International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 12, Number 2, pages 467–492, July 2022

Handling class Imbalance problem in Intrusion Detection System based on deep learning

Mariama MBOW
Department of Informatics, Kyushu University
744 Motooka Nishi-ku, Fukuoka, 819-0395, Japan
(*corresponding author) mmariamamambow@gmail.com

Hiroshi KOIDE
Research Institute for Information Technology, Cyber Security Center, Kyushu University
744 Motooka Nishi-ku, Fukuoka, 819-0395, Japan
koide@cc.kyushu-u.ac.jp

Kouichi SAKURAI Department of Informatics
Graduate School of Information Science and Electrical Engineering, Kyushu University
744 Motooka Nishi-ku, Fukuoka, 819-0395, Japan
sakurai@inf.kyushu-u.ac.jp

Received: February 15, 2022
Revised: May 5, 2022
Accepted: June 5, 2022
Communicated by Yasuyuki Nogami

Abstract

Network intrusion detection system(NIDS) is the most used tool to detect malicious network activities. The NIDS has achieved in the recent years promising results for detecting known and novel attacks, with the adoption of deep learning. However, these NIDSs still have shortcomings. Most of the datasets used for NIDS are highly imbalanced, where the number of samples that belong to normal traffic is much larger than the attack traffic. The problem of imbalanced class skews the results. It limits the deep learning classifier's performance for minority classes by misleading the classifier to be biased in favor of the majority class. To improve the detection rate for minority classes while ensuring efficiency, this study proposes a hybrid approach to handle the imbalance problem. This hybrid approach is a combination of oversampling with Synthetic Minority Over-Sampling (SMOTE) and Tomek link, an undersampling method to reduce noise. Additionally, this study uses two deep learning models such as Long Short-Term Memory Network (LSTM) and Convolutional Neural Network (CNN) to provide a better intrusion detection system. The advantage of our proposed model is tested in NSL-KDD, CICIDS2017 datasets. In addition, we evaluate the method in the most recent intrusion detection dataset, CICIDS2018 dataset. We use 10-fold cross validation in this work to train the learning models and an independent test set for evaluation. The experimental results show that in the multi-class classification with NSLKDD dataset, the proposed model reached an overall accuracy and Fscore of 99% and 99.0.2% respectively on LSTM, an overall accuracy and Fscore of 99.70% and 99.27% respectively for CNN. And with CICIDS2017 an overall accuracy and Fscore of 99.65% and 98 % respectively on LSTM, an overall accuracy and Fscore of 99.85% and 98.98% respectively for CNN. In CICIDS2018 the proposed method achieved an overall detection rate and Fscore of 95% and 94% respectively.

Keywords: Cybersecurity, Network intrusion detection, Class imbalance, Deep learning

1 Introduction

1.1 Background and motivation

The rapid development of communications and information technologies such as Internet of thing, intelligent devices, online shopping, etc, creates a constant increase of the number of network devices. This result in a growing network scope and at the same time a tremendous cyberattack risk. In the recent years, we have noticed many cyber attacks such as distributed denial of service or denial of service, brute force, botnet, cross-site scripting,etc [25]. Thus the threat of cyber intrusion has become far more serious than ever before [52], and creates an increasing concern in cyber security. According to CNBC[37], the cost of cyberattacks damages was $200,000 in 2019. This creates the need of a system that could detect such intrusion and evolves with time to meet the cybersecurity risk assessment. An intrusion is defined as any kind of unauthorised activities that aims to compromise the integrity, confidentiality, availability of security mechanisms of computer or network resources or to bypass them [15].

One of the most promising tool in cyber security for facing such threat and detecting malicious activities is an Intrusion Detection System (IDS). IDSs are generally categorized into Network IDS (NIDS) and Host IDS (HIDS) based on the data sources used to detect abnormal activities. The HIDS mainly inspect data that originate from system calls, logs etc., and can detect internal attacks that do not involve network traffic [26]. In contrast, Network-Based Intrusion Detection Systems (NIDS) collects and analyzes data captured directly from the network. This motivate us to leverage on the NIDS for detecting network attack traffic.

The detection approach used in NIDS is commonly classified into: anomaly detection and misuse detection system. In misuse detection (or signature-base detection) all known attacks are stored in a database and traffic are classified malicious if they match with those previously written in the database. This type of detection method is efficient for detecting known attacks but fails to detect unknown or zero day-attack which is a critical issue in modern networks. consequently, anomaly detection has gradually attracted researchers attention, due to its theoretical potential in the identification of known as well as unknown intrusions[32].

1.2 Related works in IDS development

The first intrusion detection system (IDS) has been proposed by Denning et al.[13]. Since then, it has received much interest in both academic and industrial domain to protect the network traffic. Many techniques have been used for developing NIDS including computing based, data mining based, statistical based, machine learning, cognitive based or knowledge based, user intention identification, etc.[24]

Machine learning techniques are one of the most used approaches due to their ability to learn patterns from data and differentiate between abnormal and normal traffic. Many classical machine learning techniques have been applied in IDS[29]. However, with the increasing network traffic and the diversification of attack categories, the traditional machine learning techniques also known as shallow learning are no longer suited to meet the demands of large-scale NIDS[52]. In recent years, Deep Learning (DL), branch of machine learning, has generated many interest in NIDS due to its ability to learn relevant features from massive data. Studies have shown that deep learning highly outperforms traditional methods[51] and can improve the efficiency of attack detection. Deep learning-based approaches used in NIDS include the deep neural network (DNN), the convolutional neural network (CNN), long short-term memory (LSTM), the recurrent neural network (RNN) etc.[32]. Literature review on machine learning and deep learning techniques are discussed in section II.

1.3 Challenging Issues

Although deep learning methods are improving the NIDS, they fail in detecting attacks with less traffic due to the class imbalance problem. Like the real-world network traffic, most state-of-the-art benchmark NIDS datasets are unbalanced (the attack traffic account for a minority compared to the normal traffic).The imbalanced data prevents the deep learning model from learning from minority classes. Therefore, this causes the minority attack classes difficult to be detected, then decreases the performance of the NIDS and leads to a high false alarm rate and a low detection rate. In recent NIDS works, insufficient attention has been paid to the problem of imbalanced data. However an efficient intrusion detection system should be able to identify all types of attack traffic[35]

1.4 Our contributions

This study aims to mitigate the class imbalanced problem for improving the detection rate of minority attack class in NIDS based deep learning. We propose a combination of data balancing method with deep learning algorithms. The main component of the data balancing is a hybrid resampling algorithm which combine SMOTE and Tomeklink [12]. SMOTE is an oversampling technique that increases minority samples while Tomek Link is an undersampling technique for cleaning up overlapping samples that occur with SMOTE. A detail is provided in section 3.

In summary, our main contributions are as follows:

1. We tackle the imbalance class problem with a combination of deep learning algorithm and a hybrid re-sampling method. The hybrid re-sampling method used is an oversampling with SMOTE and undersampling method TomekLink (SMOTETomek). SMOTE is an oversampling technique which increases minority samples with synthetic data while TomekLink is an undersampling technique used to clean overlapping samples occurred during oversampling.

2. To develop a robust intrusion detection system we leverage on two deep learning model the LSTM and 1D CNN model. We implement the NIDS on both deep learning algorithm. And then compare the best model. To reduce the model learning bias for a better generalization and low variance, we train our model in 10 folds stratified cross validation and test in an independent test set. We observe that these deep learning combined with the SMOTETomek improve the detection rate and decrease the false rate.

3. The proposed method is evaluated on a multiclass classification deep learning IDS using the benchmark datasets: NSLKDD, CICIDS2017 and finally we extend the methodology on the CICIDS2018. The motivation of choosing these datasets is NSLKDD is the most used IDS dataset, the CICIDS2017 and CICIDS2018 dataset not only contains up to date network attacks but also fulfils all the criteria of real-world attacks[33], moreover the CICIDS2018 is the most recent NIDS dataset[28] .

4. Finally we evaluate the performance of the proposed technique and compare with the baseline model and related research works. The experimental result shows that our proposed work outperform most of the proposed IDS.

1.5 Comparison with existing results

As an extended version of our previous works[30], In this study we implement the method in three dataset for a generalization purpose and provide an analysis on the impact of the imbalanced network traffic. We first compare our experimental results with the baseline result which do not handle the imbalance problem. The experimental findings showed our proposed method can detect minority classes not detected in the baseline result. Therefore the performance of the proposed method was better than the model classified without addressing the imbalanced problem. We finally, compare the proposed method with existent works in the literature. The experimental results prove the effectiveness of our proposed method. The result show our model outperforms most existing state-of-the-art models. Therefore, the proposed method can improve the detection of minority attack class and reduce false alarm rate. In term of false alarm rate, our model gives a better result mostly with the CNN.

2 Literature review

Many progress have been done in NIDS with the adoption of artificial intelligence such as machine learning and deep learning techniques. In this section we discuss related works of these techniques in NIDS.

2.1 Machine Learning

Machine learning (ML) techniques are the predominant approach used in anomly based NIDS because of their effectiveness of being able to differentiate between abnormal and normal traffic. Traditional machine learning models such as Random Forest, Decision Tree(DT), Support Vector Machine (SVM) etc., have been widely used in NIDS. In [27] Firat et al. used SVM, K-Nearest neighbor (KNN), DT algorithms to develop an IDS systems in UNSW-NB15, CIDDS-001, NSL-KDD , ISCX-2012 and CSE-CIC IDS-2018 data sets. They evaluated their model using 10 folds cross validation. Decision tree techniques like CART as classifier have been proposed in[34]for classification of attacks.Ahmad et al. [2] propose an Adaboost based decision tree classifier for binary detection in UNSW-NB15 dataset. In [17] Random-Forest (RF) classifier has been proposed using NSLKDD dataset. In [41]Soheily et al.propose a hybrid IDS based on K-means and RF(KM-RF). Ensemble model of machine learning classifiers have also been proposed for IDS. A detailed review on IDSs with traditional machine learning is given in [50, 3].
However, traditional models cannot effectively solve the massive data classification problem that arises in the face of a real network application environment[51]. One of the reason is ML-based IDS relies heavily on feature engineering to learn useful information from the network traffic. Contrary, Deep learning do not rely on feature engineering and can automatically learn complex features from the raw data due to the deep structure[50]. Thus making deep learning a suitable approach for massive data.

