Intrusion Detection in Cyber-Physical Environment Using Hybrid Naïve Bayes - Decision Table and Multi-Objective Evolutionary Feature Selection

Ranjit Panigrahi[a], Samarjeet Borah[a], Moumita Pramanik[a], Akash Kumar Bhoi[b,c], Paolo Barsocchi[c], Soumya Ranjan Nayak[d,*], Waleed Alnumay[e]

[a]*Department of Computer Applications, Sikkim Manipal Institute of Technology, Sikkim Manipal University, 737136 Majitar, Sikkim, India*

[b]*Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Sikkim Manipal University, 737136, Majitar, Sikkim, India*

[c]*Institute of Information Science and Technologies, National Research Council, 56124 Pisa, Italy*

[d]*Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India*

[e]*King Saud University, Riyadh, SA*

Abstract

Researchers are motivated to build effective Intrusion Detection Systems because of the implications of malicious actions in computing, communication, and cyber-physical systems (IDSs). In order to develop signature-based intrusion detection techniques that are suitable for use in cyber-physical environments, state-of-the-art supervised learning algorithms are devised. The main contribution of this research is the introduction of a signature-based intrusion detection model that is based on a hybrid Decision Table and Naive Bayes technique. In addition, the contribution of the suggested method is evaluated by comparing it to the existing literature in the field. In the preprocessing stage, Multi-Objective Evolutionary Feature Selection (MOEFS) feature selection has been used to select only five attack features from the recent CICIDS017 dataset. Keeping in view the class imbalance nature of CICIDS2017 dataset, adequate attack samples has been selected with more weightage to the attack classes having a smaller number of instances in the dataset. A hybrid of Decision Table and Naive Bayes models were combined to train

* Corresponding author:
Email address: nayak.soumya17@gmail.com (Soumya Ranjan Nayak)

Preprint submitted to Computer Communication December xx. 2021

and detect intrusions. Detection of botnets, port scans, Denial of Service (DoS)/Distributed Denial of Service (DDoS) attacks, such as Golden-Eye, Hulk, Slow httptest, slowloris, Heartbleed, Brute Force attacks, such as Patator (FTP), Patator (SSH), and Web attacks such as Infiltration, Web Brute Force, SQL Injection, and XSS, are all successfully detected by the proposed hybrid detection model. The proposed approach shows and accuracy 96.8% using five features of CICIDS2017, which is higher than the accuracy of methods discussed in the literatures.

Keywords: Intrusion Detection, DTNB, Signature-based, Botnet Detection, Denial of Service (DoS) attacks, Distributed Denial of Service (DDoS) attacks, Web Attacks, Class Imbalance

1. Introduction

Cyber-Physical Systems (CPS) are composed of resources such as sensors, actuators, control processing units, and many other communication devices. More and more critical infrastructures are being equipped with CPS in the modern era of computing, which enables real-time data processing Networked agents of CPS play a critical role in adopting and practicing online data processing. The sensitive data resides across many computational stations of CPS. The scattering of sensitive data across the systems attracts hackers and other automated malicious tools. These intrusions aim to sneak into the users' systems and take control of the computational activities. Most of these threats are propagated through the Internet and other network typologies. However, many security mechanisms exist [1]–[4] to counter these threats, but the unpredictable actions of these threats become a nightmare for network engineers and security experts. In many cases, these external threats stand far ahead of the existing security mechanisms, such as firewalls [5]. Firewalls and other specialized protocols [6] are not always sufficient to detect these undercover threats. The firewall follows a static access control policy and is not quickly adaptable to outside attacks [7]. Moreover, a firewall is responsible for evaluating and preventing intrusions of one entity at a time [8], [9]. These prevention schemes are not designed to monitor the collective behavior of legitimate packets sent multiple times. Therefore, as an alternate solution, Intrusion Detection Systems (IDS) come into the picture. An IDS reviews, controls, analyzes, and represents reports about any suspicious events in the system and network activities

In a typical cyber physical system, to secure the network resources, firewalls and IDSs as deployed as layers [10], where the firewall becomes the first layer of the infrastructure, and the IDS is the second layer of defense. Due to this, firewalls are being adopted as anomaly detection systems [11], and the IDSs are precisely designed as signature-based detection engines. Signature-based Intrusion Detection (SID) mechanisms are mostly used to detect known attacks. The detector detects malicious instances using well-known attack patterns. Furthermore, unsupervised learning schemes [12], [13], as well as supervised learning schemes [14]–[16], have the potential to serve as IDS engines.

IDSs present a slew of inherent issues throughout the detection process. One such issue is the sheer number of features. IDS classifiers have to develop models for a large number of characteristics from raw data, which considerably increases the time required to build the detection model. When training data consists of a huge range of instances [17], the situation deteriorates further. Additionally, it takes significant amount of time for the detection model to identify patterns in the data it receives. All of the features in the dataset are not required for an accurate classification of examples. Even when only a few features are included in a classification model, it is possible to classify the samples with the same accuracy rate. Another issue with IDS is the preparation of training data. In most training datasets, there is a high degree of class imbalance [18]. High class imbalance datasets are regarded to be skewed towards the majority classes [19]–[21]. As a result, the detection model generates a large number of spurious alarms. When building an IDS, the issue of high-class imbalance raises the question of which classification model to use for threat detection. Detection engines for IDS use classification approaches as the foundation of the detection process. The ability of an IDS to classify threats in even int the most difficult of scenarios is critical to its elegance. With high-class imbalanced datasets, an IDS classification model should be capable of detecting potential threats with the utmost degree of accuracy.

Taking into account the issues stated above, a signature-based IDS has been developed for detecting network threats. The article's primary contribution is a hybrid detection technique based on Decision Tables and Naive Bayes (DTNB) that operates efficiently on only five important features of the CICIDS2017 dataset as determined by the Multi-Objective Evolutionary Feature Selection (MOEFS) scheme. The model is capable of detecting a broad range of network attacks, including Botnet, Port Scan, Denial of Service (DoS) attacks such as GoldenEye, Hulk, Slow httptest, slowloris, and Heartbleed, Brute Force attacks such as Patator (FTP), Patator (SSH), and Web attacks such as Infiltration and SQL Injection. The ability of both decision table and naïve bayes makes the proposed hybrid approach suitable for both binary and multiclass attacks environment. The rest of the article is arranged as follows. Section 2 conducted a detailed analysis of recently proposed signature-based intrusion detection models using the CICIDS2017 dataset. Section 3 describes the hybrid intrusion detection model that is presented. Section 4 discusses the outcome and subsequent examination of the proposed detection method, followed by a conclusion in Section 5.

2. Related Work

Several researchers have attempted to lay out numerous ideas for capturing intrusion events in a host-based [18], [19], or network-based scenario [17], [24]–[26]. Most of these models have their own advantages, disadvantages, and research gaps. For instance, an artificial neural network-based network IDS [27] was proposed to detect network threats of the CICIDS2017 dataset. The model detects attacks with the best accuracy ever, at 99.9%. However, the system suffers from numerous shortcomings. First, the model uses a binary detection approach using a multilayer perceptron on a multiclass dataset. Secondly, the proposed model is silent on the feature selection scheme. The dataset contains more than 80 features; therefore, designing an IDS considering all the features is practically impossible. A deep learning strategy for IDS[28] has been proposed using the Long Short Term Memory (LSTM) network associated with Convolutional Neural Networks. The proposed LSTM and CNN hybrids have a precision of 98.44% and an accuracy of 97.16%. The LSTM+CNN hybrid model [28], like the prior artificial neural network model [27] is also silent on the feature selection technique. Moreover, the dataset considered in the LSTM+CNN hybrid model is enormous. A realistic system is not feasible unless a reasonable number of samples is considered. Shallow Neural Network (SNN) and Deep Neural Network (DNN) based IDSs are designed using neural networks [29]. Considering all the features of the CICIDS2017 dataset, the SNN and DNN model reveals the accuracy of 98.05% and 98.40%. Subsequently, with ten features, the SNN and DNN model shows 91.08% and 94.72% accuracy, respectively. It is observed from the analysis that both the SNN and DNN approach fails to detect Web Attacks, Infiltration, and Brute force attacks. These attacks appeared to be passed as false negatives. Zhang et al. [30] proposed a real-time IDS called Distributed Random Forest based System (DRFBS) for high-speed networks. The model employs a bootstrap sampling scheme for the large CICIDS2017 dataset. DRFBS model on bootstrap training sample proved to be fast, which took only 0.01s while detecting intrusions. The promptness of the model, on the other hand, compromises the detection capability of the model to a certain extent. The model exhibits a precision of 96.4% with a recall of 96.9%. A hierarchical approach of decision trees and rule-based classifiers has been proposed for designing IDS [31]. The proposed model proved to be efficient as compared to other supervised learning schemes. The hierarchical model is based on data preprocessing, training, and testing of attack and benign instances of the CICIDS2017 dataset. The detection model shows an accuracy of 96.665% accuracy, with a detection rate of 94.475% and a nominal false alarm of 1.145%. The author took 40000 instances each for training and testing instances of the CICIDS2017 dataset. When developing the training and testing model, great care was taken to ensure that all attack instances were included in both the training and testing sets. An ensemble of decision trees [32] on the top of correlation-based feature selection detects network threats with magnificent detection and accuracy rate. The detection model entrusted on 13 features of the CICIDS2017 datasets to achieve 96.8% accuracy. The model also recognizes 95.3% accuracy for all the features of the same dataset. Ensemble learning has also been implemented using Support Vector Machine (SVM) for abnormal behavior detection [33]. The created model uses a distributed approach to find anomalous behavior from large-scale network traffic data by combining deep feature extraction and multi-layer ensemble SVMs. The ensemble of SVMs shows precision and recall of 90.4% and 95.65%, respectively.

