The American Journal of Applied Sciences
ISSN 2689-0992 | Open Access

TYPE Original Research
PAGE NO. 32-54
DOI 10.37547/tajas/Volume07Issue11-05

Check for updates

Development and Evaluation of Anomaly-Based IDS model for IoT with Hybrid ML Algorithms

Nabeel Abdulrazaq Yaseen
University of Misan, Iraq

OPEN ACCESS

SUBMITED 12 October 2025
ACCEPTED 28 October 2025
PUBLISHED 19 November 2025
VOLUME Vol.07 Issue 11 2025

CITATION
Nabeel Abdulrazaq Yaseen. (2025). Development and Evaluation of Anomaly-Based IDS model for IoT with Hybrid ML Algorithms. The American Journal of Applied Sciences, 7(11), 32–54. https://doi.org/10.37547/tajas/Volume07Issue11-05

Abstract

Background: The rapidly expanding concept, the Internet of Things (IoT), connects a large number of devices to the internet for effective real-time data sharing and communication. But as IoT technology advances, new security risks also arise, and for this reason, machine learning (ML)-based intrusion detection systems (IDS), particularly anomaly-based ones, have emerged as a crucial defence against these new dangers to IoT networks.

Objective: The primary goal of this study is to develop and assess anomaly-based IDS based on ML techniques for use in Internet of Things environments. The goal is to improve the models' performance in terms of accuracy and usefulness.

Methods: The first phase of the study is a thorough review of the recent studies in the field to support the model design. The steps of the experimental stage include data preprocessing, encoding, and normalization. Data balancing was achieved using the SMOTEENN technique. The experiments and validation studies were performed on the UNSW-NB15 dataset. The employed ML algorithms for the study include DT, DNN, RF, XGBoost, and KNN, and the performance was in terms of precision, recall, accuracy, and F1-score.

Results: XGBoost recorded the best detection accuracy (96.37%), followed by RF and DNN. The models were evaluated on both normal and attack traffic, and the suggested model outperformed many existing approaches in recent literature.

Conclusion: The experimental findings confirm the suitability of hybrid and ensemble ML models in improving intrusion detection performance in IoT systems. Future research should consider integrating

real-time datasets and combining deep learning with ensemble methods to develop more robust and adaptive models.

Keywords: *IoTs; IDS; Machine Learning; Anomaly Detection; SMOTEENN.*

1. INTRODUCTION

The Internet of Things (IoT) provides a platform through which people and gadgets connect to the internet for real-world data sharing and connectivity. The impression of smartness has been improved further by the addition of computers, sensors, buildings, streets, and even communities by recent IoT technologies. The diverse range of communication devices used in many sectors, such as education, healthcare, and military, are instances of IoT devices operating in many environments to achieve several objectives. Almost every industry in contemporary civilization benefits greatly from IoT, including surveillance, agriculture, medical, and more. Moreover, IoT is a networked system built on authorized protocols that share information [1], [2]. In recent years, IoT use has also continued to rise across several industries, including health, where it has revolutionized many aspects of healthcare. However, issues surrounding the safety and privacy of IoT networks, as well as the associated energy limitations and their poor scalability, have remained areas of active research interest [3]. The safety and reliability of IoT device interactions have become increasingly important as IoT systems grow in complexity and scope [4]. Trust and reputation management systems are essential for the integrity and reliability of IoT networks.

The increases in attacks on IoT networks have mostly been blamed on the rapid advancement of IoT technology in areas such as smart grids, patient monitoring systems, smart manufacturing, and even logistics [5], [6]. Furthermore, IoT devices can self-configure, considering the enormous automated tasks that are programmed into their design. UPnP and mDNS protocols help devices find and connect to networks without requiring much effort from users to configure them for network access. These devices have plug-and-play features that can connect or power devices up to the required network protocol; they will also start up without any manual configuration. The devices can thus be dynamically configured in several situations to receive IP addresses and other network configurations [7]. IoT devices use auto-provisioning technologies to modify settings based on specific deployment scenarios and configuration data received from centralized servers and the cloud.

