The IJICS (International Journal of Informatics and Computer Science)
Vol 6 No 1, March 2022, Page 1–14
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v6i1.3890

Optimizing Attack Detection for High Dimensionality and Imbalanced Data with SMOTE, Chi-Square and Random Forest Classifier

Kurniabudi*, Abdul Harris, Veronica, Elvi Yanti

Faculty of Computer Sciences, Universitas Dinamika Bangsa, Jambi, Indonesia
Email: kbudiz@yahoo.com
Submitted: **04/03/2022**; Accepted: **30/03/2022**; Published: **31/03/2022**

Abstract−The rapid growth of the network generates a very large and varied amount of traffic which has an impact on data and information security. This study resolves two common problems in attack detection, namely high dimensionality and high-class imbalance of the network traffic. This study used the ISCX CICIDS-2017 dataset. This study used the ISCX CICIDS-2017 dataset. The CICIDS-2017 dataset is imbalance that contains very diverse types of traffic including normal traffic and several types of attacks (multi-class). This study proposes a combination of the Chi-Square feature selection technique with the Tree-Based Classifier Random Forest. In the experiment first the Chi-Square Correlation Based feature selection technique was applied to the imbalance dataset. The selected features are then validated using several Random Forest algorithms. The test was also performed comparisons with other classification algorithms such as Naïve Bayes, Bayes Network, J48, REPTree, and Adaboost. This study also examines the implementation of SMOTE to overcome the problem of high calass imbalance. The test results also show that the proposed ensemble method has a very good performance from the Accuracy, TPR, FPR, Precision, F-Measure, and ROC values.

Keywords: Attack Detection; Class Imbalanced; Feature Selection; High Dimensionality; SMOTE; Tree-Based Classifier

1. INTRODUCTION

The rapid growth of information and communication technology has made various applications, devices, and protocols are able to be connected to one data communication network. This development has increased tremendously the amount of traffic and its complexity, and took effect on network security. In fact, the amount of attack traffic is much less than the normal traffic. With the presence of various techniques and types of attacks, it becomes a challenge for intrusion detection systems to be able to recognize low scale of attack traffic on a very massive and very diverse traffic flow.

On the other hand, machine learning has been widely applied to address security issues in computer networks. Machine learning has the two big issues, i.e.: data high-dimensionality and imbalanced data [1][2]. Feature Selection (FS) is a popular technique for handling the data dimensionality. FS performs variable elimination, helps in understanding data, reduces computational requirements, reduces the "curse of dimensionality" effect, and improves prediction engine performance[3]. FS is part of dimensional reduction, which is known as the process of selecting an optimal features subset that represents the entire dataset[4]. Various feature selection techniques have been introduced and applied to deal with the high-dimensionality of data such as Information Gain[5], Gain Ratio[6], Correlation-Based[7], Chi-square[8], and others. Therefore, chi-square feature selection technique was chosen because it has the ability to rank features based on statistical significance test. Chi-square only recommends features that affect the class label[9]. One of the objectives of this research is to prove the ability of chi-square to select features on unbalanced data.

In attack detection using classification techniques, imbalanced data often causes the classification algorithms work incorrectly [10]. To overcome problems with imbalanced data, several sampling techniques have been introduced such as: over-sampling, down-sampling and learning[11]. In addition, several algorithms have been implemented such as Synthetic Minority Oversampling Technique (SMOTE)[12]. SMOTE and cluster center and nearest neighbor (CANN) algorithm were implemented by Reza et al. [13] on NSL-KDD dataset. Experimental result shows that the proposed idea is able to eliminate the limitations in detecting U2R and R2L attacks. Yan et al. [14], propose a region adaptive-SMOTE (RA-SMOTE) algorithm to solve imbalanced class problem. Three classification algorithms, i.e.: Support Vector Machines (SVM), Back Propagation neural network (BPNN), and random forests (RF) are used for evaluating the proposed algorithm. Experimental results on the NSL-KDD dataset show that the proposed algorithm is able to solve the imbalanced class problem. Then, Yan and Han [15], propose a modified local adaptive SMOTE (LA-SMOTE) algorithm and use gated recurrent units (GRU) neural network as classifier. Experiments on NSL-KDD dataset indicate that the proposed algorithm has perfect performance with low false alarm rate, besides being able to solve learning problems in imbalanced class. However, the study was carried out on dataset with limited amount of records. Thus, it is interesting to discover how SMOTE deals with a dataset, which has imbalance data, vary in traffic types, and high data dimensionality.

In this study, CICIDS-2017 dataset used because it contains huge number of records with complex traffic and high-class imbalance data, thus, CICIDS-2017 dataset represents well the actual network traffic. The choice of random forest as a classification algorithm is due to its ability to handle large datasets[14]. However, to improve random forest performance, it is necessary to eliminate input features. therefore, chi-square research was used to eliminate unimportant features. In this work, we introduce an attack detection system using combination of Chi-square and Random Forest feature selection techniques, which capable of detecting attacks on imbalanced traffic data. SMOTE method is also implemented to deal with the high-class imbalance at the data level. In fact, traffic in real network is imbalance and varies in types. This study has major contributions including: 1) implementation and testing of SMOTE on the CICIDS-2017 dataset, which has complex, high-dimensional and imbalanced data: 2) testing the chi-square feature selection technique

on the high-dimensional and imbalanced IDS dataset; and 3) Produce an attack detection method that is able to detect normal traffic and attacks on high-dimensional and imbalanced data.

2. RESEARCH METHODOLOGY

2.1 Research Framework

Figure 1 illustrates the proposed method. A combination of feature selection techniques and classification algorithms is proposed to overcome the problem of traffic detection on multi-class and high-class data. The process carried out in the experiment can be explained as follows:

a. The dataset used in this study is CICIDS-2017 dataset from ISCX.
b. Data preparation is carried out to eliminate irrelevant and unimportant features and to handle missing values.
c. For the experimental purposes, only 30% of the CICIDS-2017 dataset is used.
d. SMOTE implementation to deal with data imbalance.
e. Perform feature selection using correlation-based with Chi-square technique to reduce data dimensions, by eliminating irrelevant features.
f. The features of the selection results are then analyzed using several classification algorithms, including Naïve Bayes, Bayes Network, J48, REPTree, Random Forest, and Adaboost.and Adaboost.
g. The learning process in Weka, the training set mode is applied. With this mode, all input data will be used during the learning.
h. Compute the values of Accuracy, TPR, FPR, Precision, F-Measure, and ROC measurements for comparison purpose.

