Journal of Database Management
Volume 35 • Issue 1

Intrusion Detection System: A Comparative Study of Machine Learning-Based IDS

Amit Singh, Government of India, India
https://orcid.org/0000-0001-7351-0050

Jay Prakash, Vijay Singh Pathik Government (PG) College, India
https://orcid.org/0000-0002-6167-2412

Gaurav Kumar, Bennett University, India

Praphula Kumar Jain, GLA University, India
https://orcid.org/0000-0001-7651-4444

Loknath Sai Ambati, Oklahoma City University, USA*

ABSTRACT

The use of encrypted data, the diversity of new protocols, and the surge in the number of malicious activities worldwide have posed new challenges for intrusion detection systems (IDS). In this scenario, existing signature-based IDS are not performing well. Various researchers have proposed machine learning-based IDS to detect unknown malicious activities based on behaviour patterns. Results have shown that machine learning-based IDS perform better than signature-based IDS (SIDS) in identifying new malicious activities in the communication network. In this paper, the authors have analyzed the IDS dataset that contains the most current common attacks and evaluated the performance of network intrusion detection systems by adopting two data resampling techniques and 10 machine learning classifiers. It has been observed that the top three IDS models—KNeighbors, XGBoost, and AdaBoost—outperform binary-class classification with 99.49%, 99.14%, and 98.75% accuracy, and XGBoost, KNneighbors, and GaussianNB outperform in multi-class classification with 99.30%, 98.88%, and 96.66% accuracy.

KEYWORDS

Anomaly-Based Intrusion Detection Systems, Cyberattack, Cybersecurity, Intrusion Detection Systems, Machine Learning

1. INTRODUCTION

Because of the Covid-19 pandemic, individuals stayed at home and avoided physical gatherings, and social separation has become the new normal. The usage of new paradigms in corporate transactions, work-from-home culture, and online educational delivery has increased people's reliance on mobile and electronic devices. The use of communication networks and cloud-based processing systems

DOI: 10.4018/JDM.338276
*Corresponding Author

have increased manifold. This change in the pandemic era promotes new threats and lures intruders to exploit vulnerabilities in the data communication network. Organizations usually use diversified protocols to encrypt their data and maintain confidentiality. Volume, heterogeneity of protocols, and encryption have posed several new challenges before the IDS system in detecting malicious activities (Resende & Drummond, 2018; Senthilkumar et al., 2021). An intruder attempts to gain unauthorized access to a system or network with malafide intentions and disrupt the normal execution (Butun et al., 2014; Liao et al., 2013; Low, 2005; Mitchell & Chen, 2014). Several times intruders aim to steal or corrupt sensitive data. In 2020, Emsisoft reported that local governments, universities, and private organizations had spent $144 million in response to the worst ransomware attack (Novinson, 2020). The WHO reported that cyber-attack increased five-fold during the Covid-19 pandemic (WHO, 2020). According to the McAfee quarterly threat report 2020, fraudsters are taking advantage of the pandemic by using Covid-19-themed malicious apps, phishing campaigns, and malware (McAfee, 2020). The report also highlights that in quarter one (Q1), new malware targeting mobile devices surged by 71%, with overall malware increasing by roughly 12% over the previous four quarters (McAfee, 2020).

IDS provides security solutions against malicious attacks or security breaches. It can be a software or hardware device that detects harmful activity to maintain system security (Babu et al., 2023; Liao et al., 2013). It identifies all forms of suspicious network traffic and malicious computer activity that a firewall might miss. Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS) are two popular categories of IDS that have widely been used to provide security solutions (Axelsson, 2000; Baskerville & Portougal, 2003; Hodo et al., 2017). The SIDS relies on previously known signatures and faces challenges in identifying an unknown and obfuscated malicious attack (Amouri et al., 2020; Atli, 2017; Khraisat et al., 2019; Lin et al., 2015; Low, 2005; Vinayakumar et al., 2019; Wu & Banzhaf, 2010). Therefore, SIDS cannot prevent every intruder based on previously learned indicators of compromises; however, they can detect and prevent similar attacks from happening in the future. As the number of cyber-attacks has increased exponentially and attackers are using evolved techniques to conceal attack patterns, it becomes almost infeasible to identify intruders using SIDS (Amouri et al., 2020; Khraisat et al., 2019; Vimala et al., 2019; Warsi & Dubey, 2019; Wu & Banzhaf, 2010).

Many scholars use AIDS because of its ability to overcome the limitation of SIDS. An AIDS is a typical computer system model created using statistical-based methods, machine learning algorithms, or knowledge-based methods. These methods are designed and developed to detect abnormal behaviour in computer systems. The typical usage pattern is base-lined, and alarms are generated when usage deviates from the expected behaviour. The key benefit of using AIDS is detecting zero-day attacks because it does not rely on a signature database to detect abnormal user behaviour (Alazab et al., 2012; Laughlin et al., 2020). AIDS is further categorized into three main groups: Statistics-based, Knowledge-based, and Machine learning-based. Researchers have investigated many approaches to improve intrusion detection in the last few decades, from data mining and machine learning to time series modelling. The Machine learning-based IDS can learn the attacks' behaviour and pattern, and future attacks can be predicted using trained machine learning models.

Machine Learning is a technique for extracting knowledge from massive amounts of data. It comprises a set of rules, methods, or complex "transfer functions" that can be used to discover intriguing patterns or estimate behaviour in a wide range of applications (Abu Al-Haija et al., 2022; Choudhury et al., 2023; Dua & Du, 2016; Mangal et al., 2023; Prasad Yadav et al., 2023; Sinha & Sharma, 2021). The machine learning techniques use training data to acquire complex pattern-matching capabilities. Researchers (Hamzah & Othman, 2021; Hasan et al., 2016; Mehmood et al., 2021; Niyaz et al., 2015; Shams & Rizaner, 2018) widely use the Support Vector Machine (SVM) for Network Intrusion Detection Systems (NIDS) and different clustering algorithms such as K-means and Exception Maximization (EM) for both NIDS and anomaly detection (Bennett & Demiriz, 1999; Laughlin et al., 2020; Maseer et al., 2021; Syarif et al., 2012; Wazid & Das, 2016). They are mainly concerned with the detection effect and lack practical issues such as detection

efficiency and data management. In this paper, the authors have tried to address some problems and highlight the performance of different machine-learning models in IDS. The contributions of this paper are as follows:

1. A new and still under-analyzed IDS dataset containing the most recent common attacks has been used for the analysis. This dataset is more representative of the current threat landscape than older datasets, which can help improve intrusion detection systems' accuracy.
2. The authors have adopted two data re-sampling techniques to balance the dataset, and some pre-processing steps are performed to fix the problems that may exist in the datasets. This is important because imbalanced datasets can lead to biased results, and the researcher's approach helps ensure that their study results are more accurate.
3. The authors proposed to use the ten widely used Machine learning classifiers on intrusion detection systems to find out the best model. This allows for a more comprehensive evaluation of different machine learning approaches, and the study's results can help inform the development of more effective intrusion detection systems in the future.

Overall, the contributions made by this paper provide valuable insights into the performance of Machine learning-based IDS and contribute to the advancement of intrusion detection methodologies. The findings have practical implications for organizations seeking to strengthen their security measures in the face of evolving cyber threats while contributing to network security research's theoretical foundation.

The rest of the paper is structured as follows. Section 2 briefly overviews the work related to the Intrusion Detection System. The machine learning-based intrusion detection approach, data pre-processing, and balancing techniques are explained in section 3. The experimental analysis and results are discussed in section 4. Section 5 concludes the paper with future scope.

2. RELATED WORK

Recently, many research and practical ideas based on artificial intelligence and machine learning have been published to overcome the challenges in intrusion detection systems. The authors (Sharafaldin et al., 2018) used the CICIDS2017 dataset and examined the performance of the selected features with Naive-Bayes, KNN, ID3, RF, Adaboost, MLP, and QDA. Feature selection is an essential process in building IDS systems. Varghese and Muniyal (Varghese & Muniyal, 2017) studied the efficacy of seven different algorithms concerning two different feature selection strategies on the NSLKDD dataset. The authors have used Principal Component Analysis (PCA) and Correlation-based Feature Selection (CFS) for selecting features. Then, the performance of J48, NBTree, Random Forest, LibSVM, Bagging with REPTree, PART, and Multilayer Perceptron (MLP) classifiers were evaluated using ten-fold cross-validation. Effendy et al. (Effendy et al., 2017) also used the NSL-KDD dataset and Information Gain Ratio (IGR) for selecting features. The authors assessed the Naive-Bayes classifier with accuracy as a key performance indicator. The authors (Acharya & Singh, 2018) used intelligent water drops (IWD) nature-inspired algorithm to select the feature and a support vector machine as a classifier to evaluate the selected features. Alazzam et al. (Alazzam et al., 2020) used the pigeon-inspired optimizer technique, and Tawil et al. (Tawil & Sabri, 2021) used the Moth Flame Optimization technique to choose the relevant features in designing the IDS system. The authors (Naseri & Gharehchopogh, 2022) presented a binary version of the Farmland Fertility Algorithm (FFA) called BFFA to select the feature used in IDS classification. The authors (Biswas, 2018) considered the amalgamation of feature selection techniques and classifiers to design an accurate network intrusion detection system. They used the NSL-KDD dataset and applied four feature selection methods to evaluate the performance of five classifiers using a five-fold cross-validation strategy. The authors (Imrana et al., 2021) proposed a bidirectional Long-Short-Term-Memory (BiDLSTM)

based intrusion detection system to handle especially User-to-Root (U2R) and Remote-to-Local (R2L) attacks. Their proposed model improves the detection accuracy rate of U2R and R2L attacks more than the conventional LSTM.

