

Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction

Md. Alamin Talukder^{1,2*}, Md. Manowarul Islam², Md Ashraf Uddin³, Khondokar Fida Hasan⁴, Selina Sharmin², Salem A. Alyami⁵ and Mohammad Ali Moni^{6*}

¹Department of Computer Science and Engineering, International University of Business Agriculture and Technology, Dhaka, Bangladesh.

²Department of Computer Science and Engineering, Jagannath University, Dhaka, Bangladesh.

³School of Information Technology, Deakin University, Waurn Ponds Campus, Geelong, Australia.

⁴School of Professional Studies, University of New South Wales (UNSW), 37 Constitution Avenue, ACT 2601, Australia.

⁵Department of Mathematics and Statistics, Faculty of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), 11432, Saudi Arabia.

⁶Artificial Intelligence & Data Science, School of Health and Rehabilitation Sciences, Faculty of Health and Behavioural Sciences, The University of Queensland St Lucia, QLD 4072, Australia..

*Corresponding author(s). E-mail(s):

alamintalukder.cse.jnu@gmail.com; m.moni@uq.edu.au;

Contributing authors: manowar@cse.jnu.ac.bd;

ashraf.uddin@deakin.edu.au; fida.hasan@unsw.edu.au;

selina@cse.jnu.ac.bd; saalyami@imamu.edu.sa;

Abstract

Cybersecurity has emerged as a critical global concern. Intrusion Detection Systems (IDS) play a critical role in protecting interconnected networks by detecting malicious actors and activities. Machine Learning (ML)-based behavior analysis within the IDS has considerable potential for detecting dynamic cyber threats, identifying abnormalities, and identifying malicious conduct within the network. However, as the number of data grows, dimension reduction becomes an increasingly difficult task when training ML models. Addressing this, our paper introduces a novel ML-based network intrusion detection model that uses Random Oversampling (RO) to address data imbalance and Stacking Feature Embedding based on clustering results, as well as Principal Component Analysis (PCA) for dimension reduction and is specifically designed for large and imbalanced datasets. This model's performance is carefully evaluated using three cutting-edge benchmark datasets: UNSW-NB15, CIC-IDS-2017, and CIC-IDS-2018. On the UNSW-NB15 dataset, our trials show that the RF and ET models achieve accuracy rates of 99.59% and 99.95%, respectively. Furthermore, using the CIC-IDS2017 dataset, DT, RF, and ET models reach 99.99% accuracy, while DT and RF models obtain 99.94% accuracy on CIC-IDS2018. These performance results continuously outperform the state-of-art, indicating significant progress in the field of network intrusion detection. This achievement demonstrates the efficacy of the suggested methodology, which can be used practically to accurately monitor and identify network traffic intrusions, thereby blocking possible threats.

Keywords: Intrusion detection system; Feature extraction; Random oversampling; Principal component analysis; Machine learning

1 Introduction

Cybersecurity, in the current era, has emerged as an international imperative, driven by the critical need to protect systems from unwanted, unauthorized, and unforeseen interference (Mueller, 2021). These interferences can range from data breaches and information theft to threats that undermine the integrity and functionality of systems. Safeguarding against such threats is paramount in ensuring the smooth operation of systems, protecting sensitive data, and preserving user trust (Marwala, 2023). Intrusion Detection Systems (IDS) have traditionally served as a cornerstone of perimeter security (George et al, 2023; Nguyen et al, 2023).

These systems are crafted with the purpose of uncovering and responding to suspicious or malicious activities within a network or system. Nevertheless, the conventional signature-based intrusion detection methods, reliant on established attack patterns and signatures, have been found lacking in the face of ever-evolving and sophisticated cyber threats. These solutions are engineered to identify and react to questionable or potentially harmful actions

occurring within a network or system. Nevertheless, conventional intrusion detection methods, which hinge on established attack patterns and signatures, have demonstrated their inadequacy when confronted with ever-changing and increasingly complex cyber threats (Khan et al, 2022; Talukder et al, 2023a).

To address the shortcomings of traditional IDS, the cybersecurity community has turned its attention to Machine Learning (ML) as a promising solution. ML-enabled IDS leverages behavior analysis to detect anomalies and threats, offering the potential for significantly higher accuracy and faster detection times (Schmitt, 2023; Preuveneers and Joosen, 2021). This paradigm shift in intrusion detection holds the promise of not only bolstering security but also reshaping the privacy landscape. This shift towards ML-enabled intrusion detection has sparked concerns regarding both privacy and the field of data science (Singh and Singh, 2023; Mohammadi et al, 2019). ML algorithms, while effective at identifying threats, often require access to sensitive data. Balancing the need for security with privacy concerns is a challenge that demands innovative and ethical solutions (Allahrakha, 2023).

ML in cybersecurity serves as a powerful tool to augment the ability of systems to understand diverse patterns and forecast potential data threats (Sarker et al, 2020). It optimizes processing and training procedures to construct models that can effectively safeguard systems against dubious and spyware activities (Hussain et al, 2020; Talukder et al, 2023a). It is a transformative technology that empowers systems to learn and adapt from data, making intelligent decisions without being explicitly programmed (Mishra and Tyagi, 2022). In the context of IDS, ML algorithms utilize historical and real-time data to identify patterns of normal behavior and anomalies that may indicate security threats. By training on diverse datasets, these algorithms become proficient at recognizing new and emerging attack vectors. ML enhances IDS systems by providing faster and more accurate threat detection, reducing false positives, and adapting to evolving threats (Jayalaxmi et al, 2022). It empowers security systems to efficiently safeguard networks and data against unauthorized access and malicious activity (Kafi and Akter, 2023).

In today's landscape, the optimization of processing and training procedures is imperative for constructing models that can effectively safeguard systems against dubious and spyware activities (Sarker et al, 2020). However, it's worth noting that many contemporary ML-IDS solutions tend to be limited by their reliance on small, outdated and balanced datasets for model development (Istiaque et al, 2021; Cholakoska et al, 2021; Narayanasami et al, 2021). The focus on these smaller, often outdated datasets, coupled with imbalances in the data distribution, while facilitating preprocessing and training with diverse ML algorithms, raises questions regarding the practical applicability of these models in real-world scenarios, specifically when dealing with big data. The achievable accuracy of such models often hinges on the intricacies of dataset preprocessing and the selection of suitable algorithms, adding an additional layer of complexity to their effectiveness (Norwahidayah et al, 2021; Bhati and Rai, 2021).

Therefore, we need to develop and validate ML-based intrusion detection for large, imbalanced datasets where all potential attack scenarios are encompassed. To address this gap, Our research places a significant emphasis on constructing a robust and well-structured framework that accommodates the detection of network intrusion in a more efficient manner on substantial datasets. We employ data preprocessing techniques, including data normalization, feature resampling, Stacking Feature Embedded and dimension reduction techniques, to address the unique challenges posed by big datasets. The Key techniques of our approach are as follows:

- Stacking Feature Embedded (SFE): This technique enhances detection accuracy by introducing meta-features, providing a deeper insight into data patterns and anomalies.
- Random Oversampling (RO): By mitigating class imbalance issues, RO ensures equitable consideration of minority classes, resulting in a more balanced and reliable intrusion detection system.
- Principal Component Analysis (PCA): PCA optimizes the feature space, reducing dimensionality while preserving vital information, thus enhancing the efficiency and effectiveness of our ML models.

This comprehensive approach seeks to bridge the gap in intrusion detection, accommodating the intricacies of large, imbalanced datasets, and improving the robustness of security measures in the face of evolving threats.

Our proposed model's performance is rigorously evaluated across a spectrum of ML classifiers, including Decision Tree (DT), Random Forest (RF), Extra Tree (ET), and Extreme Gradient Boosting (XGB). These classifiers are trained using a reduced feature set. We assess our model using a comprehensive set of performance indicators, encompassing precision, recall, f1-score, confusion matrix, accuracy and the roc curve. The ML algorithms integrated into our framework demonstrate an exceptional ability to detect attacks, consistently achieving accuracy rates exceeding 99.9%. This thorough performance evaluation ensures the robustness and reliability of our intrusion detection system, highlighting its effectiveness in identifying and countering potential threats. In summary, this paper's contribution can be encapsulated as follows:

- We proposed a novel intrusion detection approach using efficient preprocessing, oversampling management, stacking feature embedding, and dimensionality reduction to enhance intrusion detection performance.
- We addressed the issue of imbalanced data by implementing a random oversampling strategy to ensure balanced consideration of minority and majority classes, leading to more robust intrusion detection.
- The introduction of Stacking Feature Embedded (SFE) enhances detection accuracy by introducing meta-features, providing a comprehensive understanding of data patterns and anomalies.
- Our utilization of Principal Component Analysis (PCA) for feature extraction optimizes intrusion detection performance while minimizing the dimensionality of the original dataset.

- Evaluation of our approach with various ML algorithms on popular large and imbalanced datasets demonstrates its effectiveness in significantly improving intrusion detection accuracy and robustness.

The subsequent sections of this paper offer a concise overview of our proposed work. In Section 2, we delve into the literature review, providing insights into the existing body of knowledge. Section 3 is dedicated to the detailed description of our proposed methodology. Section 4 outlines the experimental setup and evaluation procedures. Finally, in Section 7, we draw the conclusions from our findings and explore avenues for future research.

2 Literature Review

ML strategies have been widely utilized in cybersecurity over the last several decades due to their capacity to retrieve hidden patterns on the variations between malevolent and legitimate patterns (Zhang et al, 2020; Das et al, 2023). ML is an effective research tool for detecting any anomalies in the network stream of traffic (Bhavani et al, 2020). As a result, previous researchers explored a variety of algorithms based on ML as well as hybrid and DL models in IDS.

2.1 ML Approaches

Moualla et al (2021) proposed a revolutionary network IDS model that plays a crucial role in network security and combats existing cyberattacks on networks utilizing the UNSW-NB15 data as a baseline. It was a dynamically scalable multiclass ML-based network with several phases. The imbalance was handled by SMOTE technique, after that based on the Gini Impurity criterion, it employed the ET Classifier and finally, a pre-trained Extreme Learning Machine (ELM) was utilized to classify each of the attacks using binary. Using the outputs of the ELM classifier as inputs to a fully connected layer, a logistic regression layer was employed to produce soft judgments for all classes and attained 98.43% accuracy.

Kasongo and Sun (2020b) presented a filter-based feature-dropping technique on the UNSW-NB15 IDS dataset, employing the XGB algorithm, and assessed its performance using various predictive algorithms, including Decision Tree (DT), Artificial Neural Network (ANN), Logistic Regression (LR), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). They demonstrated that their approach led to a significant enhancement in binary accuracy, increasing it from 88.13% to 90.85%. The overall accuracy rates for binary were 90.85% for DT, 84.4% for ANN, 77.64% for LR, 84.46% for KNN, and 60.89% for SVM. For multiclass, the accuracy rates were 67.57% for DT, 77.51% for ANN, 65.29% for LR, 72.30% for KNN, and 53.95% for SVM, all of which were evaluated using the 19 optimal selected features.

Nimbalkar and Kshirsagar (2021) offered a feature selection method for IDS using Information Grain (IG) and Grain Ratio (GR) where they selected

50% of the top most features to build their model detecting Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. The studies were carried out using well-known datasets, such as KDDCUP'99 and BOT-IOT. They selected 16 and 19 features for BOT-IOT and KDDCUP'99 datasets, respectively and then trained the model using the JRip classifier to reach the desired performance. They achieved 99.99% and 99.57% accuracy for BOT-IOT and KDDCUP'99 datasets, respectively.

Kumar et al (2020) presented IDS model that detects intrusions based on misuse to protect networks from modern threats, such as DoS attacks or exploits, probes, generics and so on. Intrusion detection rate (IDR) and false alarm rate (FAR) were determined using the UNSW-NB15 dataset. The IG and C5 classifier were utilized where, IG was used to pick 13 out of 47 features for feature selection and C5 gave 99.37% accuracy rate.

Ahmad et al (2021) proposed a feature clustering-based ML model where clusters were applied for Flow, Message Queuing Telemetry Transport (MQTT) and Transmission Control Protocol (TCP). The clustering process eliminated the overfitting that was the curse of dimensionality and data-set inequity. Various supervised ML methods were utilized on the clusters, including RF and SVM. They employed the UNSW-NB15 dataset to train and test the model and found that RF produced 98.67% accuracy in binary and 97.37% accuracy in multiclass.

Ahmad et al (2021) introduced an innovative ML model based on feature clustering, with distinct clusters created for Flow, Message Queuing Telemetry Transport (MQTT), and Transmission Control Protocol (TCP). This clustering approach effectively addressed the challenges of overfitting arising from high dimensionality and dataset imbalances. They applied a range of supervised machine learning methods, including RF and SVM, to these clusters. Using the UNSW-NB15 dataset for model training and evaluation, their results showed that RF achieved impressive accuracy, reaching 98.67% in binary classification and 97.37% in multi-class classification.

Kshirsagar and Kumar (2021) presented a filter-based feature selection technique that leveraged IG Ratio (IGR), Correlation Ratio (CR), and ReliefF (ReF). This method generated a feature subset by considering the average weight of each classifier, complemented by a Subset Combination Strategy (SCS). For CICIDS 2017 dataset, the number of features was lowered from 77 to 24 and for KDDCUP'99, it was cut from 41 to 12. For CIC-IDS2017, it achieved a 99.95% accuracy rate in 133.66 sec using PART and for the KDDCUP'99 dataset, it achieved a 99.32% accuracy rate and took 11.22 sec.

In order to minimize the volume of the dataset, (Mugabo et al, 2021) employed evolutionary approach-based feature selection (EFS) and a concurrent MapReduce technique was applied to partition the input data into the most crucial characteristics. After that, their model was classified for either normal or attack using the RF classifier. They used the popular KDDCUP'99 dataset to evaluate performance and classify normal and deviant behavior. They selected 15 features to assess their model's performance and its accuracy was found almost 93.9%.

Talita et al (2021) developed an innovative approach that integrates Particle Swarm Optimization (PSO) for feature selection with the Naive Bayes (NB) classification algorithm, which was applied to the KDDCUP'99 dataset. This dataset comprised of over 400 thousand records and featured more than 40 characteristics. To optimize computational resources and memory usage, PSO was employed to select the most relevant 38 features from the original set of over 40. The outcome of this method yielded an impressive accuracy rate of 99.12%, demonstrating superior efficiency in terms of both computation time and classification accuracy when compared to other feature selection techniques.

Seth et al (2021) developed an IDS model that reduces prediction delay by minimizing the model's sophistication using a hybrid feature selection (HFS) strategy. A quick gradient boosting technique called Light Gradient Boosting Machine (LightGBM) was used to build the model. This approach cut prediction latency by 44.52% to 2.25% and model development time by 52.68% to 17.94% using the CIC-IDS2018 dataset. It can achieve excellent accuracy by combining attribute choosing and LightGBM. The developed model achieved 97.73%, 96%, 99.3%, accuracy, sensitivity, precision rate respectively and a relatively low prediction latency.

Hammad et al (2021) presented a model using t-SNERF to identify the cyber-attacks where t-SNERF was used for feature correlations, data reduction and trained the model using RF. To evaluate the model UNSW-NB15, CIC-IDS2017 and Phishing were employed. The offered innovative methodology outperformed current methods. The accuracy rate was 100% for UNSW-NB15, 99.70% for Phishing and 99.78% for CIC-IDS2017.

Guezzaz et al (2021) focused on enhancing the reliability of Network Intrusion Detection (NID) through the utilization of the DT (Decision Tree) algorithm. Their approach entailed two key steps: data quality improvement and feature selection based on entropy decision, followed by the construction of a dependable NID system using the DT classifier. This proposed model underwent evaluation using two well-known datasets, NSL-KDD and CIC-IDS2017, and yielded impressive results. Specifically, the model achieved a remarkable accuracy of 99.42% on the NSL-KDD dataset and 98.80% on the CIC-IDS2017 dataset.

Stiawan et al (2020) offered a strategy for analyzing integral and essential aspects of massive network data, increasing traffic anomaly detection accuracy and speed. They used the CIC-IDS2017 dataset and picked important and significant features using IG, as well as sorting and grouping features according to their minimal weight values and then train the dataset using various ML classifiers. The number of relevant and meaningful attributes generated by IG has a considerable impact on accuracy and execution time. A number of ML algorithms, such as Random Tree (RT), RF, NB, Bayes Net (BN) and J48 were used to train the model but RF provided the best accuracy. In the RF classifier, they used 22 relevant selected features, had the best accuracy of

99.86%, whereas the J48 classifier algorithm, which used 52 relevant selected features but took longer to execute, had the highest accuracy of 99.87%.