Smart IoT systems can use ML algorithms for self-optimizing configurations over time and adapting to their users' behaviors. Self-configuration helps IoT devices to adapt better and operate more efficiently; as a result, it cuts down human intervention and improves the usability of devices across multiple networks. Within the context of IoT systems, traditional intrusion detection systems do not offer sufficient security solutions due to their limited bandwidth and global connectivity [8]. There is a need to customize intrusion detection systems for IoT systems, so that the deployed sensors do not get abused, to automate and facilitate the stopping of intrusions by monitoring suspicious activity on the IoT network and, in a timely manner, reporting such activity to the network administrator [9]. The IDS is capable of enlightening the network administrator about novel attack vectors that may be of aid to him/her. There exist stratified requirements to revise IoT systems to identify newer attack instances [10].

The three existing formats of intrusion detection systems include "signature-based, anomaly-based, and specification-based IDS". For the signature-based systems, attacks can be detected by signatures and preset attack patterns, while in 'anomaly' based systems, attacks are flagged and classified by deviations from normal behaviour. A specification-based system, on the other hand, is bound by the system administrator's commands and governs them heuristically [11]. The complexity of the systems under consideration, in addition to the dynamic and multitude of layers that are characteristic of their ecosystem, contributes to the fact that ensuring they contain the newest trends in intruder detection is an arduous burden [12]. Hence, scholars have tried to improve their effectiveness by employing flexible techniques like DL and ML models in their frameworks [13].

There are two types of ML models-single classifiers, which use a single classifier as their basis, and multi-classifiers, which use several classification models simultaneously [14]. IDS models are classified as either binary or multiclass models; in the former, traffic is divided into two groups - normal (0) and abnormal (1),

while the latter identifies the attack types. When a multiclass model is trained on datasets with insufficient attack cases, the performance is mostly poor on unknown attacks due to its complexity compared to the binary model [15]. Furthermore, security attacks can be classified as either active or passive [16]. Active attacks are encountered during the run-time conditions and, as such, they may interfere with or damage the physical device. Unlike passive attacks, it is often difficult to perform and detect active attacks. The most prevalent type of active attack is denial of service [16]; the other types of active attacks are packet replay, message modification, and spoofing. A passive attack keeps an eye on the data for the specific target [17].

Scholars have proposed various types of anomaly-based IDSs for the conventional network architecture. Nevertheless, these solutions are unsuitable for direct implementation in IoT gadgets due to their poor bandwidth, low computing power, and low storage capacity [18]. Hence, researchers have proposed to improve existing models or algorithms or create better models or algorithms to overcome this limitation. In addition, several studies that apply ML models on anomaly-based IDSs have been reported, but optimal performances have not been reported [19], [20]. Anomaly-based IDSs enhance the efficacy of IoT devices that operate over limited network resources and successfully detect anomalies, according to [21]. Most of the efficiency is assessed using the detection accuracy, false alarm rate, and detection rate. Furthermore, the efficiency of the models is also determined by the amount of energy and memory used [22]. Therefore, any suggested solution must prioritize increasing detection efficacy while reducing energy and storage resource usage during the detection process. This study aims to assess the suitability of ML approaches in building anomaly-based IDS for IoT environments; specifically, it seeks to

1. Assess ML algorithms for effectiveness in detecting and classifying IoT attacks.
2. Compare recent contributions to anomaly-based IDS models from 2019 to 2024.
3. Compare the proposed models for performance against some existing methods using different evaluation metrics.
4. Identify current gaps and propose directions for future research.

This approach highlights the model's practical development and evaluation, which was refined with real-world data (UNSW-NB15) and ML algorithms to assess model performance and test claimed advancements over preceding works. By combining systematic analysis with empirical evaluation, the study manages to address both theoretical and practical implications, which increases the value of the contribution, the central contributions of this paper are as follows:

1. Using the Min-Max normalization method to transform all the feature values to the same scale.
2. Employing data cleaning techniques on the utilized dataset to modify or eliminate redundant and erroneous entries, as well as harshly formatted datasets.
3. Using label encoding as the utilized dataset contains varying labels per column; these labels were either numerical or alphabetical characters. An ML model cannot accept this type of data in its raw form; therefore, label encoding is used to label the data so that the model can understand it.
4. Identifying network attacks based on the anomaly-based IDS model using the effectiveness of DT, DNN, RF, XGBoost, and KNN.
5. Comparing the proposed models with existing ones in terms of performance.

This work is arranged in sections as follows: Section 2 presents the literature survey of the analyzed studies. Section 3 introduces the suggested model. Section 4 contains the evaluation results, while Section 5 contains the conclusion and recommendations for further research.