Ammar and Faisal (Aldallal & Alisa, 2021) proposed a hybrid model of Support Vector Machine (SVM) and Genetic Algorithm (GA) intrusion detection system with innovative fitness functions to evaluate the system accuracy in the cloud computing environment. The proposed approach was evaluated on the CICIDS2017 dataset and benchmarked with KDD CUP 99 and NSL-KDD. The results showed that the proposed model outperformed benchmarks by 5.74%. The authors (Imran et al., 2021) proposed an ensemble of automated machine learning and Kalman filter prediction approaches to improve anomaly detection accuracy in a network intrusion environment. The proposed model was evaluated on the UNSW-NB15 and CICIDS2017 datasets and observed intrusion detection accuracy of 98.80% for the UNSW-NB15 dataset and 97.02% for the CICIDS2017 dataset.

The authors (Al-Omari et al., 2021; Sarker et al., 2020) presented a machine learning-based security model called Intrusion Detection Tree (IntruDTree) that considers the importance of security features and then builds a tree-based generalized intrusion detection model based on the selected essential features. A survey on machine learning approaches for Cyber Security Intrusion Detection was published in 2016 using KDD 1999 and DARPA 1998 datasets (Buczak & Guven, 2016). Similar work was also published by (Sultana et al., 2019) and (da Costa et al., 2019), focusing only on reviewing current literature. All these works correlate with ours, but our work used different machine learning-based IDS models and executed them on the recently available dataset. After that, the results were compared to the existing work to assess and analyze the performance.

The authors (Abdulhammed et al., 2019) used two machine learning methods, Auto Encoder (AE) and Principal Component Analysis (PCA), for dimensionality reduction and RF, Bayesian Network, Linear Discriminant Analysis (LDA) and Quadratic Discriminate Analysis (QDA) classifiers for designing an IDS. The proposed methodology reduced the CICIDS2017 dataset's feature dimensions from 81 to 10 while maintaining an accuracy of 99.6% for multi-class and binary classification. The above-discussed literature considered the outdated dataset for developing IDS, focusing more on prediction accuracy and less on prediction latency. The authors (Seth et al., 2021) used the latest CICIDS 2018 dataset, considering the modern-day attack, to build the IDS. They proposed hybrid feature selection methods and used the Light Gradient Boosting Machine Learning (LightGBM) classifier to design the IDS. The proposed model gives 97.73% accuracy and achieves 1.5% higher accuracy than the existing models.

3. MACHINE LEARNING-BASED IDS MODELS

Many researchers and organizations use a variety of algorithms and techniques, including Support Vector Machine (SVM), Naive Bayes (NB), Decision Trees (DT), Logistic Regression (LR), K-Nearest-Neighbor (KNN), clustering, and various ensemble methods, to extract knowledge from intrusion datasets. Each record in supervised learning IDS has a network or host data source and an associated labelled output value, such as Malicious or Benign. To discover the intrinsic link between the input data and the labelled output value for the specified features, a model is developed using supervised learning techniques. In the testing rounds, the trained model categorizes the unknown input as Malicious or Benign. Each classifier has its strengths and weaknesses. A natural way to create a robust classifier is to combine many weak classifiers. Multiple classifiers are trained using ensemble techniques, and the classifiers then vote to determine the final results. Boosting, Bagging, and Stacking are just a few ensemble approaches proposed to improve performance. The term "boosting" refers to a group of algorithms that can improve the performance of weak learners. Training the same classifier on a different subset of the same dataset is called bagging. Stacking combines various classifications via a meta-classifier (Aburomman & Ibne Reaz, 2016). According to Jabbar*et al*. combination of Random Forests and the Average One-Dependence Estimator (AODE) may be used to overcome the

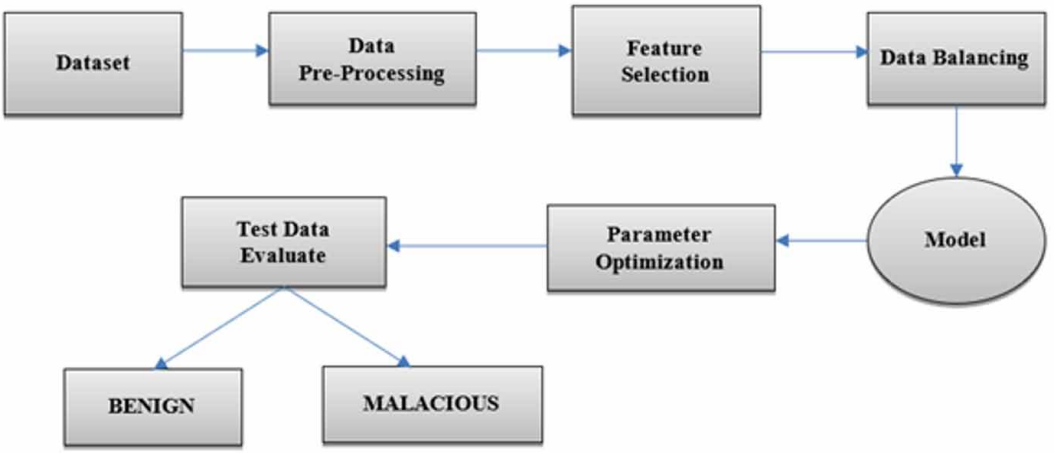
Table 1. Related work

Authors, Year	Dataset	Methodology	Accuracy
(Varghese & Muniyal, 2017)	NSLKDD	j48, NBTree, Random Forest, LibSVM, Bagging with REPTree, PART, and Multilayer Perceptron (MLP)	Random Forest (99.7%)
(Sharafaldin et al., 2018)	CICIDS2017	K-Nearest Neighbors (KNN), Random Forest (RF), ID3, Adaboost, Multilayer Perceptron (MLP), Naive-Bayes (NB), Quadratic Discriminant Analysis (QDA)	Random Forest (98%)
(Abdulhammed et al., 2019)	CICIDS2017	AE and PCA for dimensionality reduction with RF, Bayesian Network, LDA and QDA Classifiers	99.6% for both multi-class and binary classification
(Alazzam et al., 2020)	KDDCUP 99, NLS-KDD, and UNSW-NB15	Pigeon-inspired optimizer technique with Decision Tree Classifier	96% for KDDCUP 99, 88.3% for NLS-KDD and 91.7% for UNSW-NB15
(Imran et al., 2021)	UNSW-NB15 and CICIDS2017	Ensemble of automated machine learning and Kalman filter prediction approaches	98.80% for the UNSW-NB15 dataset and 97.02% for the CICIDS2017
(Aldallal & Alisa, 2021)	CICIDS2017, KDD CUP 99, and NSL-KDD	Hybrid model of Support Vector Machine (SVM) and Genetic Algorithm (GA)	Model outperformed in terms of accuracy by a maximum of 5.14% using CICIDS2017, a maximum of 4.97% using the KDD CUP 99 dataset, and a maximum of 5.74% using the NSL-KDD dataset.
(Imrana et al., 2021) Imrana et al., 2021	NSL-KDD	Bidirectional Long-Short-Term-Memory (BiDLSTM)	BiDLSTM model achieves a higher detection accuracy compared to the conventional LSTM
(Seth et al., 2021)	CICIDS2018	Light Gradient Boosting Machine Learning (LightGBM) classifier	97.73%
(Naseri & Gharehchopogh, 2022)	NSL-KDD and UNSW-NB15	Binay version of the Farmland Fertility Algorithm (BFFA) with K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Adaboost and Naive Bayes (NB)	99.6% for NSL-KDD (Hybrid) and 99% for UNSW-NB15 (DT)

issue of attribute dependence in Naïve Bayes. Random Forest enhances precision and reduces false alarms (Jabbar et al., 2017). The hybrid models are designed in many stages in combination with different classification models. Ensemble and hybrid classifiers tend to outperform single classifiers in terms of performance. The key points lie in selecting which classifiers to combine and how they are connected. The present work analyzes the top 10 popular machine Learning classifiers such as Adaboost, Decision Tree (DT), GaussianNB, KNeighbours, Logistic, Multinomial NB, Random Forest (RF), Stochastic Gradient Descent (SGD) Classifier, Support Vector Machine (SVM), and XGBoost on intrusion detection systems to find out the best model. The process flow of creating Machine learning-based IDS is shown in Figure 1.

This paper uses the IDS dataset containing the most recent common attacks. The dataset is highly imbalanced. Two data re-sampling techniques are used to balance the dataset. Afterwards, some pre-processing steps are performed to fix the problems that may exist in the datasets. These data pre-processing steps are discussed in the subsequent sections.

Figure 1. The life cycle of machine learning-based IDS



3.1 The Data Pre-Processing Steps

In machine learning, data pre-processing steps transform or encode data into suitable formats so that machines can quickly parse it. The datasets may require treating missing or inconsistent values, feature scaling, feature selection, and data imbalance problems.

3.1.1 Missing or Inconsistent Values

The presence of missing values in a dataset is quite common. Missing values must be evaluated for rectification, whether they occurred during data collection or validation. It can be solved by eliminating rows with missing data or filling them with estimated values.

3.1.2 Feature Scaling

Feature Scaling is a part of data pre-processing. It normalizes the independent features in a defined range to handle highly fluctuating magnitudes or values. There are different strategies for performing feature scaling.

Min-Max Normalization: This approach re-scales a feature or observation value in a range between Zero and One. Its formula is:

$$X_{new} = \frac{X_i - \min\left(X\right)}{\max\left(X\right) - \min\left(X\right)}$$

Standardization: It is a very effective re-scaling strategy where a feature value has a distribution with zero mean value, and variance equals to one.

$$X_{new} = \frac{X_i - X_{mean}}{Standard\ Deviation}$$

If we do not perform feature scaling, a machine learning model will weigh larger values lower and smaller ones higher, regardless of the unit of measurement.