2.2 Deep learning

In the recent years, IDS has experienced a rapid improvement with the adoption of deep learning technologies. In[51], the authors propose a deep learning approach for intrusion detection using recurrent neural networks(RNN-IDS) on the benchmark NSL-KDD dataset. They compared their model with traditional machine learning such as ANN,SVM, RF, J48 and other methods in their literature. The RNN-IDS achieves an accuracy of 83.28% and 81.29% in binary and multiclass classification respectively and outperform the compared traditional machine learning models.

Osama Faker et al. [16] integrated Deep Learning and Big Data techniques to enhance the performance of intrusion detection systems and propose and NIDS based k-means homogeneity metric for future selection. They used Deep Feed-Forward Neural Network (DNN) and traditional ML models which are two ensemble techniques, Gradient Boosting Tree (GBT) and Random Forest. They evaluate the proposed method with 5-fold cross validation in UNSW NB15 and CICIDS2017 datasets. For multiclass classification, they remove the normal traffic and only the attack traffic are used to evaluate the proposed method. In their experiment DNN achieved the highest accuracy of 99.56%. However, they did not report the false alarm and detection rate. And only the result of the cross validation was reported but did not set an independent test data after cross validation. In [32] Long Short-Term Memory (LSTM) combined with genetic algorithm (GA) for optimal feature selection is studied on NSLK-KDD. The results show that their proposed method achieved an accuracy of 93.88%. However, they did not report the detection rate. Kaur et al. [25] develop an image-based deep learning model, a 2D CNN to implement an NIDS in CICIDS2017 and CICIDS2018 datasets. In their experiment they remove attack classes with lesser traffic to have a balanced data. In both datasets they achieve an accuracy of 99% and 97% in CICIDS 2017 and CICIDS 2018 respectively in the testing accuracy. However, the experimental result shows the test accuracy is greater than the training accuracy which is an underfitting problem. Moreover, they removed the minority attack classes to prevent the imbalance problem. In [8], authors study the implementation of network intrusion detection system in various deep learning framework: Tensorflow, pytorch, fast.ai etc. with the CICIDS2018 benchmark dataset. In their experiment in multi-class classification, they achieved an accuracy of 99%. However their method fails to detect minority attack classes which means the model was biased towards the majority class to achieve a high accuracy, due to the highly imbalanced class of the dataset.

Although Deep learning techniques have shown improvement in the development of IDS, the proposed NIDSs fail to achieve a good performance(low false alarm rate and high detection rate). One of the reasons is most of those works ignore the imbalanced data in IDS datasets. Leevy et al. [28] studied a survey on NIDS based on the CICIDS2018; in their observation, authors mentioned most of the works presented a high accuracy and did not address the class imbalance which biased the results and they are not being able to detect the attack with minority traffic.

2.3 Imbalanced Data

A dataset is said to be imbalanced when some classes are very underrepresented compared to others. This uneven distribution makes the learning algorithms less effective, by degrading the detection rate especially in predicting minority class examples[49]. Like the real-world network traffic, IDS datasets contain very little attack traffic compared to normal traffic, which creates an unbalanced classification problem. However, the imbalanced problem in intrusion detection remains largely unexplored and is still counted in the list of existing challenges[52, 1]. Awad et al.[6]proposed a stratified sampling with weighted Extreme Learning Machine (ELM) on UNB ISCX2012 dataset. Their method achieve a good accuracy and f score however they did not extend their work in other datasets for a generalization. Zhang et al.[52]studied the combination of SMOTE oversampling and clustering under-sampling based on Gaussian mixture model on CICIDS2017. Although their model achieve a good performance, they did not present the false alarm rate(FAR) which is an important metric for anomaly NIDS. In their work Zhu et al. [55]studied an improved NSGS-III called I-NSGA-III feature selection algorithm using bias-selection based on probabilities for solving the imbalance problem in NSL-KDD. Though, their studies work only on the improvement of the detection rate and did not present the accuracy and FAR. To handle the class imbalance problem in CICIDS2017, Abdulhammed et al. [1] proposed a uniform distribution based balancing(UDBB). Toupas et al.[46] propose SMOTE combined with Edited Nearest Neighbors (SMOTE-ENN) and Deep Neural Network (DNN)algorithm. However, they report only the result of the cross validation and did not test in an independent test set. Jiang et al.[23] improve the intrusion detection process in NSL-KDD by using random oversampling method for minority class and under sampling majority class. However, random(traditional) oversampling tend to create overfitting problem[11], moreover researchers present only the result of the accuracy. Zhang et al.[54] propose SMOTE-ENN algorithm in NSL-KDD.Even though they use the necessary metrics for testing the IDS performance, their proposed method yields a high false alarm rate. Sinha et al. [40]proposed a CNN-BiLSTM model in NSL-KDD and UNSW-NB15, with 10-fold cross validation and used the traditional oversampling method to balance the data. Their proposed model achieved better performance than many state-of-the-art Network Intrusion presented in their related work in NSL-KDD with an accuracy of 99.22% and detection rate of 98.88% in NSL-KDD. However, they show only the result of the cross validation and did not set an independent test data after cross validation. To handle imbalanced class problem in NIDS for industrial IoT, Zhang et al. [53] propose PWG-IDS which use the pretraining Wasserstein generative adversarial network with gradient penalty(WGAN-GP) for data generation on minority class sample. The pretrained is used to reduce the number of iteration and can also generate more realistic sample than GAN. The proposed method is implemented on NSL-KDD and CICIDS2018 dataset. For the classification algorithm they used LightGBM. To reduce the training time, authors selected a subset on the CICIDS208 dataset. The experimental results achieved an overall accuracy of 99% and 96% on NSLKDD and CICIDS2018 respectively. The pretraining mechanism improve the convergence time of GAN moreover it has improved the performance of their models. However, this method requires a separate pretaining for each dataset and lack of generalization capability. The problem generalization capability is on their future problem.Also authors did not present the result of the detection rate for the minority classes and the false alarm rate. Vu et al.[48] proposed

GAN to handle the imbalanced data in intrusion detection system using three benchmark dataset NSLKDD, UNSW-NB15 and CICIDS2017. their proposed method improved the NIDS performance compared to their baseline method. Although GAN can generate sample, it's convergence tends to be difficult when the number of class samples is very small. Moreover often the generated sample with GAN differs from the real sample distribution[53]. Gupta et al.[19] propose CSE-IDS. There method is a three layer NIDS. To handle the class imbalance, They propose a combination of cost sensitive Deep learning and ensemble learning algorithm with data level technique. Two data level approach were used Random Oversampling(ROS) and SVM-SMOTE. In the first layer the cost sensitive with DNN is used to perform a binary classification of normal and attack traffic. In the second layer, the XGBoost algorithm is used preceded by data oversampling. This layer is used to separate the majority attack classes with the minority attack classes. Finally in the third layer, the data is resampling then random forest algorithms is used to classify the different type of minority attack classes. Authors compare their proposed method with the related works presented in their paper. However their method outperforms the proposed approach.

Related works in NIDS that consider the imbalance problem highly improve the performance of IDS compared to others. However most of these works do not present a complete evaluation of the IDS performance. For instance works in [52, 55, 23] did not evaluate the false alarm rate which is an important metric for anomaly NIDS performance. Moreover most of the related works evaluated their methods in one dataset also none of the studies have analyzed the combination of SMOTE + TomekLink.

3 Proposed method

Our purpose is to implement an anomaly network intrusion system in imbalanced data that detects and classifies with high accuracy as well as high detection rate of each type of attack and moreover with low False alarm rate. Firstly we mitigate the problem of class imbalanced and then proceed on the attack detection.

To mitigate the imbalanced problem in IDS, we propose a combination of Deep learning algorithm with Synthetic Minority Over-sampling Technique (SMOTE) and Tomek link for oversampling and undersampling classes. SMOTE and Tomek link is a method proposed by Batista et al. [9] to solve the problem of SMOTE. By using SMOTE the generation of synthetic samples for minority class could be overlapping with the majority class and creating more false positive[11], they proposed an oversampling with an overlapping cleaning procedure called Tomek link. Therefore we adopt the SMOTETomek method in network traffic for balancing the training data and improve the performance of the detection rate of each attack class. In the best of our knowledge, SMOTETomek has not been tested in NIDS. This Tomek link method can significantly reduce the overlap caused by oversampling with SMOTE and then increase the performance of the IDS. To reduce bias and have a better generalization ability we train our model with 10 fold cross validation. In addition we use an independent test set to evaluate the effectiveness of our model on unseen data. Fig. 1 shows the architecture of our proposed method. It consist of mainly 3 steps: data preprocessing step, data balancing for the training data and detection. Additional detail is provided in the followings.