Bamakan et al.[34] established an accurate and robust approach for detecting intrusions using Classification and Regression Ramp Loss K-Support (Ramp-KSVCR). KSVCR's Ramp loss function addresses skewed attack distributions. The model may be readily scaled up with minimal training time. This procedure was determined to be the fastest and most accurate. The correct feature selection system can boost detection rate and accuracy even more. Kabir et al. [35] presented an IDS based on Least Square Support Vector Machine (LS-SVM). The entire dataset is given random subclasses at the start of the process. The IDS uses a selection of samples to accurately represent the entire dataset. LS-SVM then discovers intrusions in the subset of data utilized to detect intrusions. The suggested model was tested on both binary and multiclass KDD99 datasets. In spite of their high accuracy rate, the KDD99 dataset used here contains earlier assaults. A recent attack dataset may thus be utilized to analyze system performance. Similarly, A LSSVM was also explored by Ambusaidi et al. [36] to design an effective IDS. Feature selection was carried out to speed up classification. Using world-class benchmark datasets, LSSVM-IDS yields accuracy levels of 99.94%.

Similarly, due to the low time complexity decision trees are frequently used to detect intrusions [37], [38]. A Snort-based intrusion detection system using on decision trees [38] has been proposed where the Snort priority levels were determined by analyzing real-world attacks on high-speed networks. Only three features in the ISCXIDS2012 dataset are capable of detecting threats with a 99.99% success rate. Akyol et al. [39] have developed an intrusion detection system with multiple layers of detection. The C4.5-based decision tree classifier and the MCP classifier were used as part of the hybrid strategy. The authors also presented a new feature selection method based on the discernibility function. To test this, they used the KDD'99Cup and ISCX datasets to evaluate the performance of their approach. There are only 0.03% false alarms with their hybrid method, which yields a detection rate and accuracy of 99.50%. NeuroC4.5, a C4.5 decision tree built on top of a neural network, outperforms the regular C4.5 decision tree in detecting threats [40]. In order to detect network breaches in real time, classification rules derived from audit data using NeuroC4.5 have been used. According to the KDD99 dataset, the NeuroC4.5 is 94.55% accurate in detecting Denial of Service (DoS) attacks. A Non-Symmetric Deep Auto-Encoder (NDAE) based unsupervised feature learning was augmented by Random Forest classifiers [41] smartly detect intrusion using the features of KDD Cup '99 and NSLKDD datasets. Less training time is required for the proposed NDAE architecture. However, the authors acknowledge that the method is not ideal and could be improved.

In a nutshell it is observed that there exists many potential supervised, unsupervised and hybrid IDSs to counter intrusions in a cyber physical system environment. However, the industry is interested in several more hybrid models for developing cutting-edge IDS.

3. Materials and Methods

In this section first the problems associated with typical IDSs and the overall method we used to counter such challenges has been outlined. Subsequently, various steps of the proposed intrusion detection model such as data preprocessing and sampling procedures are explained in detail.

3.1. Problem definition and the proposed method

The detection process for IDSs is riddled with man difficulties. An example of this is that there are so many features of attacks. IDS classifiers must create effective models from raw data from large number of features. On the other hand, attacks having a large number of features may hamper the IDSs' detection performances. Similarly, an IDS trained on a large number of examples may suffer to detect any incoming attack pattern. Preparation of training data is yet another issue with IDS. The high-class imbalance issue can be found in the majority of training datasets. According to this classification and detection model, datasets with a large class imbalance are likely to be skewed toward the majority classes. Therefore, an effective IDS must be able to identify and classify threats even in the most challenging of situations.

As a proposed solution to the problem discussed above an IDS framework has been proposed and presented in Figure 1.

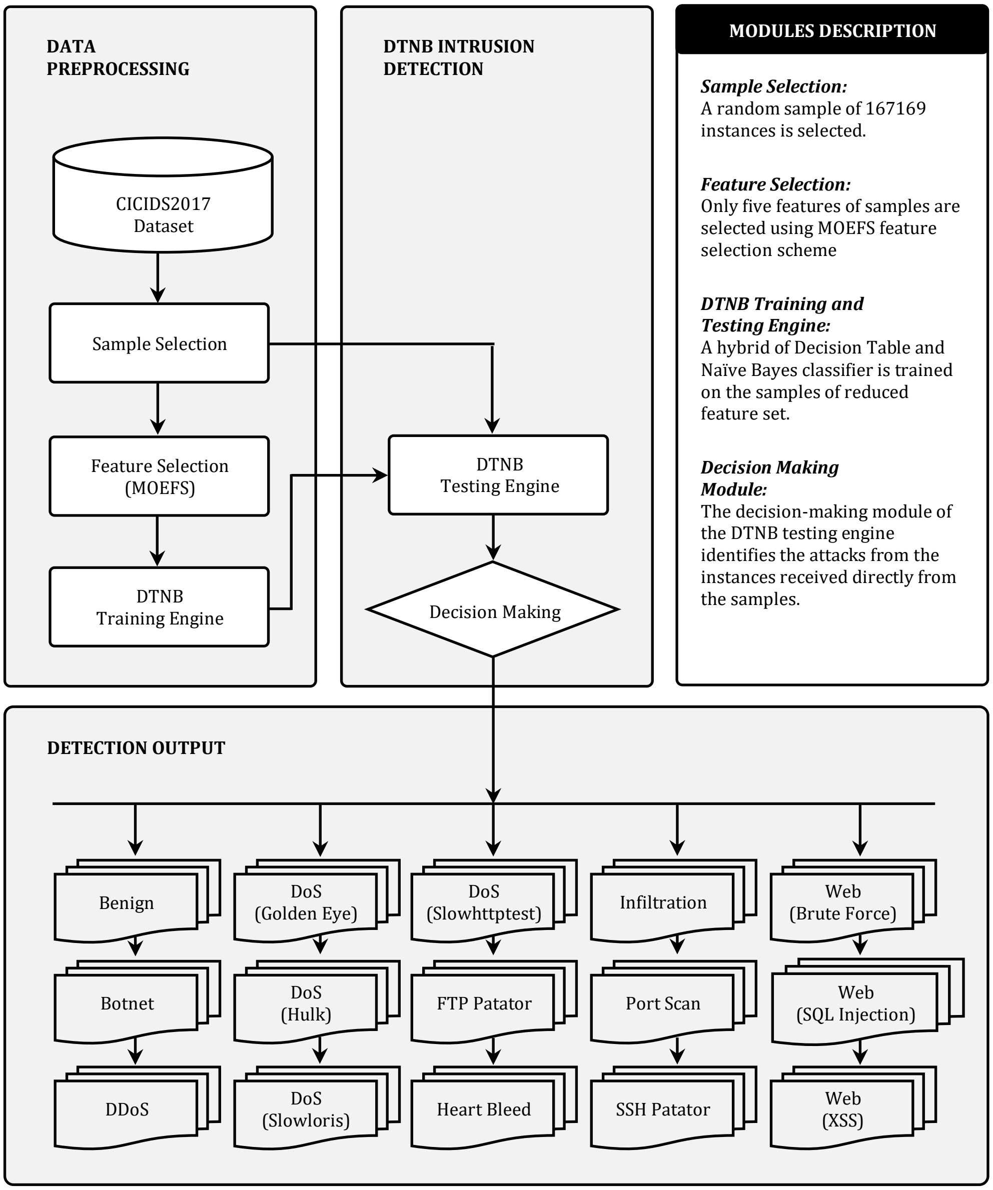
DATA PREPROCESSING
CICIDS2017 Dataset
Sample Selection
Feature Selection (MOEFS)
DTNB Training Engine
DTNB INTRUSION DETECTION
DTNB Testing Engine
Decision Making
MODULES DESCRIPTION
Sample Selection:
A random sample of 167169 instances is selected.
Feature Selection:
Only five features of samples are selected using MOEFS feature selection scheme
DTNB Training and Testing Engine:
A hybrid of Decision Table and Naïve Bayes classifier is trained on the samples of reduced feature set.
Decision Making Module:
The decision-making module of the DTNB testing engine identifies the attacks from the instances received directly from the samples.
DETECTION OUTPUT
Benign
Botnet
DDoS
DoS (Golden Eye)
DoS (Hulk)
DoS (Slowloris)
DoS (Slowhttptest)
FTP Patator
Heart Bleed
Infiltration
Port Scan
SSH Patator
Web (Brute Force)
Web (SQL Injection)
Web (XSS)

Figure 1: A schematic diagram of the proposed DTNB intrusion detection model.