The main objective of the experiment is to produce the best performing an attack detection method on high dimensionality and imbalanced data.

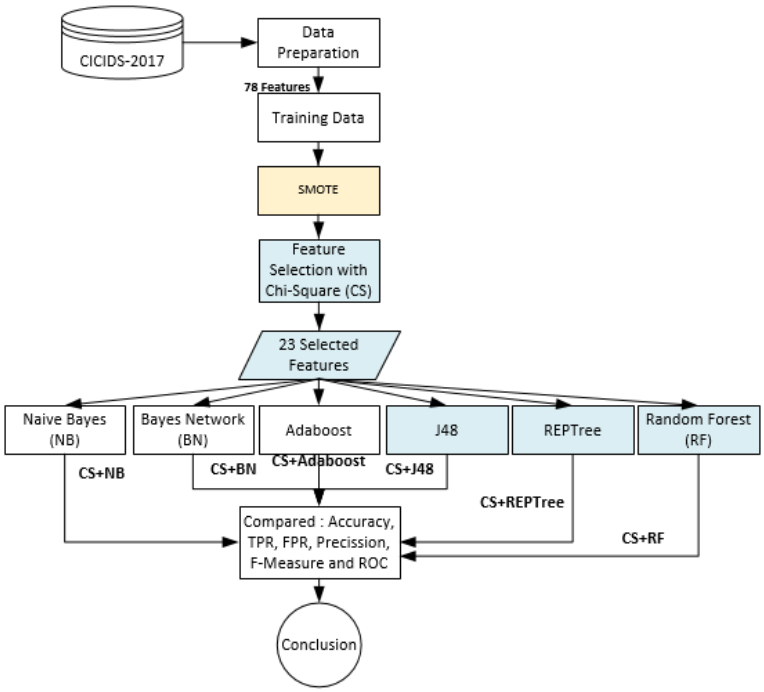

Figure 1. Proposed method

2.2 Dataset

CICIDS-2017 dataset is obtained from ISCX UNB [17]. The dataset is a collection of traffic data captured on weekdays, as a result of fishing traffic for six days with eight observation sessions with different scenarios as presented in Table 1 The dataset consists of normal traffic (benign) and attack traffic (attack) as shown in Table 2 Each record of the CICIDS-2017 dataset has 78 traffic features. The types of attacks contained in the dataset are common types such as Web attack, Brute force, DoS, DDoS, Infiltration, Heartbleed, Bot, and Scan attacks [18].

Table 1. ISCX CICIDS-2017 Dataset Profile

Activities (Day)	File Name	Number of Record	Number of Attribute	Type of Traffic

Monday	Monday WorkingHours,pcap_ISCX .csv	529,918	78	Benign (529,918)
Tuesday	Tuesday WorkingHours,pcap_ISCX .csv	445,909	78	Benign (432,074), SSH-Patator (5897), FTP-Patator (7938)
Wednesday	Wednesday workingHours,pcap_ISCX. csv	692,703	78	Benign (440,031), DoS Hulk (231073, DoS GoldenEye (10293), DoS Slowloris (5796), DoS Slowhttptest (5499), Heartbleed (11)
Thursday	Thursday-WorkingHours Morning-WebAttacks,pcap_ ISCX.csv	170,366	78	Benign (168186), Web Attack-Brute Force (1507), Web Attack-Sql Injection (21), Web Attack-XSS (652)
Thursday	Thursday-WorkingHours Afternoon-Infilteration,pcap_ ISCX.csv	288,602	78	Benign (288566), Infiltration (36)
Friday	Friday-WorkingHours Morning,pcap_ISCX.csv	191,033	78	Benign (189067), Bot (1966)
Friday	Friday-WorkingHours-Afternoon PortScan,pcap_ISCX.csv	286,467	78	Benign (127537), Portscan (158930)
Friday	Friday-WorkingHours-Afternoon DDos,pcap_ISCX.csv	225,745	78	Benign (97718), DDoS (128027)

Table 2. Types of Traffic on the CICIDS-2017 Dataset

No	Type of Traffic	Amount	% of the total data
1	BENIGN	2,273,097	80.30
2	DDoS	128,027	4.52
3	PortScan	158,930	5.61
4	Bot	1,966	0.07
5	Web Attack Brute Force	1,507	0.05
6	Web Attack XSS	652	0.02
7	Web Attack Sql Injection	21	0.00
8	Infiltration	36	0.00
9	DoS slowloris	5,796	0.20
10	DoS Slowhttptest	5,499	0.19
11	DoS Hulk	231,073	8.16
12	DoS GoldenEye	10,293	0.36
13	Heartbleed	11	0.00
14	FTP-Patator	7,938	0.28
15	SSH-Patator	5,897	0.21
	Total	2,830,743	

2.3 Machine Learning Algorithm

We describe briefly machine learning algorithms used in this work, as follows.

a. Naïve Bayes. Bayesian classification is a statistical classification that can predict the probability of class membership, the Bayesian classification is based on the Bayes theorem [19]. Bayesian classification is better known as the Naïve Bayes classification. Naïve Bayes assumes that the effect of attribute values on the class is independent of other attribute values. Several anomaly detection studies using Naive Bayes include research [20].
b. Bayesian Network (BN). BN is a model that encodes probabilistic relationships between variables of interest. The accuracy of this method depends on assumptions that are usually based on the model behavior of the target system. So, any significant deviation from the assumptions will cause a decrease in detection accuracy [21]. Several studies that detect anomalies using the Bayesian network include research [22].
c. REPTree. REPTree is one of the Decision Tree algorithms; REPTree uses tree regression logic and creates several trees in different iterations. After it selects the best of all the resulting trees, it will be considered a representative. In

tree pruning, the measure used is the mean square error in the predictions made by the tree. Basically, the Reduced Error Pruning Tree ("REPT") is a Fast Decision Tree Learner and builds a decision tree based on information gain or reducing variance. Some studies using the REPTree algorithm include them [23].