2.2 Deep Learning Approaches

[Aleesa et al \(2021\)](#) explored the application of DL models for binary and multiclass classification in the context of IDS. They conducted their evaluations using the UNSW-NB15 dataset. Specifically, the study assessed the effectiveness of three distinct types of neural network models: Deep Neural Network (DNN), Recurrent Neural Network (RNN), and ANN. They used data cleaning techniques, such as handling missing values and categorical values, followed by min-max normalization in order to make it more accurate. The efficiency was evaluated on accuracy; where, the ANN, RNN and DNN provided 99.26%, 85.42%, 99.22% for binary classification and 97.89%, 85.4% and 95.9% for multilabel classification respectively.

[Choudhary and Kesswani \(2020\)](#) introduced an IDS based on DNN for identifying IoT-related attacks. They evaluated the effectiveness of the DNN-based approach by testing it on three widely employed datasets: UNSW-NB15, KDDCUP'99 and NSL-KDD. The outcomes indicated that the DNN-based method achieved an accuracy rate of 91.50% when applied to each of these datasets.

In the study by [Al and Dener \(2021\)](#), a more effective IDS model was introduced, utilizing a combination of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) architectures. To address the issue of imbalanced datasets, they applied the Synthetic Minority Oversampling Technique (SMOTE) in conjunction with Tomek-Link, referred to as the STL method. The research implementation was carried out using PySpark, and two distinct datasets, namely CICIDS-001 for multiclass and UNSW-NB15 for binary classification, were used to evaluate the model's performance. The proposed method was benchmarked against various popular ML and DL algorithms. In multilabel classification, the proposed approach performed an outstanding accuracy of 99.83%, while in binary classification, it demonstrated a high accuracy of 99.17%.

[Adeyemo et al \(2019\)](#) explored the effectiveness of network IDS using DL and ensemble techniques. They applied an LSTM model, a homogeneous approach with an optimized bagged RF and a heterogeneous approach with four standard classifiers. The evaluation was conducted on the UNSW-NB15 dataset, split into two configurations: two and multi-attack datasets. The results showed that the LSTM achieved an 80% detection accuracy for the binary dataset and 72% for the multi-attack dataset. The homogeneous ensemble method reached impressive accuracy rates of 98% for the binary dataset and 87.4% for the multi-attack dataset. Similarly, the heterogeneous ensemble method also performed well, with a detection accuracy of 97% for the binary and 85.23% for the multi-attack dataset. This research highlights the promising performance of ensemble methods in the context of Intrusion Detection Systems.

Kim et al (2020) developed a DL model primarily focusing on detecting denial of service (DoS) attacks. The KDDCUP'99 and CSE-CIC-IDS2018 were employed to assess the model's performance. Notably, the latter dataset contained more sophisticated DoS attacks than the former. The study concentrated on the DoS category within both datasets and harnessed a CNN for model development. A comparative analysis was conducted between the CNN and RNN. In the case of KDDCUP'99, the CNN demonstrated impressive accuracy rates, surpassing 99% accuracy in binary and multiclass, while the RNN achieved 99% accuracy in binary and 93% accuracy in multiclass. For the CSE-CIC-IDS2018 dataset, the CNN model exhibited an average accuracy of 91.5%, whereas the RNN model averaged 65% accuracy. The CNN model consistently outperformed the RNN model in identifying specific DoS attacks characterized by similar attributes.

2.3 Hybrid Approaches

Bhardwaj et al (2021) presented a hybrid strategy that combines a DNN model with Ant Colony Optimization (ACO) for learning premium hyperparameters for successful DNN classification in a cloud setting. DNN detects attacks more accurately by the usage of ideal settings. They used the CIC-IDS2017 dataset and got well performance. The detection and accuracy performance are both superior to state-of-the-art approaches, at 95.74% and 98.25%, respectively.

Khan (2021) introduced a novel approach known as the Hybrid Convolutional Recurrent Neural Network for IDS (HCRNNIDS). This approach represents a hybrid IDS paradigm that leverages DL techniques to predict and classify malicious intrusions on the internet. In this model, the CNN was responsible for gathering local information through convolution, while the RNN was employed to capture temporal features, thereby enhancing the effectiveness and predictive capabilities of the ID system. To assess the model's performance, the researchers conducted simulations using publicly available ID data, focusing on the contemporary and reputable CSE-CIC-DS2018 dataset. Through the application of a 10-fold cross-validation methodology, the proposed hybrid model exhibited significant improvements over traditional ID approaches. It achieved a remarkable level of accuracy in fraudulent detection and prevention, reaching up to 97.75% for the CIC-IDS2018 dataset.

Kasongo and Sun (2020a) presented a Wrapper-based Feature Extraction Unit (WFEU) that leverages the Extra Trees technique to create a reduced optimum feature vector for a Feed-Forward Deep Neural Network (FFDNN) wireless IDS system. The proficiency was studied using the UNSW-NB15 and AWID ID datasets. Several ML algorithms, such as RF and SVM are compared to WFEU-FFDNN as well as NB and DT. Using the UNSW-NB15, the WFEU produced an ideal feature vector consisting of 22 features and achieved an overall accuracy of 87.10% for binary and 77.16% for multiclass. On the other hand, for AWID, 26 features were selected and revealed an accuracy of 99.66% for binary and 99.77% for multiclass.

[Kasongo and Sun \(2020a\)](#) introduced a Wrapper-based Feature Extraction Unit (WFEU), which harnessed the power of the Extra Trees technique to construct an optimized and reduced feature vector tailored for a Feed-Forward Deep Neural Network (FFDNN) wireless IDS. To evaluate its efficiency, they conducted experiments on the UNSW-NB15 and AWID intrusion detection datasets. Comparative analyses were performed against various ML algorithms, including RF and SVM, along with conventional approaches such as NB and DT. On the UNSW-NB15 dataset, the WFEU method yielded an optimal feature vector consisting of 22 features, achieving an impressive overall accuracy of 87.10% for binary and 77.16% for multiclass. For the AWID dataset, a feature set of 26 features was selected, resulting in remarkable accuracy rates of 99.66% for binary and 99.77% for multiclass.

[Zhang et al \(2020\)](#) proposed a novel strategy for addressing imbalanced intrusion detection, surpassing existing intrusion detection algorithms. They introduced the SGM-CNN model, which combined Synthetic Minority Over-sampling Technique (SMOTE) with a Gaussian Mixture Model (GMM). The model was validated using the UNSW-NB15 and CIC-IDS2017 datasets. On the UNSW-NB15 dataset, the model exhibited remarkable accuracy rates of 99.74% for binary and 96.54% for multiclass. Furthermore, for the CIC-IDS2017 dataset, achieved an impressive detection rate of 99.85%.

[Hassan et al \(2020\)](#) introduced a hybrid DL model for efficient network intrusion detection, combining CNN and Weight-Dropped Long Short-Term Memory (WDLSTM). The CNN was employed to extract crucial features from IDS, while the WDLSTM was utilized to capture long-term dependencies and mitigate gradient vanishing issues. Their experiments focused on the UNSW-NB15 dataset. The CNN-WDLSTM model exhibited an overall accuracy rate of 97.17% for binary and 98.43% for multiclass.

The summary of related papers can be found in [Table 1](#)

2.4 Limitations of the Existing Works

One notable limitation in the existing works is their reliance on older datasets such as KDDCUP'99 and NSLKDD, which lack recent attack scenarios featured in datasets like CIC-IDS2018. Consequently, these works may not effectively address contemporary and evolving cyber threats. Furthermore, many of these works overlook the importance of data balancing techniques. This omission leads to variations in their model's performance, particularly in terms of false positive and negative as well as true positive and negative rates. Moreover, the performance scores of existing works are inadequate for efficient network attack detection. As a result, their models may not be suitable for real-world scenarios where data imbalance is a common issue. Additionally, most of these papers employ the use of the complete feature set for conducting their experiments. This approach can be time-consuming and computationally intensive, making it less practical for real-time anomaly detection in large-scale network environments. Another notable limitation is the absence of time complexity analysis in the majority of these papers. Understanding

SL. NO	ML Technique	Algorithm	Author	Dataset	Accuracy (In %)		
					Binary	Multi-class	
1	ML	RF	Ahmad et al (2021)	UNSW-NB15	98.67	97.37	
2		ELM	Moualla et al (2021)		98.43		
3		DT	Kasongo and Sun (2020b)		90.85	67.57	
		ANN			84.4	77.51	
		LR			77.64	65.29	
		KNN			84.46	72.30	
		SVM			60.89	53.95	
4		C5	Kumar et al (2020)			99.3	
5		PART	Kshirsagar and Kumar (2021)		CIC-IDS2017	99.95	
					KDDCUP'99	99.32	
6	MapReduce+RF	Mugabo et al (2021)		93.9			
7	PSO+NB	Talita et al (2021)		99.12			
8	HFS+LightGBM	Seth et al (2021)	CIC-IDS2018		97.73		
9	IG+GR+JRip	Nimbalkar and Kshirsagar (2021)	KDDCUP'99	99.57			
			BOT-IOT	99.99			
10	t-SNERF	Hammad et al (2021)	UNSW-NB15	100			
			CIC-IDS2017	99.78			
			Phishing	99.70			
11	DT	Guezzaz et al (2021)	NSL-KDD	99.42			
12	IG+RF	Stiawan et al (2020)	CIC-IDS2017	98.80			
				99.86			
	IG+J48			99.87			
13	Deep Learning	DNN	Aleesa et al (2021)	UNSW-NB15	99.92	95.9	
					RNN	85.42	85.4
					ANN	99.26	97.89
14	DNN	Choudhary and Kesswani (2020)	KDDCUP'99	91.50			
			NSL-KDD				
			UNSW-NB15				
15	LSTM+CNN	Al and Dener (2021)	CICIDS-001	99.83			
			UNSW-NB15	99.17			
16	CNN	Hassan et al (2020)	KDDCUP'99	99			
			CIC-IDS2018	91.50			
17	DNN+ACO	Bhardwaj et al (2021)	CIC-IDS2017	98.25			
18	CNN+RNN	Khan (2021)	CIC-IDS2018	97.75			
19	Hybrid Learning	WFEU+FFDN	Kasongo and Sun (2020a)	AWID	99.66	99.77	
				UNSW-NB15	87.10	77.16	
20	SGM+CNN	Zhang et al (2020)		99.74	96.54		
			CIC-IDS2017		99.85		
21	WDLSTM+CNN	Hassan et al (2020)	UNSW-NB15	97.17	98.43		

Table 1 Related work summary of various ML Techniques

the computational demands of their algorithms is crucial for assessing their feasibility in real-world applications. Lastly, No existing works explore the Stacking Feature Embedded approach, a method that can potentially enhance

intrusion detection accuracy by incorporating meta-features. The omission of this approach limits the comprehensiveness and effectiveness of their intrusion detection models.

3 Methodology

This section describes our proposed framework and data preprocessing techniques, including feature resampling and scaling, stacking feature embedded and feature extraction. We also provide a brief recap of the ML algorithms employed for intrusion detection. The proposed framework for stacking feature embedded and dimension reduction for intrusion detection on big and imbalanced datasets can ensure a reliable secure network by detecting the receiving packets as normal or attack packets. The schematic block diagram of our proposed paradigm is illustrated in Figure 1. The approach is structured into the following phases:

- Step-1: Initially, preprocessing is accomplished by handling the missing value, removing space from column names, dropping the duplicate rows, merging the similar classes with low instances and reducing the size of the dataset by converting data types int64 to int32 and float64 to float 32. This step is crucial for data quality improvement. It addresses missing values, data type conversions, and other data cleaning tasks to prepare the dataset for analysis.
- Step-2: In the feature scaling step, we use standardization for input features and label encoding for output features. Standardization of input features and label encoding of output features ensure that the data is on the same scale, which is essential for ML algorithms.
- Step-3: In the feature resemble step, we use Random Oversampling (RO) by adding random samples to the minority class to solve the problem of the imbalanced dataset and make it a balanced dataset. To address the class imbalance using RO in the dataset, making it more suitable for training ML models.
- Step-4: In the stacking feature embedded step, we utilize the clustering results as meta-features, enhancing the original features within the IDS dataset. Using clustering results as meta-features enriches the dataset with additional information derived from underlying patterns and structures, which can enhance model performance.
- Step-5: In the feature extraction step, we reduce the dimensionality of the dataset by extracting the feature using Principal Component Analysis (PCA). Dimensionality reduction using PCA can help improve model efficiency by reducing the number of features while retaining essential information.
- Step-6: In this phase, we perform data splitting to create separate training and testing subsets from the pre-processed dataset. We employ a k-fold cross-validation technique with k set to 10. This step is crucial to enhance

accuracy and evaluate the model’s performance effectively. It ensures that the model is rigorously tested on different subsets of the data.

- Step-7: In this step, we evaluate the model’s performance using four established ML algorithms: DT, RF, ET, and XGB Classification. We employ k-fold cross-validation for these models, aiming to identify the most suitable ML models for IDS.
- Step-8: In the final phase, we assess the model’s performance using various performance metrics, including Precision, Recall, Confusion Matrix, Accuracy, F1-score, and ROC Curve. These metrics serve as benchmarks for comparing our model’s performance with existing models, enabling us to select the best-performing model for the IDS task.

3.1 Data Collection

We have utilized two benchmark big datasets for our research schemes, namely UNSW-NB15 (Moustafa and Slay, 2015) and CIC-IDS2017 (Sharafaldin et al, 2018b). Both datasets are realistic to the IDS environments and have up-to-date attack categories to detect attacks. The following sections give a closer look at each of the datasets.

3.1.1 UNSW-NB15

UNSW-NB 15 is a fairly recent dataset that comprises a vast quantity of internet traffic patterns with 9 types of malicious activities, as opposed to KDD’98, NSL-KDD, KDDCUP’ 99, CAIDA, Kyoto 2006 + and ISCX dataset (Protić et al, 2023). It includes current minimal imprint assaults as well as contemporary Netflow for both regular and unusual situations. The IXIA perforectStrom apparatus was utilized within the Cyber Run Lab of the ACCS (Australian Center for Cyber Security) to make synthetic modern assault and genuine advanced ordinary behaviors for creating the crude network packets of the UNSW-NB15 dataset (Moustafa and Slay, 2015). By catching 100 GB of the crude activity utilizing the Tcpcap instrument. To create completely 49 features with the class, 12 algorithms are created by employing Argus and Bro-IDS devices. Within the four CSV records, two million and 540,044 records are put away. A setup is conducted on the dataset by dividing it into a training set and a testing set. It has a total of 257673 entries were 175341 entries in the training set and 82332 entries in the testing set. With their respective class labels, the dataset includes both real-world modern typical behavior and staged attack actions from the present day. It incorporates 9 distinct contemporary new attacks as well as a large range of real-world activities (Moustafa and Slay, 2016). Among the sorts of attacks are fuzzier, backdoor, analysis, reconnaissance, exploit, generic, DoS, shellcode and worm attacks. The data are very unbalanced in this dataset. Figures 2 and 3 demonstrate the distribution of benign and attack data for binary and multi labels before preprocessing, after preprocessing and after oversampling, respectively, while Table 2 briefly

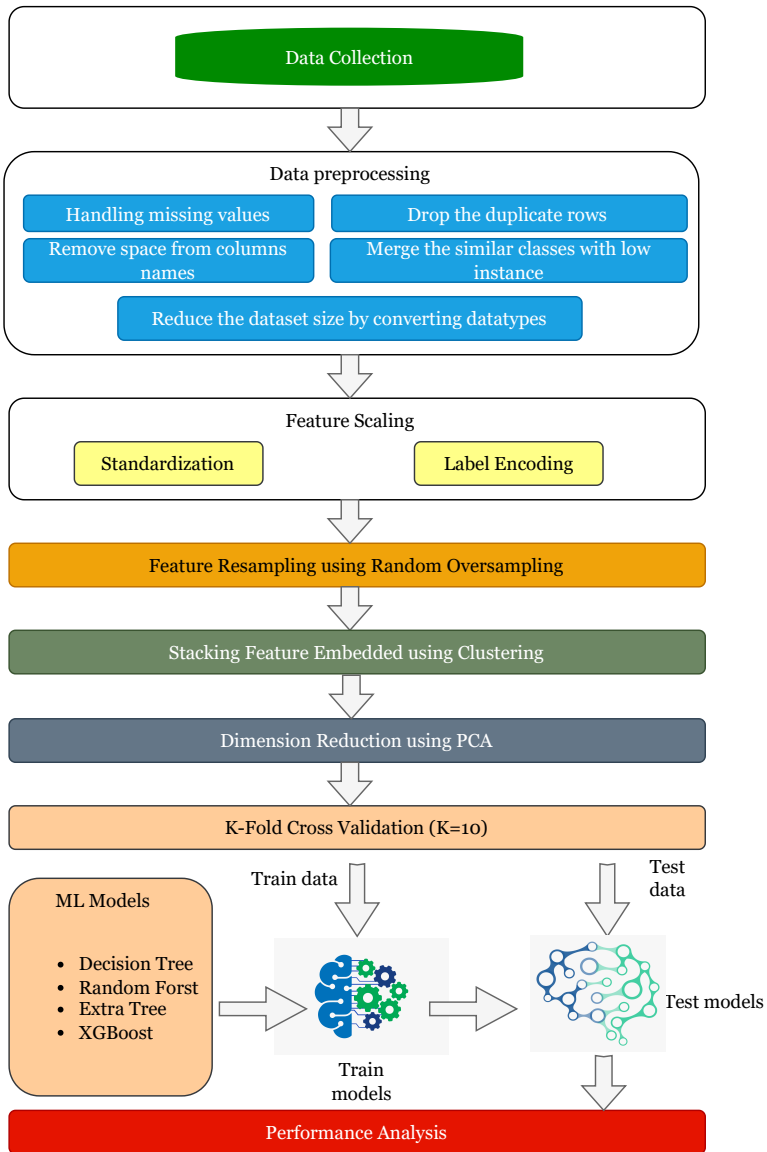


Fig. 1 Proposed framework for stacking feature embedded with PCA for intrusion detection.

describes all of the attack classes in this dataset. In Table 3 shows the frequency distribution of attack categories.