The idea behind the proposed model is that a reasonable number of samples of the CICIDS2017 dataset have been selected for feature selection. The feature selection module identifies the best possible features and contributes the maximum amount towards the detection process. The DTNB training engine has been trained in three ways; (a) using all the selected samples, (b) using 66% of the samples, where 34% of the samples are kept reserved for testing the detection engine later, and (c) through 10-fold cross-validation. Overall, the detection model detects instances as one of the 14 attacks or labeled benign, as shown in figure 1. The suggested model's three primary steps, sample selection, feature selection, and intrusion detection, are explained in the following section.

3.2. Sample Selection

The data preprocessing module starts with sample selection. The CICIDS2017 [42] intrusion dataset released by the Canadian Institute of Cyber Security was used in this study. The CICIDS2017 dataset contains benign and 14 recent attack information. The dataset claims to meet all 11 requirements of an IDS as described by Gharib et al. [43]. Using these IDS dataset design criteria, CICIDS2017 looks to be the most relevant dataset. The original CICIDS2017 dataset contained five days of normal and attack traffic data in eight separate files. There are 3119345 instances in the dataset and 15 class labels (one normal, 14 attack labels) when the day-by-day files are combined. In addition, 288602 instances of the combined files were discovered to have missing class labels and 203 occurrences with missing content. A unique set of 2830540 instances was realized by deleting the missing instances. The benign and attack instances' contribution to the dataset is presented in Table 1.

Table 1: Benign and attack instances contribution in CICIDS2017 dataset.

Benign & Attack Labels	Number of Instances	Imbalance Ratio
Benign	2359087	5.004
Bot	1966	0.001
Brute Force (Web Attack)	1507	0.001
Distributed Denial of Service (DDoS)	41835	0.015
GoldenEye (DoS)	10293	0.004
Heartbleed	11	0.000
Hulk (DoS)	231072	0.089
Infiltration	36	0.000
Patator (FTP)	7938	0.003
Patator (SSH)	5897	0.002
Port Scan	158930	0.059
Slowhttptest (DoS)	5499	0.002
Slowloris (DoS)	5796	0.002
Sql Injection (Web Attack)	21	0.000
XSS (Web Attack)	652	0.000

The last column of Table 1 represents the class imbalance ratio of the corresponding attack and benign class. Considering each class as positive and the rest of the classes as negative, the Class Imbalance Ratio (CIR) of a class C_k of the dataset having n number of instances can be estimated as:

$$CIR = \frac{|C_k|}{n-|C_k|} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (1)$$

The CICIDS2017 dataset has a high-class imbalance in nature, as shown in Table 1. The benign instances contribute 2359087 instances with a CIR of 5.004, whereas the minority classes like Heartbleed, Infiltration, SQL Injection and XSS hold approximately 0 CIR. Therefore, a reasonable number of samples are selected from vast instances of CICIIDS2017 using reciprocal down sampling technique presented in equation 2. The required number of samples is selected in such a way that the attack instances of the sample maintain the same ratio of contribution as that of the original dataset. This is possible through equation 2.

$$W_c[P] = 100 - \left[\frac{sfC[p]}{|stepS_c|} * 100\right] \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (2)$$

Where $W_c[P]$ represents the weight allocated to each attach class (p) for sample contribution, $stepS_c =$ stepwise total instances for all classes, $sfC[p]$ represents the number of instances for each class. Using equation 2 a reasonable number of 167169 has been selected. The selected samples maintain the same contribution ratio as that of the original dataset, but it also helps to avoid the difficulty of processing the entire sample set.

3.3. Feature Selection

After obtaining reasonable samples, a new feature selection scheme known as Multi-Objective Evolutionary Feature Selection (MOEFS) [44] was used to pick a subset of features. It should be mentioned that the CICIDS2017 dataset has a total of 83 features, all of which are statistically insignificant for classification tasks. Therefore, the MOEFS algorithm was utilized, which selected five statistically relevant features from the CICIDS2017 dataset's 83 features. The "Fwd IAT Mean", "Bwd IAT Total", "Subflow Bwd Bytes", "Init win bytes forward" and "min seg size forward" are the chosen features of the sample set. Table 2 shows the relevance of the selected features.

Table 2: Features selected through MOEFS with its corresponding meaning.

Features Selected	Feature Description
Fwd IAT Mean	The average time taken for two packets to be transmitted forward in the same direction.
Bwd IAT Total	The sum of all Inter Arrival Times in the reverse direction.
Subflow Bwd Bytes	An average number of bytes in a sub-flow in the reverse direction.
Init win bytes forward	Total number of bytes transferred in the forward direction in the initial window
min seg size forward	Minimum segment size recorded in the forward direction

3.4. Detection Approach

The selected features of MOEFS feature selection are then sent to a hybrid classification scheme called DTNB [45]. The DTNB approach is the heart of this detection engine. The class probability of DTNB is predicted as −

$$Q(y|X) = \alpha \times Q_{DT}(y|X^{\top}) \times Q_{NB}(y|X^{\perp})/Q(y) \qquad \ldots\ldots\ldots\ldots\ldots\ldots \qquad (3)$$

where,

$Q_{DT}(y|X^{\top})$	=	Class probability estimates of Decision Table
$Q_{NB}(y|X^{\perp})$	=	Class probability estimates of Naïve Bayes
α	=	Normalization constant
$Q(y)$	=	Prior probability of the class
$X^{\top}$	=	Set of attributes in the Decision Table
$X^{\perp}$	=	Set of attributes in Naïve Bayes

4. Results and discussions

The proposed methodology was implemented on the Param Shavak supercomputing facility established by India's Centre for Development of Advanced Computing (CDAC) using Java on the CentOS platform. The supercomputer has 64 GB of RAM and two multicore CPUs, each with 12 cores and 2.3 teraflops of capability. The samples of the CICIDS2017 dataset have been trained and tested in three ways. First, the entire samples are split into 66% training and 34% testing instances. In the second stage, the entire training set is considered for both training and testing purposes. Finally, 10-fold cross-validation was used to evaluate the suggested model's performance. Performance is measured using classification accuracy, false alarm rate, precision, and area under the curve. Finally, the suggested model is compared to currently available state-of-the-art IDSs.

4.1. Analysis through all the instances of the sample

In this analysis, the entire sample of 167169 has been treated as both training and test instances. The detection model is trained through all the sample instances, followed by the testing of the same instances to identify the attack labels. The detailed observations of this test are presented in Table 3.

Table 3: Performance of the proposed model through entire training instances as test instances

Benign & Attack Labels	Precision	Recall	FPR
Benign	0.997	0.922	0.003
Bot	0.908	0.954	0.000
Brute Force (Web Attack)	0.676	0.963	0.001
Distributed Denial of Service (DDoS)	0.745	0.975	0.017
GoldenEye (DoS)	0.996	0.966	0.000
Heartbleed	0.000	0.000	0.000
Hulk (DoS)	0.999	0.997	0.000
Infiltration	1.000	0.286	0.000
Patator (FTP)	0.992	0.977	0.000
Patator (SSH)	1.000	0.980	0.000
Port Scan	0.997	0.999	0.001
Slowhttptest (DoS)	0.769	0.909	0.002
Slowloris (DoS)	0.914	0.740	0.000
Sql Injection (Web Attack)	0.000	0.000	0.000
XSS (Web Attack)	0.004	0.108	0.021

Observing Table 3, it is found that the system can detect almost all attacks and benign instances successfully, except for Heartbleed and Web-SQL Injection. The system seems to be failed at these two attacks. This inference can be visualized through the classification diagram presented in Figure 2. The X axes represent the actual attack classes, and the Y axes represent the detected attacks by the detector. Again, in the diagram, the colored cross symbol indicates correctly detected attacks, and the small square represents missed attacks, where the detector fails to predict the attacks correctly. A similar interpretation is applicable to Figure 4 and Figure 6.