3.1 Datasets description

The NSL-KDD dataset [44] is a publicly available dataset proposed by Tavalee et al. as a refined version of the widely used KDD'99 dataset. They improved a number of shortcomings found in the KDD'99 by removing all the redundant records and partitioning the records into various difficulty level. The NSL-KDD dataset contains KDDTrain+, KDDTrain+_20 Percent, KDDTest+, and KDDTest_21. In this work, our entire dataset contains the KDDTrain+ and KDD Test+. The dataset contains 41 features fig. 13 and one(1) column of class label. Records of classes are grouped into four categories of attacks DoS, Probing,R2L and U2R attacks shown in Table 1, and the rest of records represent normal traffic flow. Though the dataset is outdated, it is still applied as a benchmark dataset for comparing different intrusion detection methods. However this dataset has imbalanced class where the normal class is the majority class with 51.88% of the entire dataset and the minority

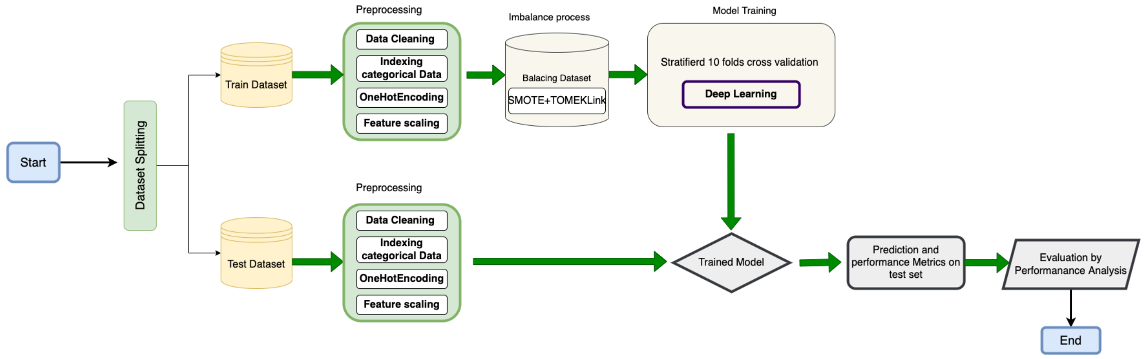

Figure 1: Architecture of our workflow model

classes like U2R and R2L represent 2.61% and 0.08% respectively of the entire dataset. we split the data in 80% training and 20% testing instances with a stratified splits to preserve the same percentage for each target class as in the full set provided in the dataset.The distribution of each class in the train and test set is shown in Table 2 with their prevalence ratio.

Table 1: Category of attacks

Class category	Attack Type
DoS	*mailbomb, neptune, apache2, back, land, pod, processtable, teardrop, smurf, udpstorm, worm*
Probe	*ipsweep, nmap, mscan, saint, portsweep, satan*
R2L	*guess_passwd, ftp_write, http-tunnel, multihop, imap, named, phf, sendmail, Snmpgetattack, spy,warezclient, snmpguess, warezmaster, , xsnoop, xlock*
U2R	*buffer_overflow, loadmodule, perl, ps, sqlattack,rootkit, xterm*

Table 2: Distribution of samples in each class in NSL KDD dataset

class Label	Number Total of instance	% w.r.t. total dataset instances
Normal	77054	51.88%
DoS	53387	35.95%
Probe	14077	9.48%
U2R	3880	2.61%
R2L	119	0.08%

The CICIDS2017 dataset[38] is a public available dataset proposed by the Canadian Cyber Security Institute. It contains network traffic samples with benign and 15 attack classes and it is divided into 8 different files. We merged the 8 files into a single one. The benign traffic account for 80.30% of the entire dataset and attack traffic account for 19.70% where the least attack sample account for 0.0003%. The dataset contains up to date network attacks and also fulfils all the criteria of real-world attacks for IDS [33]. It contains 80 features including the class label.The dataset is extremely imbalanced. In our experiment, to reduce training time, after cleaning the dataset, we first randomly reduce the benign, DoS Hulk and PortScan traffic. We then split the training and test data into 60%, 40% respectively with a stratified split. The distribution is shown in Table 3.

The CICIDS2018 dataset [39] is the most recent public [18] dataset proposed by the Canadian Cyber Security Institute. The CICIDS2018 is much larger than the CICIDS2017 and highly imbalanced. It contains 16,233,002 instances with benign and 15 attack classes and is distributed over 10 CSV files which can be downloaded from [18]. In the dataset, nine files consist of 79 independent features including label and the remaining consists of 84 independent features including label column. We merged the 10 files into a single one. Considering the computing resource overhead, after data cleaning we select a subset of the data while maintaining the imbalance problem. The distribution is shown in 4. We then split the training and test data into 60%, 40% respectively with a stratified split.

Table 3: Distribution of samples in each class in CICIDS2017 dataset

class Label	**Number total of instance**	**% w.r.t. total dataset instances**
BENIGN	227310	71.32%
DoS Hulk	23107	7.25%
PortScan	15893	4.02%
DDoS	12803	3.23%
DoS GoldenEye	10293	2.49%
FTP-Patator	7938	1.85%
SSH-Patator	5897	1.81%
DoS slowloris	5796	1.72%
DoS Slowhttptest	5499	3.4%
Bot	1966	0.62%
Web AttackBrute Force	1507	0.47%
Web Attack XSS	652	0.20%
Infiltration	36	0.011%
Web Attack Sql Injection	21	0.006%
Heartbleed	11	0.003%

3.2 Preprocessing

Data preprocessing is a critical step used to make the data understandable by the deep learning model. In this work the preprocessing consist 3 steps: cleaning, numericalization, normalization.

3.2.1 Data cleaning

The CICIDS2017 and CICIDS2018 datasets contains missing values(NaN) and infinity values. We fill the missing values by zero and replace values with infinity with the maximum value of their column in the CICIDS2017 and replace values with infinity with the maximum value of their column plus one in the CICIDS2018.

In CICIDS2018, the columns 'Flow ID', 'Src IP', 'Src Port', 'Dst IP' are present in one CSV file, therefore we remove these features. We remove 'Timestamp' feature also. After cleaning, the dataset consists of 79 features including the class label.

Table 4: Distribution of samples in each class in CICIDS2018 dataset

class Label	Number total of instance	% w.r.t. total dataset instances
BENIGN	100000	23.56%
DDOS attack-HOIC	68596	16.16%
DDOS attack-LOIC-HTTP	57613	13.57%
DoS attacks-Hulk	46189	10.88%
DoS attacks-GoldenEye	41504	9.78%
Bot	28615	6.74%
FTP-BruteForce	19333	4.56%
SSH-Bruteforce	18756	4.42%
Infiltration	16193	3.82%
DoS attacks-SlowHTTPTest	13987	3.30%
DoS attacks-Slowloris	10988	2.59%
DDOS attack-LOIC-UDP	1729	0.41%
Brute Force -Web	609	0.14%
Brute Force -XSS	230	0.05%
SQL Injection	87	0.02%

3.2.2 Numericalization

Deep learning works with numeric values, therefore categorical variables need to be converted into numerical. The NSLKDD dataset has 38 numerical features and 3 categorical features 'protocol','service' and 'flag'. We convert these categorical values into numeric using one-hot encoding. Therefore the transformation changes the features from 41 to 121 features. One-hot encoding is also applied to the class label then transforming the classification into 5-class classification problem: the 4 attacks and the normal classes.

In the CICIDS2017 all features are numeric. We applied the one-hot encoding to the class label to build a multiclass-classification.

In CICIDS2018 the feature types are object we consider all feature as numerical and convert their data type as float. Then, we applied the one-hot encoding to the class label to build a multiclass-classification.

3.2.3 Data scaling

Data scaling is an important pre-processsing step in deep learning that map the values of all numerical feature within a standard range. This process improve the performance of the learning model. There are different scales in the NIDS datasets. In our experiment we use the two most popular techniques used for scaling numerical data in NIDS dataset and compare their performance. Normalization with Max scaler (1) and StandardScaler (2) using the scikit-learn library. In our experimental findings, the best result has been achieved on LSTM with Standard scaler and on CNN with MinMax scaler.

$$x_{scaled} = \frac{x - Min(x)}{Max(x) - Min(x)} \tag{1}$$

$$x_{standard} = \frac{x - \mu}{\delta} \tag{2}$$

3.3 Stratified K-fold cross validation

Stratification is arranging data to preserve the same percentage for each target class as they appear in the whole dataset. This technique is especially important with imbalanced datasets to ensure that the same ratio of imbalance is maintained in training and test set. Cross-validation is a model validation technique applied to analyze the generalization capability of the model into an independent dataset. K-fold cross validation technique is a good approach to prevent overfitting and reduce the bias of machine learning techniques. The procedure involves splitting the training dataset into k folds. k-1 folds are used for training the model, and the holdout kth fold is used as the validation set. This process is repeated k times until each of the folds is given an opportunity to be used as the holdout validation set[22]. This paper uses stratified 10-fold cross validation for training the model[56].

3.4 Data balancing

As shown in Table 2, 3 and 4, the datasets have imbalanced data. For instance in CICIDS2017, the normal traffic having the majority class account for 71.32% of the entire dataset. In contrast the Heartbleed and Web Attack Sql Injection classes account for 0.003% and 0.006% respectively. In this situation, IDS model will be trained to be biased towards the more frequent traffic to maximize the overall accuracy rate. Therefore, we might have a high accuracy while minority attack are not detected. Hence it will deteriorate the performance of the IDS. For instance, Works in [8] achieved an accuracy of 99% in CICIDS2018, however, the detection rate of infiltration attack which has less traffic, was 0%. This condition is also known as the accuracy paradox where the accuracy value does not reflect the exact performance of the model[14]. Data level method such as oversampling or undersampling is a widely used method for imbalance problem [7]. Undersampling reduces sample from the majority class. This process might create a loss of important data. In contrast, Oversampling creates additional data in the minority class. SMOTE[11] solves the problem of overfitting that occurs with traditional random oversampling (ROS) method which create a copy of data. However, SMOTE method creates classes overlapping and can introduce additional noise[11]. We use the hybrid SMOTETomek method to overcome the problem with SMOTE.