3.1.2 CIC-IDS2017

Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) serve as vital defenses against the relentless and increasingly sophisticated

Attack Categories	Brief description
Fuzzers	By supplying a vast volume of random data, the insider tries to crash a software, operating system, or network.
Backdoor	Cyber attackers can get illegal access to websites using this form of software. By focusing on vulnerable entry points, the intruders were able to disseminate malware throughout the system.
Analysis	Pay special attention to malware attacks and computer intrusions in which attackers gain permissions by utilizing their technological capabilities.
Reconnaissance	Gathers data on system flaws that can be used to gain control of the system.
Exploit	A piece of software that exploits security flaws and vulnerabilities. An attacker can gain unrestricted access with this attack.
Generic	Has the ability to decrypt all block ciphers without having to know the cipher's structure.
DoS	User access to machines and network resources can be suspended by an attacker. By delivering too much confusing traffic, the attacker overwhelms the network.
Shellcode	It is a sequence of instructions that executes software commands to harm a machine.
Worm	It includes security flaws that attack the host machine and spread throughout the network. It is capable of exploiting many applications' security flaws.

Table 2 Briefly describe all of the attack classes in the UNSW-NB15 dataset

Attack Categories	Count	% (percentage)
Normal	93000	36.09
Generic	58871	22.85
Exploits	44525	17.28
Fuzzers	24246	9.41
DoS	16353	6.35
Reconnaissance	13987	5.43
Analysis	2677	1.04
Backdoor	2329	0.90
Shellcode	1511	0.59
Worms	174	0.07
Total	257673	100

Table 3 The frequency distribution of attack categories of the UNSW-NB15 dataset

landscape of network attacks. An ongoing challenge in the field is the scarcity of dependable test and validation datasets, which hinders the consistent and accurate evaluation of anomaly-based intrusion detection methods, as noted by Sharafaldin et al. in 2018 (Sharafaldin et al, 2018b). One promising solution to this issue lies in the CIC-IDS2017 dataset, which features a comprehensive collection of benign network traffic and a diverse range of contemporary, frequently encountered cyber attacks. This dataset closely mirrors real-world scenarios by utilizing PCAPs. Moreover, it provides a detailed summary of traffic monitoring through the utilization of CICFlowMeter, offering labeled network flows with key information, including timestamps, source and destination IP addresses, source and destination ports, protocols, and delineated attack vectors, all meticulously documented in CSV files, as described by Sharafaldin et al. in their 2018 work on intrusion detection (Sharafaldin et al, 2018a). To create various kinds of assaults such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS presented the B-Profile technology, which profiles the conceptual activity of individual contacts and produces lifelike benign baseline flow. According to

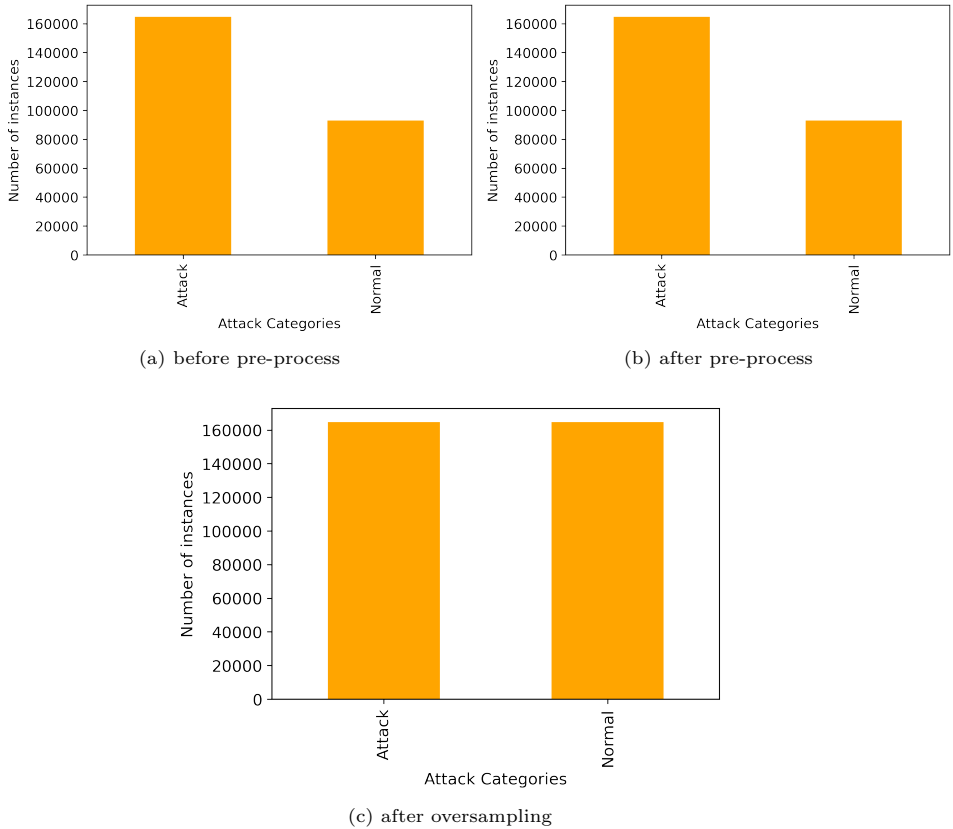


Fig. 2 Binary frequency distribution of UNSW-NB15 dataset

the dataset's appraisal methodology which was proposed (Gharib et al, 2016), there are 11 requirements that must be met in order to create a trustworthy benchmark dataset. only this dataset meets all of the requirements and hardly any of the prior IDS datasets were capable of covering those 11 requirements. The distribution of binary and multi categories of attack labels before preprocessing, after preprocessing and after oversampling is depicted in Figure4 and Figure5 and attack categories are shown in Table 4.

3.1.3 CIC-IDS2018

The CSE-CIC-IDS2018 dataset (Sharafaldin et al, 2018b), a collaborative initiative by the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC), endeavors to meet the pressing need for comprehensive datasets suitable for rigorously testing intrusion detection systems, with a specific focus on network-based anomaly detection. Anomaly detection holds significant promise for identifying emerging threats, but its

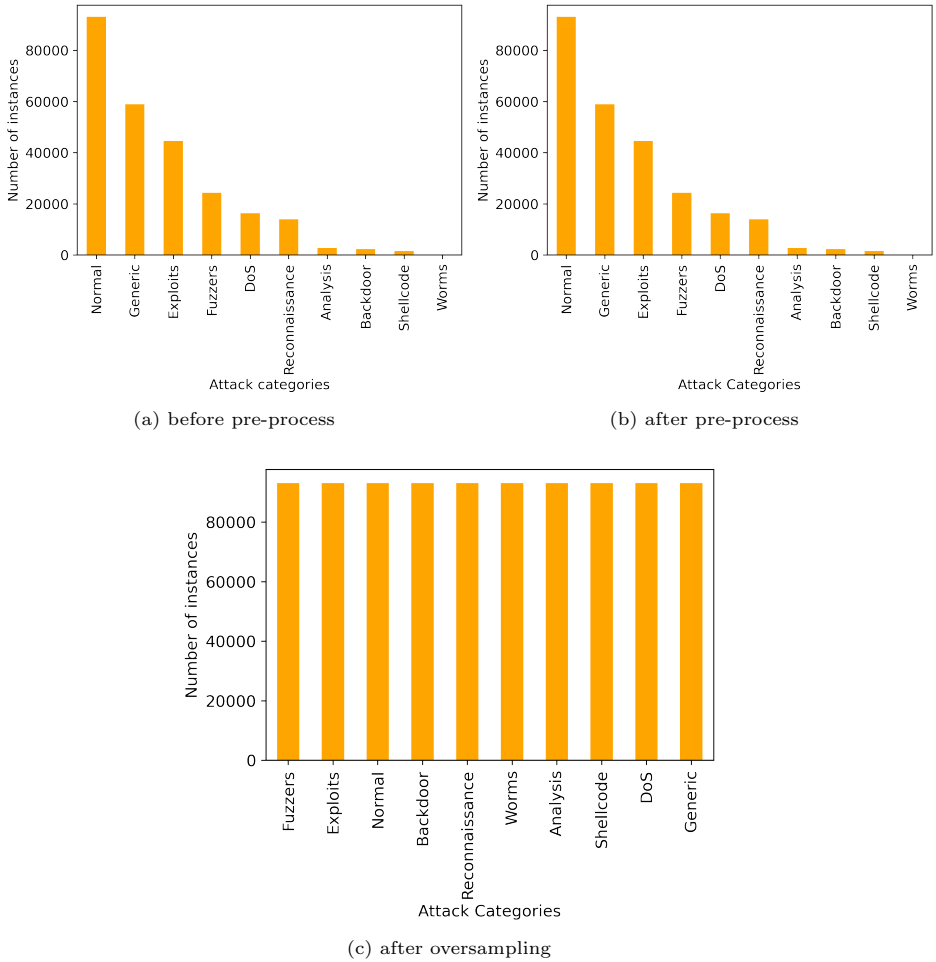


Fig. 3 multilabel frequency distribution of UNSW-NB15 dataset

Attack Categories	Count	% (percentage)
BENIGN	2273097	80.3
DoS Hulk	231073	8.16
PortScan	158930	5.61
DDoS	128027	4.52
DoS GoldenEye	10293	0.36
FTP-Patator	7938	0.28
SSH-Patator	5897	0.21
DoS slowloris	5796	0.2
DoS Slowhttptest	5499	0.19
Web Attack	2180	0.08
Bot	1966	0.07
Infiltration	36	0.01
Heartbleed	11	0.01
Total	2830743	100

Table 4 The frequency distribution of attack categories of the CIC-IDS2017 dataset

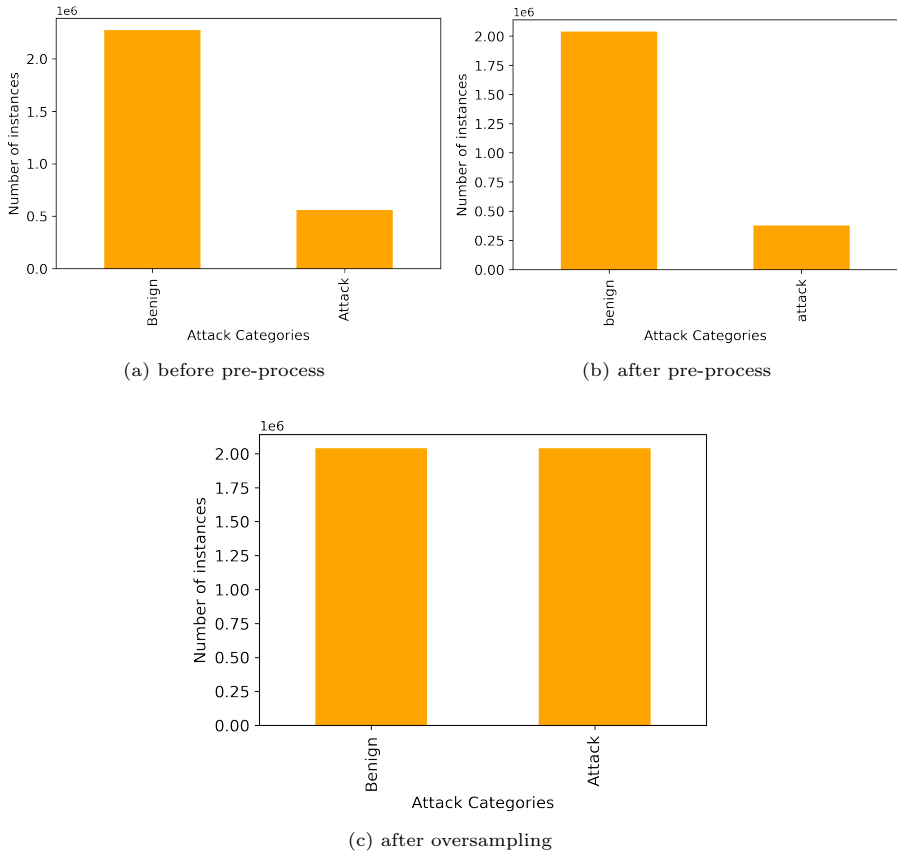


Fig. 4 Binary frequency distribution of CIC-IDS2017 dataset

practical implementation has been hindered by inherent complexities, demanding extensive testing and evaluation. Conventional datasets used for these purposes have shown limitations, stemming from privacy constraints, excessive anonymization, and a lack of representation of contemporary threat trends. This project seeks to overcome these limitations by introducing a structured approach for crafting benchmark datasets. This approach revolves around the creation of user profiles that offer abstract representations of network events and behaviors. These profiles are thoughtfully aggregated to construct datasets that exhibit distinctive features, encompassing a wide range of evaluation scenarios. The final dataset encompasses seven distinct attack scenarios, namely Brute-force, Heartbleed, Botnet, Denial of Service (DoS), Distributed Denial of Service (DDoS), Web attacks, and network infiltration. The attack infrastructure comprises 50 machines, while the target organization consists of 5 departments, incorporating 420 machines and 30 servers. The dataset includes meticulously collected network traffic and system logs from each machine, along with the extraction of 80 features through the application of

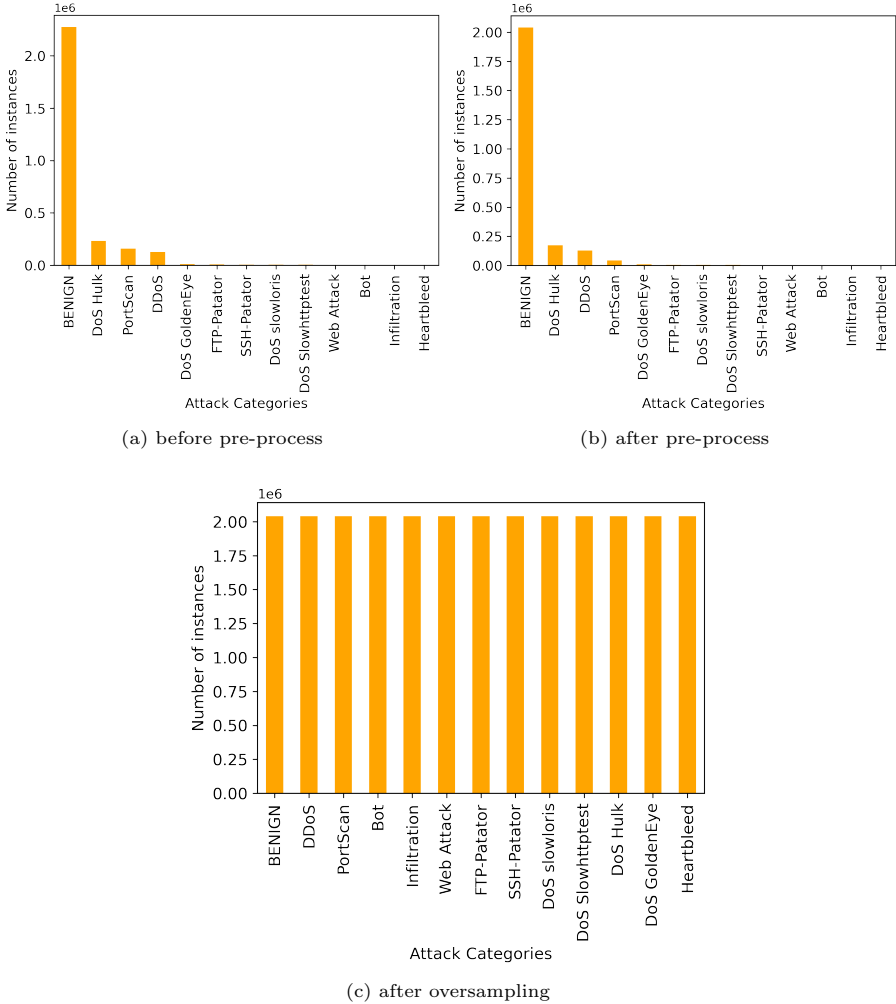


Fig. 5 multilabel frequency distribution of the CIC-IDS2017 dataset

CICFlowMeter-V3. This dataset constitutes an invaluable resource for the systematic evaluation of intrusion detection systems and offers a response to the growing demand for dynamic, adaptable, and comprehensive datasets within the domain of cybersecurity. It holds substantial promise for contributing to the advancement of intrusion detection research and its practical implementation in real-world security scenarios. In our research, we have sampled 10% of the dataset from each class to accommodate computational resource constraints. Our experimental dataset comprises 933,277 data points with 80 distinct features and encompasses 15 attack classes. The frequency distribution of attack categories for the CIC-IDS2018 dataset is detailed in Table 5.

