, Precision, F-Measure, and ROC values for all types of traffic reach 1.000 and with a very low FPR value of 0.000.

Table 8. Detection Performance for Approach-1 on Testing Dataset

| Class | TPR | FPR | Precision | F-Measure | ROC |
|---|---|---|---|---|---|
| BENIGN | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DdoS | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| PortScan | 1.000 | 0.000 | 0.999 | 1.000 | 1.000 |
| Bot | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack Brute Force | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack XSS | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack Sql Injection | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Infiltration | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS slowloris | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS Slowhttpte | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS Hulk | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS GoldenEye | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Heartbleed | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| FTP-Patator | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| SSH-Patator | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |

### 4.4 Measuring TPR, FPR, Precision, F-Measure, and ROC for Approach-2

Through Approach-2, 28 relevant features have been generated. The 28 features are used as input to detect attacks using the Random Forest algorithm. Based on the experimental results using training dataset, the Random Forest's performance in detecting attacks is shown in Table 9. The results of expereiments with testing dataset are presented in Table 10. The experimental results using both training dataset and testing dataset show that with the features generated through Approach-2, the Random Forest algorithm can detect both normal and attack traffics in imbalanced dataset.

Table 9. Detection Performance for Approach-2 on Training Dataset

| Class | TPR | FPR | Precision | F-Measure | ROC |
|---|---|---|---|---|---|
| BENIGN | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DDoS | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| PortScan | 1.000 | 0.000 | 0.999 | 0.999 | 1.000 |
| Bot | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack Brute Force | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack XSS | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack Sql Injection | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Infiltration | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS slowloris | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS Slowhttpte | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS Hulk | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS GoldenEye | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Heartbleed | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| FTP-Patator | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| SSH-Patator | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |

Table 10. Detection Performance for Approach-2 on Testing Dataset

| Class | TPR | FPR | Precision | F-Measure | ROC |
|---|---|---|---|---|---|
| BENIGN | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DdoS | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| PortScan | 1.000 | 0.000 | 0.999 | 1.000 | 1.000 |
| Bot | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack Brute Force | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack XSS | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Web Attack Sql Injection | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Infiltration | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS slowloris | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS Slowhttpte | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS Hulk | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| DoS GoldenEye | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| Heartbleed | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| FTP-Patator | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |
| SSH-Patator | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 |

The experimental results show that the performance of Approach-2 is not very significant when compared to the performance of Approach-1. It is because the features generated through Approach-1 are also belonging to Approach-2. From the 28 features produced in Approach-2, 22 of them are in Approach-1. The features that produced by Approach-2 but not by Approach-1 are *Flow IAT Mean, Avg Fwd Segment Size, Fwd Packet Length Mean, Bwd Packet Length Std, and Flow Bytes/s*.

### 4.5. Accuracy Testing

The detection engine performance can also be measured by accuracy. Accuracy shows how the machine's ability to predict traffic according to its actual conditions. In other words, machine capabilities to classify exactly a class. Table 11 shows Random Forest's performance on accuracy. As explained in the previous section, in this experiment, several test modes were used, i.e.: Full Train, 10-fold, 5-fold, split 10 to split 90. The experimental results show that by using the features generated through Approach-1, the accuracy of Random Forest algorithm in predicting normal and attack traffics is excellent with an average accuracy value of 99.842% using training dataset and 99.830% using testing dataset.

Table 11. Result of Validation Test of Approach-1 Applied to 15 Class Dataset

| | Total Instances | | Accuracy | |
|---|---|---|---|---|
| Test Mode | Training | Testing | Training | Testing |
| Use Training Set | 594456 | 254767 | 99.989 | 99.994 |
| 10-Fold | 594456 | 254767 | 99.847 | 99.829 |
| 5-Fold | 594456 | 254767 | 99.844 | 99.831 |
| Split 10 | 535010 | 229290 | 99.762 | 99.829 |
| Split 20 | 475565 | 203814 | 99.803 | 99.745 |
| Split 30 | 416119 | 178337 | 99.817 | 99.772 |
| Split 40 | 356674 | 152860 | 99.833 | 99.779 |
| Split 50 | 297228 | 127383 | 99.831 | 99.821 |
| Split 60 | 237782 | 101907 | 99.831 | 99.825 |
| Split 70 | 178337 | 76430 | 99.844 | 99.837 |
| Split 80 | 118891 | 50953 | 99.847 | 99.835 |
| Split 90 | 59446 | 25477 | 99.860 | 99.859 |
| **Average** | | | **99.842** | **99.830** |