3.4.1 SMOTE

The Synthetic Minority Oversampling Technique(SMOTE) oversamples minority classes by generating synthetic minority example along the line segments between the k minority class nearest neighbors[11]. By generating similar patterns to the existing minority points, SMOTE increases the performance of minority classes[36]. However, when generating similar examples, SMOTE does not take into consideration that neighboring examples may come from other classes. This can therefore create an overlapping of classes and introduce additional noise..

3.4.2 SMOTE + TL

Tomek link(TL) is defined as a pair of examples that belong to different classes, one from the majority the other from the minority and are each other's nearest neighbors[45]. The combination of SMOTE and Tomek link is used to resolve the problem of class overlapping in SMOTE. This technique works by, first applying SMOTE for oversampling the minority class, and then identify the Tomek links and remove them from the data set.

When using oversampling method, we should be careful about the process. By Oversampling the training set before the cross validation process similar patterns may appear in both training and validation set which will lead to an overoptimistic results[36]. Therefore We perform the imbalance process in the training sets during the cross validation at each iteration Fig. 2 shows our proposed imbalance processing method during the training.

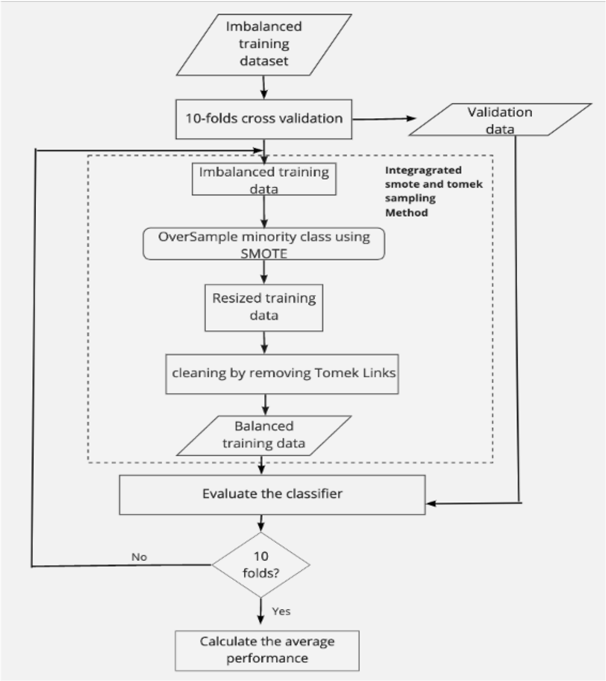
Imbalanced training dataset
10-folds cross validation
Validation data
Imbalanced training data
Integragrated smote and tomek sampling Method
OverSample minority class using SMOTE
Resized training data
cleaning by removing Tomek Links
Balanced training data
Evaluate the classifier
No
10 folds?
Yes
Calculate the average performance

Figure 2: Imbalance processing method

3.5 Building the Deep Learning models for IDS

We now describe our deep learning models for the network intrusion detection system(NIDS). Two well-known deep learning models are used to build the NIDS in each datasets: LSTM and 1DCNN. The motivation for choosing these two models comes from their efficiency proved in other domains.

3.5.1 LSTM

Recurrent Neural Network (RNN)[47] is a class of deep learning which extends the capabilities of the traditional neural network and is designed to model the sequence data of neural networks. RNN contains a looped connection in the hidden layer and has the property of reusing information already given. However RNN has difficulty to learn from long sequence because of vanishing and exploding gradients. This has led to the development of Long short-term memory (LSTM)[21].

The LSTM model is proposed by Hochreiter et al. [21] to solve the vanishing gradient problem of RNN. It is composed of a cell, an input gate, output gate and forget gate.

In this work the LSTM is used in the NSLKDD and CICIDS2017 dataset as follow: for each dataset we use a simple model with three hidden layers. The first hidden layer is a LSTM layer, and followed by two fully connected dense layers. After several trials the number of neurons for hidden layer is chosen as follows (129,109,98) and (78,70,63) for NSLKDD and CICIDS2017 respectively. Rectified Linear Unit (ReLU) and hyperbolic tangent (Tanh) has been used during hyperparameter confirugation. However the Tanh has achieved the best result. Therefore we use the Tanh as activation function in the input layer as well as hidden layer and the softmax function in the output layer. Though dropout is often used in deep learning, in our work it made no difference for the NSLKDD therefore we only used the dropout for the CICIDS2017 with a value of 10%. To compile the model Adam optimizer is used as an optimization algorithm with 0.001 for learning rate and categorical cross entropy is used as loss function.

3.5.2 1-D CNN

The Convolutional Neural Network (CNN) is a deep learning model widely used in computer vision. CNN is composed of a convolutional layer, pooling layer and dense layer. While the two dimensional CNN has been successfully applied in image recognition, the one dimensional CNN(1D CNN) is more suitable for sequence data[31].

We leverage on the 1D CNN model for creating an efficient NIDS, inspired by the ability of CNN to learn suitable feature representations of the input data. In NSLKDD we design a 4 layers 1-D CNN. the first 2 layers are convolutional layers with 64 convolutional filters, a kernel of size 3 with Relu activation function. Each convolutional layer is followed by a Max-Pooling layer with a pooling size 2 and a Batch Normalization. The max pooling will reduce the spatial size which result in decreasing the computational complexity and avoid overfitting. Batch normalization will help the training time to converge faster. The last 2 layers are two fully connected for classification. Also we use a dropout of 20% before the last fully connected layer. The same process is applied in the CICIDS2017. The implementation of CICIDS2018 is discussed in section 4.3.1

4 Experiment and result

In this section, we conduct the experiment to evaluate the performance and advantage of the combination of deep learning with SMOTETomek proposed in this paper. First the metrics used for evaluation as presented, then the proposed method is compared with the baseline model and existing state-of-the-art approach. The experiments show our proposed method significantly improve the detection of minority attack classes and outperforms most of the proposed state-of-the-art methods.

4.1 Environment Setup

TensorFlow and Keras in an Anaconda environment were used to implement the models. Regarding the hardware environment, our experiments were performed in Linux-Ubuntu 20.4 with a GPU NVIDIA GeForce GTX 1060.

4.2 Evaluation Metrics

Metrics based on the confusion matrix are used to evaluate our models. A confusion matrix[24], is a specific table that allows the visualization of the performance of an algorithm by providing information about the Actual and Predicted class. It is a largely used metric for supervised learning. The terminologies used in the confusion matrix include:

- True positive (TP): The data instances correctly predicted to be positive.
- False negative (FN): The data instances wrongly predicted to be negative.
- True negative (TN): The data instances correctly predicted to be negative.
- False positive (FP): The data instances wrongly predicted to be positive.

When data are imbalanced, Accuracy is not the appropriate metric. Therefore in addition to the Accuracy we also consider the metrics required for class imbalance namely Precision, Recall(or Detection rate) and F1 score. Moreover, the false alarm rate (FAR) also called false positive rate is considered. As shown in Equations (2),(3),(4),(5),(6). For each of the metrics, we evaluate their results using the weighted average method. This method is more appropriate for evaluating the model performance on imbalanced dataset in multiclass classification[52]. We have also calculated the ROC Area Under Curve (AUC) for each class as well as the macro and micro average AUC Fig. 8,9, 10 and 11, 12

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = DetectionRate = \frac{TP}{TP + FN} \tag{5}$$

$$F1Score = 2.\frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

$$FAR = \frac{FP}{FP + TN} \tag{7}$$

4.3 Experimental Results

Experiments are separated into evaluating the performance of the proposed model in each dataset compared with the baseline model and a comparison with the state-of-the-art methods recently proposed. Two baseline models are proposed in each dataset. LSTM and CNN have been recently widely and successfully used to implement NIDS. Therefore, LSTM and 1D-CNN are chosen in this work. The baseline methods are implemented with these two deep learning.

4.3.1 Performance comparison before and after resampling

Table 5: Experimental Results

Algorithm	NSLKDD				CICIDS2017			
	Accuracy	Precision	Detection rate	F1_score	Accuracy	Precision	Detection rate	F1_score
LSTM	96.51	96.43	96.51	96.40	95	94.47	94.98	94.66
CNN	98.29	98.28	98.29	98.28	92.30	93	92.30	92
LSTM+SMOTETomek	99	99.09	99.98	99.02	99.65	98	97.38	98
CNN+SMOTETomek	99.70	99.29	99.25	99.27	99.85	99.16	98.94	98.98

From the above table5, we can see that although the CNN and LSTM model can achieve a good performance, their F1 score are lower. After mitigating the imbalanced data with the SMOTETomek algorithm, their performance as well as F1 score have significantly improved. It shows that these deep learning model can not handle imbalanced data.

In NSLKDD From the experiment, If we take a closer look in the class detection fig. 3, fig.4, we can observe the baseline models can not detect well the minority classes U2R and R2L in the NSLKDD. The F1_score is also shown in fig.4.However with the combination of the SMOTETomek, we achieve a good detection rate of the minority attack traffic. Therefore, With the proposed method, in LSTM NIDS we obtain an overall accuracy of 99.57%, a detection rate of 98.93%, Fscore of 98.98% and with a FAR of 0.002. Table 6shows the performance result of the proposed method in each class.

With the CNN we obtain an overall accuracy of 99.70%, a detection rate of 99.25%, F1score of 99.27% and a FAR of 0.001. Table 7 shows the performance result of the proposed method in each class.