3.1.3 Feature Selection

"Feature selection" is also named "Feature Learning" or "Feature Engineering", which is the most crucial stage during pre-processing. It simplifies the data, eliminates data redundancy, reduces computational difficulty, improves the detection rate, and reduces false alarms of machine learning models. Only essential features are selected based on their correlation scores with the consequence variable. Feature selection plays a critical role in building any IDS, so the chosen features highly affect the accuracy and reduce false alerts. Each feature has specific characteristics for addressing different areas of threat detection. Features containing basic information about the software or network are considered naïve, and when they represent deeper details, they are considered rich. Three approaches, Filter, Wrapper, and Embedded, are used for Feature selection, as shown in Table 2.

3.1.4 Imbalanced Learning

Most machine learning predictive models work based on the assumption that an equal number of classes are in each sample. But when the distribution of classes is imbalanced, for example, the minority class contains a hundred samples, and the majority class contains hundreds of thousands of samples, this results in the machine learning models having poor performance, specifically for the minority class, and for the majority class the performance might be misleading. Imbalanced Learning is an open-source python toolbox with various techniques for handling imbalanced data classification. Some of the categories of handling imbalanced data such as Random Under-Sampling (RUS), which reduces the samples from the majority class; Random Over-Sampling (ROS), which creates duplicate copies of samples from the minority class; Synthetic Minority Oversampling Technique (SMOTE), where the synthetic sample is created, and Tomek Links which removes the noise from the data, are discussed with advantage and limitation in the following sub-sections. These strategies are used to fine-tune the class distribution of a data set.

Let the imbalanced dataset is represented by x, the minority class sample is represented by $x_{min,}$ and x_{max} represents the majority class samples. The balancing ratio of dataset x is defined as:

$$r_x = \frac{xmin}{xmax}$$

The balancing process is equivalent to re-sample x into a new dataset x_{res} such that $r_x > r_{xres}$.

i) **Random Under Sampling (RUS):** In RUS, the number of samples of the majority class (x_{max}) is reduced, i.e., removing some of the observations from the majority class until the majority and minority class are balanced out. The drawback of under-sampling is that we are removing the data that may be valuable.

Table 2. Feature selection approaches

Approach	Description	Advantage	Disadvantage
Filter (Hamon, 2013)	Selects the top essential features regardless of the model	Low Execution Time and over-fitting	May choose a redundant variable
Wrapper (Phuong et al., 2006)	Create subsets by combining related variables	Consider interactions	Over-fitting risk and high Execution time
Embedded (Hernandez et al., 2007)	Examine more depth interaction than Wrapper	Optimal subset results	----------

ii) **Random Over Sampling (ROS):** Contrary to under-sampling, more copies of data are added into the minority class such that new samples are generated in x_{min} to reach the balancing ratio $r_{xres.}$ It is a worthy choice when we don't have tons of data to work with, but at the same time, it also causes over-fitting and poor generalization of minority sample results.
iii) **Synthetic Minority Oversampling Technique (SMOTE):** Over-sampling method creates duplicate samples in the minority class that does not add new information to the existing data set. SMOTE solves these issues by creating synthetic samples. It chooses a random sample from a minority class and finds its k-neighbours minority class. A synthetic sample is created randomly between two samples in a feature space. This technique can be used to create as many synthetic examples for the minority class (x_{min}) to reach the balancing ratio $r_{xres.}$ This strategy may produce noisy samples by inserting new points between marginal outliers and inliers.
iv) **Tomek's Link:** This is a cleaning method to eliminate the noise generated in the majority class while creating new samples in the minority class. This is an under-sampling strategy for reducing the unwanted samples from the majority class.

This paper uses the SMOTE oversampling strategy to balance the CICIDS2018 dataset and Tomek's links to clean the unwanted samples.

4. EXPERIMENTAL SETUP AND RESULTS

The experimental setup and execution are performed on Microsoft Windows 10 environment with Intel Core i5 2.2 GHz, RAM 4GB and 500GB HDD. All models are implemented in Python 3.7.x with the help of Scikit-learn (v0.22.X), Pandas (v1.0.3), Numpy (v1.18.2), Matplotlib (v3.2.1), Seaborn (v0.10.0), XGboost (v0.90), Scipy (v1.4.1), and Imblearn (v0.4.3). All the models are trained and tested against the CICIDS2018 IDS dataset. A detailed description of the CICIDS2018 dataset is discussed next.

4.1 The CICIDS2018 Dataset

Sharafaldin et al. (Sharafaldin et al., 2018) analyzed the properties of eleven IDS datasets since 1998 and showed that most are outdated and unreliable. Some issues are i) existing datasets suffered from the lack of traffic diversity and volumes, and ii) datasets do not cover the diversity of known attacks.

The CICIDS2018 dataset is publicly available for networking security and intrusion detection research from the Canadian Institute of Cyber-security. More than 80 network flow features are extracted from the traffic data generated over five days. They also delivered the network flow dataset as CSV files with 85 features and class labels. Seven different attack scenarios, such as Brute-force, Heart bleed, Web attacks, DoS, DDoS, Botnet, and infiltration of the network from inside, are included in the final dataset. There are 50 machines in the attacker's infrastructure, while 420 devices and 30 servers in the victim organization's infrastructure are spread across five departments. The CICIDS2018 dataset consists of corresponding profiles and labelled network flows, including full packet payloads in PCAP format and CSV files for Machine and deep learning purposes, as shown in Table 3.

The timestamp, source and destination ports, source and destination IPs, protocols, and attacks are all labelled in this dataset. This dataset also includes complete network architecture, including a modem, a firewall, routers, switches, and nodes with different operating systems, i.e., open-source operating system Linux, Apple's macOS, Microsoft Windows 10, Windows 8, Windows 7, and Windows XP. The dataset set is captured daily from the network traffic and generated in a PCAP file. After that, the PCAP file is converted into a CSV file. The five days of CSV file is analyzed, containing 3119345 rows and 85 columns. Some columns' names are mentioned in Figure 2.

Table 3. CICDS2018 dataset CSV

File Name	Class
Monday-WorkingHours.pcap ISCX.csv	BENIGN
Tuesday-WorkingHours.pcap ISCX.csv	BENIGN, SSH-Patator, FTP-Patator
Wednesday-workingHours.pcap ISCX.csv	BENIGN, DoSslowloris, DoSSlowhttptest, DoS Hulk, DoSGoldenEye, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap ISCX.csv	BENIGN, Web Attack \x96 Brute Force, Web Attack \x96 XSS, Web Attack \x96 Sql Injection
Thursday-WorkingHours-Afternoon-Infilteration.pcap ISCX.csv	BENIGN, Infiltration
Friday-WorkingHours-Morning.pcap ISCX.csv	BENIGN, Bot
Friday-WorkingHours-Afternoon-DDos.pcap ISCX.csv	BENIGN, DDoS
Friday-WorkingHours-Afternoon-PortScan.pcap ISCX.csv	BENIGN, PortScan