Attack Categories	Count	(%) Percentage
Benign	658454	70.553
DDOS attack-HOIC	68601	7.351
DDoS attacks-LOIC-HTTP	57619	6.174
DoS attacks-Hulk	46191	4.949
Bot	28619	3.067
FTP-BruteForce	19336	2.072
SSH-Bruteforce	18759	2.01
Infiltration	16193	1.735
DoS attacks-SlowHTTPTest	13989	1.499
DoS attacks-GoldenEye	4151	0.445
DoS attacks-Slowloris	1099	0.118
DDOS attack-LOIC-UDP	173	0.019
Brute Force -Web	61	0.007
Brute Force -XSS	23	0.002
SQL Injection	9	0.001
Total	933277	100

Table 5 The frequency distribution of attack categories of the CIC-IDS2018 Dataset

The distribution of attack categories in the bar chart, both before preprocessing and after preprocessing, as well as after applying oversampling techniques, is illustrated in Figure 6.

3.2 Data preprocessing

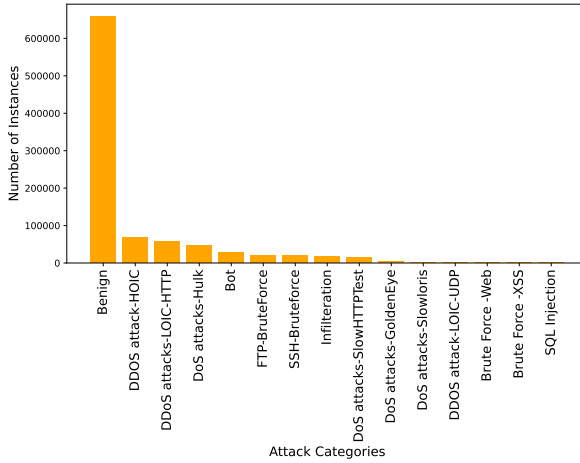
Data preprocessing is a crucial part of any ML model. Models without preprocessing can create problems with invalid, overfitting, generating error models, providing low accuracy and much more. So, preprocessing is a very significant part of an ML model. To analyze our model, we have used some preprocessing techniques such as: handling the missing value by eradicating rows containing null, -inf and inf values, removing space from columns names to work with columns smoothly and dropping the duplicate rows by keeping the first one and delete the rest from the dataset, merge the similar classes with low instance from output columns and finally, reduce the dataset size by converting data types from int64 to int32 and float64 to float32 to train models with less dataset size but same dataset entries.

3.3 Feature Scaling to normalize the features

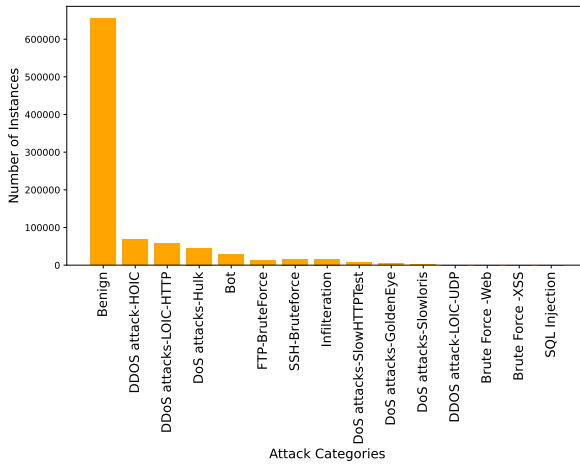
Feature scaling is a crucial preprocessing step aimed at normalizing the values of features within a consistent range. In our approach, we have employed both standardization and label encoding techniques to achieve this objective.

3.3.1 Standardization

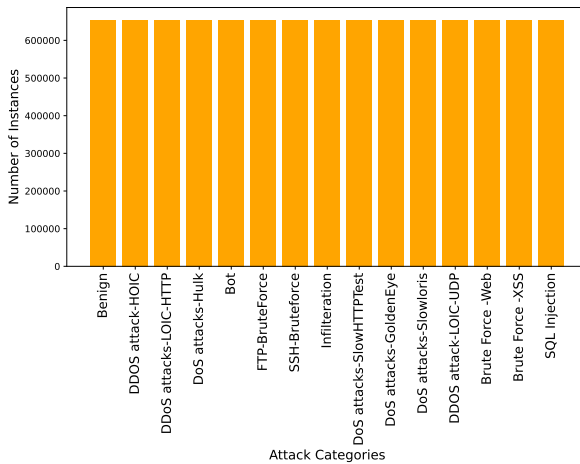
Standardization, also known as z-score normalization, is a pivotal method for feature scaling. It involves subtracting the mean from each feature's value and dividing it by the standard deviation. This technique is especially effective when there is a substantial disparity in feature values within the input data.



(a) before pre-process



(b) after pre-process



(c) after oversampling

Fig. 6 The frequency distribution of CIC-IDS2018 Dataset

Post-standardization, all features share a common scale, boasting a mean (μ) of zero and a standard deviation (σ) of one. This process significantly enhances the accuracy of our predictive models. Equation 1 presents the mathematical representation of the Z-score normalization.

$$x_{new} = \frac{x - \mu}{\sigma} \quad (1)$$

In this equation, x represents the original feature value, x_{new} signifies the standardized value, μ corresponds to the mean of the original feature, and σ denotes the standard deviation of the original feature.

3.3.2 Label Encoding

Label encoding is the practice of converting categorical data into numerical values, facilitating their utilization in machine learning algorithms. To train a machine learning model, we must transform categorical values into numerical representations to facilitate the model-building process during the training phase. This is achieved by replacing categorical values with integers ranging from 0 to (n-1), where 'n' represents the total number of unique classes. For instance, if there are 11 different categorical classes, we assign integers from 0 to 10 in place of these classes. Table 6 exemplifies the label encoding process.

Attack types	Label Encoding
Normal	0
Generic	1
Exploits	2
Fuzzers	3
DoS	4
Reconnaissance	5
Analysis	6
Backdoor	7
SSH-Patator	8
Shellcode	9
Worms	10

Table 6 Label encoding process

3.4 Feature Resampling using Random Oversampling (RO)

Feature resampling is a process to rebalance the feature from the imbalanced features in a dataset. The Random oversampling (RO) delivers a naive method to rebalance the class spreading for an imbalanced dataset. It performs an arbitrarily replicating instances from the minority group and incorporating them into the training part. For instance, if the ratio of the dataset's class is 20:80, then 20 belongs to the minority and 80 belongs to the majority class. It is efficient for skewed distribution algorithms and for a class that can stimulate to fit for the model by replicating instances. In this proposed framework we

considered big imbalanced datasets where RO is very crucial to balance the dataset for improving the performance without occurring overfitting problem. The RO process is depicted in Figure 7.

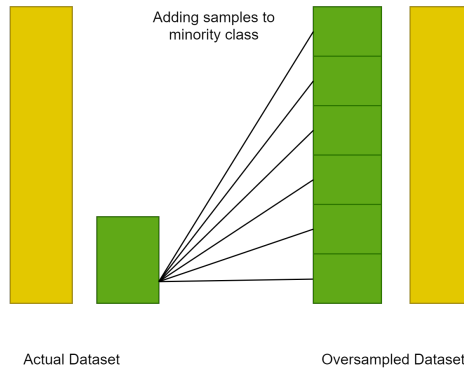


Fig. 7 Random oversampling process

3.5 Stacking Feature Embedded using Clustering with PCA

Within our experimental framework, we introduce a novel methodology known as Stacking Feature Embedded with PCA (SFE-PCA). This approach combines clustering and dimensionality reduction techniques to improve the performance of our ML models.

In the "Stacking Feature Embedded" phase, we first employ clustering methods, such as K-Means and Gaussian Mixture (GM) Clustering, to group data points based on their intrinsic patterns. The clustered results are then embedded as meta-dataset points into the original feature space. This augmentation adds a layer of complexity to our dataset, capturing finer details that might be missed by conventional approaches. Subsequently, Principal Component Analysis (PCA) is applied to this enriched feature set. PCA allows us to reduce the dimensionality while retaining the most informative features. This step ensures that we maintain a set of highly relevant and discriminative features, optimizing the input for our machine learning models.

The integration of clustering and PCA into the SFE-PCA approach aims to strike a balance between detailed feature representation and computational efficiency. By capturing essential information through clustering and refining it with PCA, we aim to empower our ML models with a more focused and effective feature set, ultimately contributing to improved performance and precision in our experimental results.

3.5.1 Stacking Feature Embedded using Clustering

The proposed SFE methodology serves as a fundamental component of our experimental framework. It is designed to address the challenges posed by big and imbalanced datasets, particularly in the context of machine learning-based network intrusion detection. This approach combines the strengths of clustering techniques and feature embedding to improve detection accuracy. The SFE process is illustrated in Figure 8. The following are the working principles of this approach:

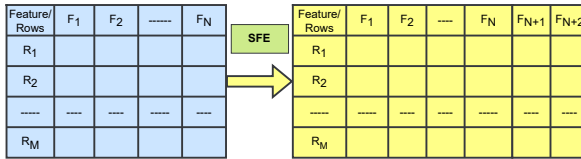


Fig. 8 SFE process

- **Cluster Formation:** The process begins with the application of two clustering methods: K-Means and Gaussian Mixture Clustering. These techniques group data points into coherent clusters based on shared characteristics and patterns. Clustering reveals the underlying structure in the data, enabling a more comprehensive understanding.
- **Feature Embedding:** The output generated by the clustering phase is then embedded into the original feature space. This integration creates a set of additional features, often referred to as meta-dataset points. These new features capture nuanced information that enriches the overall dataset.
- **Enhanced Data Representation:** The dataset now includes the original features alongside the newly embedded meta-dataset points. This augmented representation offers a more comprehensive view of the data, enabling the detection of subtle anomalies and patterns.

The approach was adopted to address the limitations of traditional intrusion detection methods when dealing with big and imbalanced data. By integrating clustering techniques and feature embedding, our objectives encompass several key aspects. Firstly, we seek to enrich the reliability and accuracy of intrusion detection, providing a more robust defense against cyber threats. Additionally, our approach enables the capture of fine-grained details within network traffic data, improving our ability to discern subtle anomalies. Moreover, it facilitates the detection of previously undetected threats, contributing to a more comprehensive security posture. Lastly, by incorporating these techniques, we seek to improve the overall performance and precision of our ML models, making them more effective in safeguarding network environments.

This approach represents a crucial advancement in the field, promising to contribute to the development of more robust and effective intrusion detection systems for real-world network security challenges.

3.5.2 Feature Extraction using PCA

The curse of a high dimensional dataset makes a model more complex and leads to overfitting that fallout an ill performance. It is essential to reduce dimension for getting reduced dataset, less computation time, quickly visualize the data and remove redundant features from the dataset.

Feature reduction in a dataset involves the generation of new features from existing ones, with the aim of preserving the essential information present in the original features. PCA is a statistical technique that employs an orthogonal transformation to convert a set of correlated variables into a set of uncorrelated variables. In both exploratory data analysis and the development of predictive machine learning models, PCA stands as a fundamental and widely employed tool. Additionally, It serves as an invaluable unsupervised statistical method for exploring the relationships between a set of variables. It differs from regression in that it seeks to create a line of best fit, which is often referred to as a form of generic factor analysis. To reduce the dimension of the features from n to k , the following steps should be preceded:

1. Equation 2 is used to equalize the data's initial attribute values by the mean and variance.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (2)$$

Here n represents the instances number and x_i represents the data points.

2. Substitute x_i by $x_i - \mu$
3. Each vector $x_{j(i)}$ should be rescaled to have unit variance.

$$\sigma_j^2 = \frac{1}{n} \sum_{i=1}^n (x_{j(i)})^2 \quad (3)$$

4. Substitute $x_{j(i)}$ by $\frac{x_{j(i)}}{\sigma}$
5. The Covariance Matrix C_M should be calculated as follows:

$$C_M = \frac{1}{n} \sum_{i=1}^n x_i \cdot (x_i)^T \quad (4)$$

6. Determine Eigen-vectors and their related Eigen-values of C_M .
7. To generate w , sort the Eigen-vectors by decreasing their Eigen-values and choose k Eigen-vectors with the largest Eigen-values.
8. Equation 5 is used to convert the data onto the new subspace using w .

$$y = w^T * x \quad (5)$$

where x represents one sample as a $d \times 1$ dimensional vector and y represents the converted $k \times 1$ dimensional vector in the resulting subspace.

The number of features D_p that each data point represents determines the computational complexity of running the designed PCA (Zou et al, 2006).

$$O(D_p^3) \quad (6)$$

The reduction ratio (RR) is the number of output dimensions divided by the number of input dimensions. (Vasan and Surendiran, 2016). The efficiency of PCA is inverse to RR. The lower the RR value, the higher the PCA's efficiency.

In our proposed framework we adopted PCA to reduce the dimension of our datasets to get better performance with less number of features than the original. The reduced features contain the most important information of the datasets to produce the better performance to detect intrusion efficiently. During our proposed work, we considered the RR is 10:45 or 22.22% for UNSW-NB15 and 10:79 or 12.65% for the CIC-IDS2017 dataset, which is used to provide higher accuracy with a lower false rate. Several existing works took 13-22 or 28.88%-48.88% PCA for UNSW-NB15 (Kumar et al, 2020; Kasongo and Sun, 2020a) and 22-52 or 27%-65.82% PCA for CIC-IDS2017 datasets (Al-Janabi and Ismail, 2021; Stiawan et al, 2020). In this study we considered 10 PCA for both datasets to check the performance at these lower RR to prove the efficiency of our approach. The PCA process is depicted in Figure 9.

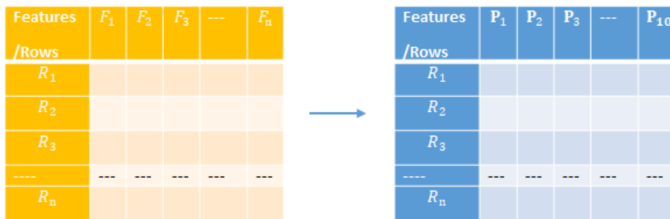


Fig. 9 Dimension reduction process using PCA

3.6 ML Algorithms

In this section, we have leveraged supervised ML algorithms to assess our performance in binary and multilabel classification tasks using our datasets. The detection of intrusion in the context of network security using ML is outlined as follows:

3.6.1 Decision Tree (DT)

A non-parametric supervised ML technique called the DT is used to solve problems with regression and classification. The prediction of the value of the output of a dataset is generated gripping decision rules from dataset features. It's easy to comprehend and interpret and it can be visualized. It can handle multi-output problems (Ahmim et al, 2019). It is widely used in IDS. Decision node, having multiple branches and confirmed to make the decision and Leaf nodes, not contain any branches and the output of those decisions are the two nodes in DT. The starting decision node is called the root node. To build a decision tree, Attribute Selection Measure (ASM) is performed on information

gain and the Gini index to select the feature (Sharmin et al, 2023). The change of entropy based on a feature after splitting is called IG. Based on the value of IG, we have separated the node and constructed the decision tree based on the value of IG. The measure of purity or impurity creating a DT is called GI. To create binary splits, GI is used. The lower GI should be preferred as compared to the higher GI. Pruning is the practice of deleting nodes from a tree that is no longer needed to achieve the best decision tree possible, which is accomplished through Cost Complexity Pruning and Reduced Error Pruning. Equ. 7 and 8 show GI and IG respectively.

$$Gini(D) = 1 - \sum_{i=1}^n (p_i)^2 \quad (7)$$

$$Gain(A) = Entropy(D) - Entropy_A(D) \quad (8)$$

where,

$$Entropy(D) = - \sum_{i=1}^n p_i \log_2(p_i) \quad (9)$$

$$Entropy_A(D) = \sum_{i=1}^n \frac{D_i}{D} \times Entropy(D_i) \quad (10)$$

and the probability of a data point in the subset of D_i of a dataset D is denoted by (P_i) .