In CICIDS2017 dataset, with the LSTM model, during the experiment we noticed the training time was too long, therefore, to reduce the computation cost we proceeded on a feature selection. In [10] Binbusayyis et al. proposed a feature selection on CICIDS2017 using different evaluation measures. As a result they proposed 6 features relevant for the CICIDS2017 such as *Flow Duration, Bwd Packet Len Mean, Flow IAT Max, Average Packet Size, Init_Win_bytes_backward, and Init_Win_bytes_forward.* A detailed approach can be found in their paper. In this experiment we employed these 6 features proposed to build the LSTM model to reduced the training time of the LSTM. Then the performance of the proposed method was compared with the state-of-the-art models that do not address the imbalance problem. Fig.5 depicts the attack detection rate obtained by the proposed method and the baseline method. The proposed method outperforms the state-of-the-art model and achieve the highest performance for each attack detection. We the proposed method,

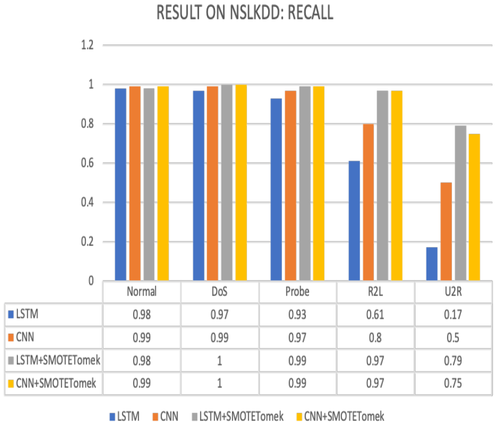

Figure 3: Recall values NSLKDD

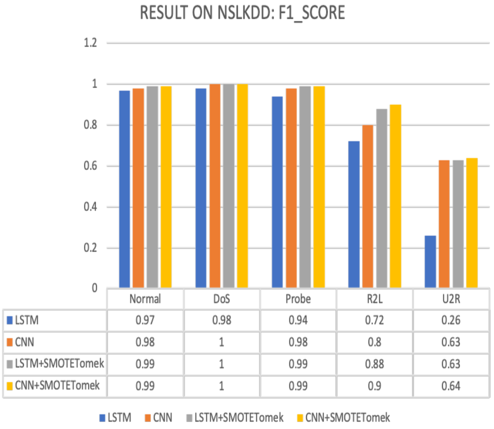

Figure 4: F1 score values NSLKDD

we obtain an overall accuracy and Fscore of 99.82%, 98.65% respectively a detection rate of 98.70% and a FAR of 0.001. The performance of each class is shown in Table 9

With the CNN model, no feature selection was performed. Based on the model results, our proposed method outperforms also its counterpart fig5 and fig6. We achieve an accuracy of 99.85%, a detection rate of 98.94% and F1score of 98.98%. The model gives a false alarm rate of 0.0007. The performance of each class is shown in Table 10

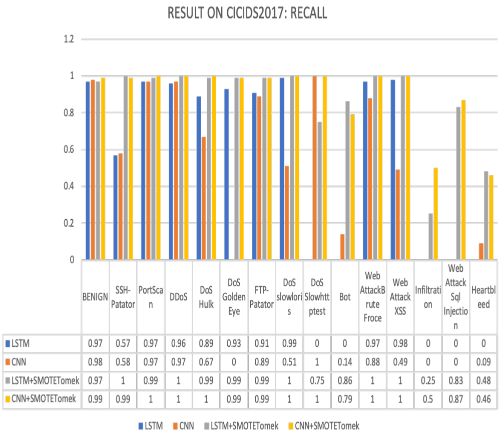

Figure 5: Recall values CICIDS2017

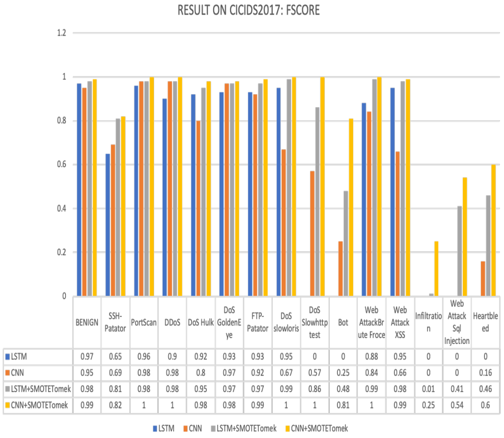

Figure 6: F1 score values CICIDS2017

In our experimental results, the 1D-CNN has shown better result than the LSTM in the NSL-KDD and CICIDS2017 datasets. Therefore in CICIDS2018 dataset, we implement the NIDS using the 1-D CNN. With the CNN model, no feature selection was performed. We design a 5 layers 1-D CNN. The first 3 layers are convolutional layers with 64 convolutional filters, a kernel of size 3 with Relu activation function. The first convolutional layer is followed by Max-Pooling layer with a pooling size 2 and a Batch Normalization. The max pooling will reduce the spatial size which result in decreasing the computational complexity and avoid overfitting. Batch normalization will help the training time to converge faster. The second and third convolutionnal layers are stacked and then followed by Max-Pooling layer with a pooling size 2 and a Batch Normalization. The last 2 layers are two fully connected for classification. Also we use a dropout of 40% before the last fully connected layer.

Based on the model results we obtain an accuracy of 95%, a detection rate of 95% and F1score of 95%. The model gives a false alarm rate of 0.003. Fig.7 shows the performance of the model compared with the baseline which do not handle the imlabalced data. The performance of each class traffic or the CICIDS2018 dataset is shown in Table 12

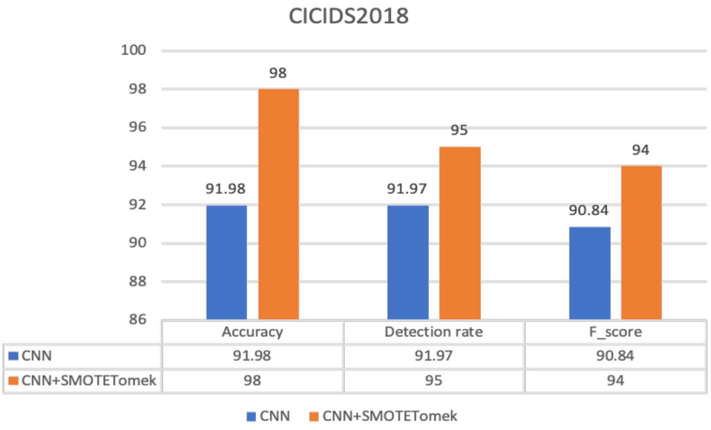

Figure 7: Experiment Results on CICIDS2018

Furthumore, the performance of the proposed method is compared with recent works in the literature that address the class imbalanced in NSLKDD [51, 40, 55, 42] and CICIDS2017 datasets [52, 1, 46, 5] shown in Table 10 and Table 11. The obtained results highlight our proposed method outperforms most of the recent works and effectively improve the NIDS performance. The comparison are discussed in the followed section.

4.3.2 Discussion

From the experimental result we can observe that:

1. In NSLKDD, the minority traffic are the R2L and U2R attack. R2L attack account 0.08% and U2R account 2.61%. However with our proposed model we were able to detect these attacks with a high detection rate. for instance with the U2R we achieve a detection rate of 95% with an accuracy of 99.52% in CNN Table (7)

   The comparison of results on NSLKDD is shown on Table 10. Table 10, is a comparison with relevant works in the literature that consider the imbalanced problem and works that do not consider the imbalanced problem in NSL-KDD. Based on the results, the best performance are achieved by works taking into consideration the imbalanced problem [55],[40], and our proposed method. However, by using a simple CNN model our proposed approach achieves the best performance.

2. In CICIDS2017 datasets, the minority traffic are heartbleed with 0.003% instance, web attack sql injection with 0.006%, web attack xss with 0.20%, Tab 3. We can observe that our proposed model can detect these minority traffic. For instance we achieve a detection rate of 100% with both LSTM and CNN on the web attack xss traffic Table 8 and Table 9; and a detection rate of 87% for web attack sql injection with CNN Table 9 and 83% on LSTM Table8.

   Table 11 shows the comparison results on CICIDS2017 dataset. Here, to have a qualitative comparison, we only consider previous work that studied the imbalance problem. With a simple CNN model, our method was able to outperform most state-of-the-art methods. However in term of detection rate the SMOTE-SGM with CNN [52] achieved the best result but they did not present the false alarm rate result contrary to our work. In our future work we will investigate in a robust model and also explore other deep learning models.

3. Both deep learning models combined with the resampling method, clearly improve the performance of the IDS while being very simple models. However, in all experiments, the CNN showed to be better than the LSTM. To show the advantage of our model and the usefulness of handling the imbalanced problem, we compare the proposed work with the existing method. The result shows that our model can reduce the false alarm rate(FAR), hence gives a promising approach for intrusion detection systems. However, the detection rate for the minority classes still needs improvement.

4. After evaluating the proposed method on NSL-KDD and CICIDS2017, we extended the work on the most recent NIDS dataset, the CICIDS2018 using the CNN model. Based on the result we can notice the hybrid method improve the detection of attack traffic with less instances. Moreover the model achieve a false rate of 0.003%. The roc curve is shown in fig.12

5. In this work we implemented LSTM and CNN separately. However, some recent works[4, 43, 20] have shown the hybrid deep learning model can achieve a better performance in NIDS. Therefore, in our future work we will implement the proposed method in a hybrid deep learning model which combine LSTM and CNN with the association of big data technique to be able to evaluate our proposed model in large scale dataset with less training time.