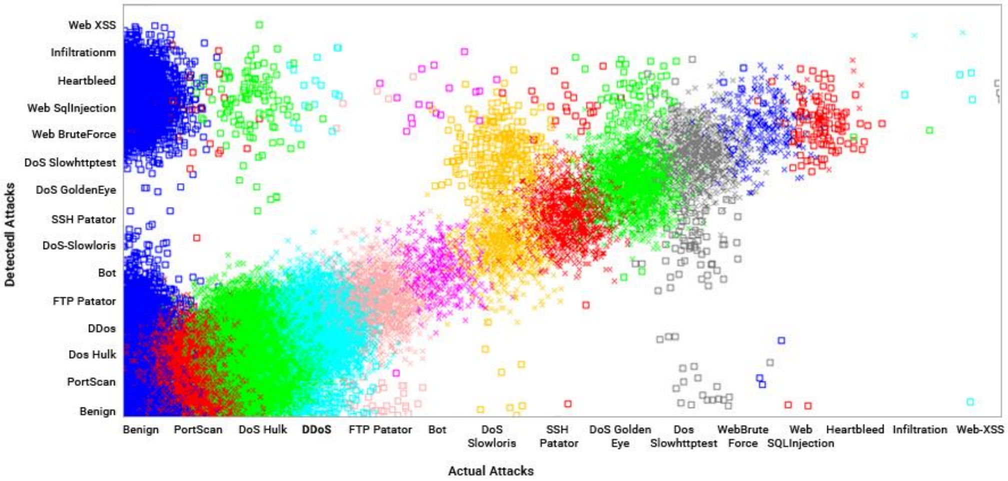

Figure 2: Classification of attacks and benign instances considering entire samples

From Figure 2, it is evident that the proposed model performs satisfactorily in majority attack classes but poorly in minority class instances. One of the reasons behind this may be the class imbalance issue, which was not properly addressed during the preprocessing stage. Although, an attempt has been made to counter class imbalance problem before the feature selection happens, but it is not appearing to be sufficient in this case. Therefore, at the second stage of this analysis, the Receiver Operating Curve (ROC) curve of all the attacks has been plotted to understand whether the system fails at the minority class attacks like Heartbleed and Web-SQL Injection. Table 4 shows the Area Under Curve (AUC) of the ROC of benign and attack labels.

Table 4: Area under curve of benign and other attacks detected under 167169 training and test instances

Labels (Attacks and Benign)	AUC
Benign	0.989
Bot	0.999
Distributed Denial of Service (DDoS)	0.993
GoldenEye (DoS)	1.000
Slowhttptest (DoS)	0.998
Slowloris (DoS)	0.999
Hulk (DoS)	1.000
Patator (FTP)	1.000
Heartbleed	0.990
Infiltration	0.994
Port Scan	1.000
Patator (SSH)	1.000
Brute Force (Web Attack)	0.999
Sql Injection (Web Attack)	0.990
XSS (Web Attack)	0.998

The ROC Curves of all the attacks and benign instances have been presented in Figure 3. According to Figure 3, the proposed model shows better ROC values for minor attack classes, which seems to contradict the inference observed in Table 3 and Figure 2. According to Table 3. For an instance, the precision, recall and false positive rate ascertained for minor attack labels like Heartbleed attack is not encouraging, whereas the receiver operating curve shows an impressive ROC of 0.990. This contradiction is probably due to the presence of class imbalance issue in the dataset. Therefore, the second step of the experiment was conducted and analyzed in section 4.2 to determine the actual cause of the conflict.

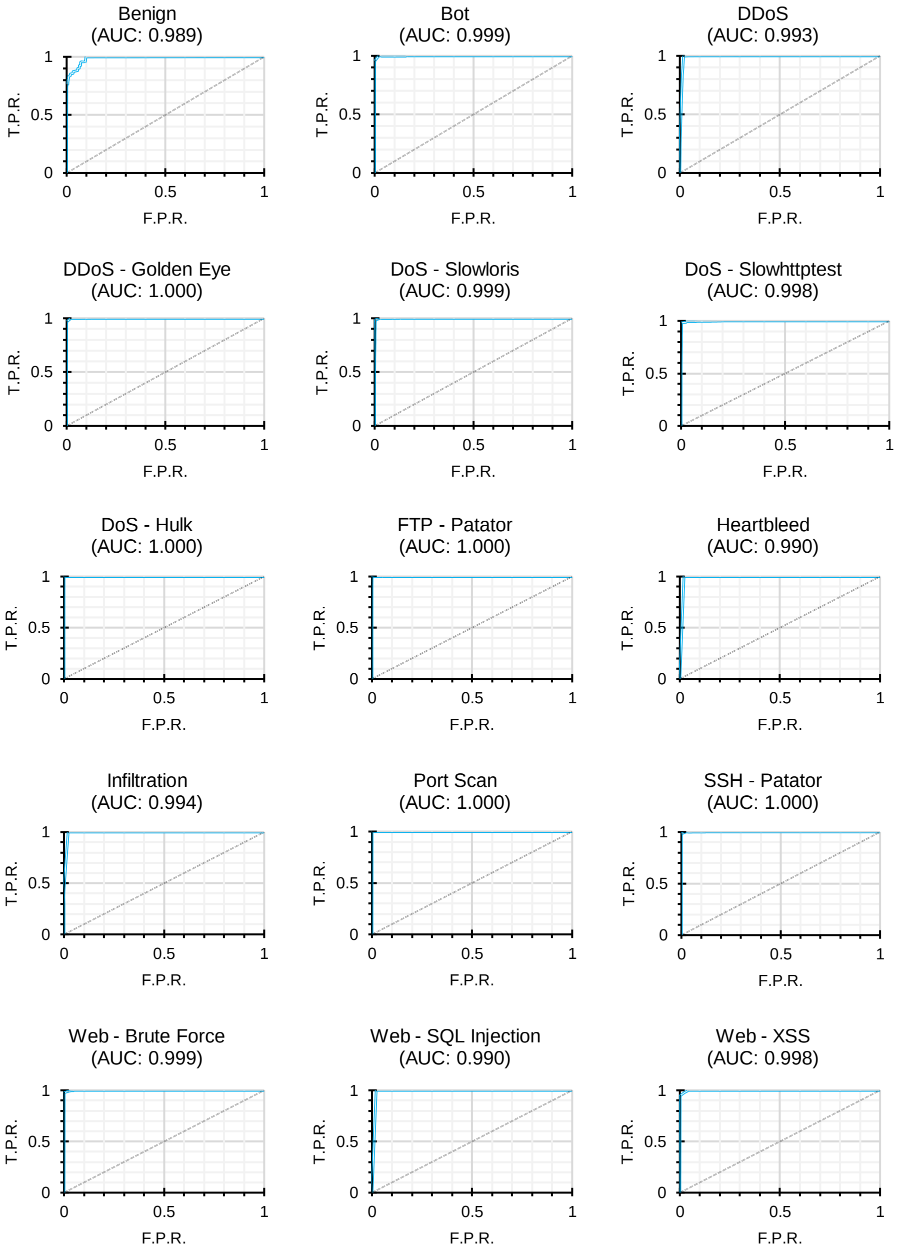

Figure 3: Receiver operating curve of benign and other attacks detected under 167169 training and test instances

4.2. Analysis through training and testing instance split

It should be noted that considering all the instances as training and test instances is not ideal. In a real environment, the incoming instances may not be known to the attack detection module. Therefore, in this analysis, the total of 167169 samples is split into 66% and 34% of training and testing instances respectively [46]–[49]. Thus, a total of 110332 training instances and 56837 testing instances are prepared. This analysis is to observe any possible deviation compared to the earlier analysis of section 4.1. The performance outcome of the proposed model through 56837 testing instances is presented in Table 5.

Table 5: Performance of the proposed model through 66% training and 34% testing instances

Benign & Attack Labels	Precision	Recall	FPR
Benign	0.992	0.933	0.007
Bot	0.898	0.820	0.000
Brute Force (Web Attack)	0.664	0.908	0.001
Distributed Denial of Service (DDoS)	0.740	0.976	0.018
GoldenEye (DoS)	1.000	0.947	0.000
Heartbleed	0.000	0.000	0.000
Hulk (DoS)	0.999	0.995	0.000
Infiltration	0.000	0.000	0.000
Patator (FTP)	0.993	0.972	0.000
Patator (SSH)	0.997	0.935	0.000
Port Scan	0.997	0.998	0.001
Slowhttptest (DoS)	0.725	0.934	0.002
Slowloris (DoS)	0.964	0.634	0.000
Sql Injection (Web Attack)	0.000	0.000	0.000
XSS (Web Attack)	0.001	0.024	0.015

Table 5 shows that the proposed DTNB+MOEFS model can detect all DoS attacks and Bot, Port Scan, Patator (SSH), and Patator (FTP). Brute Force and Slowhttptest attacks are also moderately detected by the system. At the same time, the suggested model is ineffective in detecting Heartbleed, Infiltration, and SQL Injection attacks, which supports the previous observation (Table 3). A classification diagram provided in Figure 4 demonstrates, on the other hand, that splitting instances into training and test detects attacks better than the detector, which relies on all the instances.