3.6.2 Random Forest (RF)

Random Forest (RF) is a renowned supervised ML technique rooted in the concept of ensemble learning which involves the amalgamation of multiple classifiers to tackle complex problems and enhance the overall performance of the model. RF serves as a meta predictor that leverages averaging to enhance predictive accuracy, all the while mitigating overfitting concerns by adapting various decision tree classifiers to diverse subsets of the dataset. A bootstrap randomized resampling method creates each decision tree (Breiman, 2001). It requires the least amount of training time compared to other algorithms and estimates output with high accuracy; it also operates efficiently on large datasets. It improves the model's accuracy and eliminates the problem of overfitting. The algorithm gathers the prediction results from each tree, sets up a voting mechanism and then performs a majority vote among the classifiers to make a classification decision. It builds a forecast using the results of many decision trees, which improves prediction accuracy (Uddin et al, 2023).

3.6.3 Extra Tree (ET)

An ensemble ML technique and meta-estimator, Extra-Tree is also called Extremely Randomized Trees. In order to increase the model's prediction accuracy, it applies a series of randomized decision trees, referred to as extra-trees, to various sub-samples of datasets and averages them. This prevents overfitting. It's an ensemble model, just like bagging and random forest in an

ensemble decision tree. From the training datasets, it creates a huge number of unpruned decision trees in order to function. For regression and classification, respectively, the majority vote and average are used to predict the decision tree. It builds decision trees using the entire learning sample, and divides the nodes by randomly choosing all of the cut-points (Geurts and Ernst, 2006).

3.6.4 Extreme Gradient Boosting (XGB)

A supervised ML method that uses gradient-boosted decision trees to improve speed and performance. XGB has remarkable speed as compared to other gradient boosting implementations (Chen and He, 2015). The combination of residuals from earlier models, which new models form—leads to its ultimate forecasts. This method employs a gradient descent to lessen loss and improve model performance. When looking for cutting-edge solutions for various ML problems, data scientists have come to appreciate it, as a scalable end-to-end tree-boosting technique. (Chen and Guestrin, 2016). Within XGB, the objective functions are composed of two key components: the training loss and regularization, with θ representing the optimal settings for the training data x_i and the associated labels y_i . Equation 11 illustrates the objective functions employed in XGBoost.

$$O(\theta) = L(\theta) + \Omega(\theta) \quad (11)$$

In this context, L represents the training loss function, which is a metric for evaluating the model's performance in predicting the training datasets.

For instance, a straightforward example of a training loss function that represents Mean Squared Error (MSE):

$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2 \quad (12)$$

The logistic loss function is a frequently used loss function in logistic regression:

$$L(\theta) = \sum_i [y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i) \ln(1 + e^{\hat{y}_i})] \quad (13)$$

Ω is the regularization that controls the complexity of the model, which helps us to avoid overfitting which is given by

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T \|w\|^2 \quad (14)$$

In this case, γ denotes encouraging pruning, T denotes the number of terminal nodes, w denotes the leaf weights, and λ is expected to lower the outcome's sensitivity (Chen and He, 2015).

	Actual positive	Actual negative
Predicted positive	TP	FP
Predicted negative	FN	TN

Table 7 Confusion Matrix

4 Experimental setup and Evaluations

In this section, we have first covered the environment setup and performance evaluation measures in this section. Then, we have included descriptions for the CIC-IDS2017, CIC-IDS2018, and UNSW-NB15 benchmark datasets. We have employed four classification methods for our experiments: DT, RF, ET, and XGB. Data on binary and multilabel classification are used to examine the performance.

4.1 Environment setup

The experiments are conducted in a robust computing environment, utilizing a high-performance 2X-large virtual machine instance. This instance boasts 8 cores, allowing for efficient concurrent task handling and enhanced multi-threading capabilities. With 64 GB of RAM, the system is well-equipped to accommodate memory-intensive applications, and it offers a generous 40 GB of disk space for data storage. The experiments are seamlessly executed using the Jupyter notebook through Anaconda Navigator. To support our performance evaluation, we leverage the Python programming language and a suite of indispensable libraries, including TensorFlow, Keras, Pandas, Scikit-learn, NumPy, Seaborn, Matplotlib, Imbalanced-learn etc.

4.2 Performance Evaluation Metrics

Several measures, such as accuracy, precision, recall, F1-score, ROC curve, and confusion matrix, are used to evaluate the performance of our proposed model. The following defines these performance matrices:

4.2.1 Confusion Matrix

The Confusion Matrix is a valuable tool for evaluating ML classification performance. It is a tabular representation containing four combinations of predicted and actual values: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) (Talukder et al, 2023b). Table 7 illustrates a confusion matrix where TP represents correctly anticipated positive values, TN indicates accurately projected negative values, FP corresponds to incorrectly forecasted positive values, and FN signifies inaccurately predicted negative values. This matrix is essential for assessing Recall, F1-score, Accuracy and Precision.

4.2.2 Accuracy

Accuracy is a fundamental performance metric, representing the proportion of correctly predicted observations to the total observations. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (15)$$

4.2.3 Precision

Precision quantifies the ratio of correctly predicted positive values to the total number of predicted positive values:

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

4.2.4 Recall

Recall measures the ratio of correctly predicted positive values to all actual positive values:

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

4.2.5 F1-Score

The F1-score represents the harmonic mean of precision and recall for classification problems:

$$F1 - Score = 2 \cdot \frac{(Precision \cdot Recall)}{(Precision + Recall)} \quad (18)$$

4.2.6 ROC Curve

ROC curves are commonly employed two-dimensional plots for assessing the significance of classifiers (?). These graphs provide a clear visualization of how a classifier's sensitivity and specificity trade-off at various classification thresholds. This feature is valuable for selecting classifiers that align with specific user requirements, often associated with variable error costs and accuracy expectations, as noted in studies by (Sameera et al, 2016; Vergara et al, 2008). The Area Under the Curve (AUC) represents the degree of discrimination in the ROC curve, while the ROC curve itself is a probability curve that assesses the model's ability to distinguish between different categories. The true positive rate is plotted on the Y-axis, and the false positive rate is on the X-axis. An AUC value approaching 1 suggests that the model excels at distinguishing between class labels, while an AUC value approaching 0 indicates poor predictive performance, implying that the results mirror randomness. This method serves as a means to visualize the classification's efficiency, as emphasized by (Gorunescu, 2011). In essence, classifiers with higher ROC curves are considered superior, as supported by (Yulianto et al, 2019).

4.3 K-fold Cross-Validation (CV)

K-fold CV is a standard method that partitions the training set into k smaller sets or folds. The model is trained on each fold, and it is tested on the remaining data after that. Using k-fold CV, the performance measure is obtained as the average of these values. We use k-fold CV in our studies, where k is 10 and the dataset is split into 90% training and 10% testing for each fold. Figure 10 illustrates the k-fold CV process.

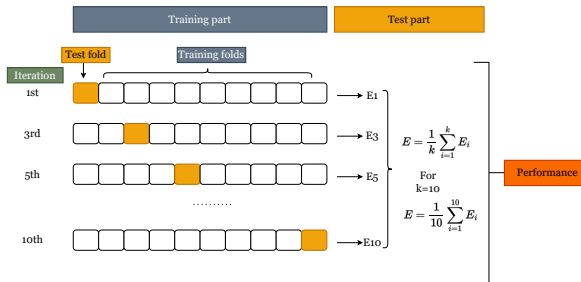


Fig. 10 K-fold cross-validation process

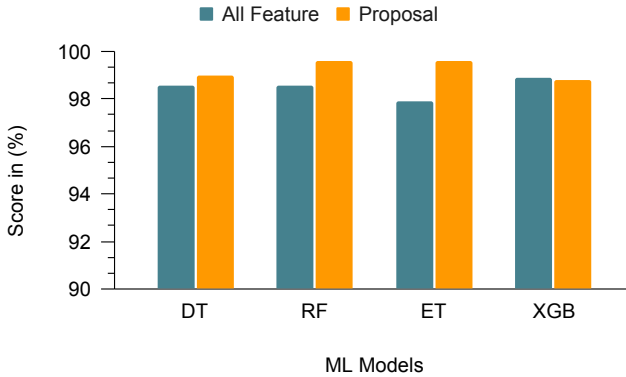
5 Result Analysis

In our analysis of the results, we evaluated the performance of four ML models for Intrusion IDS such as DT, RF, ET and XGB. Our focus was on assessing key performance metrics by considering "All Features" and a novel "Proposal" feature set in our evaluation.

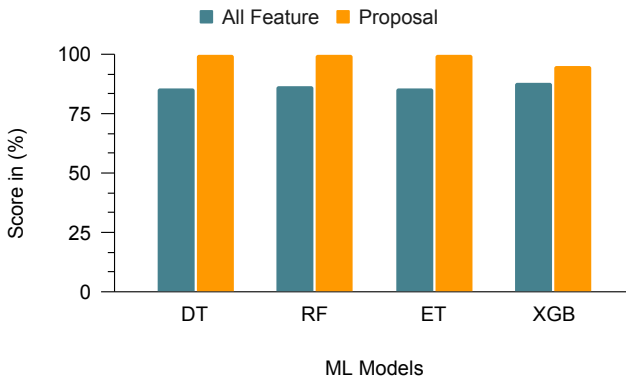
5.1 Results of UNSW-NB15 Dataset

The performance results of binary and multilabel classification on the UNSW-NB15 dataset are presented in Figure 11, as well as in Table 8 and Table 9. These figures and tables showcase the experimental results for two distinct scenarios: **All Features**: In this case, "All Features" represent the dataset where we did not oversample any features. We preprocessed, scaled, and applied these features directly to the machine learning models for training and performance evaluation. **Proposal Features**: Here, the "Proposal" refers to a methodology that encompasses various preprocessing steps that have been undertaken for evaluation. These results provide a comprehensive view of our model's performance and the impact of feature selection and preprocessing on IDS using the UNSW-NB15 dataset.

In the bar chart, it is evident that our proposed model exhibits a noteworthy increase in accuracy rates for binary and multilabel classification. Notably, the rate of accuracy improvement is more pronounced in the context of multilabel classification when compared to binary classification



(a) Binary



(b) multilabel

Fig. 11 Binary and multilabel Accuracy performance Bar Chart for All Features and Proposal Features for UNSW-NB15 Dataset

ML	Accuracy		Precision		Recall		F1-score	
	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal
DT	98.56	98.97	98.42	98.98	98.44	98.97	98.44	98.97
RF	98.55	99.59	98.46	99.59	98.43	99.59	98.43	99.59
ET	97.9	99.59	97.71	99.59	97.73	99.59	97.73	99.59
XGB	98.9	98.81	98.71	98.81	98.82	98.81	98.82	98.81

Table 8 Performance Analysis of Binary Classification for UNSW-NB15 Dataset

The binary result analysis, as presented in Table 8, showcases the performance evaluation of various ML algorithms using both the "All Feature" and "Proposed" feature sets. Notably, among these algorithms, DT consistently emerges as the top performer in terms of accuracy, precision, recall, and F1-score. With the "Proposed" feature set, RF demonstrates outstanding performance, achieving an impressive accuracy of 99.59%. This places RF at the forefront, surpassing other algorithms, including DT with an accuracy of

ML	Accuracy		Precision		Recall		F1-score	
	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal
DT	85.38	99.79	60.24	99.79	60.63	99.79	60.63	99.79
RF	86.42	99.95	62.45	99.95	58.06	99.95	58.06	99.95
ET	85.55	99.95	59.49	99.95	56.15	99.95	56.15	99.95
XGB	87.73	95.04	76.92	95.29	66.59	95.03	66.59	95.03

Table 9 Performance metrics of Multilabel Classification for UNSW-NB15 Dataset

98.97%, XGB with 98.81%, and ET with 99.59%. This remarkable increase in accuracy is consistently reflected in other metrics. The success of RF can be attributed to its ensemble learning approach, which harnesses the strengths of multiple decision trees to create a robust and highly accurate model.

The multilabel performance analysis, as shown in Table 9, evaluates the performance of various ML algorithms on multiclass using both the "All Feature" and "Proposal" feature sets. Among these algorithms, RF consistently emerges as the top performer in terms of accuracy, precision, recall, and F1-score. Notably, with the "Proposal" feature set, RF and ET achieve an impressive 99.95% accuracy, surpassing all other algorithms, including DT with an accuracy of 98.97% and XGB with 95.04%. This substantial accuracy enhancement extends to precision, recall, and F1-score metrics, underscoring RF's and ET's success attributed to their ensemble learning approach, which leverages the strengths of multiple decision trees to create a robust model.

The binary confusion matrix is displayed in Figure 12. A successful predictive model is characterized by a low number of Type 1 (FP) and Type 2 (FN) errors in the confusion matrix. For RF, the TP rate stands impressively at 49.71%, while the TN rate is equally strong at 49.88%. Additionally, the FP and FN rates are remarkably low, at 0.15% and 0.26%, respectively. These findings highlight RF's robust performance in accurately identifying positive cases (intrusions) and negative cases (non-intrusions), making it a compelling choice for intrusion detection. For ET, the TP rate is an outstanding 49.77%, and the TN rate is equally impressive at 49.82%. Furthermore, the FP and FN rates are notably low, at 0.08% and 0.32%, respectively. These results underscore the exceptional performance of ET in accurately identifying both positive cases (intrusions) and negative cases (non-intrusions), making it a compelling choice for intrusion detection.

Among all the evaluated models, it is evident that both RF and ET outperform the others, showcasing superior performance in terms of TP and TN rates for IDS. They consistently deliver higher TP and TN rates, demonstrating their effectiveness in accurately identifying intrusions while maintaining a low rate of false positives and false negatives.

The ROC Curve is depicted in Figure 14 for binary and multilabel classification. The ROC Curves clearly illustrate that the AUC (Area Under the Curve) values are approaching 1, indicating a strong predictive model's ability to distinguish between classes. In the binary classification scenario, the AUC scores are 98.97% for DT, 99.98% for RF, 99.98% for ET, and 99.93% for XGB, with RF and ET algorithms leading in AUC score. In the multiclass classification, the AUC scores are 99.88% for DT, 100% for RF, 100% for ET,

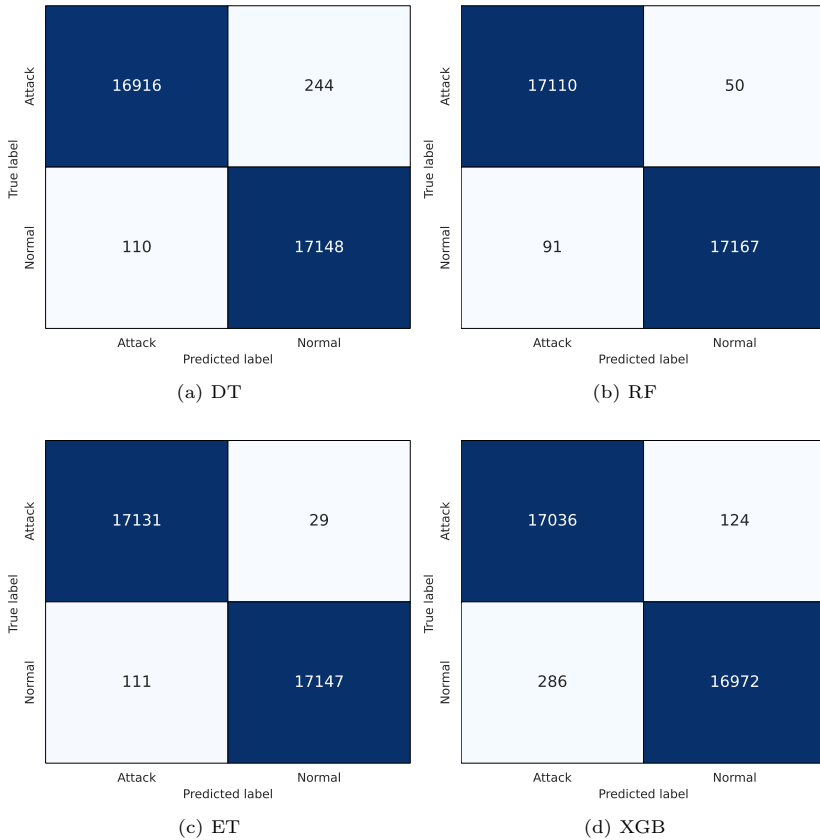


Fig. 12 Binary Confusion Matrix for UNSW-NB15 Dataset

and 99.83% for XGB, with RF and ET again displaying superior AUC scores compared to the others. These high AUC scores, close to 1, indicate the strong predictive performance of the models on the UNSW-NB15 dataset, further validating their effectiveness.