Table 6: Performance evaluation of the proposed method on the NSLKDD dataset using LSTM

class Label	Accurary	Precision	Detection rate	F1 Score
Normal	99.89%	1.00	0.98	0.99
DoS	99.06%	1.00	1.00	1.00
Probe	99.77%	0.99	0.99	0.99
U2R	99.33%	0.81	0.97	0.88
R2L	99.92%	0.53	0.79	0.63
Average	99%	98.98%	99.09%	99.02%

Table 7: Performance evaluation of the proposed method on the NSLKDD dataset using CNN

class Label	Accurary	Precision	Detection rate	F1 Score
Normal	99.30%	1.00	0.99	0.99
DoS	99.91%	1.00	1.00	1.00
Probe	99.84%	0.99	0.99	0.99
U2R	99.52%	0.88	0.95	0.91
R2L	99.92%	0.55	0.71	0.62
Average	99.70%	99.29%	99.25%	99.27%

5 Conclusion and Future research direction

The main focus of this work is to mitigate the class imbalance problem which affects seriously the detection rate in NIDS based on deep learning. We propose a training process technique that combine a hybrid imbalance processing method with SMOTETomek and deep learning. Two deep learning model have been implemented: CNN and LSTM. SMOTETomek method is used as the resampling technique during the training process. This technique combine SMOTE for oversampling and TomekLink for undersampling, which remove the overlapping data created with SMOTE. Based on the experimental results, the proposed model can improve the detection rate of minority attack instances and can also improve the overall accuracy. Moreover the model can decrease the false alarm

Table 8: Performance evaluation of the proposed method on the CICIDS2017 dataset using LSTM

class Label	Accurary	Precision	Detection rate	F1 Score
BENIGN	97.79%	1.00	0.97	0.98
SSH-Patator	99.70%	0.68	1.00	0.81
PortScan	99.86%	0.98	0.99	0.98
DDoS	99.85%	0.96	1.00	0.98
DoS Hulk	99.28%	0.92	0.99	0.95
DoS GoldenEye	99.89%	0.98	0.99	0.97
FTP-Patator	99.88%	0.95	0.99	0.97
DoS slowloris	99.96%	0.99	1.00	0.99
DoS Slowhttptest	99.99%	1.00	0.75	0.86
Bot	99.97%	0.33	0.86	0.48
Web AttackBrute Force	99.91%	0.99	1.00	0.99
Web Attack XSS	99.90%	0.95	1.00	0.98
Infiltration	99.74%	0.01	0.25	0.01
Web Attack Sql Injection	99.51%	0.27	0.83	0.41
Heartbleed	99.45%	0.43	0.48	0.46
Average	99.65%	98.13%	98%	98%

rate. To show the advantage of our model and the usefulness of handling the imbalanced problem, we compare the proposed work with our baseline models and the existing proposed approach. The result shows that our model can reduce the false alarm rate(FAR), hence gives a promising approach for intrusion detection systems. However, the detection rate for the minority classes still needs improvement. In our future work, we plan to investigate other imbalance processing technique and future selection to improve the NIDS. Moreover, in this work we implemented LSTM and CNN separately. However, some recent works have shown the hybrid deep learning model can achieve a better performance in NIDS. Therefore, in our future work we will implement the proposed method in a hybrid deep learning model which combine LSTM and CNN with the association of big data technique to be able to evaluate our proposed model in large scale dataset with less training time.

Table 9: Performance evaluation of the proposed method on the CICIDS2017 dataset using CNN

class Label	Accurary	Precision	Detection rate	F1 Score
BENIGN	99.26%	1.00	0.99	0.99
SSH-Patator	99.73%	0.70	0.99	0.82
PortScan	99.98%	1.00	1.00	1.00
DDoS	99.97%	0.99	1.00	1.00
DoS Hulk	99.69%	0.96	1.00	0.98
DoS GoldenEye	99.93%	0.97	0.99	0.98
FTP-Patator	99.97%	0.99	0.99	0.99
DoS slowloris	99.99%	1.00	1.00	1.00
DoS Slowhttptest	100%	1.00	1.00	1.00
Bot	99.99%	0.85	0.79	0.81
Web AttackBrute Force	99.95%	0.99	1.00	1.00
Web Attack XSS	99.97%	0.99	1.00	0.99
Infiltration	99.98%	0.17	0.50	0.25
Web Attack Sql Injection	99.69%	0.39	0.87	0.54
Heartbleed	99.70%	0.86	0.46	0.60
Average	99.85%	99.16%	98.94%	98.98%

Table 10: Comparison result with previous studies in NSLKDD

model	Accurary	Detection rate	FAR
LSTM RNN with GA[32]	93.88%	-	0.005
RNNIDS[51]	81.29%	-	-
BAT[42]	84.25%	84.15%	0.003
I-NSGA-III[55]	-	99.21%	-
CNN-BILSTM[40]	99.22%	98.88%	0.004
our proposed work with LSTM	99%	99.09%	0.002
our proposed work with CNN	99.70%	99.25%	0.001

Table 11: Comparison result of the proposed methods and previous studies in CICIDS2017

method	classifier	Accuracy	Detection rate	F1 score	FAR
SMOTEENN[46]	DNN	99.95%	95.62%	94.1%	0.0005
SMOTE-SGM[52]	CNN	99.85%	99.85	99.86	-
UDBB[1]	RF	98.8%	98.8	94.67	0.001
SMOTE - BMCD [5]	RF	99.32%	-	98.36%	-
	MLP	94.82%	-	80.63%	-
	NB	75.35%	-	90.82%	-
our work(SMOTETomek)	LSTM	99.65%	98%	98%	0.001
our work(SMOTETomek)	CNN	99.85%	98.94%	98.98%	0.0007

Table 12: Performance evaluation of the proposed method on the CICIDS2018 dataset using CNN

class Label	Accuracy	Precision	Detection rate	F1 Score	FAR
BENIGN	96.90%	0.89	0.99	0.94	
Bot	99.99%	1.00	1.00	1.00	
Brute Force -Web	99.94%	0.77	0.87	0.82	
Brute Force -XSS	99.96%	0.71	0.68	0.70	
DDOS attack-HOIC	100%	1.00	1.00	1.00	
DDOS attack-LOIC-UDP	99.98%	0.96	1.00	0.98	
DDoS attacks-LOIC-HTTP	99.97%	1.00	1.00	1.00	
DoS attacks-GoldenEye	99.99%	1.00	1.00	1.00	
DoS attacks-Hulk	99.99%	1.00	1.00	1.00	
DoS attacks-SlowHTTPTest	97.83%	0.76	0.50	0.60	
DoS attacks-Slowloris	99.99%	1.00	1.00	1.00	
FTP-BruteForce	97.82%	0.71	0.89	0.79	
Infilteration	96.93%	0.82	0.25	0.39	
SQL Injection	99.98%	1.00	0.46	0.63	
SSH-Bruteforce	99.99%	1.00	1.00	1.00	
Average	98.17%	95%	95%	94%	0.003

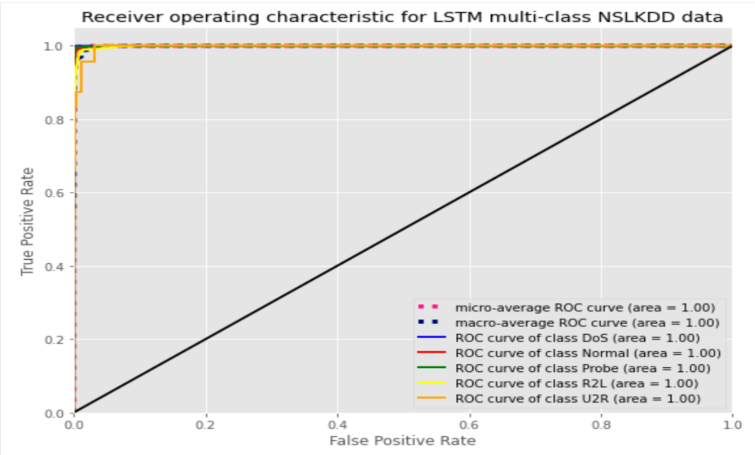

Figure 8: ROC-AUC Curve LSTM in NSL-KDD

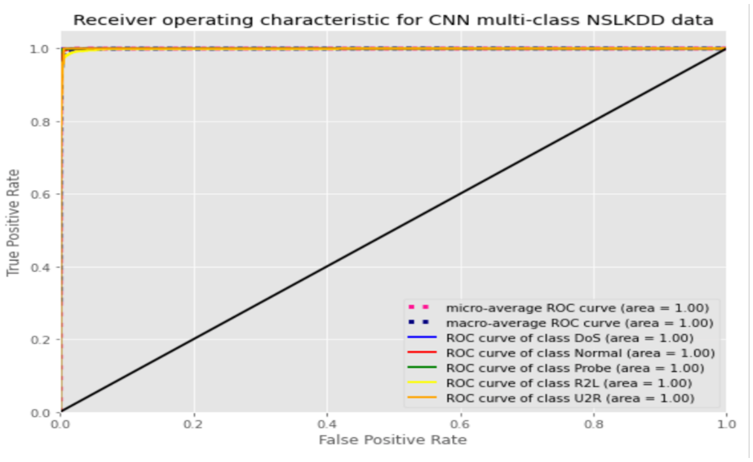

Figure 9: ROC-AUC Curve CNN in NSL-KDD

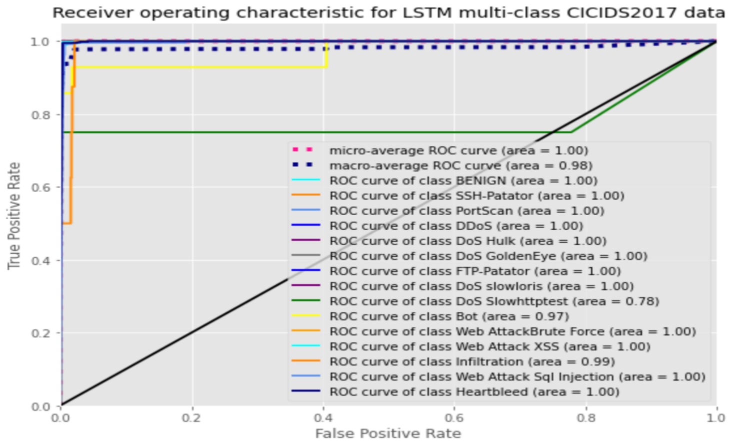

Figure 10: ROC-AUC Curve LSTM in CICIDS2017

Acknowledgment

This research is supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT). This research is also supported by JSPS KAKENHI Grant Number 21K11888 and Hitachi