d. J48 or C4.5. J48 is a machine learning algorithm that is widely used and is included in the decision tree algorithm. This algorithm builds a decision tree from a set of training data with the concept of entropy[19]. This algorithm differs from IDE3 in that it builds a decision tree, where J48 or C4.5 can accept both continuous and categorical attributes. [24]. This algorithm differs from IDE3 in that it builds a decision tree, where J48 or C4.5 can accept both continuous and categorical attributes [25].

e. Random Forest (RF). RF is an 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 [26]. The Random Forest algorithm was proposed by Breich in 2001[27]. Some anomaly detection studies using Random Forest include research conducted by [28].

2.4 Chi-Square Feature Selection Technique

Chi-square feature selection is a method to eliminate features that are not relevant to the statistical approach. Chi-square calculates the weight value between features and class. This method in the feature value by calculating the square statistical value according to the class [29]. Some studies that apply chi-square for feature selection include research [30]. Chi-square is calculated using Eq. 1.

$$x^2(f,c) = \frac{N*(WZ-XY)^2}{(W+x)+(W+Z)+(W+X)+(Y+Z)} \quad (1)$$

Where,

W = How many times the feature 't' and class label 'c' appears
X = How many times 't' is present without 'c'
Y = How many times 'c' is present without 't'
Z = How many times other than 'c' or 't' are present
N = Total number of records

2.5 SMOTE

SMOTE algorithm was proposed by [31] to solve problems with imbalanced data. Data imbalance occurs when classification categories are not evenly represented. Often under real conditions, the data is dominated by “normal” examples and very few “abnormal” or data” examples. SMOTE increases the number of minority data by creating new synthetic data by repeating the minority sample [13]. SMOTE uses an over-sampling approach, where the minority class is over-sampled by creating “synthetic” data. Several IDS studies using SMOTE, research [32].

2.6 Experiment Setup

All experiments were carried out on computer with the following specifications: Intel core i7 with 2.70 GHz processor, 8 GB RAM and running Windows 10 operating system. For feature selection, classification, and analysis, Weka 3.8 software with a heap size configuration of 3072 MB was used. Weka is a software that was first developed at the University of Waikato [33].

2.6 Performance Measurement

This study examines the performance of the proposed anomaly detection method. In several IDS and anomaly detection studies, to measure the performance of a detection system a measurement metric is used which consists of: (i) accuracy, (ii) true positive rate, (iii) false alarm rate, (iv) precision, (v) F-measure , and (vi) receiver operating characteristics (ROC) curve. As a basis for evaluating the “interesting accuracy using the Confusion matrix as in table 3 [34].

Table 3. Binary Confusion Matrix

		Prediction	
		Normal	Attack
Actual	Normal	TP	FP
	Attack	FN	TN

In the context of IDS Tab. 3 can be explained as follows:

a. FP (False Positive): defined as the actual number of normal detected as an attack
b. FN (False Negative): is defined as prediction error, whereas the actual attack is detected as normal
c. TP (True Positive): defined as predictive accuracy, actual normal is detected as normal
d. TN (True Negativity): defined as the actual attack detected as an attack

Based on the definition generated by the confusion matrix table, detection performance metrics are defined as follow.

a. Accuracy: defined as the level of closeness between categorization values and actual values, Often used to measure the effectiveness of classification algorithms, also known as Classification Rate (CR)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

b. Recall or Detection Rate (DR): which is defined as actual positive categorized correctly as a positive class, Also known as True Positive Rate (TPR) and Sensitivity

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

c. Precission: defined as a measure of the estimated probability of a correct positive prediction, also known as Positive Predictive Value (PPV)

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

d. False Positive Rate (FPR): which is defined as negative actual is categorized as a positive class, normal traffic is considered an attack, also known as the False Acceptance Rate (FAR) or fall-out.

$$False\ Positive\ Rate = \frac{FP}{FP+TN} \quad (5)$$

e. ROC (Receiving Operating Curve) : This curve is used to evaluate the performance of the classification algorithm [35]. The X-axis represents the FAR value and the y-axis represents the Sensitivity value.

3. RESULT AND DISCUSSION

3.1 Data Preparation

This work uses only 30% of the data from the MachineLearningCSV version of the CICIDS-2017 dataset. Table 4 depicts the dataset profile. Referring to the dataset profile, it can be seen that the dataset contains multi-class and high class-imbalanced data. There is a data imbalance, i.e.: the distribution of data for each class is not uniform. There are several minority classes such as XSS Web Attack and SQL Injection Web Attack data that each only has 8 rows of data, and Heartbleed attack only has 5 rows of data. Based on the percentage of the major data, these three attacks are very small. When compared with normal traffic, the overall percentage of attack traffic is very small. It can be concluded that the dataset represents high-class imbalance data.

Table 4. The 30% of CICIDS-2017 data profile

No.	Label	Number of Instances	Fraction from Total Instance
1	Benign	681,995	80.308
2	DDoS	38,427	4.525
3	PortScan	47,487	5.592
4	Bot	574	0.068
5	Web Attack Brute Force	455	0.054
6	Web Attack XSS	202	0.024
7	Web Attack Sql Injection	8	0.001
8	Infiltration	8	0.001
9	DoS Slowloris	1,739	0.206
10	DoS Slowhttptest	1,605	0.189
11	DoS Hulk	69,259	8.156
12	DoS GoldenEye	3,206	0.376
13	Heartbleed	5	0.001
14	FTP-Patator	2,422	0.285
15	SSH-Patator	1,831	0.216
	Total	**849,223**	

3.2 Feature Selection with Chi-Square

Chi-square technique is a statistical approach and is capable of eliminating irrelevant features using. Several previous studies have shown good performance of this feature selection technique. Table 5 presents the features selection results using Chi-Square.