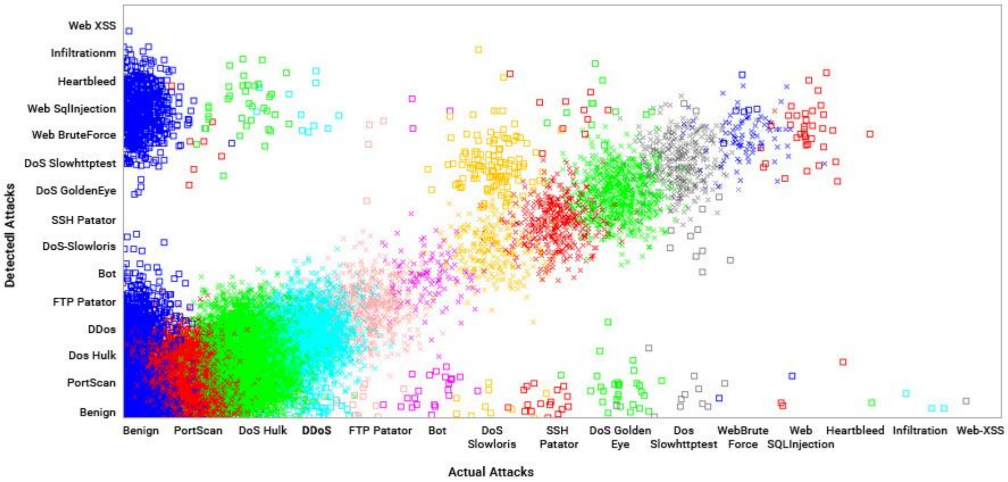

Figure 4: Classification of attacks and benign instances considering 66% training and 34% testing samples

To analyze further, the ROC of each attack classes has been plotted again. The ROC curve of attack classes is demonstrated in Figure 5. The AUC of benign and attack labels are also presented in Table 6.

Table 6: Receiver operating curve of benign and other attacks detected under 110332 training and 56837 test instances.

Labels (Attacks and Benign)	AUC
Benign	0.989
Bot	0.938
Distributed Denial of Service (DDoS)	0.993
GoldenEye (DoS)	0.992
Slowhttptest (DoS)	0.992
Slowloris (DoS)	0.990
Hulk (DoS)	1.000
Patator (FTP)	0.999
Heartbleed	0.012
Infiltration	0.012
Port Scan	1.000
Patator (SSH)	0.986
Brute Force (Web Attack)	0.992
Sql Injection (Web Attack)	0.012
XSS (Web Attack)	0.978

The ROC of each attack agrees with the inferences as those of Table 3 and Table 5. All the minor class attacks, i.e., Heartbleed, Infiltration, and Web-SQL Injection, are challenging to detect. The DoS-Hulk and Port Scan attacks are detected correctly with a ROC value of 1.00.

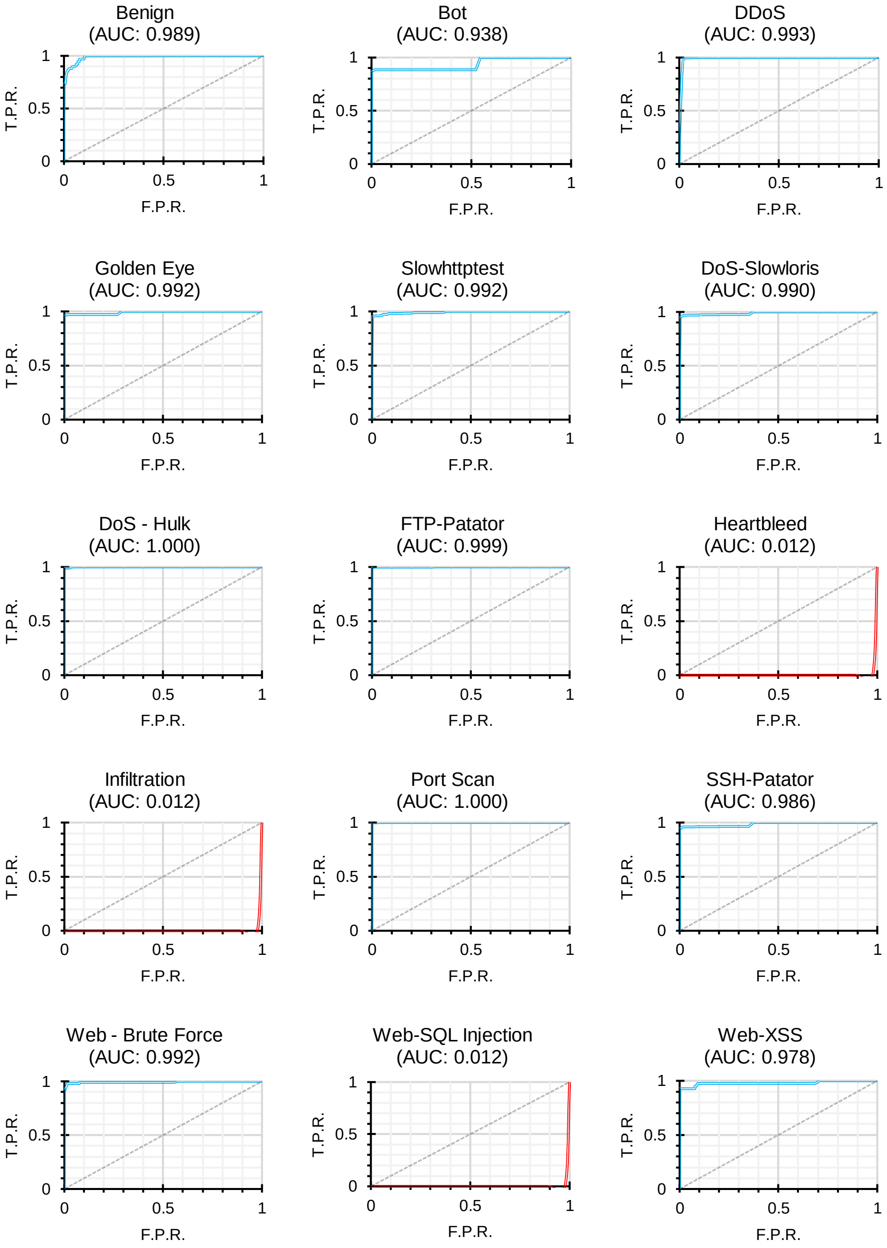

Figure 5: Receiver operating curve of benign and other attacks detected under 110332 training and 56837 test instances.

4.3. Analysis through 10-fold cross-validation

The entire sample set of 167169 occurrences was randomly divided into ten blocks for 10-fold cross-validation, with one block used to train the model and the remaining nine blocks sent for testing. Finally, the test results of all the nine blocks of samples are averaged to determine the model's performance. The *k*-fold cross-validation has been considered as a practical approach for understanding the performance of an IDS. Table 5 summarizes the results of the 10-fold cross-validation for each attack class, and Figure 7 depicts the ROC curves for each assault class.

Table 7: Performance of the proposed model through 10-fold cross-validation

Benign & Attack Labels	Precision	Recall	FPR
Benign	0.991	0.949	0.008
Bot	0.324	0.906	0.004
Brute Force (Web Attack)	0.659	0.917	0.001
Distributed Denial of Service (DDoS)	0.743	0.971	0.017
GoldenEye (DoS)	0.998	0.939	0.000
Heartbleed	0.000	0.000	0.000
Hulk (DoS)	0.999	0.995	0.000
Infiltration	0.000	0.000	0.000
Patator (FTP)	0.990	0.972	0.000
Patator (SSH)	0.999	0.964	0.000
Port Scan	0.997	0.998	0.001
Slowhttptest (DoS)	0.739	0.882	0.002
Slowloris (DoS)	0.893	0.698	0.001
Sql Injection (Web Attack)	0.000	0.000	0.002
XSS (Web Attack)	0.023	0.046	0.002

The 10-fold cross-validation shows almost similar results as that of the 66%-34% training and testing split. The model again detects all the network attacks, including DoS and DDoS attacks, except Heartbleed, Infiltration, and Web-SQL Injection. The visualization of the classification diagram presented in Figure 6 also reveals the exact inference. However, the false alarms are significantly reduced as compared to the detection model shown in Section 4.1 and Section 4.2.