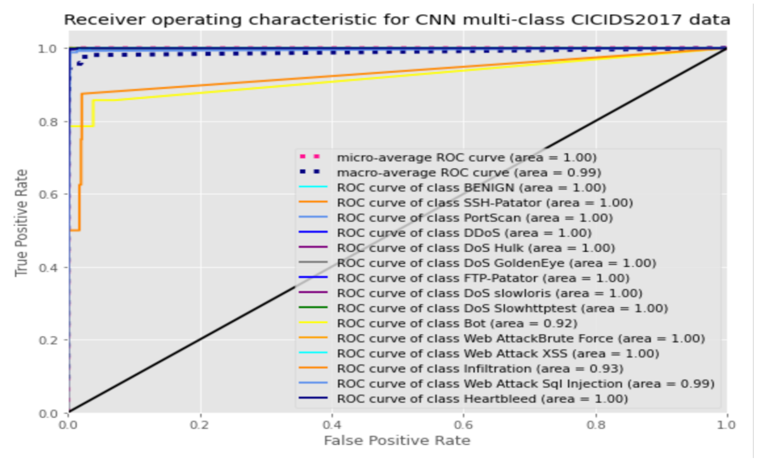

Figure 11: ROC-AUC Curve CNN in CICIDS2017

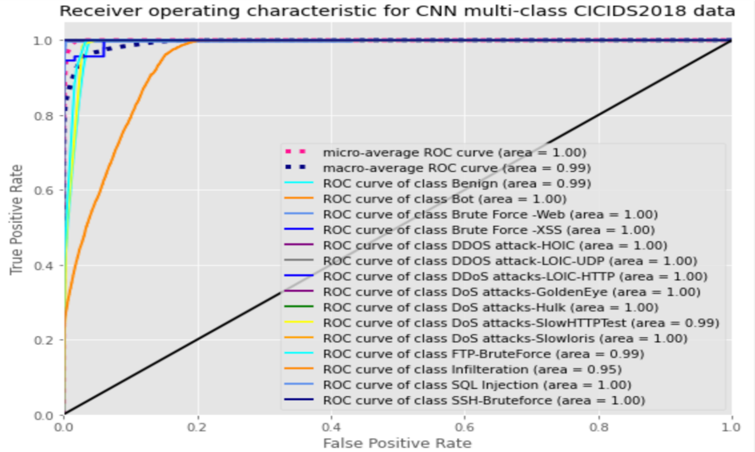

Figure 12: ROC-AUC Curve CNN in CICIDS2018

Systems, Ltd.

References

[1] Razan Abdulhammed, Hassan Musafer, Ali Alessa, Miad Faezipour, and Abdelshakour Abuzneid. Features dimensionality reduction approaches for machine learning based network intrusion detection. *Electronics*, 8(3), 2019.

[2] Iftikhar Ahmad, Qazi Emad Ul Haq, Muhammad Imran, Madini O. Alassafi, and Rayed A. AlGhamdi. An efficient network intrusion detection and classification system. *Mathematics*, 10(3), 2022.

[3] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. Network intrusion detection system: A systematic study of machine learning and deep learning approache. 2020.

[4] Samed Al and Murat Dener. Stl-hdl: A new hybrid network intrusion detection system for imbalanced dataset on big data environment. *Comput. Secur.*, 110(C), nov 2021.

[5] Amer Abulmajeed Abdulrahman Alsameraee and Mahmood Khalel Ibrahem. Toward constructing a balanced intrusion detection dataset. *Samarra Journal of Pure and Applied Science*, 2021.

[6] Mohammed Awad and Alaeddin Alabdallah. Addressing imbalanced classes problem of intrusion detection system using weighted extreme learning machine. *Electronic*, 2019.

[7] Sikha Bagui and Kunqi Li. Resampling imbalanced data for network intrusion detection datasets. *Journal of Big Data*, 8:1–41, 2021.

[8] Ram B. Basnet, Riad Shash, Clayton Johnson, Lucas Walgren, and Tenzin Doleck. Towards detecting and classifying network intrusion traffic using deep learning frameworks. *J. Internet Serv. Inf. Secur.*, 9:1–17, 2019.

[9] Gustavo E. A. P. A. Batista, Ana Lúcia Cetertich Bazzan, and Maria Carolina Monard. Balancing training data for automated annotation of keywords: a case study. In *WOB*, 2003.

[10] Adel Binbusayyis and Thavavel Vaiyapuri. Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach. *IEEE Access*, 7:106495–106513, 2019.

[11] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. 16(1):321–357, jun. 2002.

[12] The imbalanced-learn developers Copyright 2014-2022. Smotetomek. `https://imbalanced-learn.org/dev/references/generated/imblearn.combine.SMOTETomek.html`, 2022.

[13] D.E. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, 1987.

[14] Wisam Elmasry, Akhan Akbulut, and Abdul Halim Zaim. Empirical study on multiclass classification-based network intrusion detection. *Computational Intelligence*, 35:919 – 954, 2019.

[15] Osama Faker and Erdogan Dogdu. Intrusion detection using big data and deep learning techniques. New York, NY, USA, 2019. Association for Computing Machinery.

[16] Osama Faker and Erdogan Dogdu. Intrusion detection using big data and deep learning techniques. *Proceedings of the 2019 ACM Southeast Conference*, 2019.

[17] Nabila Farnaaz and M.A. Jabbar. Random forest modeling for network intrusion detection system. *Procedia Computer Science*, 89:213–217, 2016. Twelfth International Conference on Communication Networks, ICCN 2016, August 19– 21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India.

[18] Canadian Institute for Cybersecurity. A realistic cyber defense dataset (cse-cic-ids2018). `https://registry.opendata.aws/cse-cic-ids2018`, 2022.

[19] Neha Gupta, Vinita Jindal, and Punam Bedi. Cse-ids: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. *Comput. Secur.*, 112(C), jan 2022.

[20] Mohammad Mehedi Hassan, Abdu Gumaei, Ahmed Alsanad, Majed Alrubaian, and Giancarlo Fortino. A hybrid deep learning model for efficient intrusion detection in big data environment. *Information Sciences*, 513:386–396, 2020.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997.

[22] Brownlee Jason. How to fix k-fold cross-validation for imbalanced classification, 2020.

[23] Jianguo Jiang, Qiwen Wang, Zhixin Shi, Bin Lv, and Biao Qi. Rst-rf: A hybrid model based on rough set theory and random forest for network intrusion detection. ICCSP 2018, page 77–81, New York, NY, USA, 2018. Association for Computing Machinery.

[24] V. Jyothsna and K. Munivara Prasad. Anomaly-based intrusion detection system. 2019.

[25] Gurdip Kaur, Arash Habibi Lashkari, and Abir Rahali. Intrusion traffic detection and characterization using deep image learning. *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 55–62, 2020.

[26] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecur.*, 2:20, 2019.

[27] Ilhan Firat Kilincer, Fatih Ertam, and Abdulkadir Sengur. Machine learning methods for cyber security intrusion detection: Datasets and comparative study. *Computer Networks*, 188:107840, 2021.

[28] Joffrey L. Leevy and Taghi M. Khoshgoftaar. A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data. *Journal of Big Data*, 7:1–19, 2020.

[29] Ahmed M. Mahfouz, Deepak Venugopal, and Sajjan G. Shiva. Comparative analysis of ml classifiers for network intrusion detection. In *ICICT*, 2019.

[30] Mariama Mbow, Hiroshi Koide, and Kouichi Sakurai. An intrusion detection system for imbalanced dataset based on deep learning. In *2021 Ninth International Symposium on Computing and Networking (CANDAR)*, pages 38–47, 2021.

[31] Aziz Meliboev, Jumabek Alikhanov, and Wooseong Kim. 1d cnn based network intrusion detection with normalization on imbalanced data. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 218–224, 2020.

[32] Pramita Sree Muhuri, Prosenjit Chatterjee, Xiaohong Yuan, Kaushik Roy, and Albert Esterline. Using a long short-term memory recurrent neural network (lstm-rnn) to classify network attacks. *Information*, 11(5), 2020.

[33] Ranjit Panigrahi and Samarjeet Borah. A detailed analysis of cicids2017 dataset for designing intrusion detection systems. *International Journal of Engineering & Technology*, 7(3.24), 2018.

[34] Panagiotis I. Radoglou-Grammatikis and Panagiotis G. Sarigiannidis. An anomaly-based intrusion detection system for the smart grid based on cart decision tree. In *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–5, 2018.

[35] Sireesha Rodda and Uma Shankar Rao Erothi. Class imbalance problem in the network intrusion detection systems. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 2685–2688, 2016.

[36] Miriam Seoane Santos, Jastin Pompeu Soares, Pedro Henrigues Abreu, Helder Araujo, and Joao Santos. Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]. 13(4):59–76, nov 2018.