Table 5. The selected features by Chi-Square

No.	Feature ID	Feature Names
1	41	Packet Length Std
2	13	Total Length of Bwd Packets
3	65	Subflow Bwd Bytes
4	42	Packet Length Variance
5	39	Max Packet Length
6	1	Bwd Packet Length Std
7	12	Total Length of Fwd Packets
8	63	Subflow Fwd Bytes
9	18	Bwd Packet Length Max
10	52	Average Packet Size
11	14	Fwd Packet Length Max
12	54	Avg Bwd Segment Size
13	20	Bwd Packet Length Mean
14	67	Init_Win_bytes_backward
15	40	Packet Length Mean
16	22	Flow IAT Max
17	17	Fwd Packet Length Std
18	26	Fwd IAT Max
19	9	Flow Duration
20	8	Destination Port
21	53	Avg Fwd Segment Size
22	16	Fwd Packet Length Mean
23	66	Init_Win_bytes_forward

This technique generates a list of features that are considered relevant. Basically, Chi-square also analyzes features based on the weight of the feature's dependence on other features and on class. Furthermore, the weight values are sorted from the highest to the lowest weight. Similar to Information Gain, in Chi-Square expert intervention is also required to determine the minimum weight value which is the basis for determining the number of relevant features. To determine the weight, repeated testing is carried out by reducing the features with the lowest weight, so that a set of features is the most ideal for use in detecting attacks. Thus, in this study 23 features are the most ideal number of features for detecting attacks.

3.3 SMOTE Implementation

As explained in the previous section, this study uses SMOTE to solve the problem of imbalanced data. SMOTE does oversample for classes that have the least amount of data. In this study, based on the data used, the data for the Heartbleed attack type class had the least amount of data (minor) compared to other attack classes. In the experiment, SMOTE was implemented with oversampling of 100%. Table 6 shows the data profile after applying SMOTE.

Table 6. Results of SMOTE implementation

No.	Label	Number of Instances
1	Benign	681,995
2	DDoS	38,427
3	PortScan	47,487
4	Bot	574
5	Web Attack Brute Force	455
6	Web Attack XSS	202
7	Web Attack Sql Injection	8
8	Infiltration	8
9	DoS Slowloris	1,739
10	DoS Slowhttptest	1,605
11	DoS Hulk	69,259
12	DoS GoldenEye	3,206
13	Heartbleed	10
14	FTP-Patator	2,422
15	SSH-Patator	1,831

After SMOTE was applied, the amount of Heartbleed attack data which was previously 5 now changes to 10.

3.4 Detection Performance

After applying the SMOTE technique to the imbalanced data, the data is used to detect normal traffic and attacks traffic with the following classification algorithms: Naïve Bayes, Bayes Network, J48, REPTree, and Adaboost. Then, the detection performance is measured for the following metrics: TPR, FPR, Precision, F-Measure, and ROC values.

3.4.1 Experiment with Naïve Bayes Classification Algorithm

Naïve Bayes is a fairly simple machine learning algorithm because it relies on probability and statistics. The classification for the selected features in Weka-3-8 software, is implemented in the following command:

```
Started weka.classifiers.bayes.NaiveBayes
Command: weka.classifiers.bayes.NaiveBayes
Finished weka.classifiers.bayes.NaiveBayes
```

In Table 7, the results of detection performance testing using the Naïve Bayes algorithm are presented. TPR values show that Naïve Bayes are only able to detect very well the traffic of Portscan attacks, Web Attack XSS, Infiltration, HeartBleed and FPT-Patator. Naïve Bayes is also quite good at detecting DDoS and DoS GoldenEye attacks. Several other types of traffic can also be detected but with a low TPR. Naïve Bayes, has not done well at detecting Web Brute Force attack.

Table 7. Detection Performance Using Naïve Bayes

Class	TP Rate	FP Rate	Precision	F-Measure	ROC
BENIGN	0.344	0.001	1.000	0.512	0.960
DDoS	0.782	0.002	0.954	0.860	0.998
PortScan	0.991	0.225	0.207	0.343	0.986
Bot	0.352	0.298	0.001	0.002	0.686
Web Attack Brute Force	0.004	0.004	0.001	0.001	0.983
Web Attack XSS	0.955	0.004	0.050	0.094	0.983
Web Attack Sql Injection	0.500	0.000	0.013	0.025	0.999
Infiltration	1.000	0.010	0.001	0.002	0.996
DoS slowloris	0.525	0.006	0.157	0.241	0.976
DoS Slowhttpte	0.189	0.011	0.032	0.055	0.981
DoS Hulk	0.597	0.007	0.889	0.715	0.966
DoS GoldenEye	0.700	0.020	0.115	0.198	0.967
Heartbleed	1.000	0.000	0.769	0.870	1.000
FTP-Patator	0.998	0.001	0.827	0.904	1.000
SSH-Patator	0.515	0.002	0.398	0.449	0.998

For FPR values, it is often used to detect false alarms or where a positive classification condition is doubtful or incorrect. The experimental results show that the detection results of PortScan and Bot attacks are higher than other types of traffic.

The precision value is often used to measure the accuracy of the classification results; the results show very variation precision values. For Benign, DDoS, DoS Hulk, HeartBleed, and FTP-Patator traffics show good precision values, i.e.: more than 0.700.

The F-measure value is often used to assess the accuracy of the classification results by considering Precision and Recall. The experimental results exhibit a fairly high F-Measure value is obtained for DDoS, DoS Hulk, HeartBleed, and FTP-Patator attacks. The ROC value is also used to see the performance of the classification engine, and the results show excellent performance of Naïve Bayes, although for Bot attack, it has the lowest value compared to other types of traffic.

By paying attention to the five measurement matrices, by applying SMOTE to high-class imbalance data, Naïve Bayes algorithm is able to detect DDoS, HeartBleed, and FTP-Patator attack traffic types.