Figure 2. CICDS2018 dataset columns

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2830743 entries, 0 to 692702
Data columns (total 85 columns):
Flow ID                        object
 Source IP                     object
 Source Port                   float64
 Destination IP                object
 Destination Port              float64
 Protocol                      float64
 Timestamp                     object
 Flow Duration                 float64
 Total Fwd Packets             float64
 Total Backward Packets        float64
Total Length of Fwd Packets    float64
 Total Length of Bwd Packets   float64
 Fwd Packet Length Max         float64
 Fwd Packet Length Min         float64
 Fwd Packet Length Mean        float64
 Fwd Packet Length Std         float64
Bwd Packet Length Max          float64
 Bwd Packet Length Min         float64
 Bwd Packet Length Mean        float64
 Bwd Packet Length Std         float64
Flow Bytes/s                   object
 Flow Packets/s                object
 Flow IAT Mean                 float64
 Flow IAT Std                  float64
 Flow IAT Max                  float64
 Flow IAT Min                  float64
Fwd IAT Total                  float64
```

The dataset contains NULL values, as shown in Figure 3. After that, data is pre-processed, and continuous NULL values are removed from CICDS2018 datasets. The NULL values are eliminated by dropping that row from the dataset, as shown in Figure 4.

The correlation map has been created using Pearson's Correlation Coefficient (r) between the feature and target variable, as shown in Figure 5. A correlation map represents the relationships between variables with each other or the target variables. The increases in one feature value increase the target variable's value, representing the positive correlation. And the increase in one feature value decreases the value of the target variable, representing the negative correlation. The feature score is computed using the univariate selection method, as shown in Figure 6. A subset of features is selected based on their score.

The CICIDS2018 is an imbalanced dataset as it has more Benign type samples than Malware type samples. It can be seen in Figure **7** that the count of Malware type samples is much less as compared to Benign type samples. The SMOTE Tomek method is applied to the imbalance CICIDS2018data to convert it into balanced data for Binary classifiers, shown in Figure **8**.

Similarly, the CICIDS2018 is an imbalanced dataset in the multi-class, as shown in Figure 9. The Oversampling Technique is applied to the imbalance CICIDS2018 data to convert it into balanced data for multi-class classifiers, as shown in Figure 10.

4.2 Performance Measure

This section discusses the classification metrics for IDS. Table 4 shows the confusion matrix for a two-class classifier that can be used to evaluate an IDS's performance. Each column of the confusion matrix indicates the samples in a predicted class, while each row shows the samples in an actual

Figure 3. CICDS2018 with NULL values

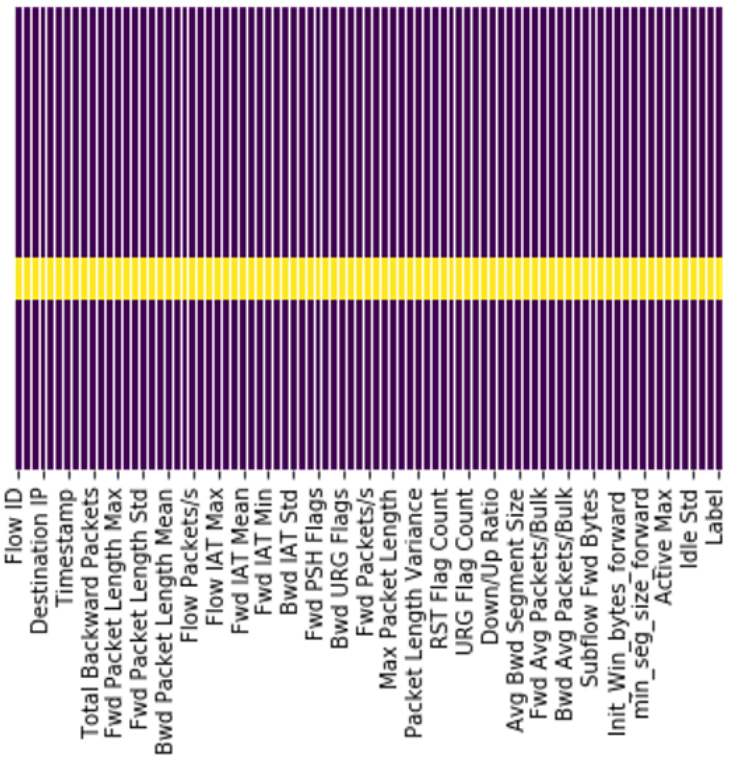


Figure 4. CICDS2018 without NULL values

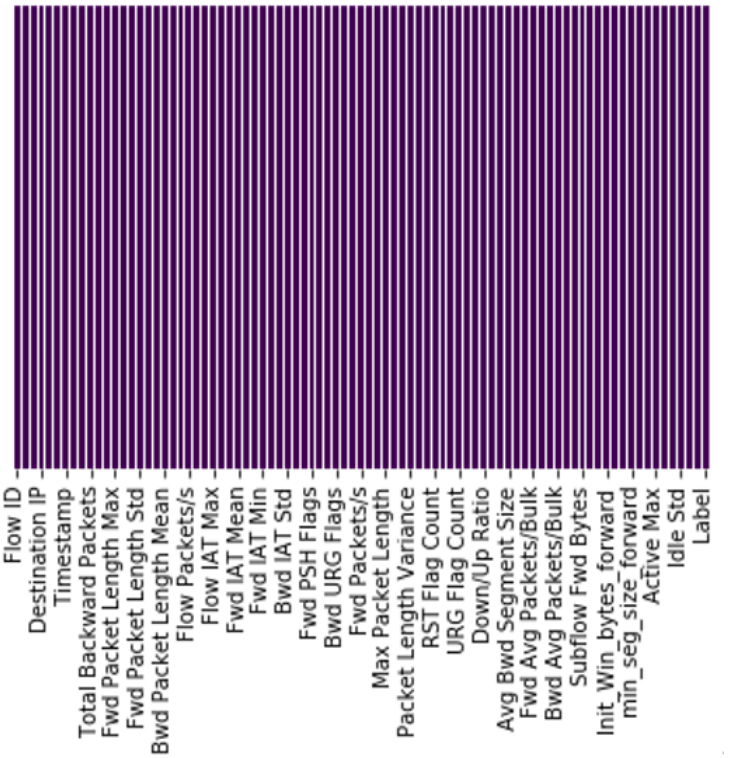
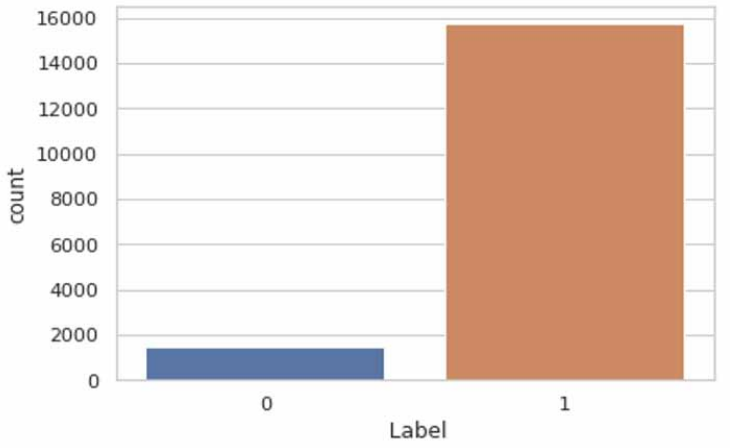


Figure 5. CICDS2018 with correlation map

Figure 6. CICDS2018 with feature score

10	Total Length of Bwd Packets	7922.214863
70	Subflow Bwd Bytes	7922.214863
35	Fwd PSH Flags	4749.538023
49	SYN Flag Count	4749.538023
53	URG Flag Count	4727.620879
5	Timestamp	4403.359213
72	Init_Win_bytes_backward	4218.061208
57	Average Packet Size	4074.591212
25	Fwd IAT Total	4010.421601
45	Packet Length Mean	3978.064168
6	Flow Duration	3739.596487
17	Bwd Packet Length Mean	3701.417384
59	Avg Bwd Segment Size	3701.417384
18	Bwd Packet Length Std	3610.224579
3	Destination Port	3500.491924
81	Idle Max	3394.822015
46	Packet Length Std	3306.982131
33	Bwd IAT Max	3190.369015
28	Fwd IAT Max	3108.090223
23	Flow IAT Max	3073.457762
30	Bwd IAT Total	3006.250257

Figure 7. Imbalance data binary class



class. The diagonal of the confusion matrix represents the correct classification of samples, while the non-diagonal represents the incorrect classification. The main aspects to consider when measuring the accuracy are:

- True Positive (TP): The classifier correctly predicts the intrusions attack.
- True Negative (TN): The classifier correctly predicts the non-intrusions instances.
- False Positive (FP): The classifier in-correctly predicts the intrusions attack.
- False Negative (FN): The classifier correctly predicts the non-intrusions instances.

Figure 8. Balanced data binary class

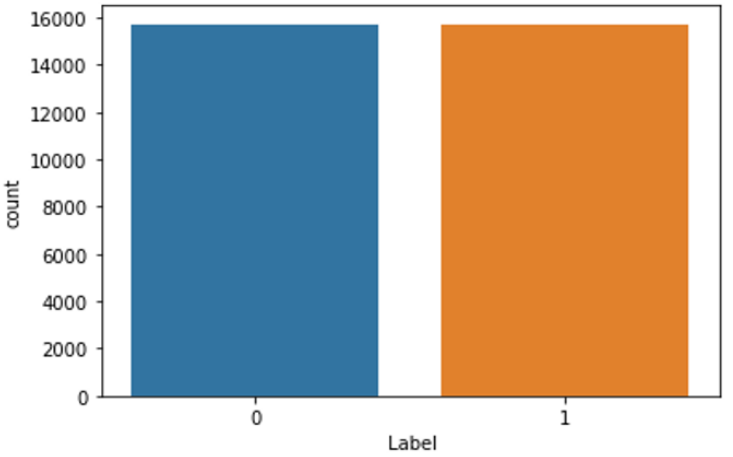
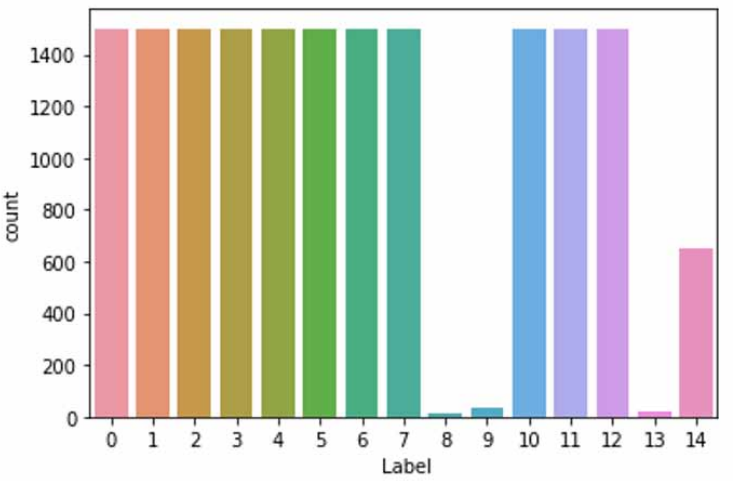


Figure 9. Imbalance data multi-class



This paper uses popular performance measures, including overall accuracy, decision rates, precision, Recall, and F1-score (Aminanto et al., 2017; Atli, 2017; Hodo et al., 2017), which are briefly discussed.

Accuracy: Accuracy is the most intuitive performance measure of a classification model. It is the ratio of the total correctly predicted samples and the total number of samples in the dataset, as shown in Equation 1. High accuracy means the model is performing well. Accuracy is a valuable measurement only when the dataset is well-balanced.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Figure 10. Balanced data multi-class

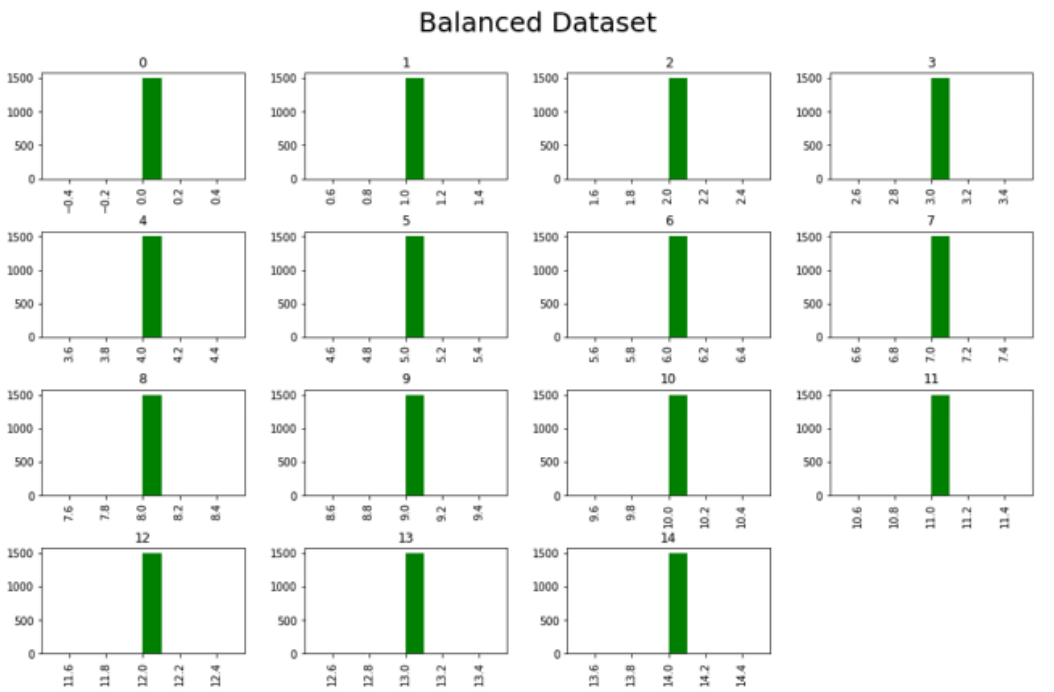


Table 4. Confusion matrix

	Predicted Class		
	Class	**Normal**	**Attack**
Actual Class	Normal	TN	FP
	Attack	FN	TP

Precision: Precision is also a performance measure of correctly classifying data points out of total data points predicted by the classification model, as shown in Equation 2. The higher precision value indicates the better performance of the model. Precision is also known as a positive predictive value (PPV). Precision is an excellent measure to determine when the cost of false positives is high.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall: It measures the sensitivity of the model. The Recall is a performance measure of correctly retrieving the data points. In other words, the Recall is the ratio of the total correct class predicted and the actual data points in the dataset, as shown in Equation 3. The Recall is also known as the true positive rate (TPR). The higher recall value indicates the better performance of the model. It is a good metric of measurement when there is a high cost associated with False Negative.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1–Score: It is an instrumental performance measurement technique widely used when the model produces high Recall and low precision, or low recall and high precision, i.e., uneven class distribution (a large number of actual negative classes). F1-Score uses harmonic instead of arithmetic to punish extreme values shown in Equation 4.

$$F_1 - score = 2.\frac{precision.recall}{precision + recall} \tag{4}$$

AUC-ROC Curves: The Area under the Curve (AUC) and Receiver Operating Characteristics (ROC) curve is an approach for measuring the performance of a classification model on different threshold settings. The curve is plotted between TPR on the y-axis and the FPR on the x-axis to measure the performance of a classifier. The higher AUC means the classification has high accuracy. It is used to know the capability of a classification model to separate the classes.