Literature Review

The core studies on which the anomaly-based intrusion detection model has been proposed are presented in this section. Articles from 2019 to 2024 examined several machine learning models applied to IoT security through literature. An examination of key models, datasets, evaluation results, and feature engineering strategies is performed to develop the implemented solution. This section discusses the application of various machine learning techniques for network attack detection on traditional and IoT networks; it covers practical implementations and their performances. These methods work efficiently when the features are

chosen properly and the data is pre-processed enough for correct classification.

A study presented by[23] has developed the WFEU-FFDNN wireless IDS model, which implements a wrapper-based feature selection technique with the fully connected feedforward deep neural network (FFDNN) model for classification purposes. Utilizing the UNSW-NB15 and AWID datasets gave 87.10 % and 99.66 % accuracies on binary classification and 77.16 % and 99.77 % on multi-class classification. Other studies, like [24], designed a DL model for anomaly detection using an AutoEncoder (AE), where the model was built with three encoder and three decoder layers. Their model achieved an accuracy ranging from 84 % to 100 % on the KDD Cup '99 dataset and 95 % on a steel company dataset from Korea. Similarly, [25] used discriminative CNN and other DL models to process the CICIDS2018 and Bot-IoT datasets, further demonstrating the effectiveness of ML methods for intrusion detection. The accuracy was 97.3 % on the CSE-CIC-IDS2018 and 98.3 % on the Bot-IoT datasets for the discriminative CNN and the autoencoder models, respectively. In the work of [26], an unsupervised feature learning based on the autoencoder on KDD Cup '99 and NSL-KDD datasets was developed. The accuracy on the NSL-KDD and KDD Cup 99 datasets was 85 % and 97.85 %, respectively. The DNN model proposed by [27], configured with the ReLU activation function and three hidden layers, was trained with 150 epochs. The experiments showed that the model accurately predicted over 95 % of the different classes of attacks on the dataset, outperforming many other models. [28] developed a DL based model on the BoT–IoT dataset for intrusion detection systems. It was constructed with two dense layers using ReLU with 512 neurons per layer. The features were encoded and later, with the transfer learning idea, were used in the multiclass model for the same encoding employed in the binary classification. In addition to this, various hyperparameters were adjusted, such as learning rate, hidden layers, epochs, weight decay, and dropout regularization. The multiclass feedforward neural networks (FNN) model produced 99.79 % accuracy on four classes and 99 % on the binary class. As cited in [29], the authors constructed a DL model utilizing various deep architectures (CNN, DNN, MLP, and autoencoder) for IoT network intrusion detection. These models were applied to the UNSW-NB15 and NSL-KDD99 datasets, which revealed the MLP model to have the best performance (F1-score = 99.28 % and 95.76 % on the respective datasets) while the DNN model had the highest accuracy of 99.24 %.

[30] presented a DL-based Deep Feature Embedding Learning (DFEL) framework, which attained 93.13 % accuracy on the considered dataset. [31] constructed a novel framework for adversarial attack on DL-based IDS in the IoT network. In this framework, slight modifications to designated attack packets, as some of their attributes were shown to dramatically impact the models' prediction and increase the risks of DoS attacks. The model's Attack Success Rate (ASR) was greater than 95%. In another work, [32] suggested a CNN-IDS model that included a pooling layer, convolution layers, and five hidden layers. The dataset's superfluous features were eliminated using dimensionality reduction techniques. Models were trained on 10% of the KDDCup99 dataset. [33] introduced a DL-based IDS model using a hybrid LSTM-CNN for feature extraction. The impact of an unbalanced dataset on performance was reduced using the weight optimization technique. An overall model accuracy of 98.67% was demonstrated by evaluations on the CICIDS2017 dataset. Additionally, a DL-based CNN model for DoS attack detection was published in another paper by [34]; the CICIDS2018 and KDD Cup99 datasets were used to compare the model's performance to that of the RNN model. On the KDD Cup99, the accuracy was 99.5%, whereas on the CICIDS2018 dataset, it was 91.5%. [35] introduced a FNN model that used the information gain filter approach for feature selection. On the NSL-KDD dataset, model performance was compared to SVM and DT, and various neuron counts and learning rates were used for model tuning. The best accuracy of 99 % and 88 % for binary classification on the training and testing datasets, respectively, was attained by the model with three hidden layers, 0.005 learning rate, and 30 neurons per layer.