3.4.2 Experiment with Bayes Network Classification Algorithm

The classification for the selected features is implemented in Weka-3-8 software in the following command:

```
Started weka.classifiers.bayes.BayesNet
Command: weka.classifiers.bayes.BayesNet -D -Q
weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E
weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Finished weka.classifiers.bayes.BayesNet
```

Table 8 presents the results of the Bayes Network performance. Considering the values of TPR, FPR, Precision, F-Measure, and ROC, the features selection results using Bayes Network and Chi-square algorithm are able to classify well Benign, DDoS, PortScan, DoS Slowloris, DoS Hulk, DoS GoldenEye, FTP-Patator and SSH-Patator traffic types.

Table 8. Detection Performance Using Bayes Network

Class	TP Rate	FP Rate	Precision	F-Measure	ROC
BENIGN	0.897	0.001	1.000	0.945	0.997
DDoS	0.998	0.000	0.997	0.997	1.000
PortScan	0.995	0.001	0.988	0.991	1.000
Bot	1.000	0.026	0.025	0.049	1.000
Web Attack Brute Force	0.996	0.042	0.013	0.025	0.996
Web Attack XSS	0.040	0.002	0.004	0.007	0.981
Web Attack Sql Injection	0.500	0.001	0.006	0.011	0.878
Infiltration	0.875	0.002	0.005	0.009	0.999
DoS slowloris	0.994	0.000	0.985	0.989	1.000
DoS Slowhttpte	0.990	0.009	0.174	0.296	0.999
DoS Hulk	0.990	0.000	0.995	0.993	1.000
DoS GoldenEye	0.807	0.000	0.998	0.892	1.000
Heartbleed	1.000	0.002	0.007	0.013	1.000
FTP-Patator	0.998	0.000	0.949	0.973	1.000
SSH-Patator	0.998	0.000	0.820	0.900	1.000

3.4.3 Experiment with J48 Classification Algorithm

The classification for the selected features is implemented in Weka-3-8 software in the following command:

```
Started weka.classifiers.trees.J48
Command: weka.classifiers.trees.J48 -C 0.25 -M 2
Finished weka.classifiers.trees.J48
```

Table 9 presents the results of the traffic detection experiment using the J48 algorithm. The experimental results show that using the selected features and Chi-square algorithm, the J48 algorithm is able to detect almost all traffics. Although some types of attacks have low TPR values, i.e.: Web Attack XSS and Web Attack SQL, overall J48 algorithm goes beyond Naïve Bayes and Bayes Network.

Table 9. Detection Performance Using J48

Class	TP Rate	FP Rate	Precision	F-Measure	ROC
BENIGN	0.999	0.001	1.000	0.999	1.000
DDoS	0.999	0.000	1.000	1.000	1.000
PortScan	1.000	0.000	0.994	0.997	1.000
Bot	0.735	0.000	0.981	0.841	1.000
Web Attack Brute Force	0.899	0.000	0.773	0.831	1.000
Web Attack XSS	0.406	0.000	0.678	0.508	1.000
Web Attack Sql Injection	0.500	0.000	1.000	0.667	1.000
Infiltration	0.625	0.000	1.000	0.769	0.998
DoS slowloris	0.994	0.000	0.999	0.997	1.000
DoS Slowhttpte	0.993	0.000	0.995	0.994	1.000
DoS Hulk	1.000	0.000	0.996	0.998	1.000
DoS GoldenEye	0.998	0.000	0.995	0.996	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	0.999	0.000	0.999	0.999	1.000

3.4.4 Experiment with REPTree Classification Algorithm

The classification for the selected features is implemented in Weka-3-8 software in the following command:

```
Started weka.classifiers.trees.REPTree
Command: weka.classifiers.trees.REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0
Finished weka.classifiers.trees.REPTree
```

The results of REPTree's performance experiment in detecting traffic are presented in Table 10 The performance of the REPTree algorithm is almost equal to that of the J48 algorithm, where REPTree is able to detect all traffics, except for the traffic of the XSS Web attack and the Web sql Injection attack.

Table 10. Detection Performance Using REPTree

Class	TP Rate	FP Rate	Precision	F-Measure	ROC
BENIGN	0.999	0.002	0.999	0.999	1.000
DDoS	0.999	0.000	1.000	0.999	1.000
PortScan	0.999	0.000	0.994	0.997	1.000
Bot	0.746	0.000	0.973	0.844	1.000
Web Attack Brute Force	0.824	0.000	0.752	0.786	1.000
Web Attack XSS	0.322	0.000	0.684	0.438	0.999
Web Attack Sql Injection	0.000	0.000	?	?	0.992
Infiltration	1.000	0.000	1.000	1.000	1.000
DoS slowloris	0.994	0.000	0.995	0.995	1.000
DoS Slowhttpte	0.992	0.000	0.992	0.992	1.000
DoS Hulk	0.999	0.000	0.997	0.998	1.000
DoS GoldenEye	0.998	0.000	0.994	0.996	1.000
Heartbleed	0.700	0.000	1.000	0.824	0.999
FTP-Patator	0.998	0.000	0.999	0.999	1.000
SSH-Patator	0.999	0.000	0.998	0.999	1.000

3.4.5 Experiment with Adaboost Classification Algorithm

The classification for the selected features is implemented in Weka-3-8 software in the following command:

```
Started weka.classifiers.meta.AdaBoostM1
Command: weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W
weka.classifiers.trees.DecisionStump
Finished weka.classifiers.meta.AdaBoostM1
```

All classification experiments use training set mode which means using all data as input.

The results of the Adaboost performance experiment are presented in Table 11 The experimental results show that with the features of the Chi-square selection results, the Adaboost algorithm has not worked well. Only two types of traffic can be detected, i.e.: Benign and DoS Hulk, and even then, the FPR value is quite high compared to the performance of other techniques.