The matrices mentioned above can be used to measure the performance of both binary and multi-class IDS in which incidents are classified as either Benign or Malicious or family of Malicious.

4.3 Results Analysis

The final quantitative for each class label is assigned after noise clean-up, as shown in Table 5. It can be observed from the table that the dataset is highly unbalanced. The NULL values are removed from the dataset; the missing values are treated carefully and filled with valid data. Then feature scaling/transformation is performed using MinMaxScaler techniques because the dataset contains varying magnitudes, values, or units. The fixed range values are provided in each column of the datasets.

The Univariate Selection method is used to compute the score of each feature on the whole dataset. The top 50 features are selected based on the score shown in Table 6. The scikit-learn library provides the *SelectKBes*t class to extract the best features of a given dataset. *SelectKBes*t class performs statistical tests to select features with the strongest relationship with the output or

Table 5. Data clean-up

Class Name	The Value Assigned for Class	Original Sample	Final Samples
BENIGN	0	2273097	2273097
DoS Hulk	4	231073	231073
PortScan	10	158930	158930
DDoS	2	128027	128027
DoSGoldenEye	3	10293	10293
FTP-Patator	7	7938	7938
SSH-Patator	11	5897	5897
DoSslowloris	6	5796	5796
DoSSlowhttptest	5	5499	5499
Bot	1	1966	1966
Web Attack – Brute Force	12	1507	1507
Web Attack – XSS	14	652	652
Infiltration	9	36	36
Web Attack – Sql Injection	13	21	21
Heartbleed	8	11	11
		Null Value: 288602	----------
Total	----------	**3119345**	**2830743**

Table 6. Top-50 features selected

The Top 50 Features Based on Their High Score Are Arranged in Descending Order
Total Length of Bwd Packets, SubflowBwd Bytes, Fwd PSH Flags, SYN Flag Count, URG Flag Count, Timestamp, Init_Win_bytes_backward, Average Packet Size, Fwd IAT Total, Packet Length Mean, Flow Duration, Bwd Packet Length Mean, AvgBwd Segment Size, Bwd Packet Length Std, Destination Port, Idle Max, Packet Length Std, Bwd IAT Max, Fwd IAT Max, Flow IAT Max, Bwd IAT Total, Bwd Packet Length Max, Bwd IAT Mean, Fwd Header Length, Fwd Header Length.1, Idle Mean, Bwd IAT Min, ACK Flag Count, Flow IAT Std, Flow IAT Mean, Idle Min, Max Packet Length, Bwd IAT Std, Total Fwd Packets, SubFlowFwd Packets, Packet Length Variance, Bwd Header Length, Bwd Packet Length Min, Down/Up Ratio, Fwd IAT Std, Fwd IAT Mean, Active Min, Fwd IAT Min, Total Backward Packets, SubflowBwd Packets, Init_Win_bytes_forward, Idle Std, Active Mean, PSH Flag Count, Total Length of Fwd Packets

target variable. In this class, the Chi-Square method is used on the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution. Table 6 lists our 50 features selected. The final considered dataset has 50 feature columns and one column with class labels.

Imbalanced Learning: Imbalanced classification is a classification problem when unequal classes are in the training dataset. The imbalanced class distribution may vary, but modelling severely imbalanced data may require more specialized techniques. The dataset is classified into two sections Binary classification and Multi-class classification. Both classifications contain an Imbalanced dataset.

a) **Binary Classification:** Binary or binomial classification uses classification rules to classify elements of a given set into two groups. The IDS dataset contains a target labelled as Benign and Malware in the form of Binary classification. In this dataset target is Imbalanced. The SMOTETomek method is used to balance the dataset present in imblearn.combine library.
b) **Multi-class Classification:** In machine learning, multi-class or multinomial classification is the problem of classifying instances into one of three or more classes. That means those data sets that contain more than two targets or labels. The IDS dataset target or label had Benign and some family of malware, so the dataset was classified as multi-class classification, and these were imbalanced. So for balancing the dataset, the RandomOverSampler method presents in imblearn.over_sampling library is used.

The hyper-parameter techniques, i.e., GridSearchCV and RandomizedSearchCV, are employed to search for the best parameter for all classifiers according to the dataset. The target in the dataset is classified using a binary-class and multi-class classifier. So, ten popular machine-learning classification models are used based on binary-class and multi-class classifiers. The results of these models are evaluated on various factors such as Score, Precision, Recall, F1_score, Accuracy, and Total time (in seconds) taken by each algorithm.

4.3.1 Binary Classifier

The Binary classifier classifies target samples in the CICIDS2018 dataset into two classes. All classifiers and their accuracy, precision, Recall, f1 score, and time are shown in Table 7. It can be observed from the table that the top three best classifiers are KNeighbors (99.49%), XGBoost (99.14%) and AdaBoost (98.75%).

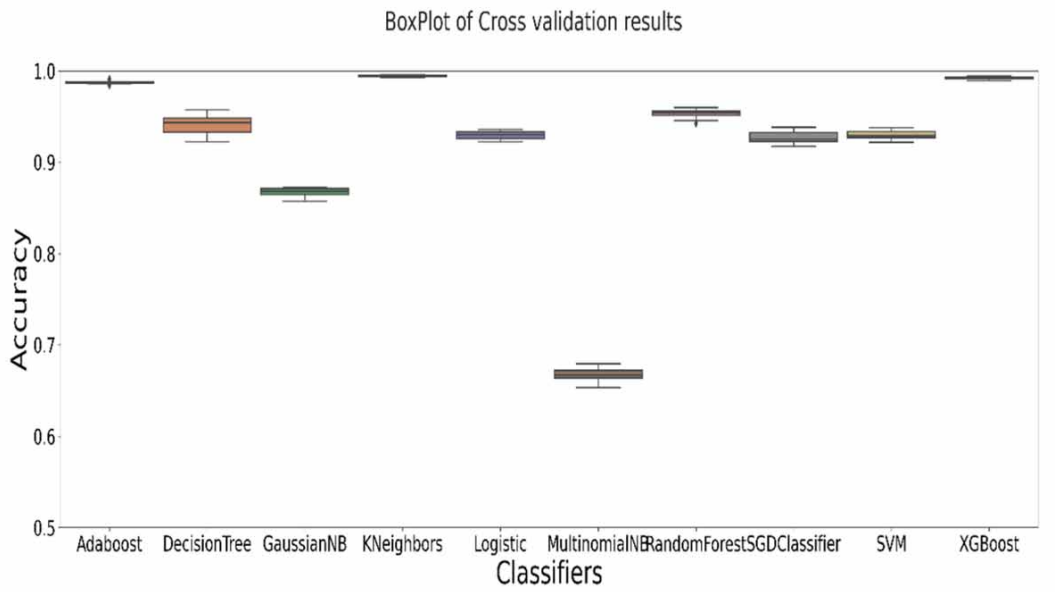
Box plotting is an excellent tool for identifying outliers and comparing distributions. The Box plots chart is shown in Figure 11. It helps us better understand and visualize how values are spaced out in different data sets.

The ROC curve of the Binary Classifier is shown in Figure 12. The accuracy of a testing model is evaluated based on how well the model distinguishes between malware and Benign. The ROC

Table 7. Performance comparison of binary classifiers

	Score	Precision	Recall	F1_Score	Accuracy	Time (s)
Adaboost	0.987587	0.987587	0.987587	0.987587	98.75873	6.26027
Decision Tree	0.943674	0.943674	0.943674	0.943674	94.36737	0.35466
GaussianNB	0.863565	0.863565	0.863565	0.863565	86.35652	0.02468
KNeighbors	0.994993	0.994993	0.994993	0.994993	99.49932	2.03291
Logistic	0.927402	0.927402	0.927402	0.927402	92.74016	0.45754
MultinomialNB	0.649630	0.649630	0.649630	0.649630	64.96297	0.01359
RandomForest	0.951393	0.951393	0.951393	0.951393	95.13925	2.58871
SGDClassifier	0.924272	0.924272	0.924272	0.924272	92.42724	0.06629
SVM	0.935851	0.935851	0.935851	0.935851	93.58506	13.5986
XGBoost	0.991447	0.991447	0.991447	0.991447	99.14467	3.85169

Figure 11. Box-plot for binary classifier



curve is plotted considering the Sensitivity or TPR and FPR. The colour denotes the threshold value for each TPR and FPR pair. Its threshold will be around one of the given instances that has a high affinity for the class. Hence, darker will be the colour in the ROC for a higher threshold of instances.

The AUC (Area under the Curve) measures the proportion of correctly classified test data. AUC value one represents a perfect test, whereas 0.5 represents a minor accurate test. In

Figure 12, KNeighbors, XGBoost and AdaBoost are close to 1 and have a larger area under the curve than all other classifiers. It can be observed from Figure 12 that KNeighbors, XGBoost, and AdaBoost classified most of the samples correctly and have a higher percentage of accuracy than other classifiers.