5.2 Results of CIC-IDS2017 Dataset

The performance results for binary and multilabel classification on the CIC-IDS2017 dataset are presented in Figure 15, along with detailed metrics in Table 10 and Table 11. These figures and tables display the experimental results for two distinct scenarios: **All Features**: In this case, "All Features" represent the dataset where features are neither oversampled nor modified extensively. These features are preprocessed, scaled, and then used to train and evaluate the ML models. **Proposal Features**: The "Proposal" indicates a set of features that have undergone various preprocessing steps and modifications as part of our proposed methodology for evaluation. The inclusion of both

Normal	10462	0	0	0	0	0	0	0	0	0
Generic	19	10485	1	3	77	0	6	0	0	0
Exploits	0	0	10502	0	0	0	9	0	37	0
Fuzzers	0	0	0	10862	0	0	0	0	0	0
DoS	0	12	0	0	10483	0	0	0	0	0
Reconnaissance	0	0	0	0	0	10514	0	0	0	62
Analysis	0	0	0	0	0	0	10585	0	0	0
Backdoor	0	0	0	0	0	0	0	10657	0	0
Shellcode	0	0	0	0	0	0	0	0	10586	0
Worms	0	0	0	0	0	0	0	0	0	10462
	Normal	Generic	Exploits	Fuzzers	DoS	Reconnaissance	Analysis	Backdoor	Shellcode	Worms

(a) DT

Normal	10462	0	0	0	0	0	0	0	0	0
Generic	4	10580	0	0	7	0	0	0	0	0
Exploits	0	0	10541	0	0	0	1	0	6	0
Fuzzers	0	0	0	10862	0	0	0	0	0	0
DoS	0	12	0	0	10483	0	0	0	0	0
Reconnaissance	0	0	0	0	0	10552	0	0	0	24
Analysis	0	0	0	0	0	0	10585	0	0	0
Backdoor	0	0	0	0	0	0	0	10657	0	0
Shellcode	0	0	0	0	0	0	0	0	10586	0
Worms	0	0	0	0	0	0	0	0	0	10462
	Normal	Generic	Exploits	Fuzzers	DoS	Reconnaissance	Analysis	Backdoor	Shellcode	Worms

(b) RF

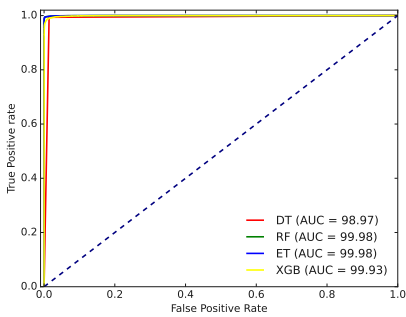
Normal	10462	0	0	0	0	0	0	0	0	0
Generic	4	10583	0	0	4	0	0	0	0	0
Exploits	0	0	10538	0	0	0	4	0	6	0
Fuzzers	0	0	0	10862	0	0	0	0	0	0
DoS	0	12	0	0	10483	0	0	0	0	0
Reconnaissance	0	0	0	0	0	10557	0	0	0	19
Analysis	0	0	0	0	0	0	10585	0	0	0
Backdoor	0	0	0	0	0	0	0	10657	0	0
Shellcode	0	0	0	0	0	0	0	0	10586	0
Worms	0	0	0	0	0	0	0	0	0	10462
	Normal	Generic	Exploits	Fuzzers	DoS	Reconnaissance	Analysis	Backdoor	Shellcode	Worms

(c) ET

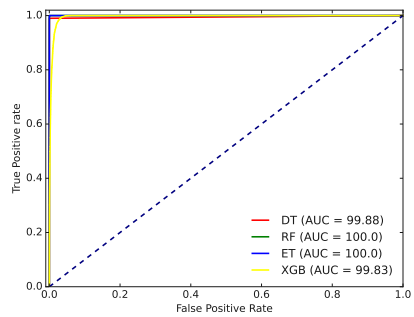
Normal	8830	840	0	153	639	0	0	0	0	0
Generic	125	10278	0	23	163	0	2	0	0	0
Exploits	2	2	10196	0	2	0	30	0	316	0
Fuzzers	18	42	0	10754	31	0	13	0	4	0
DoS	115	1202	0	9	9163	0	3	2	1	0
Reconnaissance	0	0	0	0	0	10469	0	0	0	107
Analysis	6	39	178	123	39	0	10158	0	42	0
Backdoor	0	0	0	0	0	0	0	10657	0	0
Shellcode	0	0	799	0	0	0	0	0	9787	0
Worms	0	0	0	0	0	183	0	0	0	10279
	Normal	Generic	Exploits	Fuzzers	DoS	Reconnaissance	Analysis	Backdoor	Shellcode	Worms

(d) XGB

Fig. 13 Multilabel Confusion Matrix for UNSW-NB15 Dataset



(a) Binary



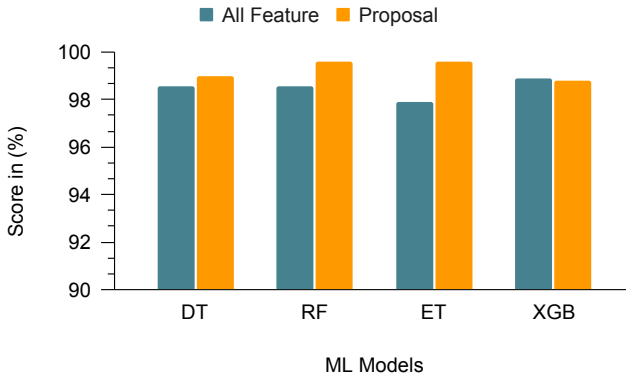
(b) multilabel

Fig. 14 Binary and Multilabel ROC Curve for UNSW-NB15 Dataset

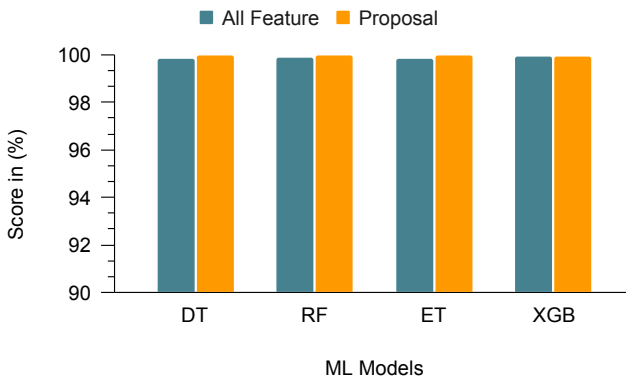
ML	Accuracy		Precision		Recall		F1-score	
	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal
DT	99.87	99.91	99.78	99.91	99.78	99.91	99.78	99.91
RF	99.9	99.94	99.82	99.94	99.82	99.94	99.82	99.94
ET	99.83	99.95	99.73	99.95	99.7	99.95	99.7	99.95
XGB	99.92	99.65	99.83	99.65	99.86	99.65	99.86	99.65

Table 10 Performance metrics for Binary Classification for CIC-IDS2017 Dataset

scenarios allows for a comprehensive assessment of model performance on the CIC-IDS2017 dataset



(a) Binary



(b) multilabel

Fig. 15 Binary and Multilabel Accuracy Performance Bar Chart for All Features and Proposal Features for the CIC-IDS2017 Dataset

In the bar chart, it's evident that our proposed model exhibits a substantial increase in accuracy for binary and multilabel classification. Interestingly, the rate of accuracy improvement is notably higher in multilabel classification when compared to binary classification.

ML	Accuracy		Precision		Recall		F1-score	
	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal
DT	99.85	99.99	98.69	99.99	94.69	99.99	94.69	99.99
RF	99.89	99.99	98.98	99.99	94.17	99.99	94.17	99.99
ET	99.83	99.99	98.57	99.99	94.14	99.99	94.14	99.99
XGB	99.92	99.94	99.3	99.94	94.47	99.94	94.47	99.94

Table 11 Performance Analysis of Multilabel Classification for CIC-IDS2017 Dataset

In the binary classification, the accuracy rates on the proposed model are as follows: 99.91% for DT, 99.94% for RF, 99.95% for ET, and 99.65% for XGB.

The accuracy rates for multilabel classification are 99.91% for DT, 99.94% for RF, 99.95% for ET, and 99.65% for XGB on the proposed model. The confusion matrix is displayed in Figure 16 for binary classification and Figure 17 for multilabel classification. Upon examining the confusion matrix results for binary classification, it's observed that RF, ET, and XGB provide similar accuracy rates, with slight variations in their TP, FP, and FN rates.

In the binary confusion matrix, it's insightful to note that the TP rates are 50.07% for DT, RF, ET, and 49.91% for XGB. The TN rates are 49.84% for DT, 49.87% for RF, 49.88% for ET, and 49.74% for XGB. Additionally, the FP rates are 0.0% for DT, RF, ET, and 0.17% for XGB, while the FN rates are 0.08% for DT, 0.06% for RF, 0.05% for ET, and 0.18% for XGB. RF employs bootstrap repetitions and selects the best-split method, making it an effective ensemble of independent decision trees working together. ET, on the other hand, uses the entire original sample and selects the splitting operation at random, resulting in an ensemble of extra trees. This diversity in learning methods contributes to achieving the highest accuracy. For multilabel classification, DT, RF and ET exhibit similar and superior accuracy with minor variances in TP, FP, and FN rates. Notably, XGB demonstrates lower accuracy compared to the other algorithms in binary and multilabel classification scenarios.

The ROC Curve, as illustrated in Figure 18, provides a comprehensive view of the model's performance in binary and multilabel classification. The ROC Curves showcase that the Area Under the Curve (AUC) values are approaching the ideal value of 1, indicating a highly effective predictive model for distinguishing between classes. In the binary classification, the XGBoost (XGB) algorithm stands out with the highest AUC score. Specifically, the AUC scores are as follows: 99.92% for Decision Trees (DT), 99.98% for Random Forest (RF), 99.97% for Extra Trees (ET), and an impressive 99.99% for XGB. In multiclass classification, RF, ET, and XGB collectively achieve the highest AUC scores. Specifically, the AUC scores are 99.99% for DT, a perfect 100% for RF, ET, and XGB, emphasizing the exceptional predictive performance of these algorithms. These consistently high AUC scores, close to 1, underscore the strong predictive capabilities of the model on the CIC-IDS2017 dataset, indicating its suitability for the task.

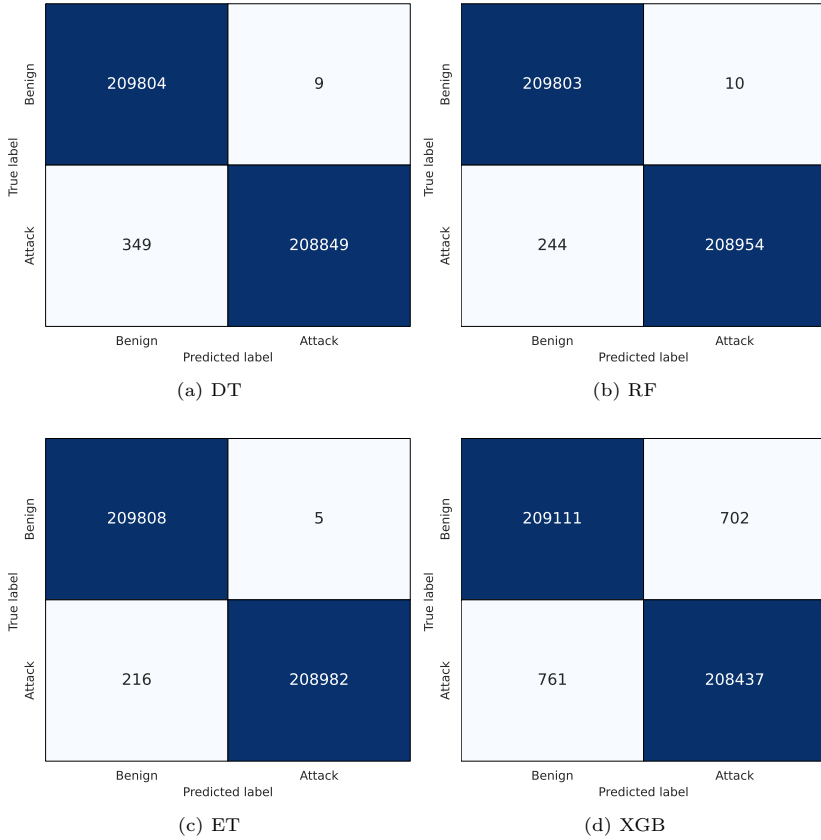


Fig. 16 Binary Confusion Matrix for CIC-IDS2017 Dataset

5.3 Results of CIC-IDS2018 Dataset

The performance results of the CIC-IDS2018 dataset are presented in Figure 19, as well as in Table 12. These figure and table showcase the experimental results for two distinct scenarios: **All Features:** In this case, "All Features" represent the dataset where we did not oversample any features. We preprocessed, scaled, and applied these features directly to the machine learning models for training and performance evaluation. **Proposal Features:** Here, the "Proposal" refers to a methodology that encompasses various preprocessing steps that have been undertaken for evaluation. These results provide a comprehensive view of our model's performance and the impact of feature selection and preprocessing on IDS using the CIC-IDS2018 dataset.

In the bar chart, it is evident that our proposed model exhibits a noteworthy increase in accuracy rates for attack classification. Notably, the rate of accuracy improvement is more pronounced in the context of the proposed feature when compared to all feature.

True label \ Predicted label	Benign	DoS Hulk	DDoS	PortScan	DoS GoldenEye	FTP-Ratator	DoS Slowloris	DoS Slowhttpstest	SSH-Ratator	Web Attack	Bot	Infiltration	Heartbleed
Benign	209523	59	13	25	60	6	11	1	0	1	99	3	6
DoS Hulk	0	208551	0	0	0	0	0	0	0	0	0	0	0
DDoS	0	0	210062	0	0	0	0	0	0	0	0	0	0
PortScan	0	0	0	209294	0	0	0	0	0	0	0	0	0
DoS GoldenEye	0	0	0	0	208817	0	0	0	0	0	0	0	0
FTP-Ratator	0	0	0	0	0	209301	0	0	0	0	0	0	0
DoS slowloris	0	0	0	0	0	0	209886	0	0	0	0	0	0
DoS Slowhttpstest	0	0	0	0	0	0	0	209621	0	0	0	0	0
SSH-Ratator	0	0	0	0	0	0	0	0	209155	0	0	0	0
Web Attack	0	0	0	0	0	0	0	0	0	209689	0	0	0
Bot	0	0	0	0	0	0	0	0	0	0	210359	0	0
Infiltration	0	0	0	0	0	0	0	0	0	0	0	209603	0
Heartbleed	0	0	0	0	0	0	0	0	0	0	0	0	209429

(a) DT

True label \ Predicted label	Benign	DoS Hulk	DDoS	PortScan	DoS GoldenEye	FTP-Ratator	DoS Slowloris	DoS Slowhttpstest	SSH-Ratator	Web Attack	Bot	Infiltration	Heartbleed
Benign	209523	57	12	7	45	4	2	0	0	0	97	6	2
DoS Hulk	0	208551	0	0	0	0	0	0	0	0	0	0	0
DDoS	0	0	210062	0	0	0	0	0	0	0	0	0	0
PortScan	0	0	0	209294	0	0	0	0	0	0	0	0	0
DoS GoldenEye	0	0	0	0	208817	0	0	0	0	0	0	0	0
FTP-Ratator	0	0	0	0	0	209301	0	0	0	0	0	0	0
DoS slowloris	0	0	0	0	0	0	209886	0	0	0	0	0	0
DoS Slowhttpstest	0	0	0	0	0	0	0	209621	0	0	0	0	0
SSH-Ratator	0	0	0	0	0	0	0	0	209155	0	0	0	0
Web Attack	0	0	0	0	0	0	0	0	0	209689	0	0	0
Bot	0	0	0	0	0	0	0	0	0	0	210359	0	0
Infiltration	0	0	0	0	0	0	0	0	0	0	0	209603	0
Heartbleed	0	0	0	0	0	0	0	0	0	0	0	0	209429