Table 11. Detection Performance Using Adaboost

Class	TP Rate	FP Rate	Precision	F-Measure	ROC
BENIGN	0.990	0.571	0.876	0.930	0.882
DDoS	0.000	0.000	?	?	0.682
PortScan	0.000	0.000	?	?	0.929
Bot	0.000	0.000	?	?	0.485
Web Attack Brute Force	0.000	0.000	?	?	0.822
Web Attack XSS	0.000	0.000	?	?	0.853
Web Attack Sql Injection	0.000	0.000	?	?	0.611
Infiltration	0.000	0.000	?	?	0.361
DoS slowloris	0.000	0.000	?	?	0.500
DoS Slowhttpte	0.000	0.000	?	?	0.639
DoS Hulk	0.665	0.042	0.587	0.624	0.918
DoS GoldenEye	0.000	0.000	?	?	0.736
Heartbleed	0.000	0.000	?	?	0.861
FTP-Patator	0.000	0.000	?	?	0.361
SSH-Patator	0.000	0.000	?	?	0.602

3.4.6 Experiment with Random Forest Classification Algorithm

The classification for the selected features is implemented in Weka-3-8 software in the following command:

```
Started weka.classifiers.trees.RandomForest
Command: weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0
-M 1.0 -V 0.001 -S 1
Finished weka.classifiers.trees.RandomForest
```

The traffic classification performance with Random Forest is presented in Tab. 12 The experimental results show that with the features of the Chi-Square selection, Random Forest is able to detect well each type of traffics, which can be seen from the values of TPR, FPR, Precision, F-Measure, and ROC of each traffic.

Table 12. Performance of Detection Using Random Forest

Class	TP Rate	FP Rate	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	0.999	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

3.5 Comparing Detection Accuracy

To see the reliability of the proposed method, a comparison of TPR, FPR, Precision, F-Measure, ROC, and Accuracy of each state-of-the-art method was compared with the proposed method.

Figure 2 presents the TPR for each method. The comparison results show that the proposed method has a TPR value that is superior to other methods. It can be concluded, the proposed method has the ability to detect normal traffic types and attack traffic on the dataset. In addition to having an excellent detection rate, the proposed method also has a low false alarm rate, as presented in Figure 3.

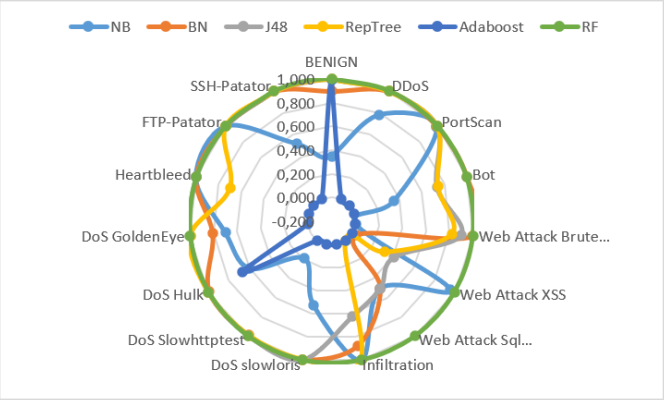

Figure 2. Comparing TPR

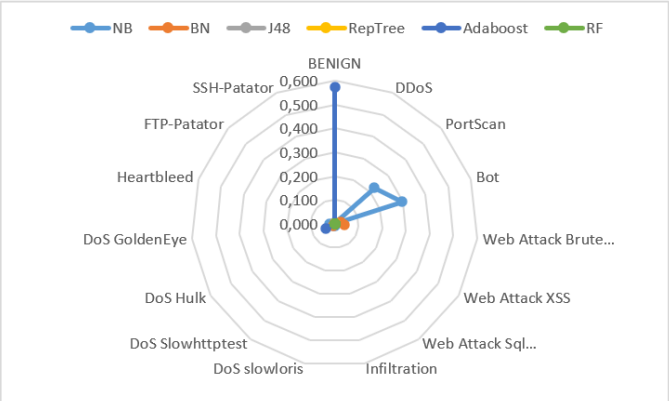

Figure 3. Comparing FPR

The reliability of the performance of the proposed method to detect attacks on high-dimensional and imbalanced data can also be seen in the results of the comparison precision value in Figure 4, the F-Measure comparison in Figure 5, and the comparison of ROC values in Figure 6. The comparison results show that the performance of the proposed method outperforms state-of-the-art methods. In addition, based on the comparison of accuracy in Figure 7, the proposed method has an accuracy of 99.98%, where this accuracy value is the highest when compared to state-of-the-art methods.