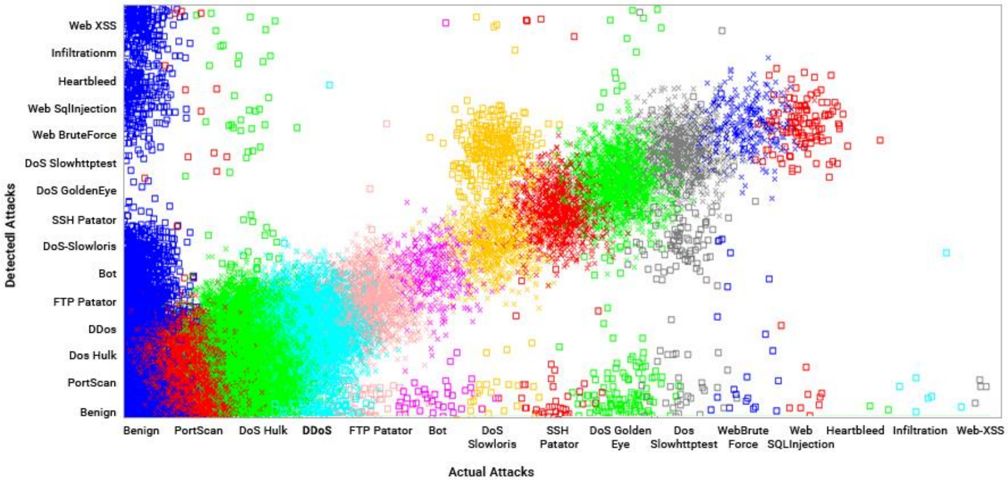

Figure 6: Classification of attacks and benign instances considering 10-fold cross-validation

The ROC curves of all attacks are drawn, showing the inferences in the same direction. under 10-fold cross-validation, the ROC curves of all the attacks demonstrate that the system detects Benign, Bot, DDoS, DoS – Golden Eye, DDoS – Hulk, DoS – Slowhttptest, DoS – Slowloris, Patator (FTP), Port Scan, Patator (SSH), Web-Brute Force, and Web-XSS threats excellently.

Table 8: Receiver operating curve of benign and other attacks detected under10-folds cross-validation

Labels (Attacks and Benign)	AUC
Benign	0.990
Bot	0.979
Distributed Denial of Service (DDoS)	0.993
GoldenEye (DoS)	0.992
Slowhttptest (DoS)	0.995
Slowloris (DoS)	0.995
Hulk (DoS)	1.000
Patator (FTP)	0.999
Heartbleed	0.002
Infiltration	0.423
Port Scan	1.000
Patator (SSH)	0.992
Brute Force (Web Attack)	0.992
Sql Injection (Web Attack)	0.004
XSS (Web Attack)	0.981

It can be seen that, the system suffers from class imbalance issue due to the minor attack classes, i.e., Heartbleed and Web-SQL Injection instances. The system shows moderate ROC for Infiltration attacks. However, this is a normal attacks scenario in a cyber physical system.

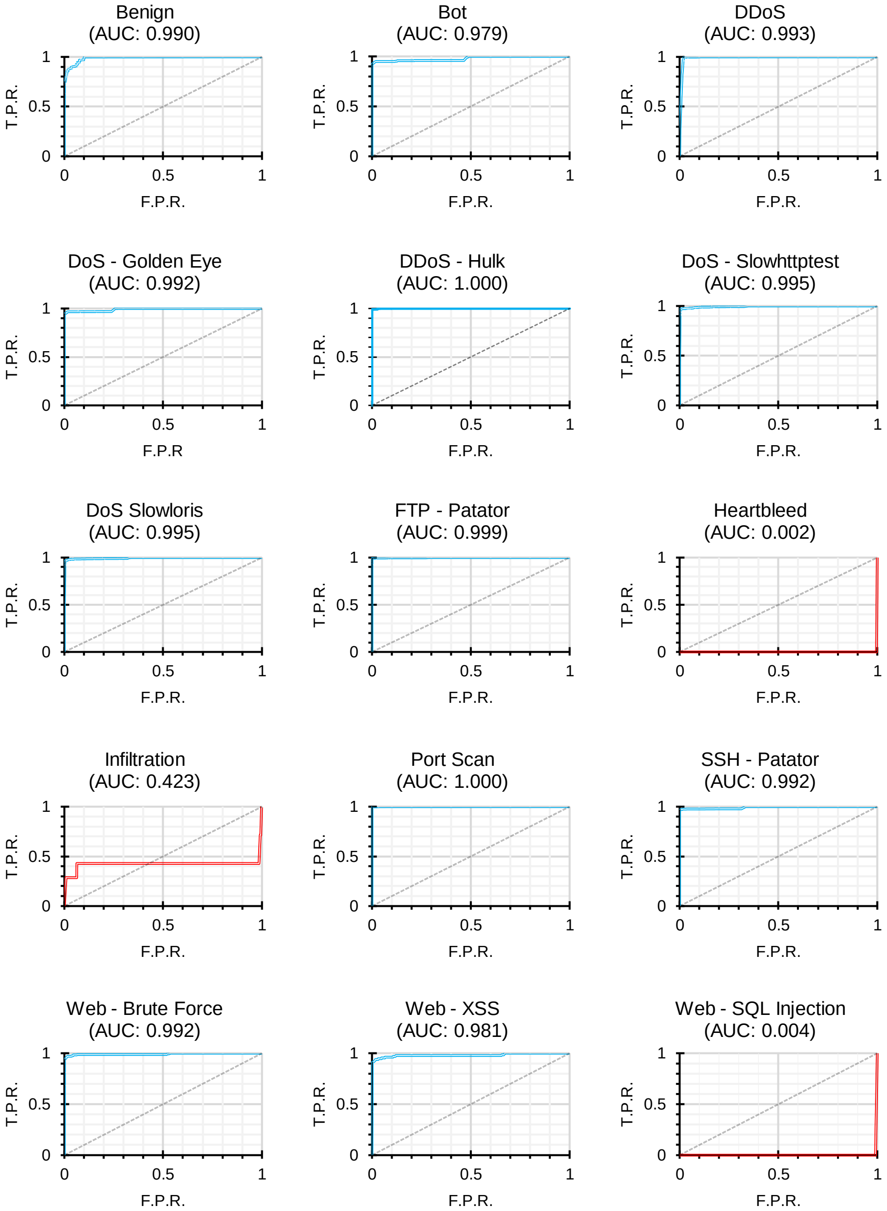

Figure 7: Receiver operating curve of benign and other attacks detected under10-folds cross-validation.

Though the proposed model detects multiple network attacks smoothly, except for a few minor class attacks, the model's overall performance needs to be analyzed along with recent state-of-the-art intrusion detection models, which necessarily reveals its true capability of detection. The following section highlights the comparison of the proposed model with other peer models.

4.4. Comparative analysis along with existing IDSs

In this analysis, the overall result of our proposed DTNB+MOEFS detection model has been analyzed along with MLP Neural Network IDS, LSTM Neural Network-based IDS, and CNN proposed by Roopak et al. [50], SNN and DNN IDS proposed by Ustebay et al. [51], Decision Tree ensemble model with CFS-BA feature selection (DT+CFS-BA) proposed by Zhou et al.[52], Decision Tree and Rule-based Model (DT+RBM) proposed by Ahmim et al. [53], Deep Belief Network (DBN) with Ensemble SVM IDS (DBN+SVM) proposed by Marir et al.[54] and Distributed Random Forest with Apache Spark IDS (DRF+AS) proposed by Zhang et al. [55].

The comparative analysis and discussion have been carried out in two stages. First, an anatomical comparison ensures the proposed detection model is at par with other peer models in terms of the architecture, preprocessing mechanisms deployed, and the attacks detected. Secondly, a more detailed comparison has been carried out on the observed performance of the detectors.

4.4.1. Anatomical analysis with other peer models

As stated earlier, this analysis aims to understand the capability of the proposed IDS with regard to other similar IDS models. An analogical table has been prepared to keep in view the architecture of the IDS, feature selection approach, number of features selected, the sample size for analysis, whether class imbalance issued addressed, and the attacks detected.