[36] developed an IDS model that used ML models for classification. Different ML techniques were compared for performance, and DT offered the best classification on various datasets. Three distinct datasets (KDD99, NSL-KDD, and UNSW-NB15) were used to assess the model's performance, and DT outperformed the others in the classification task (98 % accuracy). [37] assessed the Shamoon assault behavior using the FPSO. Shamoon focuses mostly on industrial data because fog nodes provide three different sorts of data: medical, educational, and industrial. Therefore, the study

estimated the best cost and fitness value of the Shammon assault strategy; it also tracked the movement and dispersion of Shammon attacks. According to the study's findings, Shamoon attacks are initiated by taking the shortest route feasible, meaning that identifying the first node will reveal the attack source.

It was suggested to use the IoT to create a smart health monitoring system that uses sensors to assess three important health parameters: body temperature, heart rate, and blood oxygen [38]. To guard against both internal and external cloud assaults, the data was gathered and encrypted using the AES method. After that, the encrypted data is sent to a medical facility, where servers receive it and decrypt it. According to the experimental results, the suggested approach performed better (C.I. = 95%). The consensus method for the blockchain was enhanced using a blockchain-based DL model for IoT. The suggested model outperformed the benchmark models, according to the results of the comparison study with the current consensus methodology [37]. The suggested model outperformed the benchmark models. [39] described a blockchain-based healthcare system that used a homomorphic encryption approach to protect patient health data privacy. Hyperledger Calliper is used for blockchain networks, and a graph theory-based binary search (BSS) with a hybrid DNN is used for intrusion detection in the Internet of Things. Performance was compared with benchmark models, and the suggested model provided greater security at lower computational costs.

An improved PSO was presented for jamming attack detection [40]; performance was compared with the traditional PSO and other optimization methods, and the result showed better performance of the new method in terms of coverage area and minimal fitness value.

METHODOLOGY

This study adopts an applied research methodology, focusing on the design, implementation, and evaluation of an enhanced anomaly-based IDS tailored for IoT environments. The methodological approach was carefully planned before data collection to ensure objectivity and transparency. The methodology includes:

1. A framework with structured data preprocessing, model training, and performance evaluation.
2. Specified metrics for dataset selection, feature engineering, and selection of ML algorithms (DT, DNN, RF, XGBoost, and KNN).
3. A standardized evaluation procedure, incorporating metrics such as accuracy, precision, recall, and F1-score for performance assessment.
4. Independent validation of experimental steps by two researchers specializing in cybersecurity and machine learning to minimize implementation bias.

1.1 Experimental Framework and Study Design

The research was carried out according to a predefined experimental framework to maintain the methodology, avoid bias, and make it reproducible. Before implementation, all processes regarding the selection of the dataset, preprocessing steps, design of the algorithm, and evaluation metrics were described. The research team consisted of a data scientist and a data science expert who verified every stage for uniformity and objectivity.

1.2 Dataset Justification and Preparation

The dataset utilized for this research (UNSW-NB15) was selected as it provided a variety of normal traffic and attack traffic. Moreover, the dataset is widely used in the area of intrusion detection. Datasets containing realistic, labeled, and diverse traffic types were prioritized through the selection process. The decision to use these datasets was made after looking at several benchmark datasets for their suitability. The dataset selection for this study was based on its relevance to anomaly-based intrusion detection in IoT environments. The selection of the UNSW-NB15 dataset for this work was due to its extensive coverage of different attack types and real-world traffic simulation. The preparation process included:

1. Dataset features screening for identifying relevant attributes for ML modeling.
2. Data preprocessing for the removal of noise, handling missing values, and data formatting.
3. Data normalization and label encoding to ensure all features are on the same scale and suitable for the employed algorithms.

To allow for the inclusion of only high-quality, representative data instances in the training and

validation of the model, a structured evaluation framework was implemented. The rigorous and consistent preparation of the structured data enables reproducibility of an experiment design.

1.3 Data Cleaning and Filtering Criteria

Only relevant, complete, and consistent entries of the dataset were used. Traffic samples that have a known attack type and complete features are only retained. The final dataset did not include entries with missing values or inconsistent labels, as well as redundant features, for data cleanliness and integrity.