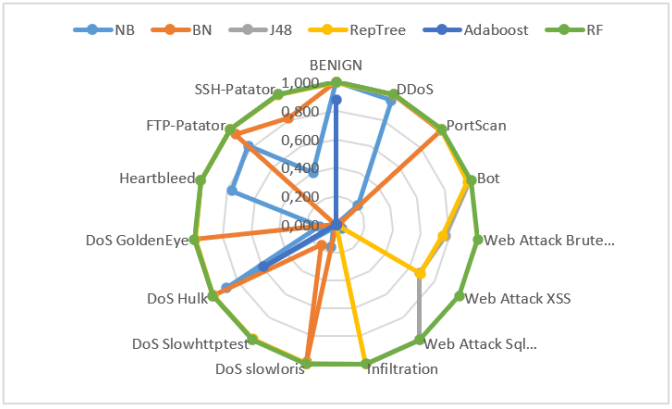

Figure 4. Comparing Precision

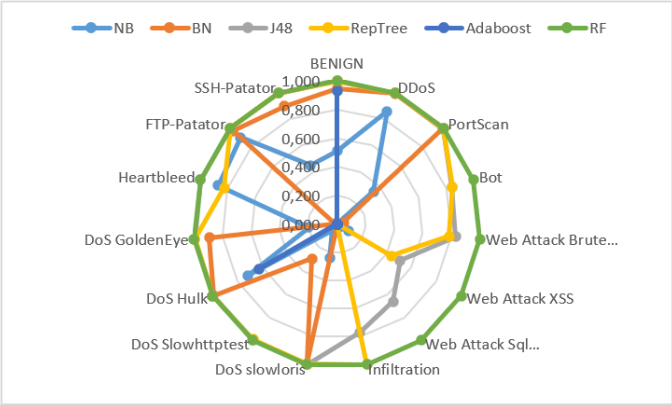

Figure 5. Comparing F-Measure

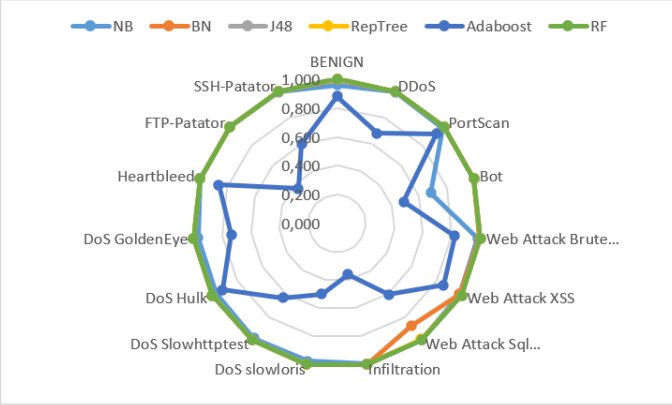

Figure 6. Comparing ROC

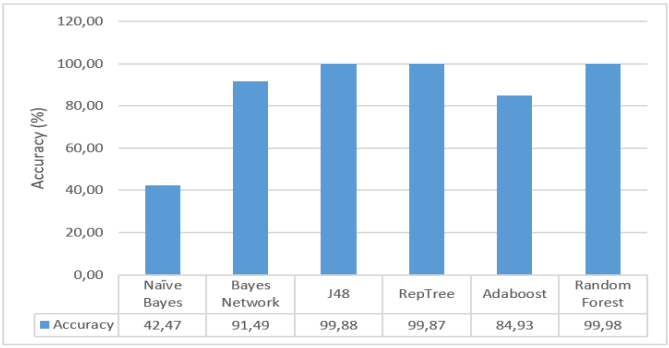

Figure 7. Comparing Accuracy

3.6 Comparison with Other Works

In addition to making comparisons with state-of-the-art methods, we also make comparisons of the proposed method with previous research works. This comparison aims to obtain external validation of the proposed method; Table 13 presents comparison measurement results of the proposed method against previous research works.

Table 13. Comparison with Previous Research Works

Authors	Techniques/ Methods	Dataset	Accuracy
(Ahmim et al.. 2019) [36]	The combination of three classification algorithms. namely REP Tree. JRip. and Forest PA	CICDS-2017	96.66%
(Abdulhammed et al. 2019) [37]	Dimensionality reduction using an auto-encoder and PCA. Validation using Random Forest	CICIDS-2017	99.60%
(Ustebay et al.) [38]	Recursive elimination of features using random forest. and Deep Learning Multilayer Perceptron (DMLP)	CICIDS-2017	91.00%
Proposed Method	SMOTE+Chi-Square + Random Forest	CICIDS-2017	99.98%

Based on the data in the table, the proposed method has a superior performance compared to previous studies, with an accuracy value of 99.98%. This accuracy exceeds the method proposed in Ahmin et al. [36], Abdulhammed et al. [37], and Ustebay et al. [38].

4. CONCLUSION

This study proposes an ensemble method for optimizing attack detection on high-dimensional and high-class imbalance data. The proposed ensemble method is a combination of feature selection techniques with state-of-the-art classification algorithms, i.e.: Naïve Bayes, Bayes Network, J48, REPTree, and Adaboost. Correlation-Based with Chi-Square feature selection technique is used to produce relevant features. The feature selection process generates 23 features, which then are used to detect normal and attacks traffic. This study also expereimented the SMOTE technique to overcome the problem of high-class imbalance in the CICIDS-2017 dataset. The CICIDS-2017 dataset used is an up-to-date dataset that represents modern network traffic. In this research, random forest classification algorithm is proposed to identify attack and normal traffic. The test results show that the proposed method has superior performance. The proposed method is also compared with state-of-the-art methods and previous studies. The comparison results show that the proposed method has superior performance than state-of-the-art methods and previous studies. Although chi-square is able to provide the best feature recommendations based on statistical tests, expert intervention is still needed to eliminate features. The next research will focus on developing automated feature selection techniques. In addition, with the rapid development of communication networks, and the possibility of the emergence of new types of attacks, a detection system must have the adaptability to recognize new types of attacks. This is also a new challenge in future IDS research.

REFERENCES

[1] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, “Machine learning on big data: Opportunities and challenges,” Neurocomputing, vol. 237, pp. 350–361, 2017, doi: 10.1016/j.neucom.2017.01.026.
[2] S. Maldonado and J. López, “Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification,” Applied Soft Computing Journal, vol. 67, pp. 94–105, 2018, doi: 10.1016/j.asoc.2018.02.051.
[3] M. W. Mwadulo, “A Review on Feature Selection Methods For Classification Tasks,” vol. 5, no. 6, pp. 395–402, 2016.
[4] Sheena, K. Kumar, and G. Kumar, “Analysis of Feature Selection Techniques: A Data Mining Approach,” International Conference on Engineering & Technology, vol. 4, no. Icaet, pp. 17–21, 2016, [Online]. Available: https://pdfs.semanticscholar.org/e6cf/aaea75a76aff400b8edc13ea402f445cd5ad.pdf
[5] T. A. Alhaj, M. M. Siraj, A. Zainal, H. T. Elshoush, and F. Elhaj, “Feature selection using information gain for improved structural-based alert correlation,” PLoS ONE, vol. 11, no. 11, pp. 1–18, 2016, doi: 10.1371/journal.pone.0166017.