Figure 12. ROC curve of binary classifier

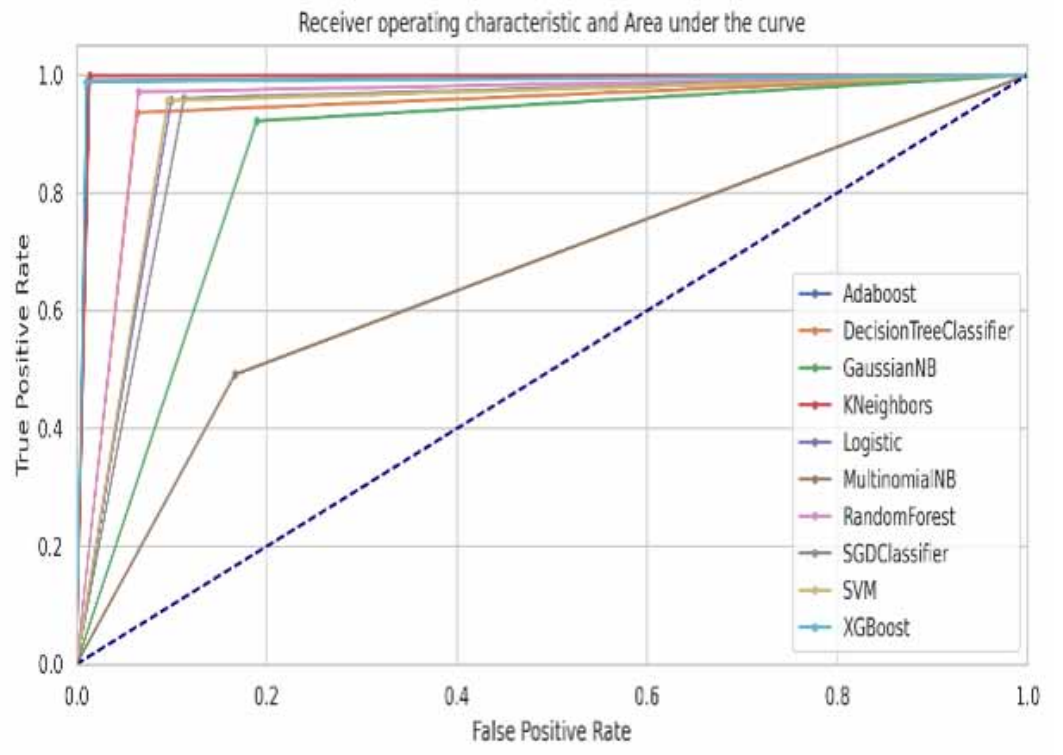


Figure 13 depicts a histogram, a bar chart used to show the frequency distribution of continuous data. Class or bin indicates the number of observations between the ranges of values. This histogram represents the Test score of precision, recall, f1_score, and score of each classifier.

4.3.2 Multi-Class Classifier

Target samples in the CICIDS2018 dataset are classified into multi-class in the multi-class classifier. The multi-class classifiers and their accuracy, precision, Recall, f1_score, and time are shown in Table 8. The box plots chart and histogram plots of the Multi-class classifier are shown in Figure 14 and Figure 15.

It can be observed that the model XGBoost, K-Neighbors, and GaussianNB perform better than other multi-class classifiers with 99.30%, 98.88% and 96.66% accuracy, respectively.

5. CONCLUSION AND FUTURE WORK

The work on IDS is important because it is essential for protecting computer networks from malicious attacks. As the amount of data processed and transferred over networks grows, so does the number of potential attack vectors. In this environment, signature-based IDS are not always effective, as they can only detect known attacks. On the other hand, Machine learning-based IDS can detect unknown attacks by learning from normal and malicious behaviour patterns. The work presented in this paper demonstrates the effectiveness of Machine learning-based IDS in detecting both known and unknown attacks. The authors evaluated the performance of ten machine learning classifiers on a dataset of common attacks. They found that the top three models (KNeighbours, XGBoost, and

Figure 13. Test score of binary classifiers

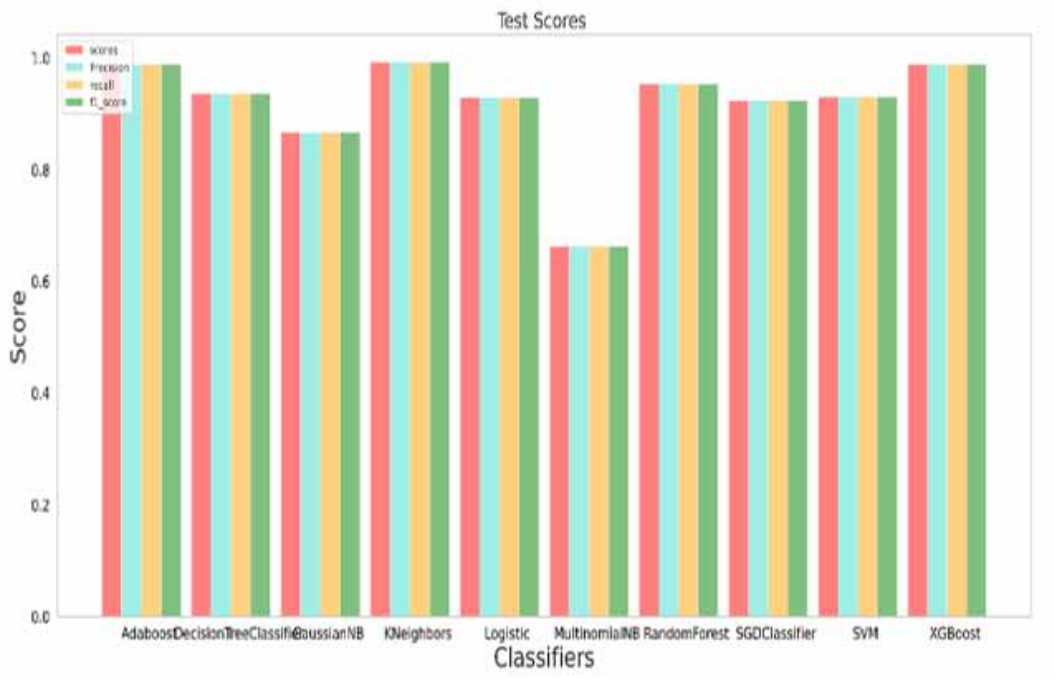


Table 8. Performance comparison of multi-class classifiers

	Score	Precision	Recall	F1_Score	Accuracy	Time
Adaboost	0.949037	0.949037	0.949037	0.949037	94.903704	4.45012
Decision Tree	0.928148	0.928148	0.928148	0.928148	92.814815	0.02235
GaussianNB	0.966667	0.966667	0.966667	0.966667	96.666667	0.03283
KNeighbors	0.988889	0.988889	0.988889	0.988889	98.888889	0.93272
Logistic	0.903407	0.903407	0.903407	0.903407	90.340741	2.39280
MultinomialNB	0.614963	0.614963	0.614963	0.614963	61.496296	0.01185
RandomForest	0.947556	0.947556	0.947556	0.947556	94.755556	2.67216
SVM	0.914370	0.914370	0.914370	0.914370	91.437037	3.47171
XGBoost	0.993037	0.993037	0.993037	0.993037	99.303704	53.44061

AdaBoost) achieved up to 99.49% accuracy in binary-class classification and 99.30% in multi-class classification. This research shed light on the practical and theoretical importance of enhancing IDS capabilities in addressing evolving cybersecurity threats.

PRACTICAL IMPORTANCE

The practical importance of this work is that it provides a valuable tool for network security professionals. Organizations can improve their ability to detect and respond to malicious attacks by using machine learning-based IDS. These findings enable the development and implementation of

Figure 14. Box-plot of multi-class classifier

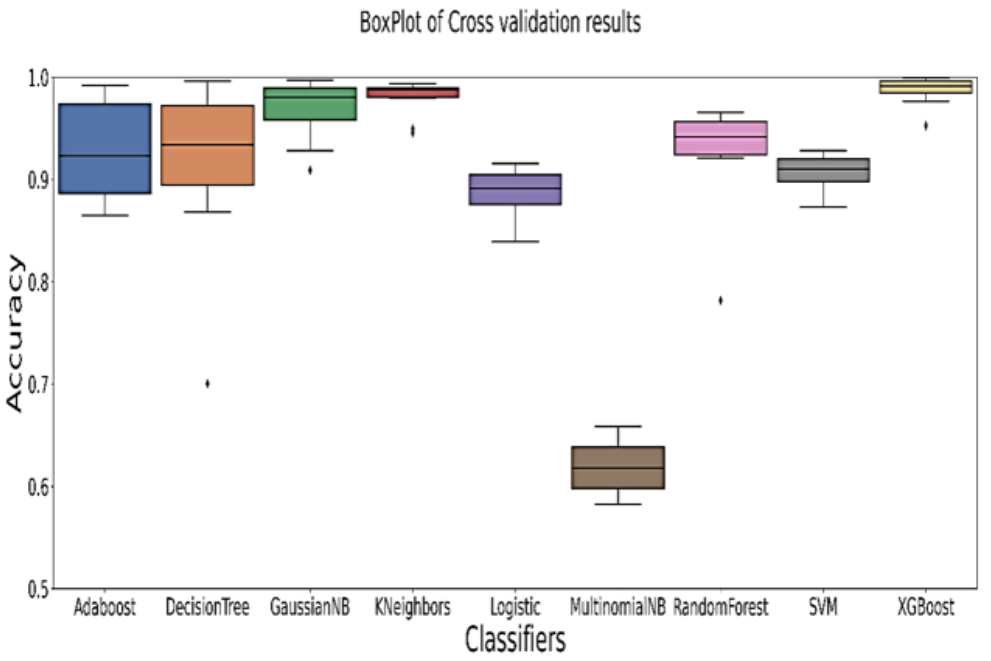


Figure 15. Test score of multi-class classifiers

IDS systems to safeguard sensitive data, protect against ransomware attacks, and mitigate cyber-attacks' impact during the Covid-19 era and beyond. The identified top-performing models, such as KNeighbours, XGBoost, and AdaBoost, offer practical guidance for organizations seeking adequate security against unknown and obfuscated malicious activities. The authors found that the XGBoost

algorithm was the most effective for detecting malicious attacks in their dataset. The finding suggests that XGBoost may be a good choice for Machine learning-based IDS in other settings.

THEORETICAL IMPORTANCE

The theoretical importance of this work is that it contributes to the body of knowledge on machine learning-based IDS. The authors' findings provide insights into the effectiveness of different machine-learning algorithms for detecting malicious attacks. This information can be used to develop more effective IDS in the future. Using statistical-based, knowledge-based, and Machine learning-based methods, researchers can enhance the accuracy and effectiveness of intrusion detection systems. As presented in this research, analyzing different machine learning classifiers expands the understanding of their performance in IDS applications, thereby contributing to the theoretical foundation of network security research. Moreover, the utilization of data re-sampling techniques and pre-processing steps ensures the robustness and reliability of the IDS dataset, facilitating the development of more accurate and efficient detection models. In addition to the practical and theoretical importance, the work on IDS also has the following benefits:

- It can help to identify new malicious activities that are not yet known to signature-based IDS.
- It can provide insights into the behaviour of malicious actors, which can be used to develop better defensive strategies.
- It can help to improve the overall security posture of an organization.

In summary, the undertaken work on IDS is of critical importance, both in practical terms, by addressing the immediate security challenges faced in the Covid-19 era, and in theoretical terms, by advancing intrusion detection methodologies and exploring the potential of machine learning techniques. By leveraging the insights gained from this research, organizations and researchers can make informed decisions and develop effective strategies to protect against evolving cyber threats, secure sensitive data, and ensure the integrity of communication networks in the face of an increasingly interconnected digital landscape.

In future work, the adversarial example that an attacker has intentionally designed to cause the model to make a mistake can be considered an input to the different machine learning models to understand the vulnerability of machine learning classifiers.

DECLARATIONS

The authors of this publication declare there is no conflict of interest. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A. (2019). Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. *Electronics (Basel)*, *8*(3), 1–27. doi:10.3390/electronics8030322

Abu Al-Haija, Q., Al Badawi, A., & Bojja, G. R. (2022). Boost-Defence for resilient IoT networks: A head-to-toe approach. *Expert Systems: International Journal of Knowledge Engineering and Neural Networks*, *39*(10), e12934. doi:10.1111/exsy.12934

Aburomman, A. A., & Ibne Reaz, M. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, *38*, 360–372. doi:10.1016/j.asoc.2015.10.011

Acharya, N., & Singh, S. (2018). An IWD-based feature selection method for intrusion detection system. *Soft Computing*, *22*(13), 4407–4416. doi:10.1007/s00500-017-2635-2

Al-Omari, M., Rawashdeh, M., Qutaishat, F., Alshira'H, M., & Ababneh, N. (2021). An Intelligent Tree-Based Intrusion Detection Model for Cyber Security. *Journal of Network and Systems Management*, *29*(2), 20. doi:10.1007/s10922-021-09591-y