Table 9: Anatomical comparison of the proposed scheme with the other state of the art schemes

IDS Mechanisms	Dataset Sample Size	Feature Selection Approach	No of Features	Normalization of Sample	Class imbalance issue addressed	Attacks Detected
DRF+AS	158930	-	13	No	Yes	DDoS, Botnet, PortScan,
DBN+SVM	-	Deep Belief Network	-	No	Yes	DDoS, PortScan
CNN	-	No	-	No	Yes	DDoS,
DT+RBM	Training instances: 40000, Testing instances: 40000	No	79	Yes	Yes	DoS, PortScan, Bot, BruteForce, WebAttack, Infiltration
DNN	Training: 1979513, Testing: 848363	AutoEncoder	10	No	No	BruteForce, DDoS, WebAttacks, Infiltration, Botnet
DT+CFS-BA	Training instances: 40000, Testing instances: 40000	CFS-BA	13	Yes	Yes	DDoS, PortScan, Bot, BruteForce, WebAttack
LSTM	-	No	-	No	Yes	DDoS
MLP	-	No	-	No	Yes	DDoS
SNN	Training: 1979513, Testing: 848363	AutoEncoder	10	No	No	BruteForce, DDoS, WebAttacks, Infiltration, Botnet
DTNB+MOEFS (Proposed)	167169 (Training: 110332 Testing: 56837)	MOEFS	5	No	Yes	Bot, Distributed Denial of Service (DDoS), GoldenEye (DoS), Slowhttptest (DoS), Slowloris (DoS), Hulk (DoS), Patator (FTP), Heartbleed, Infiltration, Port Scan, Patator (SSH), Brute Force (Web Attack), Sql Injection (Web Attack), XSS (Web Attack)

It can be seen from the Table 9 that; the proposed model provides advantages over peer models in three folds.

a) ***Advantages of relevant features:*** The proposed DTNB+MOEFS detection model takes only five features from the large CICIDS2017 dataset; nevertheless, it can detect fourteen attack classes successfully, proving to be a state-of-the-art detector over others. It is the MOEFS feature selection scheme that plays a crucial role during the preprocessing stage. The MOEFS scheme can retrieve the most potential features responsible for attack detection.

b) ***Advantages of small training set:*** The DTNB+MOEFS detection mechanism is a potential IDS since it detects all attack labels even if trained only on 110332 instances, which is far lower than the SNN DNN and DRF+AS detectors. On the other hand, the DT+RBM and DT+CFS-BA detection models require a lower number of training instances than the proposed DTNB+MOEFS detector, but then, these IDSs focus on five attack labels.

c) ***Ability to detect attacks in the presence of class imbalance:*** The attacks and benign sample of the proposed IDS has been considered with the same ratio as the original dataset. Since the class imbalance issue in the original CICIDS2017 dataset exists, the class imbalance issue is also apparent in the sample set. The detector can detect fourteen classes of attacks even if the class imbalance issue exists.

4.4.2. Detailed comparative analysis with other peer models

A more detailed comparative analysis has been conducted in terms of performance observation. The classification accuracy, precision, and detection rate have been considered for practical analysis. Table 10 summarizes the overall performance of the proposed detection and the performance of the other detectors. The missing values (-) indicate the result has not been reported in the concerned articles.

Table 10: Comparison of the proposed scheme with other IDSs in detail.

IDS Mechanisms	Detection Rate	Precision	Accuracy
DRF+AS	96.90%	96.40%	-
DBN+SVM	95.65%	90.40%	-
CNN	90.17%	98.14%	95.14%
DT+RBM	94.48%	-	96.67%
DNN	95.00%	95.00%	94.72%
DT+CFS-BA	94.04%	96.80%	96.76%

LSTM	89.89%	98.44%	96.24%
MLP	86.25%	88.47%	86.34%
SNN	91.00%	91.00%	91.08%
DTNB+MOEFS (Proposed)	96.70%	97.40%	96.80%

It is observed from Table 10 that the proposed model shows a better accuracy rate than other IDS models. Similarly, in terms of precision of detection, both the CNN and LSTM based detection model seem to be better choices than the proposed model. In contrast, the proposed model shows superior precision over the DBN+SVM, MLP, DNN, SNN, and DRF+AS intrusion detection models. However, the situation is flipped when the detection rate of the IDSs is analyzed. In terms of detection rate, the proposed model is far ahead of that of the CNN and LSTM models; thus, it seems to be a better intrusion detection model than others, though it slightly lacks in the DRF+AS models. Therefore, the proposed IDS detects a more significant number of attacks than the DRF+AS detector.

The consistent performance across accuracy, precision, and detection rate makes the proposed model more versatile and efficient than other IDSs. Another reason that makes the proposed IDS unique over other detection models is the choice of features. The proposed model shows better performance with just five features in hand, whereas other approaches under analysis required a significant number of features for intrusion detection. Moreover, the combination of both decision table and naïve bayes makes the detection process robust for both binary and multiclass attack detection scenario.

5. Conclusion

This article has selected the most informative features of the CICIDS2017 dataset to design a multiclass IDS for a cyber-physical environment. The MOEFS has been deployed and selects only five features, such as Fwd IAT Mean, Bwd IAT Total, Subflow Bwd Bytes, Init win bytes forward, and min seg size forward. On top of these features, a hybrid classification mechanism combining the efficiency of DTNB has been used for detecting threats found in network traffic. The proposed system successfully detected benign instances and 11 attack classes, viz., Bot, DDoS, DoS – Golden Eye, DDoS – Hulk, DoS – Slowhttptest, DoS – Slowloris, Patator (FTP), Port Scan, Patator (SSH), Web-Brute Force and Web-XSS of 167169 random instances from the CICIDS2017 dataset. In contrast, it suffers from detecting Heartbleed and Infiltration and Web-SQL Injection attack instances. The system exhibited 96.8% accuracy with just five features, which proved it a better detector than other peers' IDSs. Despite the performance observed, the proposed IDS also has some limitations. One of the shortcomings is the absence of a feedback approach, which could strengthen the system further towards more dynamism.

A feedback approach may empower the admin module to isolate the compromised instances or hosts from the network; thus, making the network communication process stable. During the 10-fold cross-validation test and 66%-34% train and testing instances split test, the system fails to detect minority class attack instances such as Heartbleed, Infiltration & Web-SQL Injection, which remains a challenge to the proposed IDS. It was due to the low instance participation of these minority class instances in the training module. As future work, the detector can be redesigned to work with minority classes, or the minor class attacks can be separated from the original IDS framework, and a specialized IDS can be designed to handle minority class attack instances. One way to handle this issue would be to teach class relabeling. The minority classes with similar characteristics can be merged to form new attack labels, improving attack detection further. For example, the Web – Brute Force, Web - SQL Injection, and Web – XSS attack instances can be merged to have a standard label "Web Attacks". Moreover, other recent feature selection schemes can be explored to improve the IDS's performance further.

Compliance with Ethical Standards

Disclosure of potential conflicts of interest: The authors declare that this manuscript has no conflict of interest with any other published source and has not been published previously (partly or in full). No data have been fabricated or manipulated to support our conclusions.

Research involving Human Participants and/or Animals: This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent: Informed consent was obtained from all individual participants included in the study.

Author Contributions: All authors have equally contributed to this work, read and agreed to the published version of the manuscript.

Funding: This work is supported by Researchers Supporting Project number (RSP-2021/250), King Saud University, Riyadh, Saud Arabia.

References

[1] A. Singh, J. Nagar, S. Sharma, and V. Kotiyal, “A Gaussian process regression approach to predict the k-barrier coverage probability for intrusion detection in wireless sensor networks,” *Expert Systems with Applications*, vol. 172, p. 114603, Jun. 2021, doi: 10.1016/j.eswa.2021.114603.

[2] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, and A. Razaque, “Cascaded hybrid intrusion detection model based on SOM and RBF neural networks,” *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, Nov. 2020, doi: 10.1002/cpe.5233.

[3] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, “Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, Aug. 2020, doi: 10.1109/JIOT.2020.2970501.

[4] S. Rawat, A. Srinivasan, V. Ravi, and U. Ghosh, “Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network,” *Internet Technology Letters*, p. e232.

[5] A. Jain and J. L. Rana, “Classifier Selection Models for Intrusion Detection System (Ids),” *Informatics Engineering, an International Journal (IEIJ)*, vol. 4, no. 1, 2016, doi: 10.5121/ieij.2016.4101.

[6] U. Ghosh and R. Datta, “A secure addressing scheme for large-scale managed MANETs,” *IEEE transactions on network and service management*, vol. 12, no. 3, pp. 483–495, 2015.

[7] X. Zhang, C. Li, and W. Zheng, “Intrusion prevention system design,” in *The Fourth International Conference onComputer and Information Technology, 2004. CIT’04.*, 2004, pp. 386–390.

[8] R. Kenig, “Can your firewall and IPS block DDoS attacks? | Radware Blog.” May 2013.

[9] M. A. Sayeed, M. A. Sayeed, and S. Saxena, “Intrusion detection system based on Software Defined Network firewall,” in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, Sep. 2015, pp. 379–382. doi: 10.1109/NGCT.2015.7375145.

[10] G. Gross, “Intrusion Detection Techniques, Methods & Best Practices | AT&T Cybersecurity.” Feb. 2019.

[11] H. Hu, G.-J. Ahn, and K. Kulkarni, “Detecting and Resolving Firewall Policy Anomalies,” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 3, pp. 318–331, May 2012, doi: 10.1109/TDSC.2012.20.

[12] P. Chatterjee, U. Ghosh, I. Sengupta, and S. K. Ghosh, “A trust enhanced secure clustering framework for wireless ad hoc networks,” *Wireless networks*, vol. 20, no. 7, pp. 1669–1684, 2014.