[6] O. Isaiah, A. Olutola, and O. Olayemi, "Feature or Attribute Extraction for Intrusion Detection System using Gain Ratio and Principal Component Analysis (PCA)," Communications on Applied Electronics, vol. 4, no. 3, pp. 1–4, 2016, doi: 10.5120/cae2016652032.
[7] S. Bahl and D. Dahiya, "Enhanced Intrusion Detection System for Detecting Rare Class Attacks using Correlation based Dimensionality Reduction Technique," vol. 9, no. March, 2016, doi: 10.17485/ijst/2016/v9i11/84277.
[8] V. Vijayakumar and V. Neelanarayanan, "Intrusion Detection Model Using Chi Square Feature Selection and Modified Naïve Bayes Classifier," Smart Innovation, Systems and Technologies, vol. 49. Springer International Publishing Switzerland, pp. v–vii, 2016. doi: 10.1007/978-3-319-30348-2.
[9] I. S. Thaseen, C. A. Kumar, and A. Ahmad, "Integrated Intrusion Detection Model Using Chi-Square Feature Selection and Ensemble of Classifiers," Arabian Journal for Science and Engineering, vol. 44, no. 4, pp. 3357–3368, 2019, doi: 10.1007/s13369-018-3507-5.
[10] S. Rodda and U. S. R. Erothi, "Class imbalance problem in the Network Intrusion Detection Systems," International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016, pp. 2685–2688, 2016, doi: 10.1109/ICEEOT.2016.7755181.
[11] Q. Wang, Z. Luo, J. Huang, Y. Feng, and Z. Liu, "A Novel Ensemble Method for Imbalanced Data Learning," Computational Intelligence and Neuroscience, vol. 2017, pp. 1–11, 2017.
[12] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," Expert Systems with Applications, vol. 73, pp. 220–239, 2017, doi: 10.1016/j.eswa.2016.12.035.
[13] M. Reza, S. Miri, and R. Javidan, "A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset," International Journal of Advanced Computer Science and Applications, vol. 7, no. 6, pp. 20–25, 2016, doi: 10.14569/ijacsa.2016.070603.
[14] B. Yan, G. Han, M. Sun, and S. Ye, "A novel region adaptive SMOTE algorithm for intrusion detection on imbalanced problem," 2017 3rd IEEE International Conference on Computer and Communications, ICCC 2017, vol. 2018-Janua, pp. 1281–1286, 2018, doi: 10.1109/CompComm.2017.8322749.
[15] B. Yan and G. Han, "LA-GRU: Building Combined Intrusion Detection Model Based on Imbalanced Learning and Gated Recurrent Unit Neural Network," Security and Communication Networks, vol. 2018, 2018, doi: 10.1155/2018/6026878.
[16] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems," Applied Intelligence, 2020, doi: 10.1007/s10489-020-01886-y.
[17] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy, vol. 2018-Janua, no. Cic, pp. 108–116, 2018, doi: 10.5220/0006639801080116.
[18] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," International Journal of Engineering and Technology(UAE), vol. 7, no. 3.24 Special Issue 24, pp. 479–482, 2018.
[19] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys & Tutorials, vol. PP, no. 99, p. 1, 2015, doi: 10.1109/COMST.2015.2494502.
[20] K. Goeschel, "Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis," Conference Proceedings - IEEE SOUTHEASTCON, vol. 2016-July, 2016, doi: 10.1109/SECON.2016.7506774.
[21] B. Dhruba K and K. Jugal K, Network Anomaly Detection A Machine Learning Perspective. 2014.
[22] M. Reazul, A. Rahman, and T. Samad, "A Network Intrusion Detection Framework based on Bayesian Network using Wrapper Approach," International Journal of Computer Applications, vol. 166, no. 4, pp. 13–17, 2017, doi: 10.5120/ijca2017913992.
[23] T. Ait Tchakoucht and M. Ezziyyani, "Building a fast intrusion detection system for high-speed-networks: Probe and dos attacks detection," Procedia Computer Science, vol. 127, pp. 521–530, 2018, doi: 10.1016/j.procs.2018.01.151.
[24] S. Aljawarneh, M. B. Yassein, and M. Aljundi, "An enhanced J48 classification algorithm for the anomaly intrusion detection systems," Cluster Computing, pp. 1–17, 2017, doi: 10.1007/s10586-017-1109-8.
[25] A. P. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading k-Means clustering and C4.5 decision tree algorithm," Procedia Engineering, vol. 30, no. 2011, pp. 174–182, 2012, doi: 10.1016/j.proeng.2012.01.849.
[26] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems). 2011.
[27] M. C. Belavagi and B. Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," Procedia Computer Science, vol. 89, pp. 117–123, 2016, doi: 10.1016/j.procs.2016.06.016.
[28] 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 International Conference Proceeding Series, pp. 77–81, 2018, doi: 10.1145/3199478.3199489.
[29] J. Novaković, P. Strbac, and D. Bulatović, "Toward optimal feature selection using ranking methods and classification algorithms," Yugoslav Journal of Operations Research, vol. 21, no. 1, pp. 119–135, 2011, doi: 10.2298/YJOR1101119N.
[30] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid Intelligent Intrusion Detection Scheme," Soft computing in industrial applications, pp. 293–303, 2011.
[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
[32] Y. T. Y. Wei Chang Yeh and C. M. Lai, "A Hybrid Simplified Swarm Optimization Method for Imbalanced Data Feature Selection," Australian Academy of Business and Economics Review, vol. 2, no. 3, pp. 19–21, 2016.
[33] T. Garg and S. S. Khurana, "Comparison of classification techniques for intrusion detection dataset using WEKA," International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2014, 2014, doi: 10.1109/ICRAIE.2014.6909184.
[34] R. Goel, A. Sardana, and R. C. Joshi, "Parallel Misuse and Anomaly Detection Model," vol. 14, no. 4, pp. 211–222, 2012.
[35] D. Summeet and D. Xian, Data Mining and Machine Learning in Cybersecurity. CRC Press, 2011.
[36] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A Novel Hierarchical Intrusion Detection System based on Decision Tree and Rules-based Models," 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, pp. 228–233, 2019.

[37] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," Electronics (Switzerland). MPDI, vol. 8, no. 3, p. 322, 2019, doi: 10.3390/electronics8030322.
[38] S. Ustebay, Z. Turgut, and M. A. Aydin, "Intrusion Detection System with Recursive Feature Elimination by Using Random Forest and Deep Learning Classifier," International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, IBIGDELFT 2018 - Proceedings, pp. 71–76, 2019, doi: 10.1109/IBIGDELFT.2018.8625318.