1.4 Data Preprocessing and Feature Engineering

Standardized data preprocessing techniques were applied to extract features and transform the data. The main processes included Min-Max normalization and label encoding, along with filtering out noise and duplicates. Ultimately, these processes made sure that the data going into the machine learning models was clean, on the same scale, and usable.

1.5 Information Sources

Resources from various academic databases and indexing services like IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, etc., were scanned for context and the necessary background support.

1.6 Data Items

The key data items extracted for this study included algorithm types (e.g., DT, RF, XGBoost, KNN, DNN), dataset name and structure (e.g., UNSW-NB15), preprocessing techniques (normalization, label encoding), data balancing methods (e.g., SMOTEENN), performance measures (accuracy, precision, recall, F1-score), and implementation frameworks (e.g., Scikit-learn, TensorFlow). Additional items included the number of features selected and the total number of samples per class before and after balancing.

1.7 Evaluation Metrics

The metrics used for performance evaluation are accuracy (the ratio of the rightly predicted cases to the total cases), precision (the ratio of true positive cases among all the predicted positive cases), recall (the ratio of true positive cases among all actual positive cases), and F1-score (harmonic mean of precision and recall). Where available, AUC (Area Under the Curve) and confusion matrix (CM) values were also analyzed to supplement the evaluation. Although no formal GRADE assessment was carried out considering the peculiarities of the technological domain, levels of confidence in the synthesized evidence were taken to be in the moderate and, in some cases, high range along the dimensions of data transparency, model reproducibility, and benchmark datasets. Studies that used standard datasets such as the UNSW-NB15 and made the code or parameter settings openly accessible were assigned greater certainty. Limitations in certain experimental setups were transparently acknowledged in the discussion to maintain a balanced interpretation. In this study, several metrics were employed for performance evaluation; these metrics are as follows:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

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

$$F1 = 2\,\text{Precision} \times Recall\,Precision + Recall \quad (4)$$

Environmental and Experimental Setup

The model presented was trained and tested on the utilised dataset. The data is fit for IDS evaluation as it has normal and malicious traffic. The tests were done with Python 3.9 through Scikit learn and Tensorflow libraries in a uniform computing environment (Intel Core i7, 16GB RAM). The data was divided into training and test sets, which were subject to all the pre-processing steps before training.

Proposed Model and Empirical Evaluation:

The anomaly-based intrusion detection model presented in this work is introduced in this section. Five ML algorithms (DNN, XGBoost, KNN, RF, and DT) are used to construct the model. The objective is to assess each algorithm's efficiency and detection accuracy using the same dataset. The model employs data preprocessing steps, including Min-Max normalization, label encoding, and data cleaning to improve learning performance.

It takes a number of steps to create a complete framework that can categorize IoT network attacks. The actions that are taken to complete the model suggested

in this study are briefly illustrated in Figure 1. To prepare it for the ML algorithms, the first step is to collect raw network traffic data from the UNSW-NB 15 dataset [41]; this is followed by pre-processing in many steps to make it suitable for the next steps. The dataset underwent standardization and normalization processes to get rid of redundant or unstructured data; this step also provides a standardized data format for the model construction stages. Features are rescaled during the standardization process to ensure mean and standard deviation values of 0 and 1, respectively (also known as Z-score normalization). Equations 5 and 6 are illustrations of the standardization and normalization processes.

Next, the raw network traffic data undergoes data cleaning to eliminate or change redundant and inaccurate data entries, as well as improperly formatted datasets. The next phase is label encoding; in ML, researchers often come across data that has multiple labels in one or more columns. These labels could be either numbers or characters [42]. An ML model cannot be fed these kinds of data in their unprocessed state. Label encoding is sometimes used to label the data so that the model can understand it. Furthermore, "label encoding" was used to convert labels into numerical values before feeding the ML model. For each value in the category column, a number between 0 and N-1 is introduced and used as a substitute. A Label Encoder was used to normalize labels in a way that only includes values between 0 and n_classes-1. The categorical feature encoding in machine learning is seen in Equation 7.

$$x_{std} = x_{mean} - \sigma \quad (5)$$

Where x = original data.

$$normalization\ image = (image - mean)/\ standard\ deviation \quad (6)$$

$$Frequency\ encoding = \frac{frequency\ (categorical)}{size(data)} \quad (7)$$

In this work, SMOTEENN was used to accomplish an important step known as over-sampling of the prepared data [43]. The used data are prepared for feeding into five distinct machine learning algorithms, including RF [44], DT [45], KNN [46], XGBoost [47], and DNN [48], following data pre-processing and over-sampling. These algorithms underwent testing, training, and performance evaluation. Figure 1 shows the main flowchart of the suggested IDS model.