Furthermore, using the features generated by Approach-2 the accuracy of Random Forest algorithm in predicting traffic is presented in Table 12. The experimental results also show the accuracy of Random Forest algorithm which is excellent with an average accuracy value of 99.820% for training data and 99.790% for testing data.

Table 12. Result of Validation Test of Approach-2 Applied to 15 Class Dataset

| | Total Instances | | Accuracy (%) | |
|---|---|---|---|---|
| Testing Mode | Training | Testing | Training | Testing |
| Use Training Set | 594456 | 254767 | 99.989 | 99.994 |
| 10-Fold | 594456 | 254767 | 99.829 | 99.805 |
| 5-Fold | 594456 | 254767 | 99.827 | 99.805 |
| Split 10 | 535010 | 229290 | 99.729 | 99.591 |
| Split 20 | 475565 | 203814 | 99.773 | 99.718 |

| | | | | |
|---|---|---|---|---|
| Split 30 | 416119 | 178337 | 99.788 | 99.759 |
| Split 40 | 356674 | 152860 | 99.802 | 99.772 |
| Split 50 | 297228 | 127383 | 99.807 | 99.801 |
| Split 60 | 237782 | 101907 | 99.812 | 99.805 |
| Split 70 | 178337 | 76430 | 99.825 | 99.813 |
| Split 80 | 118891 | 50953 | 99.821 | 99.806 |
| Split 90 | 59446 | 25477 | 99.844 | 99.806 |
| **Average** | | | **99.820** | **99.790** |

### 4.6. Comparison

Having done experimentations on features selection using Approach-1 and Approach-2, validation is carried out with several classification algorithms, i.e.: RF, Naïve Bayes (NB), J48, RepTree, Bayes Network (Bnet), and OneR. This validation aims to see the ability of each algorithm in detecting the type of traffic using the selected features. The classification algorithms validation was carried out using training data and testing data. Details of the Approach-1 and Approach-2 validation processes, presented in Algorithm-1 and Algorithm-2.

***Algorithm 1:*** *Approach-1 Validation*

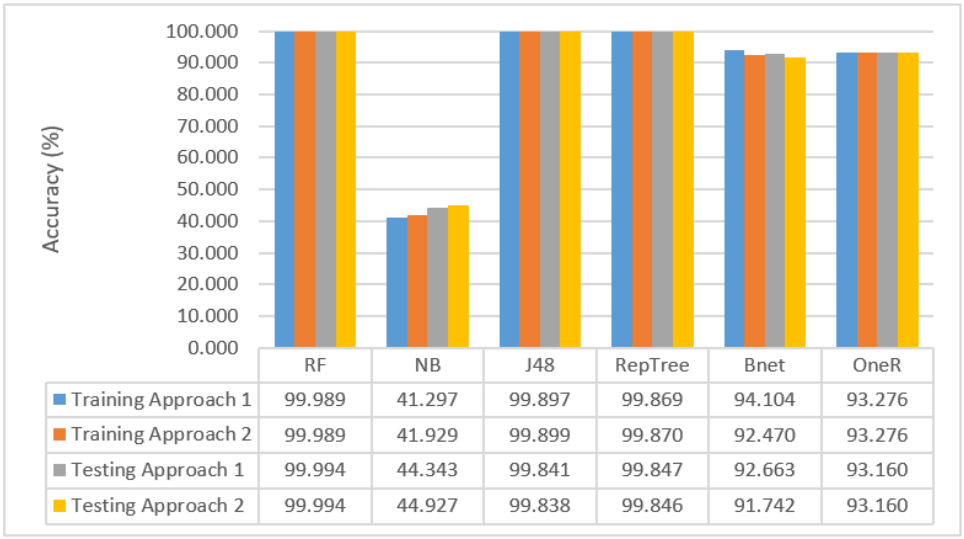
*Step-1: Procedure model ()*
*Step-2: Input Fn= CICIDS-2017 MachineLearningCSV dataset with 78 features;*
*Step-3: Use 22 Features = {f8,f9,f12,f13,f14,f18,f20,f22,f24,f26,f36,f39,f40,f41,f42,f52,f54,f55,f63,f65,f66,f67} from Fn set as Subset1;*
*Step-4: Provide Subset1 to Random Forest, Naïve Bayes (NB), J48, RepTree, Bayes Network (Bnet), and OneR using 30% of datasest;*
*Step-5: Calculate Accuracy, TPR, FPR, Precision, F-Measure, and ROC as Performance;*
*Step-6: Compare Performance of Random Forest, Naïve Bayes (NB), J48, RepTree, Bayes Network (Bnet), and OneR;*
*Step-7: Select the best Result*