Alazab, A., Hobbs, M., Abawajy, J., & Alazab, M. (2012). Using feature selection for intrusion detection system. *2012 International Symposium on Communications and Information Technologies, ISCIT 2012*, 296–301. doi:10.1109/ISCIT.2012.6380910

Alazzam, H., Sharieh, A., & Sabri, K. E. (2020). A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer. *Expert Systems with Applications*, *148*, 113249. Advance online publication. doi:10.1016/j.eswa.2020.113249

Aldallal, A., & Alisa, F. (2021). Effective intrusion detection system to secure data in cloud using machine learning. *Symmetry*, *13*(12), 2306. Advance online publication. doi:10.3390/sym13122306

Aminanto, M. E., Choi, R., Tanuwidjaja, H. C., Yoo, P. D., & Kim, K. (2017). Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. *IEEE Transactions on Information Forensics and Security*. Advance online publication. doi:10.1109/TIFS.2017.2762828

Amouri, A., Alaparthy, V. T., & Morgera, S. D. (2020). A machine learning based intrusion detection system for mobile internet of things. *Sensors (Basel)*, *20*(2), 1–6. doi:10.3390/s20020461 PMID:31947567

Atli, B. G. (2017). *Anomaly-Based Intrusion Detection by Modeling Probability Distributions of Flow Characteristics*. Aalto University.

Axelsson, S. (2000). Intrusion Detection Systems: A Survey and Taxonomy. *Technical Report*, *99*, 1–15.

Babu, E. S., Padma, B., Nayak, S. R., Mohammad, N., & Ghosh, U. (2023). Cooperative IDS for Detecting Collaborative Attacks in RPL-AODV Protocol in Internet of Everything. *Journal of Database Management*, *34*(2), 1–33. doi:10.4018/JDM.324099

Baskerville, R. L., & Portougal, V. (2003). A possibility theory framework for security evaluation in national infrastructure protection. *Journal of Database Management*, *14*(1), 1–13. Advance online publication. doi:10.4018/jdm.2003040101

Bennett, K. P., & Demiriz, A. (1999). Semi-supervised support vector machines. *Advances in Neural Information Processing Systems*.

Biswas, S. K. (2018). Intrusion Detection Using Machine Learning : A Comparison Study. *International Journal of Pure and Applied Mathematics*.

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys and Tutorials*, *18*(2), 1153–1176. doi:10.1109/COMST.2015.2494502

Butun, I., Morgera, S. D., & Sankar, R. (2014). A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, *16*(1), 266–282. doi:10.1109/SURV.2013.050113.00191

Choudhury, A., Vuppu, S., Pratap Singh, S., Kumar, M., & Nakharu Prasad Kumar, S. (2023). ECG-based heartbeat classification using exponential-political optimizer trained deep learning for arrhythmia detection. *Biomedical Signal Processing and Control*, *84*, 104816. doi:10.1016/j.bspc.2023.104816

da Costa, K. A. P., Papa, J. P., Lisboa, C. O., Munoz, R., & de Albuquerque, V. H. C. (2019). Internet of Things: A survey on Machine learning-based intrusion detection approaches. *Computer Networks*, *151*, 147–157. doi:10.1016/j.comnet.2019.01.023

Dua, S., & Du, X. (2016). *Data Mining and Machine Learning in Cybersecurity*. doi:10.1201/b10867

Effendy, D. A., Kusrini, K., & Sudarmawan, S. (2017). Classification of intrusion detection system (IDS) based on computer network. *2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 90–94. doi:10.1109/ICITISEE.2017.8285566

Hamon, J. (2013). *Combinatorial optimization for variable selection in high dimensional regression: Application in animal genetic*. Université des Sciences et Technologie de Lille, Lille I.

Hamzah, M. A., & Othman, S. H. (2021). A Review of Support Vector Machine-based Intrusion Detection System for Wireless Sensor Network with Different Kernel Functions. *International Journal of Innovative Computing*, *11*(1), 59–67. Advance online publication. doi:10.11113/ijic.v11n1.303

Hasan, M. A. M., Xu, S., Kabir, M. M. J., & Ahmad, S. (2016). Performance evaluation of different kernels for support vector machine used in intrusion detection system. *International Journal of Computer Networks and Communications*, *8*(6), 39–53. Advance online publication. doi:10.5121/ijcnc.2016.8604

Hernandez, J. C., Duval, B., & Hao, J. K. (2007). A genetic embedded approach for gene selection and classification of microarray data. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4447 LNCS, 90–101. doi:10.1007/978-3-540-71783-6_9

Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., & Atkinson, R. (2017). Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. *CoRR, abs/1701.0*.

Imran, J., Jamil, F., & Kim, D. (2021). An ensemble of a prediction and learning mechanism for improving accuracy of anomaly detection in network intrusion environments. *Sustainability (Basel)*, *13*(18), 10057. Advance online publication. doi:10.3390/su131810057

Imrana, Y., Xiang, Y., Ali, L., & Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. *Expert Systems with Applications*, *185*(July), 115524. doi:10.1016/j.eswa.2021.115524

Jabbar, M. A., Aluvalu, R., & Reddy, S. S. (2017). RFAODE: A Novel Ensemble Intrusion Detection System. *Procedia Computer Science*, *115*, 226–234. doi:10.1016/j.procs.2017.09.129

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1–22. doi:10.1186/s42400-019-0038-7

Laughlin, B., Sankaranarayanan, K., & El-Khatib, K. (2020). A service architecture using machine learning to contextualize anomaly detection. *Journal of Database Management*, *31*(1), 64–84. Advance online publication. doi:10.4018/JDM.2020010104

Liao, H. J., Richard Lin, C. H., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications. doi:10.1016/j.jnca.2012.09.004

Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, *78*(1), 13–21. doi:10.1016/j.knosys.2015.01.009

Low, C. (2005). Understanding Wireless Attacks and Detection. In *SANS Institute InfoSec Reading Room* (pp. 1–22). SANS Institute InfoSec Reading Room.

Mangal, A., Garg, H., & Bhatnagar, C. (2023). Sparse Feature based Progressive Algorithm for Co-Saliency Detection. *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 759–761.

Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, C. F. M. (2021). Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset. *IEEE Access : Practical Innovations, Open Solutions*, *9*, 22351–22370. Advance online publication. doi:10.1109/ACCESS.2021.3056614

McAfee. (2020). *McAfee Labs COVID-19 Threats Report*. Author.

Mehmood, M., Javed, T., Nebhen, J., Abbas, S., Abid, R., Bojja, G. R., & Rizwan, M. (2021). A hybrid approach for network intrusion detection. *Computers, Materials & Continua*, *70*(1), 91–107. Advance online publication. doi:10.32604/cmc.2022.019127

Mitchell, R., & Chen, I.-R. (2014). A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys*, *46*(4), 1–29. doi:10.1145/2542049

Naseri, T. S., & Gharehchopogh, F. S. (2022). A Feature Selection Based on the Farmland Fertility Algorithm for Improved Intrusion Detection Systems. *Journal of Network and Systems Management*, *30*(3), 40. Advance online publication. doi:10.1007/s10922-022-09653-9

Niyaz, Q., Sun, W., Javaid, A. Y., & Alam, M. (2015). A deep learning approach for network intrusion detection system. *EAI International Conference on Bio-Inspired Information and Communications Technologies (BICT)*. doi:10.4108/eai.3-12-2015.2262516