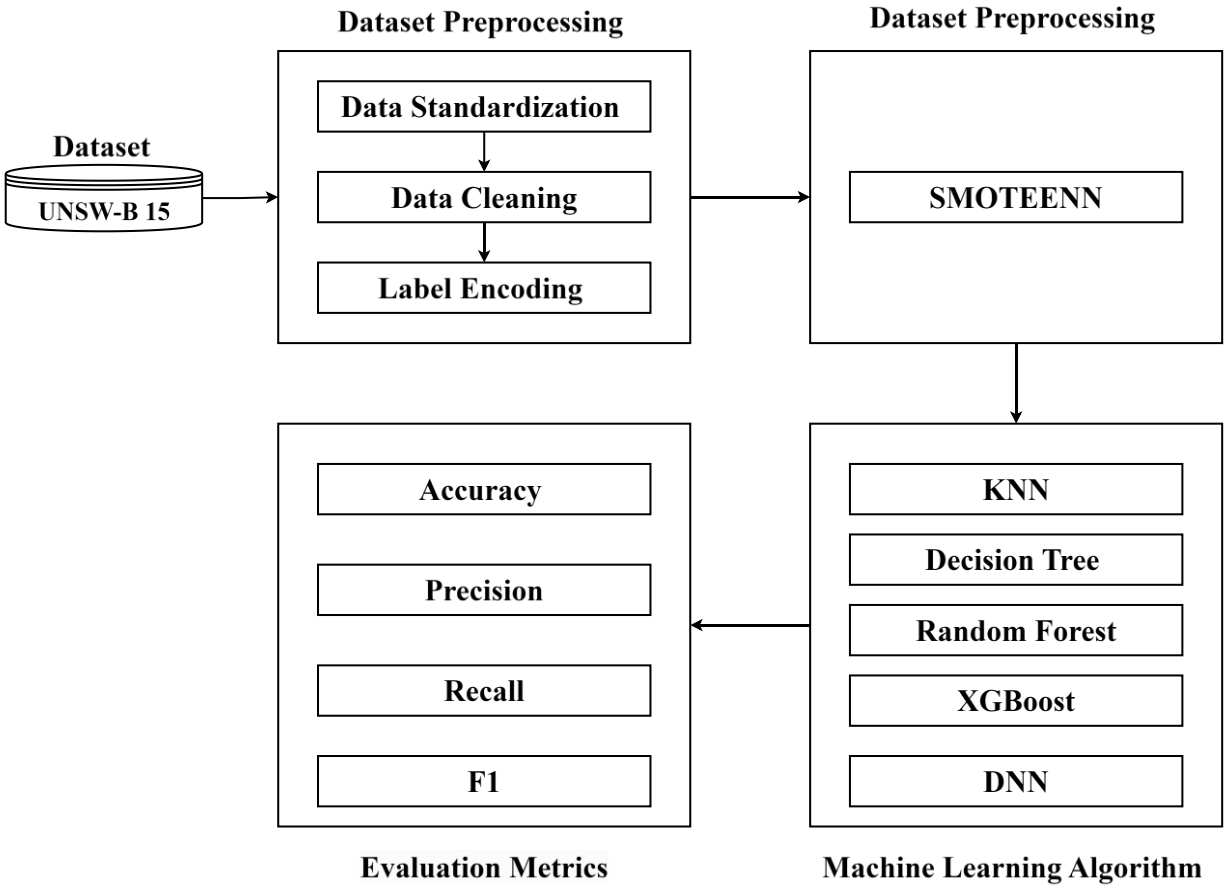

Figure 1. The flowchart of the suggested IDS

1.8 DATASET DESCRIPTION

The initial stage is to obtain a dataset to achieve attack categorization, which is the main aim of the suggested model. The best dataset for this purpose was thus found to be UNSW-NB15 [41]. The capability to generate actual traffic, the existence of real testbeds, and labelled data availability made this dataset significant for this work. The Cyber Range Lab at ACCS produced the dataset, which includes both attack and normal traffic [17]. To create this dataset, two servers were used in hybrid mode; the purpose of one server was to generate regular traffic, and the other was to generate attack traffic. Additionally, the dataset was generated using the IXIA Perfect Storm program. Data was gathered and stored in the TCPdump format. Features extraction was done using the Argus and Bro-IDS tools hosted on the Linux Ubuntu 14.0.4 OS. The dataset was divided into ten groups, nine of which were associated with attack traffic and one of which was referred to as normal. Figure 2 illustrates the number of packets in each category.