***Algorithm 2:*** *Approach-2 Validation*

*Step-1: Procedure model ()*
*Step-2: Input Fn= CICIDS-2017 MachineLearningCSV dataset with 78 features;*
*Step-3: Use 28 Features = {f1,f2,f8,f9,f12,f13,f14,f16,f18,f20,f21,f22,f24,f25,f26,f36,f39,f40,f41,f42,f52,f53,f54,f55,f63 ,f65,f66,f67} from Fn set as Subset2;*
*Step-4: Provide Subset2 to Random Forest, Naïve Bayes (NB), J48, RepTree, Bayes Network (Bnet), and OneR using 30% of datasest;*
*Step-5: Calculate Accuracy, TPR, FPR, Precision, F-Measure, and ROC as Performance;*
*Step-6: Compare Performance of Random Forest, Naïve Bayes (NB), J48, RepTree, Bayes Network (Bnet), and OneR;*
*Step-7: Select the best Result*

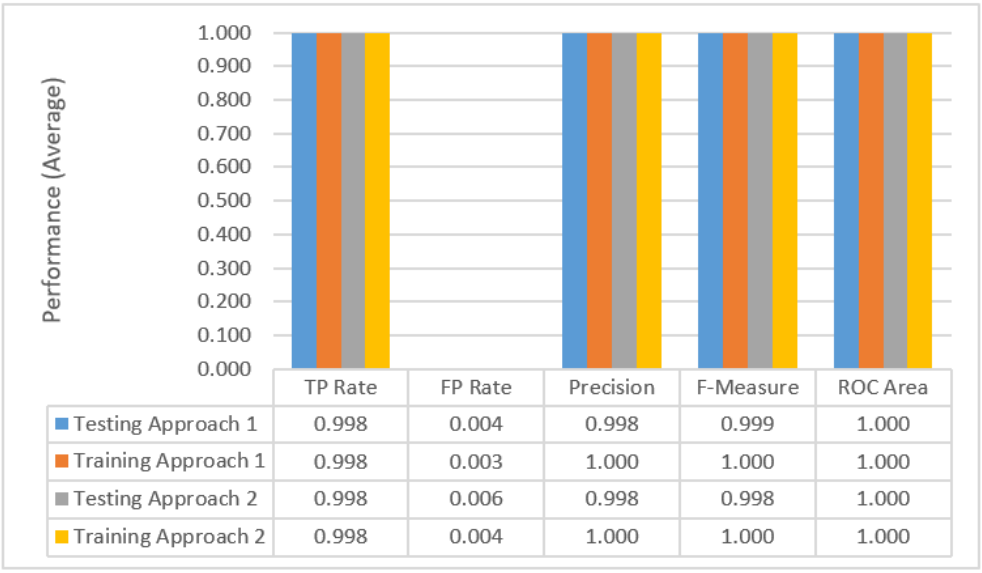
Figure 2 presents a graph of the accuracy comparison between Approach-1 and Approach-2 for different classification algorithms applied to training data and testing data. The comparison results show that the Random Forest algorithm is excellent at detecting normal and attack traffics using the features generated by Approach-1 and Approach-2.