[13] P. Casas, J. Mazel, and P. Owezarski, “Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge,” *Computer Communications*, vol. 35, no. 7, pp. 772–783, Apr. 2012, doi: 10.1016/j.comcom.2012.01.016.

[14] A. H. Almutairi and N. T. Abdelmajeed, “Innovative signature based intrusion detection system: Parallel processing and minimized database,” in *2017 International Conference on the Frontiers and Advances in Data Science (FADS)*, Oct. 2017, pp. 114–119. doi: 10.1109/FADS.2017.8253208.

[15] S. Dey, Q. Ye, and S. Sampalli, “A machine learning based intrusion detection scheme for data fusion in mobile clouds involving heterogeneous client networks,” *Information Fusion*, vol. 49, pp. 205–215, 2019.

[16] A. Saied, R. E. Overill, and T. Radzik, “Detection of known and unknown DDoS attacks using Artificial Neural Networks,” *Neurocomputing*, vol. 172, pp. 385–393, 2016.

[17] K. Narayana Rao, K. Venkata Rao, and P. R. P.V.G.D., “A hybrid Intrusion Detection System based on Sparse autoencoder and Deep Neural Network,” *Computer Communications*, vol. 180, pp. 77–88, Dec. 2021, doi: 10.1016/j.comcom.2021.08.026.

[18] C. Mera and J. W. Branch, “A survey on class imbalance learning on automatic visual inspection,” *IEEE Latin America Transactions*, vol. 12, no. 4, pp. 657–667, 2014.

[19] S. Wang, L. L. Minku, and X. Yao, “A systematic study of online class imbalance learning with concept drift,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 10, pp. 4802–4821, 2018.

[20] Q. Song, Y. Guo, and M. Shepperd, “A comprehensive investigation of the role of imbalanced learning for software defect prediction,” *IEEE Transactions on Software Engineering*, vol. 45, no. 12, pp. 1253–1269, 2018.

[21] Shuo Wang and Xin Yao, “Multiclass Imbalance Problems: Analysis and Potential Solutions,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012, doi: 10.1109/TSMCB.2012.2187280.

[22] Ch. Gayathri Harshitha, M. Kameswara Rao, and P. Neelesh Kumar, “A Novel Mechanism for Host-Based Intrusion Detection System,” 2020, pp. 527–536. doi: 10.1007/978-981-15-0029-9_42.

[23] R. Gassais, N. Ezzati-Jivan, J. M. Fernandez, D. Aloise, and M. R. Dagenais, “Multi-level host-based intrusion detection system for Internet of things,” *Journal of Cloud Computing*, vol. 9, no. 1, p. 62, Dec. 2020, doi: 10.1186/s13677-020-00206-6.

[24] N. DEMİR and G. DALKILIÇ, “Modified stacking ensemble approach to detect network intrusion,” *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, vol. 26, pp. 418–433, 2018, doi: 10.3906/elk-1702-279.

[25] P. Suresh, R. Sukumar, and S. Ayyasamy, “Efficient pattern matching algorithm for security and Binary Search Tree (BST) based memory system in Wireless Intrusion Detection System (WIDS),” *Computer Communications*, vol. 151, pp. 111–118, Feb. 2020, doi: 10.1016/j.comcom.2019.11.035.

[26] M. S. Al-Daweri, S. Abdullah, and K. A. Zainol Ariffin, “An adaptive method and a new dataset, UKM-IDS20, for the network intrusion detection system,” *Computer Communications*, vol. 180, pp. 57–76, Dec. 2021, doi: 10.1016/j.comcom.2021.09.007.

[27] V. Kanimozhi and T. P. Jacob, “Artificial Intelligence based Network Intrusion Detection with Hyper-Parameter Optimization Tuning on the Realistic Cyber Dataset CSE-CIC-IDS2018 using Cloud Computing,” in *2019 International Conference on Communication and Signal Processing (ICCSP)*, 2019, pp. 33–36.

[28] M. Roopak, G. Y. Tian, and J. Chambers, “Deep Learning Models for Cyber Security in IoT Networks,” in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 452–457.

[29] S. Ustebay, Z. Turgut, and M. A. Aydin, “Cyber Attack Detection by Using Neural Network Approaches: Shallow Neural Network, Deep Neural Network and AutoEncoder,” in *International Conference on Computer Networks*, 2019, pp. 144–155.

[30] H. Zhang, S. Dai, Y. Li, and W. Zhang, “Real-time Distributed-Random-Forest-Based Network Intrusion Detection System Using Apache Spark,” in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, 2018, pp. 1–7.

[31] 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,” in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019, pp. 228–233.

[32] Y.-Y. Zhou and G. Cheng, “An Efficient Network Intrusion Detection System Based on Feature Selection and Ensemble Classifier,” *arXiv preprint arXiv:1904.01352*, 2019.

[33] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, “Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark,” *IEEE Access*, vol. 6, pp. 59657–59671, 2018.

[34] S. M. H. Bamakan, H. Wang, and Y. Shi, “Ramp loss K-Support Vector Classification-Regression; a robust and sparse multi-class approach to the intrusion detection problem,” *Knowledge-Based Systems*, vol. 126, pp. 113–126, 2017.

[35] E. Kabir, J. Hu, H. Wang, and G. Zhuo, “A novel statistical technique for intrusion detection systems,” *Future Generation Computer Systems*, vol. 79, pp. 303–318, 2018.

[36] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, “Building an intrusion detection system using a filter-based feature selection algorithm,” *IEEE transactions on computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[37] N. ben Amor, S. Benferhat, and Z. Elouedi, “Naive bayes vs decision trees in intrusion detection systems,” in *Proceedings of the 2004 ACM symposium on Applied computing*, 2004, pp. 420–424.

[38] A. Ammar, “A decision tree classifier for intrusion detection priority tagging,” *Journal of Computer and Communications*, vol. 3, no. 04, p. 52, 2015.

[39] A. Akyol, M. Hacibeyoglu, and B. Karlik, “Design of multilevel hybrid classifier with variant feature sets for intrusion detection system,” *IEICE Transactions on Information and Systems*, vol. E99D, no. 7, pp. 1810–1821, 2016, doi: 10.1587/transinf.2015EDP7357.

[40] S. S. sivatha Sindhu, S. Geetha, S. Subashini, R. V. Priya, and A. Kannan, “An active rule approach for network intrusion detection with NeuroC4. 5 algorithm,” in *2006 Annual IEEE India Conference*, 2006, pp. 1–5.

[41] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[42] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization,” 2018, doi: 10.5220/0006639801080116.

[43] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “An evaluation framework for intrusion detection dataset,” in *2016 International Conference on Information Science and Security (ICISS)*, 2016, pp. 1–6.

[44] F. Jiménez, G. Sánchez, J. M. García, G. Sciavicco, and L. Miralles, “Multi-objective evolutionary feature selection for online sales forecasting,” *Neurocomputing*, vol. 234, pp. 75–92, 2017.

[45] M. A. Hall and E. Frank, “Combining naive bayes and decision tables.,” in *FLAIRS conference*, 2008, vol. 2118, pp. 318–319.

[46] R. Katarzyniak, *New challenges in applied intelligence technologies*, vol. 134. Springer, 2008.

[47] M. Alenezi and B. Qureshi, *5th International Symposium on Data Mining Applications*, vol. 753. Springer, 2018.

[48] K. Pabreja, “Comparison of different classification techniques for educational data,” *International Journal of Information Systems in the Service Sector (IJISSS)*, vol. 9, no. 1, pp. 54–67, 2017.

[49] R. R. Bouckaert *et al.*, “WEKA manual for version 3-9-1,” *University of Waikato, Hamilton, New Zealand*, 2016.

[50] M. Roopak, G. Y. Tian, and J. Chambers, “Deep Learning Models for Cyber Security in IoT Networks,” in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 452–457.

[51] S. Ustebay, Z. Turgut, and M. A. Aydin, “Cyber Attack Detection by Using Neural Network Approaches: Shallow Neural Network, Deep Neural Network and AutoEncoder,” in *International Conference on Computer Networks*, 2019, pp. 144–155.

[52] Y.-Y. Zhou and G. Cheng, “An Efficient Network Intrusion Detection System Based on Feature Selection and Ensemble Classifier,” *arXiv preprint arXiv:1904.01352*, 2019.

[53] 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,” in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019, pp. 228–233.

[54] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, “Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark,” *IEEE Access*, vol. 6, pp. 59657–59671, 2018.

[55] H. Zhang, S. Dai, Y. Li, and W. Zhang, “Real-time Distributed-Random-Forest-Based Network Intrusion Detection System Using Apache Spark,” in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, 2018, pp. 1–7.