(b) RF

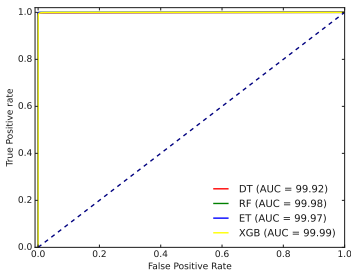
True label \ Predicted label	Benign	DoS Hulk	DDoS	PortScan	DoS GoldenEye	FTP-Ratator	DoS Slowloris	DoS Slowhttpstest	SSH-Ratator	Web Attack	Bot	Infiltration	Heartbleed
Benign	209601	40	14	3	44	5	3	0	0	0	93	3	1
DoS Hulk	0	208551	0	0	0	0	0	0	0	0	0	0	0
DDoS	0	0	210062	0	0	0	0	0	0	0	0	0	0
PortScan	0	0	0	209294	0	0	0	0	0	0	0	0	0
DoS GoldenEye	0	0	0	0	208817	0	0	0	0	0	0	0	0
FTP-Ratator	0	0	0	0	0	209301	0	0	0	0	0	0	0
DoS slowloris	0	0	0	0	0	0	209886	0	0	0	0	0	0
DoS Slowhttpstest	0	0	0	0	0	0	0	209621	0	0	0	0	0
SSH-Ratator	0	0	0	0	0	0	0	0	209155	0	0	0	0
Web Attack	0	0	0	0	0	0	0	0	0	209689	0	0	0
Bot	0	0	0	0	0	0	0	0	0	0	210359	0	0
Infiltration	0	0	0	0	0	0	0	0	0	0	0	209603	0
Heartbleed	0	0	0	0	0	0	0	0	0	0	0	0	209429

(c) ET

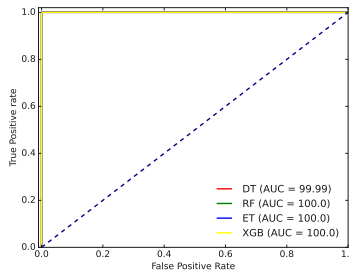
True label \ Predicted label	Benign	DoS Hulk	DDoS	PortScan	DoS GoldenEye	FTP-Ratator	DoS Slowloris	DoS Slowhttpstest	SSH-Ratator	Web Attack	Bot	Infiltration	Heartbleed
Benign	208370	763	36	94	238	16	14	2	0	1	148	69	56
DoS Hulk	0	208551	0	0	0	0	0	0	0	0	0	0	0
DDoS	4	0	210058	0	0	0	0	0	0	0	0	0	0
PortScan	0	0	0	209294	0	0	0	0	0	0	0	0	0
DoS GoldenEye	48	0	0	43	208705	0	0	0	0	17	0	4	0
FTP-Ratator	0	0	0	0	0	209301	0	0	0	0	0	0	0
DoS slowloris	0	0	0	0	0	0	209886	0	0	0	0	0	0
DoS Slowhttpstest	0	0	0	0	0	0	0	209621	0	0	0	0	0
SSH-Ratator	0	0	0	0	0	0	0	0	209155	0	0	0	0
Web Attack	0	0	0	0	0	0	0	0	0	209689	0	0	0
Bot	0	0	0	0	0	0	0	0	0	0	210323	10	8
Infiltration	0	0	0	0	0	0	0	0	0	0	0	209603	0
Heartbleed	0	0	0	0	0	0	0	0	0	0	0	0	209429

(d) XGB

Fig. 17 multilabel Confusion Matrix for CIC-IDS2017 Dataset

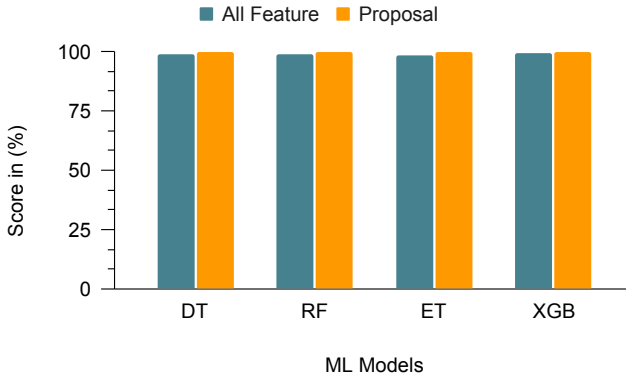


(a) Binary



(b) Multilabel

Fig. 18 Binary and multilabel ROC Curve for CIC-IDS2017 Dataset



(a) Binary

Fig. 19 Accuracy performance Bar Chart for All Features and Proposal Features for the CIC-IDS2018 Dataset

ML	Accuracy		Precision		Recall		F1-score	
	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal	All Feature	Proposal
DT	98.88	99.94	95.25	99.94	95.21	99.94	95.21	99.94
RF	98.74	99.93	98.25	99.93	91.55	99.93	91.55	99.93
ET	98.37	99.94	96.81	99.94	90.38	99.94	90.38	99.94
XGB	99.11	99.87	98.51	99.87	96.51	99.87	96.51	99.87

Table 12 Performance Analysis of CIC-IDS2018 Dataset

The result analysis, as presented in Table 12, highlights the performance assessment of various ML algorithms using both the "All Feature" and "Proposed" feature sets. Notably, among these algorithms, DT and ET consistently stand out as top performers in terms of accuracy, precision, recall, and F1-score. When considering the "Proposed" feature set, both DT and ET exhibit exceptional performance, achieving an impressive accuracy of 99.94%, surpassing other algorithms, including RF with an accuracy of 99.93% and XGB with 98.87%. This remarkable increase in accuracy extends to precision, recall, and F1-score metrics, underscoring the effectiveness of DT and ET. Their success can be attributed to their decision tree-based learning approach, which allows them to effectively model complex relationships in the data. Additionally, they demonstrate robustness when applied to the "Proposed" feature set, which may involve a more intricate feature engineering process.

The confusion matrix is displayed in Figure 20. A successful predictive model is characterized by a low number of Type 1 (FP) and Type 2 (FN) errors in the confusion matrix. For DT, the TP rate stands impressively at 49.74%, while the TN rate is equally strong at 49.76%. Additionally, the FP and FN rates are remarkably low, at 0.16% and 0.24%, respectively. These findings highlight DT's robust performance in accurately identifying positive cases (intrusions) and negative cases (non-intrusions), making it a compelling choice for intrusion detection. For ET, the TP rate is an outstanding 49.78%, and the TN rate is equally impressive at 49.82%. Furthermore, the FP and

FN rates are notably low, at 0.18% and 0.28%, respectively. These results underscore the exceptional performance of ET in accurately identifying both positive cases (intrusions) and negative cases (non-intrusions), making it a compelling choice for intrusion detection.

Among all the evaluated models, it is evident that both DT and ET outperform the others, showcasing superior performance in terms of TP and TN rates for IDS. They consistently deliver higher TP and TN rates, demonstrating their effectiveness in accurately identifying intrusions while maintaining a low rate of false positives and false negatives.

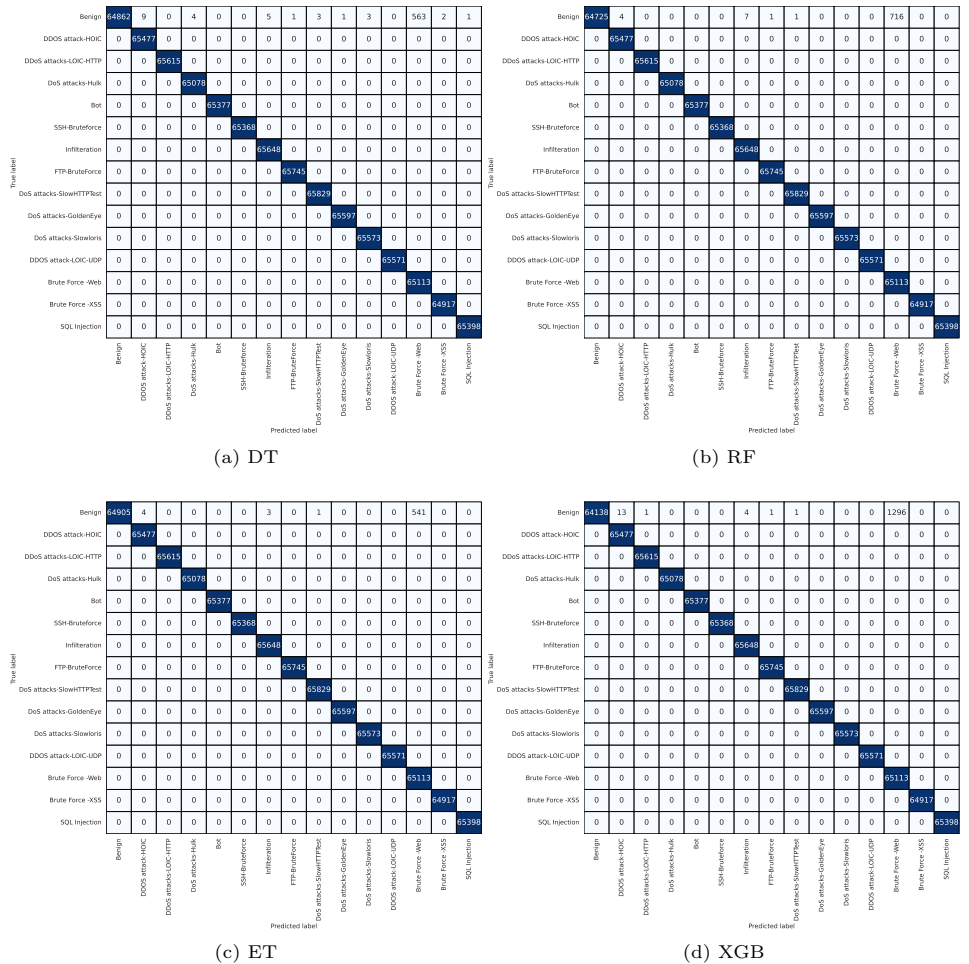


Fig. 20 Confusion Matrix for CIC-IDS2018 Dataset

In Figure 21, the ROC Curve depicts the performance of machine learning models on the CIC-IDS2018 Dataset. The ROC Curves clearly demonstrate

that the AUC values for DT, RF, ET, and XGB approach the desirable threshold of 1, indicative of an effective model for distinguishing between different classes. Notably, both RF and ET achieve the highest AUC scores. Specifically, the AUC scores are 99.99% for DT, a perfect 100% for RF and ET, and 84.85% for XGB. These high AUC scores, approaching 1, signify the strong predictive performance of the model on the CIC-IDS2018 dataset, underlining its effectiveness in class differentiation.

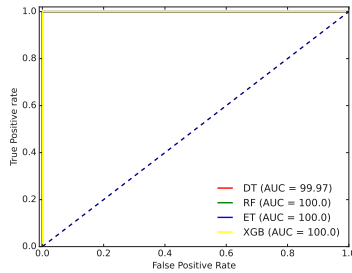


Fig. 21 ROC Curve for CIC-IDS2018 Dataset

5.4 Discussion

We have conducted a comprehensive comparative analysis of our proposed model in conjunction with other models. The comparative results are systematically presented in tabular format, as outlined in Table 13 for the UNSW-NB15 dataset, Table 14 for the CIC-IDS2017 dataset, and Table 15 for the CIC-IDS2018 dataset. These tables offer valuable insights into the performance of our model relative to other models, facilitating a detailed examination of the results across different datasets.

In our research, we address the challenge of imbalanced datasets by employing Random Oversampling (RO). Additionally, we apply the Stacking Feature Embedded (SFE) technique to augment feature sets and create metadata. Subsequently, we reduce the dimensionality to 10 features using Principal Component Analysis (PCA). Our model is then trained using popular machine learning algorithms, including DT, RF, ET, and XGB. The performance of our model is evaluated on two prominent datasets: UNSW-NB15 and CIC-IDS2017. For the UNSW-NB15 dataset, we achieve noteworthy accuracy scores. In binary classification, our model attains accuracy rates of 98.97% (DT), 99.59% (RF), 99.59% (ET), and 98.81% (XGB). In multilabel classification, the accuracy scores reach 99.79% (DT), 99.95% (RF), 99.95% (ET), and 95.04% (XGB). For the CIC-IDS2017 dataset, our model continues to excel. In binary classification, we obtain accuracy rates of 99.91% (DT), 99.94% (RF), 99.95% (ET), and 99.65% (XGB). In multilabel classification, the accuracy scores are impressive, with 99.99% for DT, RF, and ET, and 99.94% for XGB. Furthermore, in the evaluation on the CIC-IDS2018 Dataset, our

Sl. NO.	Authors	Data Balancing	Dimension Reduction	Algorithm	Selected Feature	Binary Acc(%)	Multilabel Acc(%)
1	Moualla et al (2021)	SMOTE	-	ELM	-	98.43	-
2	Kasongo and Sun (2020b)	-	XGB	DT, ANN	19	90.85 (DT)	77.51 (ANN)
3	Ahmad et al (2021)	-	MQTT+TCP	RF	-	98.67	97.37
4	Kasongo and Sun (2020a)	-	WFEU	FFDNN	22	87.10	77.16
5	Choudhary and Kesswani (2020)	-	-	DNN	-	91.50	-
6	Aleesa et al (2021)	-	-	ANN	-	-	97.89
7	Al and Dener (2021)	STL	-	LSTM+CNN	-	-	-
8	Zhang et al (2020)	SGM	-	CNN	-	-	96.54
9	Hassan et al (2020)	-	-	CNN-WDLSTM	-	97.17	98.43
10	Our Proposal	RO	SFE-PCA	RF	10	99.59	99.95
11	Our Proposal	RO	SFE-PCA	ET	10	99.59	99.95

Table 13 Comparison Analysis of UNSW-NB15 Dataset

model maintains high accuracy rates. We achieve accuracy scores of 99.94% (DT), 99.93% (RF), 99.94% (ET), and 99.87% (XGB). The results analysis reveals that the highest accuracy rates for binary and multilabel classification are 99.59% and 99.95%, achieved with RF and ET algorithms on the UNSW-NB15 dataset. In contrast, for the CIC-IDS2017 dataset, the highest accuracy rate is 99.99% for binary classification using the ET model and 99.99% for multilabel classification using DT, RF, and ET models.

In summary, our proposed model consistently outperforms existing methods in binary and multilabel classification scenarios with reduced features. The 10-feature dimension reduction (RR is 10:N, where N represents the input features) significantly enhances intrusion detection accuracy, minimizing false positive and negative rates. These results emphasize the importance of considering at least 10 features for optimal intrusion detection accuracy across various dimensional datasets.

6 Time Complexity

Time complexity, which denotes the time required for executing an operation, plays a crucial role in assessing the efficiency of algorithms ([Talukder et al,](#)

SI. NO.	Authors	Data Balancing	Dimension Reduction	Algorithm Selected Feature		Accuracy(%)
1	Kshirsagar and Kumar (2021)	-	IGR+CR +ReF	PART	-	99.95 (Binary)
2	Our Proposal	RO	SFE-PCA	ET	10	99.95 (Binary)
3	Hammad et al (2021)	-	t-SNE	RF	-	99.78
4	Al-Janabi and Ismail (2021)	-	NTLBO	LR	22	97.00
5	Guezzaz et al (2021)	-	EDFS	DT	-	98.80
6	Stiawan et al (2020)	-	IG+Ranking +Grouping	RF	22	99.86
7	Stiawan et al (2020)	-	IG+Ranking +Grouping	J48	52	99.87
8	Bhardwaj et al (2021)	-	-	DNN+AC \emptyset		98.25
9	Zhang et al (2020)	SGM	-	CNN	-	99.85
10	Our Proposal	RO	SFE-PCA	DT	10	99.99
11	Our Proposal	RO	SFE-PCA	RF	10	99.99
12	Our Proposal	RO	SFE-PCA	ET	10	99.99

Table 14 Comparison Analysis of CIC-IDS2017 Dataset

SI. NO.	Authors	Data Balancing	Dimension Reduction	Algorithm Selected Feature		Accuracy(%)
1	Seth et al (2021)	-	HFS	LightGBM24		97.73
2	Khan (2021)	-	-	CNN +RNN	-	97.75
3	Kim et al (2020)	-	-	CNN	-	91.50
4	Our Proposal	RO	SFE-PCA	DT	10	99.94
5	Our Proposal	RO	SFE-PCA	RF	10	99.94

Table 15 Comparison Analysis of CIC-IDS2018 Dataset

2022, 2024). In the context of IDS, evaluating the time complexity of ML models is paramount for efficient operation. We analyze the time complexity of key algorithms such as DT, RF, MLP, KNN, LGB and XGB models as follows.

- **DT:** The time complexity is typically $O(n \cdot m \cdot \log(m))$, with n representing data points and m as features. It constructs a tree by recursively partitioning data.