In Figure 3, a comparison of the performance of Approach-1 and Approach-2 is presented. Comparisons were made based on the mean values of TPR, FPR, Precision, F-Measure, and ROC. Based on the graph, it can be seen that the TPR values of Approach-1 and Approach-2 are the same, i.e.: 0.998. Meanwhile, for the mean value of FPR, Approach-1 is better than Approach-2. Furthermore, the average value of Precision, F-Measure, and ROC shows the same performance between Approach-1 and Approach-2.




| | RF | NB | J48 | RepTree | Bnet | OneR |
|---|---|---|---|---|---|---|
| Training Approach 1 | 99.989 | 41.297 | 99.897 | 99.869 | 94.104 | 93.276 |
| Training Approach 2 | 99.989 | 41.929 | 99.899 | 99.870 | 92.470 | 93.276 |
| Testing Approach 1 | 99.994 | 44.343 | 99.841 | 99.847 | 92.663 | 93.160 |
| Testing Approach 2 | 99.994 | 44.927 | 99.838 | 99.846 | 91.742 | 93.160 |

Figure 2. Accuracy detection for each approach




| | TP Rate | FP Rate | Precision | F-Measure | ROC Area |
|---|---|---|---|---|---|
| Testing Approach 1 | 0.998 | 0.004 | 0.998 | 0.999 | 1.000 |
| Training Approach 1 | 0.998 | 0.003 | 1.000 | 1.000 | 1.000 |
| Testing Approach 2 | 0.998 | 0.006 | 0.998 | 0.998 | 1.000 |
| Training Approach 2 | 0.998 | 0.004 | 1.000 | 1.000 | 1.000 |

Figure 3 TPR, FPR, Precision, F-Measure and ROC of each Approach

Performance comparison between the proposed approaches and previous studies is shown in Table 13. Figures in the table, show that both Approach-1 and Approach-2 have better performance than previous studies in term of Accuracy, TPR, Precision, F-Measure, and ROC.

Table 13. Comparison with State-of-the-art research on CICIDS-2017 Dataset

| Authors | Method | Accuracy | Precision | Recall | F-Score | ROC |
|---|---|---|---|---|---|---|
| [19] | SMOTE + PCA dan EFS + Adaboost | 81.83% | 81.83%, | 100% | 90.01%. | 0.902 |
| [20] | UDDB + AE dan PCA | 99.60% | 98.90% | 98.80% | 98.80% | NA |
| [21] | IGAN-IDS | 84.45% | 84.85% | 84.45% | 84.17% | 0.955 |
| [24] | Artificial Neural Network and Machine Learning algorithm | 96.24% | NA | NA | NA | NA |
| Approach-1 | Information Gain (7 Class) + Random Forest (15 Class) | 99.83 | 99.90% | 99.80% | 99.90% | 1.000 |
| Approach-2 | Information Gain (15 Class) + Random Forest (15 Class) | 99.79 | 99.80% | 99.90% | 99.90% | 1.000 |

## 5. CONCLUSION

This study has proposed two approaches to produce relevant features to be used to detect attacks on high dimensionality, multi-class, and high-class imbalanced dataset. The Random Forest algorithm was chosen as the classification method because of its ability to handle multiclass data. Based on the results of the experiments on CICIDS-2017 dataset with 15 traffic class labels, Approach-1 and Approach-2 produced 22 and 28 important features, respectively. Furthermore, experiments on validations showed that combination of Approach-1 with 22 important features and Random Forest classification algorithm worked well in detecting attacks with an average accuracy rate of 99.842% on the training dataset and 99.830% on the test dataset. In addition, the results of the experiment prove that the proposed approach is able to provide recommendations of important and relevant features. With Random Forest algorithm, the resulting features are able to detect attacks with better performance on high-dimensional and high-class imbalanced datasets. The experimental results also show that the proposed method exceeds the performance of the state-of-the-art methods in terms of Accuracy, TPR, FPR, Precision, and ROC.