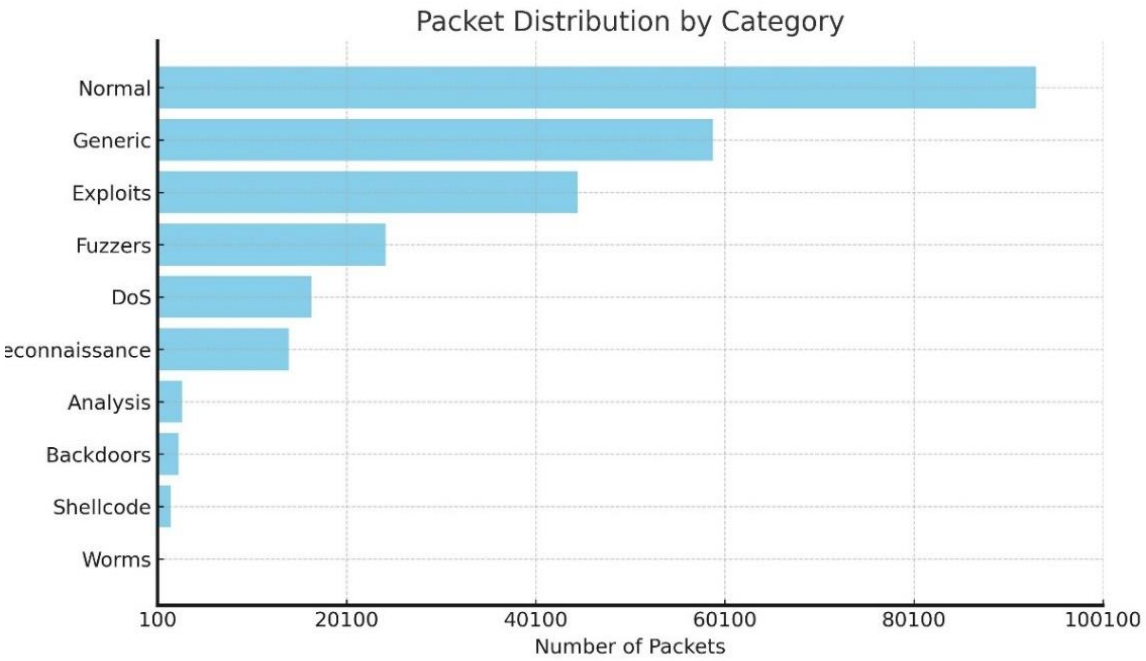

Figure 2. Attack categories in the utilized dataset and the associated number of packets

The employed dataset contains 257,673 packets, of which 93,000 are regular packets, and the remaining number are different attack packets (see Figure 3a, b, and c) for the detailed distributed categories of attacks in the utilized dataset.