[37] Steinberg Scott. Cyberattacks now cost companies $200,000 on average, putting many out of business. `https://www.cnbc.com/2019/10/13/cyberattacks-cost-small-companies-200k-putting-many-out-of-business.html`, march 2019.

[38] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, 2018.

[39] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, 2018.

[40] Jay Sinha and M. Manollas. Efficient deep cnn-bilstm model for network intrusion detection. In *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*, AIPR 2020, page 223–231, New York, NY, USA, 2020. Association for Computing Machinery.

[41] Saeid Soheily-Khah, Pierre-François Marteau, and Nicolas Béchet. Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the iscx dataset. In *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, pages 219–226, 2018.

[42] Tongtong Su, Huazhi Sun, Jinqi Zhu, Sheng Wang, and Yabo Li. Bat: Deep learning methods on network intrusion detection using nsl-kdd dataset. *IEEE Access*, 8:29575–29585, 2020.

[43] Pengfei Sun, Pengju Liu, Qi Li, Chenxi Liu, Xiangling Lu, Ruochen Hao, and Jinpeng Chen. Dl-ids: Extracting features using cnn-lstm hybrid network for intrusion detection system. *Security and Communication Networks*, 2020, 2020.

[44] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of the kdd cup 99 data set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6, 2009.

[45] Ivan Tomek. Two modifications of cnn. 1976.

[46] Petros Toupas, Dimitra Chamou, Konstantinos M. Giannoutakis, Anastasios Drosou, and Dimitrios Tzovaras. An intrusion detection system for multi-class classification based on deep neural networks. *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1253–1258, 2019.

[47] R Vinayakumar, K.P. Soman, and Prabaharan Poornachandran. Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). *International Journal of Information System Modeling and Design (IJISMD)*, 8(3):43–63, July 2017.

[48] Ly Vu and Quang Uy Nguyen. Handling imbalanced data in intrusion detection systems using generative adversarial networks. 2020.

[49] Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42:1119–1130, 2012.

[50] Yang Xin, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng Gao, Haixia Hou, and Chunhua Wang. Machine learning and deep learning methods for cybersecurity. *IEEE Access*, 6:35365–35381, 2018.

[51] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017.

[52] Hongpo Zhang, Lulu Huang, Chase Q. Wu, and Zhanbo Li. An effective convolutional neural network based on smote and gaussian mixture model for intrusion detection in imbalanced dataset. *Computer Networks*, 177:107315, 2020.

[53] Lei Zhang, Shuaimin Jiang, Xiajiong Shen, Brij B. Gupta, and Zhihong Tian. Pwg-ids: An intrusion detection model for solving class imbalance in iiot networks using generative adversarial networks, 2021.

[54] Xiaoxuan Zhang, Jing Ran, and Jize Mi. An intrusion detection system based on convolutional neural network for imbalanced network traffic. *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, pages 456–460, 2019.

[55] Yingying Zhu, Junwei Liang, Jianyong Chen, and Zhong Ming. An improved nsga-iii algorithm for feature selection used in intrusion detection. *Know.-Based Syst.*, 116(C):74–85, jan. 2017.

[56] scikit-learn developers (BSD License) © 2007 2021. Stratified k-folds cross-validator, 2021.

Table 13: List of features in NSLKDD dataset

No.	Feature name	No.	Feature name
0	DURATION	21	IS_GUEST_LOGIN
1	PROTOCOL	22	COUNT
2	SERVICE	23	SRV_COUNT
3	FLAG	24	SERROR_RATE
4	SRC_BYTES	25	SRV_SERROR_RATE
5	DST_BYTES	26	RERROR_RATE
6	LAND	27	SRV_RERROR_RATE
7	WRONG_FRAGMENT	28	SAME_SRV_RATE
8	URGENT	29	DIFF_SRV_RATE
9	HOT	30	SRV_DIFF_HOST_RATE
10	NUM_FAILED_LOGINS	31	DST_HOST_COUNT
11	LOGGED_IN	32	DST_HOST_SRV_COUNT
12	NUM_COMPROMISED	33	DST_HOST_SAME_SRV_RATE
13	ROOT_SHELL	34	DST_HOST_DIFF_SRV_RATE
14	SU_ATTEMPTED	35	DST_HOST_SAME_SRC_PORT_RATE
15	NUM_ROOT	36	DST_HOST_SRV_DIFF_HOST_RATE
16	NUM_FILE_CREATIONS	37	DST_HOST_SERROR_RATE
17	NUM_SHELLS	38	DST_HOST_SRV_SERROR_RATE
18	NUM_ACCESS_FILES	39	DST_HOST_RERROR_RATE
19	NUM_OUTBOUND_CMDS	40	DST_HOST_SRV_RERROR_RATE
20	IS_HOST_LOGIN		

Table 14: List of features in CICIDS2017 dataset

No.	**Feature name**	**No.**	**Feature name**
0	Source Port	41	Max Packet Length
1	Destination Port	42	Packet Length Mean
2	Protocol	43	Packet Length Std
3	Flow Duration	44	Packet Length Variance
4	Total Fwd Packets	45	FIN Flag Count
5	Total Backward Packets	46	SYN Flag Count
6	Total Length of Fwd Packets	47	RST Flag Count
7	Total Length of Bwd Packets	48	PSH Flag Count
8	Fwd Packet Length Max	49	ACK Flag Count
9	Fwd Packet Length Min	50	URG Flag Count
10	Fwd Packet Length Mean	51	CWE Flag Count
11	Fwd Packet Length Std	52	ECE Flag Count
12	Bwd Packet Length Max	53	Down Up Ratio
13	Bwd Packet Length Min	54	Average Packet Size
14	Bwd Packet Length Mean	55	Avg Fwd Segment Size
15	Bwd Packet Length Std	56	Avg Bwd Segment Size
16	Flow Bytess_creations	57	Fwd Avg Bytes Bulk
17	Flow Packetss	58	Fwd Avg Packets Bulk
18	Flow IAT Mean	59	Fwd Avg Bulk Rate
19	Flow IAT Std	60	Bwd Avg Bytes Bulk
20	Flow IAT Max	61	Bwd Avg Packets Bulk
21	Flow IAT Min	62	Bwd Avg Bulk Rate
22	Fwd IAT Total	63	Subflow Fwd Packets
23	Fwd IAT Mean	64	Subflow Fwd Bytes
24	Fwd IAT Std	65	Subflow Bwd Packets
25	Fwd IAT Max	66	Subflow Bwd Bytes
26	Fwd IAT Min	67	Init_Win_bytes_forward
27	Bwd IAT Total	68	Init_Win_bytes_backward
28	Bwd IAT Mean	69	act_data_pkt_fwd
29	Bwd IAT Std	70	min_seg_size_forward
30	Bwd IAT Max	71	Active Mean
31	Bwd IAT Min	72	Active Std
32	Fwd PSH Flags	73	Active Max
33	Bwd PSH Flags	74	Active Min
34	Fwd URG Flags	75	Idle Mean
35	Bwd URG Flags	76	Idle Std
36	Fwd Header Length	77	Idle Max
37	Bwd Header Length	78	Idle Min
38	Fwd Packetss		
39	Bwd Packetss		
40	Min Packet Length		

Table 15: List of features in CICIDS2018 dataset

No.	Feature name	No.	Feature name
0	Dst Port	43	Pkt Len Std
1	Protocol	44	Pkt Len Var
2	Timestamp	45	FIN Flag Cnt
3	Flow Duration	46	SYN Flag Cnt
4	Tot Fwd Pkts	47	RST Flag Cnt
5	Tot Bwd Pkts	48	PSH Flag Cnt
6	TotLen Fwd Pkts	49	ACK Flag Cnt
7	otLen Bwd Pkts	50	URG Flag Cnt
8	Fwd Pkt Len Max	51	CWE Flag Count
9	Fwd Pkt Len Min	52	ECE Flag Cnt
10	Fwd Pkt Len Mean	53	Down/Up Ratio
11	Fwd Pkt Len Std	54	Pkt Size Avg
12	Bwd Pkt Len Max	55	Fwd Seg Size Avg
13	Bwd Pkt Len Min	56	Bwd Seg Size Avg
14	Bwd Pkt Len Mean	57	Fwd Byts/b Avg
15	Bwd Pkt Len Std	58	Fwd Pkts/b Avg
16	Flow Byts/s	59	Fwd Blk Rate Avg
17	Flow Pkts/s	60	Bwd Byts/b Avg
18	Flow IAT Mean	61	Bwd Pkts/b Avg
19	Flow IAT Std	62	Bwd Blk Rate Avg
20	Flow IAT Max	63	Subflow Fwd Pkts
21	Flow IAT Min	64	Subflow Fwd Byts
22	Fwd IAT Tot	65	Subflow Bwd Pkts
23	Fwd IAT Mean	66	Subflow Bwd Byts
24	Fwd IAT Std	67	Init Fwd Win Byts
25	Fwd IAT Max	68	Init Bwd Win Byts
26	Fwd IAT Min	69	Fwd Act Data Pkts
27	Bwd IAT Tot	70	Fwd Seg Size Min
28	Bwd IAT Mean	71	Active Mean
29	Bwd IAT Std	72	Active Std
30	Bwd IAT Max	73	Active Max
31	Bwd IAT Min	74	Active Min
32	Fwd PSH Flags	75	Idle Mean
33	Bwd PSH Flags	76	Idle Std
34	Fwd URG Flags	77	Idle Max
35	Bwd URG Flags	78	Idle Min
36	Fwd Header Len	79	Label
37	Bwd Header Len	80	Flow ID
38	Fwd Pkts/s	81	Src IP
39	Bwd Pkts/s	82	Src Port
40	Pkt Len Min	83	Dst IP
41	Pkt Len Max		
42	Pkt Len Mean		