Although this research has shown surprising results, the Information Gain technique yet requires repeated experiments and validations to obtain the minimum weight for selecting important features. Therefore, in the near future, the research will focus on finding the most optimal way to produce the ideal features with involving intelligent approaches.

## ACKNOWLEDGMENTS

This reserach supported by Universitas Dinamika Bangsa through human resource development programs and collaboration with Comnets Lab Universitas Sriwijaya.

## REFERENCES

[1] K. Naidu, A. Dhenge, and K. Wankhade, "Feature selection algorithm for improving the performance of classification: A survey," *Proc. - 2014 4th Int. Conf. Commun. Syst. Netw. Technol. CSNT 2014*, pp. 468–471, 2014.

[2] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and Big Heterogeneous Data: a Survey," *J. Big Data*, vol. 2, no. 1, 2015.

[3] S. Chormunge and S. Jena, "Efficient feature subset selection algorithm for high dimensional data," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 4, pp. 1880–1888, 2016.

[4] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," *Ski. 2014 - 8th Int. Conf. Software, Knowledge, Inf. Manag. Appl.*, 2014.

[5] B. A. Tama and K. H. Rhee, "A Combination of PSO-Based Feature Selection and Tree-Based Classifiers Ensemble for Intrusion Detection Systems," *Adv. Comput. Sci. Ubiquitous Comput.*, vol. 373, pp. 489–495, 2015.

[6] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 420–432, 2016.

[7] P. Kushwaha, H. Buckchash, and B. Raman, "Anomaly based intrusion detection using filter based feature selection on KDD-CUP 99," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2017-Decem, pp. 839–844, 2017.

[8] F. Chen, Z. Ye, C. Wang, L. Yan, and R. Wang, "A feature selection approach for network intrusion detection based on tree-seed algorithm and k-nearest neighbor," *Proc. 2018 IEEE 4th Int. Symp. Wirel. Syst. within Int. Conf. Intell. Data Acquis. Adv. Comput. Syst. IDAACS-SWS 2018*, pp. 68–72, 2018.

[9] F. Gottwalt, E. Chang, and T. Dillon, "CorrCorr: A feature selection method for multivariate correlation network anomaly detection techniques," *Comput. Secur.*, vol. 83, pp. 234–245, 2019.

[10] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Comput. Networks*, vol. 174, no. October 2019, 2020.

[11] Y. Dhote, S. Agrawal, and A. J. Deen, "A Survey on Feature Selection Techniques for Internet Traffic Classification," *Proc. - 2015 Int. Conf. Comput. Intell. Commun. Networks, CICN 2015*, pp. 1375–1380, 2016.

[12] P. R. K. Varma, V. V. Kumari, and S. S. Kumar, *A Survey of Feature Selection Techniques in Intrusion Detection System: A Soft Computing Perspective*, vol. 710. Springer Singapore, 2018.

[13] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Bin Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.

[14] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[15] S. Rodda and U. S. R. Erothi, "Class imbalance problem in the Network Intrusion Detection Systems," *Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016*, pp. 2685–2688, 2016.

[16] M. Reza, S. Miri, and R. Javidan, "A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 6, pp. 20–25, 2016.

[17] B. Yan, G. Han, M. Sun, and S. Ye, "A novel region adaptive SMOTE algorithm for intrusion detection on