Novinson, M. (2020, June). *The 11 Biggest Ransomware Attacks Of 2020*. www.crn.com

Phuong, T. M., Lin, Z., & Altman, R. B. (2006). Choosing SNPs using feature selection. *Journal of Bioinformatics and Computational Biology*, *4*(2), 241–257. doi:10.1142/S0219720006001941 PMID:16819782

Prasad Yadav, D., Chauhan, S., Kada, B., & Kumar, A. (2023). Spatial attention-based dual stream transformer for concrete defect identification. *Measurement*, *218*, 113137. doi:10.1016/j.measurement.2023.113137

Resende, P. A. A., & Drummond, A. C. (2018). A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys*, *51*(3), 1–36. Advance online publication. doi:10.1145/3178582

Sarker, I. H., Abushark, Y. B., Alsolami, F., & Khan, A. I. (2020). IntruDTree: A machine learning based cyber security intrusion detection model. *Symmetry*, *12*(5), 754. Advance online publication. doi:10.3390/sym12050754

Senthilkumar, S., Brindha, K., Kryvinska, N., Bhattacharya, S., & Reddy Bojja, G. (2021). SCB-HC-ECC–Based Privacy Safeguard Protocol for Secure Cloud Storage of Smart Card–Based Health Care System. *Frontiers in Public Health*, *9*(September), 1–15. doi:10.3389/fpubh.2021.688399 PMID:34660507

Seth, S., Singh, G., & Kaur Chahal, K. (2021). A novel time efficient learning-based approach for smart intrusion detection system. *Journal of Big Data*, *8*(1), 111. Advance online publication. doi:10.1186/s40537-021-00498-8

Shams, E. A., & Rizaner, A. (2018). A novel support vector machine based intrusion detection system for mobile ad hoc networks. *Wireless Networks*, *24*(5), 1821–1829. Advance online publication. doi:10.1007/s11276-016-1439-0

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*. doi:10.5220/0006639801080116

Sinha, D., & Sharma, A. (2021). Cardiac Arrest Detection using Genetic machine Learning Algorithm. *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, 1–4. doi:10.1109/ISCON52037.2021.9702429

Sultana, N., Chilamkurti, N., Peng, W., & Alhadad, R. (2019). Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, *12*(2), 493–501. doi:10.1007/s12083-017-0630-0

Syarif, I., Prugel-Bennett, A., & Wills, G. (2012). Unsupervised Clustering Approach for Network Anomaly Detection. *Communications in Computer and Information Science*, *293*, 135–145. Advance online publication. doi:10.1007/978-3-642-30507-8_13

Tawil, A. A., & Sabri, K. E. (2021). A feature selection algorithm for intrusion detection system based on Moth Flame Optimization. *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, 377–381. doi:10.1109/ICIT52682.2021.9491690

Varghese, J. E., & Muniyal, B. (2017). An investigation of classification algorithms for intrusion detection system - A quantitative approach. *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*. doi:10.1109/ICACCI.2017.8126146

Vimala, S., Khanaa, V., & Nalini, C. (2019). A study on supervised machine learning algorithm to improvise intrusion detection systems for mobile ad hoc networks. *Cluster Computing*, *22*(S2, s2), 4065–4074. doi:10.1007/s10586-018-2686-x

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access : Practical Innovations, Open Solutions*, *7*(c), 41525–41550. doi:10.1109/ACCESS.2019.2895334

Warsi, S., & Dubey, P. P. (2019). Literature Review of Various Data Mining Based Techniques for Ids Data Classification. *International Journal of Innovative Research in Technology*, *5*(12), 68–70.

Wazid, M., & Das, A. K. (2016). An Efficient Hybrid Anomaly Detection Scheme Using K-Means Clustering for Wireless Sensor Networks. *Wireless Personal Communications*, *90*(4), 1971–2000. Advance online publication. doi:10.1007/s11277-016-3433-3

WHO. (2020). *WHO reports fivefold increase in cyber attacks, urges vigilance*. Coronavirus Disease (COVID-19) Pandemic.

Wu, S. X., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, *10*(1), 1–35. doi:10.1016/j.asoc.2009.06.019

Amit Singh is presently working as Scientist in Government of India. He has completed his Ph.D. from School of computer and Systems Sciences, JNU New Delhi. His research interests include Machine Learning, Data Mining and Cyber Security. He has several publications in international conference and journals. Dr. Singh has served as reviewer of several reputed journals as well as on the program committees of various conferences.

Jay Prakash is currently working as Assistant professor in Vijay Singh Pathik Govt.(PG) College Kairana, Shamli (Under the Department of Higher Education, Government of Uttar Pradesh, India). He has completed his PhD in computer science from Jawaharlal Nehru University, New Delhi, India. He has completed his M.Tech in Computer Science and Technology from Jawaharlal Nehru University, New Delhi, India after completing his MCA and BCA from Indira Gandhi National Open University, New Delhi, India. His research interests are databases, evolutionary algorithms for multi-objective optimization and Nature Inspired Computing.

Gaurav Kumar is currently working as an Assistant professor at GLA University Mathura, Uttar Pradesh, India. He has completed his PhD in computer science from Jawaharlal Nehru University, New Delhi, India. He has completed his M.Tech in Computer Science and Technology from Jawaharlal Nehru University, New Delhi, India. His research interests are in Decision Support Systems, Sentiment Analysis, Recommender Systems, and Cyber Security.

Praphula Kumar Jain working as a Assistant Professor at GLA University, Mathura, UP, INDIA. He obtained Ph.D. degree from the Department of Computer Science and Engineering, at Indian Institute of Technology(ISM), Dhanbad, JH, INDIA. He obtained a B.E degree in Computer Science and Engineering at the Faculty of Computer Science and Engineering, RGPV, Bhopal, MP, India, and an M.Tech degree in Computer Science and Engineering at the Indian Institute of Technology(ISM), Dhanbad, JH, INDIA. His Ph.D. research focuses on Machine Learning methods for various applications. He has three years of teaching experience and has published many technical articles in international scholarly journals such as Wireless personal communication springer, SN Applied Sciences Springer, Opsearch Springer. He has also published many conferences; few of them are RAIT, GUCON, etc. He also communicated three transaction papers on consumer recommendations prediction using machine learning techniques. He is also associated with reputed journals like the Journal of supercomputing, AI & Ethics, AI & Society, and conferences like FSDM, GLOBECOM 2020 as a reviewer. He is very much interested in social activity and providing free of cost education to underprivileged students.

Loknath Sai Ambati is an Assistant Professor of Business Analytics at Oklahoma City University and got his PhD in Management Information Systems at Dakota State University. His research interests include Social media mining, Healthcare Informatics, Deep learning and Artificial Intelligence in the realm of Information Systems. He published various research publications in both conferences and journals in the field of Information Systems.