- **RF**: Comprising multiple DTs, its time complexity is $O(t \cdot n \cdot m \cdot \log(m))$, where t is the number of trees.
- **MLP (Multi-Layer Perceptron)**: With multiple layers and neurons, its time complexity is $O(w \cdot e \cdot n)$, where w is the number of weights, e the number of epochs, and n the number of data points.
- **KNN (K-Nearest Neighbors)**: This non-parametric method has a time complexity of $O(n \cdot m)$ for training, where n is the number of data points and m is the number of features. The prediction phase can be more computationally intensive.
- **LGB (Light Gradient Boosting Machine)**: Known for efficiency and low memory usage, LGB's time complexity is $O(t \cdot n \cdot m)$, where t is the number of trees.
- **XGB (XGBoost)**: This model's time complexity varies but is generally $O(t \cdot d)$, where t is the number of trees and d the depth of the trees.

SI. No.	ML Model	Time Complexity
1	DT (Decision Trees)	$O(n \cdot m \cdot \log(m))$
2	RF (Random Forests)	$O(t \cdot n \cdot m \cdot \log(m))$
3	ET (Extra Trees)	$O(t \cdot n \cdot m \cdot \log(m))$
4	XGB (XGBoost)	$O(t \cdot d)$

Table 16 Time Complexity of ML Models in IDS

7 Conclusion

In conclusion, our research has introduced a novel approach to network intrusion detection by combining various techniques to address the challenges of imbalanced data, feature embedding, and dimension reduction. Our model leverages the Random Oversampling (RO) method to tackle data imbalance, utilizes feature embedding through Kmeans and GM clustering results, and employs Principal Component Analysis (PCA) for dimension reduction. We have evaluated our model's performance with four prominent ML algorithms, DT, RF, ET and XGB for binary and multilabel classification studies using three benchmark datasets: UNSW-NB15, CIC-IDS2017 and CIC-IDS2018. Our experimental results have demonstrated exceptional accuracy rates, with RF and ET achieving 99.59% and 99.95% accuracy, respectively, on the UNSW-NB15 dataset. Besides, our model has achieved remarkable accuracy rates, with DT, RF, and ET reaching 99.99% on the CIC-IDS2017 dataset and for CIC-IDS2018 we achieved 99.94% accuracy rate using DT and RF models. These performance scores surpass those of existing methods, indicating the effectiveness of our approach in enhancing network intrusion detection. Our proposed model brings about substantial improvements across these benchmark datasets. Specifically, we observed a significant increase in accuracy, with enhancements ranging from 1.52% to 22.19% for UNSW-NB15, 0.12% to 2.99% for CIC-IDS2017, and 1.99% to 8.44% for CIC-IDS2018 when compared to

prior research. These results highlight the remarkable advancements our model introduces in the field of intrusion detection.

Our contributions to the field of network intrusion detection are substantial. We have addressed the persistent challenge of imbalanced data, ensuring that our model can handle real-world scenarios where intrusion instances are often rare compared to benign network traffic. The incorporation of feature embedding techniques has allowed us to capture more nuanced patterns and anomalies within the data, thus improving detection accuracy. Additionally, the application of PCA for dimension reduction has not only reduced computational complexity but also enhanced the interpretability of the model. The benefits of our new model extend beyond accuracy improvements. It offers a more robust and adaptable solution for intrusion detection, capable of handling varying data distributions and network environments. By combining multiple machine learning algorithms, our model harnesses the strengths of each, providing a versatile tool for network security professionals. Furthermore, its enhanced accuracy and lower false positive rates can significantly reduce the burden of false alarms in intrusion detection systems, allowing security teams to focus on the most critical threats.

In practical terms, our model can be invaluable for IDS in safeguarding network infrastructure. Its high accuracy rates and adaptability make it well-suited for identifying both known and novel threats, enhancing the overall security posture of organizations. The reduced false positive rates contribute to a more efficient use of resources, as security teams can concentrate their efforts on genuine security incidents. Ultimately, our model represents a significant advancement in the field of network intrusion detection, offering a more reliable and efficient solution for protecting critical network assets.

The limitation of our research is that we did not employ deep learning models along with optimization techniques. While our current approach has demonstrated remarkable results, there remains untapped potential for further improving the performance of intrusion detection systems. In the future, we envision expanding our work to incorporate deep learning models, which have shown great promise in various fields, including intrusion detection. DL algorithms, such as DNN, RNN or Hybrid are capable of capturing intricate patterns and representations in complex data, which can be particularly advantageous in the realm of network security.

Declarations

Conflict of interest

The authors have no conflicts of interest to declare that they are relevant to the content of this article.

Ethics approval

Not applicable

Consent to participate

Not applicable

Consent to Publish

Not applicable

Availability of data and materials

The selected datasets are sourced from free and open-access sources such as UNSW-NB15: <https://research.unsw.edu.au/projects/unsw-nb15-dataset>, CIC-IDS2017: <https://www.unb.ca/cic/datasets/ids-2017.html> and CIC-IDS2018: <https://www.unb.ca/cic/datasets/ids-2018.html>.

Authors' contributions

Md. Alamin Talukder: Conceptualization, Data curation, Methodology, Software, Formal analysis, Visualization, Writing – original draft, Writing – review & editing. Md. Manowarul Islam: Methodology, Investigation, Validation, Project administration. Md. Ashraf Uddin: Supervision, Investigation, Resources, Validation. Khondokar Fida Hasan: Investigation, Validation, Visualization Selina Sharmin: Software, Validation. Salem A. Alyami: Investigation, Validation. Mohammad Ali Moni: Investigation, Validation, Visualization, Writing – review & editing.

Acknowledgments

This research was supported by the Deanship of Scientific Research Large Groups at King Khalid University, Kingdom of Saudi Arabia (RGP.2/219/43)

References

- Adeyemo VE, Abdullah A, JhanJhi N, et al (2019) Ensemble and deep-learning methods for two-class and multi-attack anomaly intrusion detection: an empirical study. *International Journal of Advanced Computer Science and Applications* 10(9)
- Ahmad M, Riaz Q, Zeeshan M, et al (2021) Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using unsw-nb15 data-set. *EURASIP Journal on Wireless Communications and Networking* 2021(1):1–23
- Ahmim A, Maglaras L, Ferrag MA, et al (2019) 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), IEEE, pp 228–233

- Al S, Dener M (2021) Stl-hdl: A new hybrid network intrusion detection system for imbalanced dataset on big data environment. *Computers & Security* 110:102,435
- Al-Janabi M, Ismail MA (2021) Improved intrusion detection algorithm based on tlbo and ga algorithms. *Int Arab J Inf Technol* 18(2):170–179
- Aleesa A, Younis M, Mohammed AA, et al (2021) Deep-intrusion detection system with enhanced unsw-nb15 dataset based on deep learning techniques. *Journal of Engineering Science and Technology* 16(1):711–727
- Allahrakha N (2023) Balancing cyber-security and privacy: Legal and ethical considerations in the digital age. *Legal Issues in the Digital Age* 4(2):78–121
- Bhardwaj A, Mangat V, Vig R (2021) Hybrid deep neural architecture for detection of ddos attacks in cloud computing. In: *Intelligent Systems, Technologies and Applications*. Springer, p 71–86
- Bhati BS, Rai CS (2021) Intrusion detection technique using coarse gaussian svm. *International Journal of Grid and Utility Computing* 12(1):27–32
- Bhavani TT, Rao MK, Reddy AM (2020) Network intrusion detection system using random forest and decision tree machine learning techniques. In: *First international conference on sustainable technologies for computational intelligence*, Springer, pp 637–643
- Breiman L (2001) Random forests. *Machine learning* 45(1):5–32
- Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp 785–794
- Chen T, He T (2015) Higgs boson discovery with boosted trees. In: *NIPS 2014 workshop on high-energy physics and machine learning*, PMLR, pp 69–80
- Cholakoska A, Shushlevska M, Todorov Z, et al (2021) Analysis of machine learning classification techniques for anomaly detection with nsl-kdd data set. In: *Proceedings of the Computational Methods in Systems and Software*, Springer, pp 258–267
- Choudhary S, Kesswani N (2020) Analysis of kdd-cup'99, nsl-kdd and unsw-nb15 datasets using deep learning in iot. *Procedia Computer Science* 167:1561–1573
- Das S, Gangwani P, Upadhyay H (2023) Integration of machine learning with cybersecurity: Applications and challenges. *Artificial Intelligence in Cyber Security: Theories and Applications* pp 67–81

- George AS, George AH, Baskar T (2023) Digitally immune systems: building robust defences in the age of cyber threats. *Partners Universal International Innovation Journal* 1(4):155–172
- Geurts P, Ernst D (2006) Louiswehenkel. 2006. extremely randomized trees. *Machine Learning* 63(1)
- Gharib A, Sharafaldin I, Lashkari AH, et al (2016) An evaluation framework for intrusion detection dataset. In: 2016 International Conference on Information Science and Security (ICISS), IEEE, pp 1–6
- Gorunescu F (2011) *Data Mining: Concepts, models and techniques*, vol 12. Springer Science & Business Media
- Guezzaz A, Benkirane S, Azrou M, et al (2021) A reliable network intrusion detection approach using decision tree with enhanced data quality. *Security and Communication Networks* 2021
- Hammad M, Hewahi N, Elmedany W (2021) T-snerf: A novel high accuracy machine learning approach for intrusion detection systems. *IET Information Security* 15(2):178–190
- Hassan MM, Gumaie A, Alsanad A, et al (2020) A hybrid deep learning model for efficient intrusion detection in big data environment. *Information Sciences* 513:386–396
- Hussain F, Hussain R, Hassan SA, et al (2020) Machine learning in iot security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials* 22(3):1686–1721
- Istiaque SM, Khan AI, Al Hassan Z, et al (2021) Performance evaluation of a smart intrusion detection system (ids) model. *European Journal of Engineering and Technology Research* 6(2):148–152
- Jayalaxmi P, Saha R, Kumar G, et al (2022) Machine and deep learning solutions for intrusion detection and prevention in iots: A survey. *IEEE Access*
- Kafi MA, Akter N (2023) Securing financial information in the digital realm: Case studies in cybersecurity for accounting data protection. *American Journal of Trade and Policy* 10(1):15–26
- Kasongo SM, Sun Y (2020a) A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Computers & Security* 92:101,752

- Kasongo SM, Sun Y (2020b) Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset. *Journal of Big Data* 7(1):1–20
- Khan AR, Kashif M, Jhaveri RH, et al (2022) Deep learning for intrusion detection and security of internet of things (iot): current analysis, challenges, and possible solutions. *Security and Communication Networks* 2022
- Khan MA (2021) Hcrnmid: Hybrid convolutional recurrent neural network-based network intrusion detection system. *Processes* 9(5):834
- Kim J, Kim J, Kim H, et al (2020) Cnn-based network intrusion detection against denial-of-service attacks. *Electronics* 9(6):916
- Kshirsagar D, Kumar S (2021) An efficient feature reduction method for the detection of dos attack. *ICT Express*
- Kumar V, Das AK, Sinha D (2020) Statistical analysis of the unsw-nb15 dataset for intrusion detection. In: *Computational Intelligence in Pattern Recognition*. Springer, p 279–294
- Marwala T (2023) Cybersecurity in politics. In: *Artificial Intelligence, Game Theory and Mechanism Design in Politics*. Springer, p 135–155
- Mishra S, Tyagi AK (2022) The role of machine learning techniques in internet of things-based cloud applications. *Artificial intelligence-based internet of things systems* pp 105–135
- Mohammadi S, Mirvaziri H, Ghazizadeh-Ahsae M, et al (2019) Cyber intrusion detection by combined feature selection algorithm. *Journal of information security and applications* 44:80–88
- Moualla S, Khorzom K, Jafar A (2021) Improving the performance of machine learning-based network intrusion detection systems on the unsw-nb15 dataset. *Computational Intelligence and Neuroscience* 2021
- Moustafa N, Slay J (2015) Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: *2015 military communications and information systems conference (MilCIS), IEEE*, pp 1–6
- Moustafa N, Slay J (2016) The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. *Information Security Journal: A Global Perspective* 25(1-3):18–31

- Mueller S (2021) Facing the 2020 pandemic: What does cyberbiosecurity want us to know to safeguard the future? *Biosafety and health* 3(1):11–21
- Mugabo E, Zhang QY, Ngaboyindekwe A, et al (2021) Intrusion detection method based on mapreduce for evolutionary feature selection in mobile cloud computing. *International Journal of Network Security* 23(1):106–115
- Narayanasami S, Sengan S, Khurram S, et al (2021) Biological feature selection and classification techniques for intrusion detection on bat. *Wireless Personal Communications* pp 1–23
- Nguyen H, Lim Y, Seo M, et al (2023) Strengthening information security through zero trust architecture: A case study in south korea. In: *International Conference on Intelligent Systems and Data Science*, Springer, pp 63–77
- Nimbalkar P, Kshirsagar D (2021) Feature selection for intrusion detection system in internet-of-things (iot). *ICT Express* 7(2):177–181
- Norwahidayah S, Farahah N, Amirah A, et al (2021) Performances of artificial neural network (ann) and particle swarm optimization (pso) using kdd cup ‘99 dataset in intrusion detection system (ids). In: *Journal of Physics: Conference Series*, IOP Publishing, p 012061
- Preuveneers D, Joosen W (2021) Sharing machine learning models as indicators of compromise for cyber threat intelligence. *Journal of Cybersecurity and Privacy* 1(1):140–163
- Protić D, Stanković M, Prodanović R, et al (2023) Numerical feature selection and hyperbolic tangent feature scaling in machine learning-based detection of anomalies in the computer network behavior. *Electronics* 12(19):4158
- Sameera G, Vardhan RV, Sarma K (2016) Binary classification using multivariate receiver operating characteristic curve for continuous data. *Journal of biopharmaceutical statistics* 26(3):421–431
- Sarker IH, Kayes A, Badsha S, et al (2020) Cybersecurity data science: an overview from machine learning perspective. *Journal of Big data* 7:1–29
- Schmitt M (2023) Securing the digital world: Protecting smart infrastructures and digital industries with artificial intelligence (ai)-enabled malware and intrusion detection. *Journal of Industrial Information Integration* 36:100,520
- 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):1–28

- Sharafaldin I, Lashkari AH, Ghorbani AA (2018a) Intrusion detection evaluation dataset (cic-ids2017). Proceedings of the of Canadian Institute for Cybersecurity
- Sharafaldin I, Lashkari AH, Ghorbani AA (2018b) Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* 1:108–116
- Sharmin S, Ahammad T, Talukder MA, et al (2023) A hybrid dependable deep feature extraction and ensemble-based machine learning approach for breast cancer detection. *IEEE Access*
- Singh P, Singh P (2023) Artificial intelligence: The backbone of national security in 21st century. *Tuijin Jishu/Journal of Propulsion Technology* 44(4):2022–2038
- Stiawan D, Idris MYB, Bamhdi AM, et al (2020) Cicans-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access* 8:132,911–132,921
- Talita A, Nataza O, Rustam Z (2021) Naïve bayes classifier and particle swarm optimization feature selection method for classifying intrusion detection system dataset. In: *Journal of Physics: Conference Series*, IOP Publishing, p 012021
- Talukder MA, Islam MM, Uddin MA, et al (2022) Machine learning-based lung and colon cancer detection using deep feature extraction and ensemble learning. *Expert Systems with Applications* 205:117,695
- Talukder MA, Hasan KF, Islam MM, et al (2023a) A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications* 72:103,405
- Talukder MA, Islam MM, Uddin MA, et al (2023b) An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning. *Expert Systems with Applications* p 120534
- Talukder MA, Layek MA, Kazi M, et al (2024) Empowering covid-19 detection: Optimizing performance through fine-tuned efficientnet deep learning architecture. *Computers in Biology and Medicine* 168:107,789
- Uddin N, Ahamed MKU, Uddin MA, et al (2023) An ensemble machine learning based bank loan approval predictions system with a smart application. *International Journal of Cognitive Computing in Engineering* 4:327–339
- Vasan KK, Surendiran B (2016) Dimensionality reduction using principal component analysis for network intrusion detection. *Perspectives in Science*

8:510–512

Vergara IA, Norambuena T, Ferrada E, et al (2008) Star: a simple tool for the statistical comparison of roc curves. *BMC bioinformatics* 9(1):1–5

Yulianto A, Sukarno P, Suwastika NA (2019) Improving adaboost-based intrusion detection system (ids) performance on cic ids 2017 dataset. In: *Journal of Physics: Conference Series*, IOP Publishing, p 012018

Zhang H, Huang L, Wu CQ, et al (2020) An effective convolutional neural network based on smote and gaussian mixture model for intrusion detection in imbalanced dataset. *Computer Networks* 177:107,315

Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. *Journal of computational and graphical statistics* 15(2):265–286