imbalanced problem," *2017 3rd IEEE Int. Conf. Comput. Commun. ICCC 2017*, vol. 2018-Janua, pp. 1281–1286, 2018.
[18] J. H. Seo and Y. H. Kim, "Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection," *Comput. Intell. Neurosci.*, vol. 2018, 2018.
[19] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019.
[20] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electron. (Switzerland). MPDI*, vol. 8, no. 3, p. 322, 2019.
[21] S. Huang and K. Lei, "IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," *Ad Hoc Networks*, vol. 105, 2020.
[22] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems," *Appl. Intell.*, 2020.
[23] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSP 2018 - Proc. 4th Int. Conf. Inf. Syst. Secur. Priv.*, vol. 2018-Janua, no. Cic, pp. 108–116, 2018.
[24] Z. Pelletier and M. Abualkibash, "Evaluating the CIC IDS-2017 Dataset Using Machine Learning Methods and Creating Multiple Predictive Models in the Statistical Computing Language R," *Int. Res. J. Adv. Eng. Sci.*, vol. 5, no. 2, pp. 187–191, 2020.
[25] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *Int. J. Eng. Technol.*, vol. 7, no. 3.24 Special Issue 24, pp. 479–482, 2018.
[26] M. N. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 4537, no. c, pp. 1–1, 2020.
[27] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc.*, 2015.
[28] A. I. Madbouly and T. M. Barakat, "Enhanced relevant feature selection model for intrusion detection systems," *Int. J. Intell. Eng. Informatics*, vol. 4, no. 1, p. 21, 2016.
[29] I. Syarif, "Feature Selection of Network Intrusion Data using Genetic Algorithm and Particle Swarm Optimization," *Emit. Int. J. Eng. Technol.*, vol. 4, no. 2, pp. 277–290, 2016.
[30] K. Singh and B. Nagpal, "Random Forest Algorithm in Intrusion Detection System : A Survey," vol. 3, no. 5, pp. 673–676, 2018.
[31] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*. 2011.
[32] M. C. Belavagi and B. Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," *Procedia Comput. Sci.*, vol. 89, pp. 117–123, 2016.
[33] J. Jiang, Q. Wang, Z. Shi, B. Lv, and B. Qi, "RST-RF: A hybrid model based on rough set theory and random forest for network intrusion detection," *ACM Int. Conf. Proceeding Ser.*, pp. 77–81, 2018.
[34] A. Abd and A. Hadi, "Performance Analysis of Big Data Intrusion Detection System over Random Forest Algorithm," *Int. J. Appl. Eng. Res.*, vol. 13, no. 2, pp. 1520–1527, 2018.
[35] R. K. Singh, S. Dalal, V. K. Chauhan, and D. Kumar, "Optimization of FAR in Intrusion Detection System by Using Random Forest Algorithm," *SSRN Electron. J.*, pp. 3–6, 2019.
[36] D. Summeet and D. Xian, *Data Mining and Machine Learning in Cybersecurity*. CRC Press, 2011.

## BIOGRAPHY OF AUTHORS

KURNIABUDI received his master degree in Computer Science from Universitas Putra Indonesia YPTK Padang, West Sumatera, Indonesia. Currently he is a PhD candidate at Faculty of Engineering, Universitas Sriwijaya. He is currently a senior lecturer at Faculty of Computer Science, Universitas Dinamika Bangsa, Indonesia. His research interests include technology adoption, information technology, information security, and network security.

DERIS STIAWAN received the PhD degree in Computer Engineering from Universiti Teknologi Malaysia, Malaysia. He is currently an Associate Professor at Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer network, Intrusion Detection/ Prevention System, and heterogeneous network.

DARMAWIJOYO received his Doctor of Mathematics from Delft University of Technology, Netherlands. He is currently an Associate Professor at the Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Sriwijaya. His research interests include problem solving, applied mathematics, modeling, and mathematical thinking.

MOHD YAZID IDRIS is an Associate Professor at School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia. He obtained his M.Sc and Ph.D. in the area of Software Engineering, and Information Technology (IT) Security in 1998 and 2008 respectively. In software engineering, he focuses on the research of designing and development of mobile and telecommunication software. His main research activity in IT security is in the area of Intrusion Prevention and Detection (IPD).

BEDINE KERIM obtained his Ph.D in Computer Science from university of Le Havre, Le Havre, France. He is currently assistant professor in the college of computer science and information technology at Albaha University-KSA. He is curious researcher and rigorous academic with good background in artificial intelligence, mathematical modeling, game theory, machine learning, cloud computing, Fuzzy Logic.

RAHMAT BUDIARTO received B.Sc. degree from Bandung Institute of Technology in 1986, M.Eng. and Dr.Eng. in Computer Science from Nagoya Institute of Technology in 1995 and 1998, respectively. Currently, he is a full Professor at College of Computer Science and IT, Albaha University, Saudi Arabia. His research interests include intelligent systems, brain modeling, IPv6, network security, Wireless sensor networks, and MANETs.