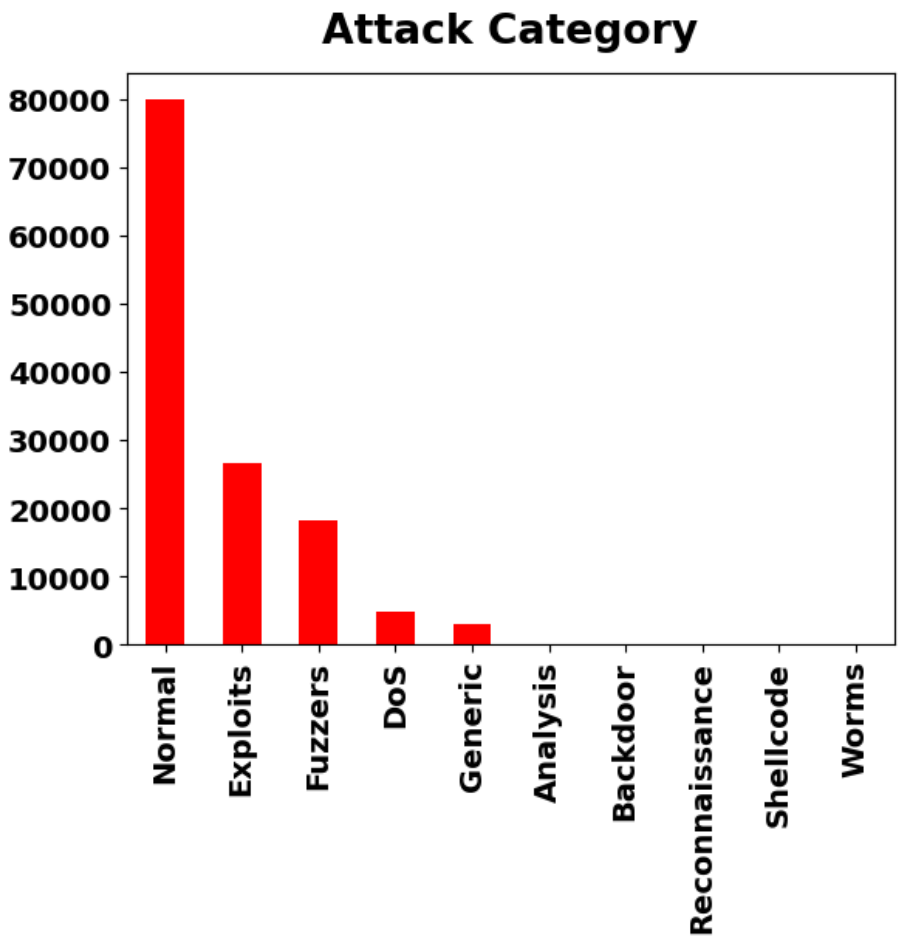

Figure 3 (a). Types of attack in the utilized dataset

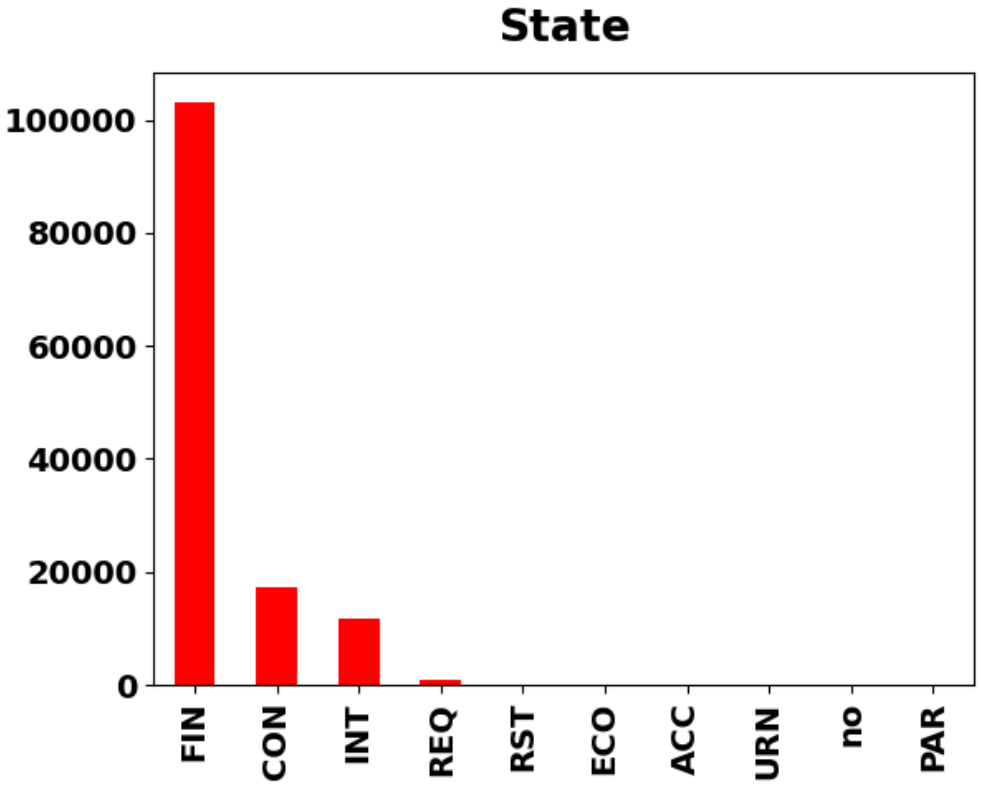

Figure 3 (b). States of the attack state in the utilized dataset

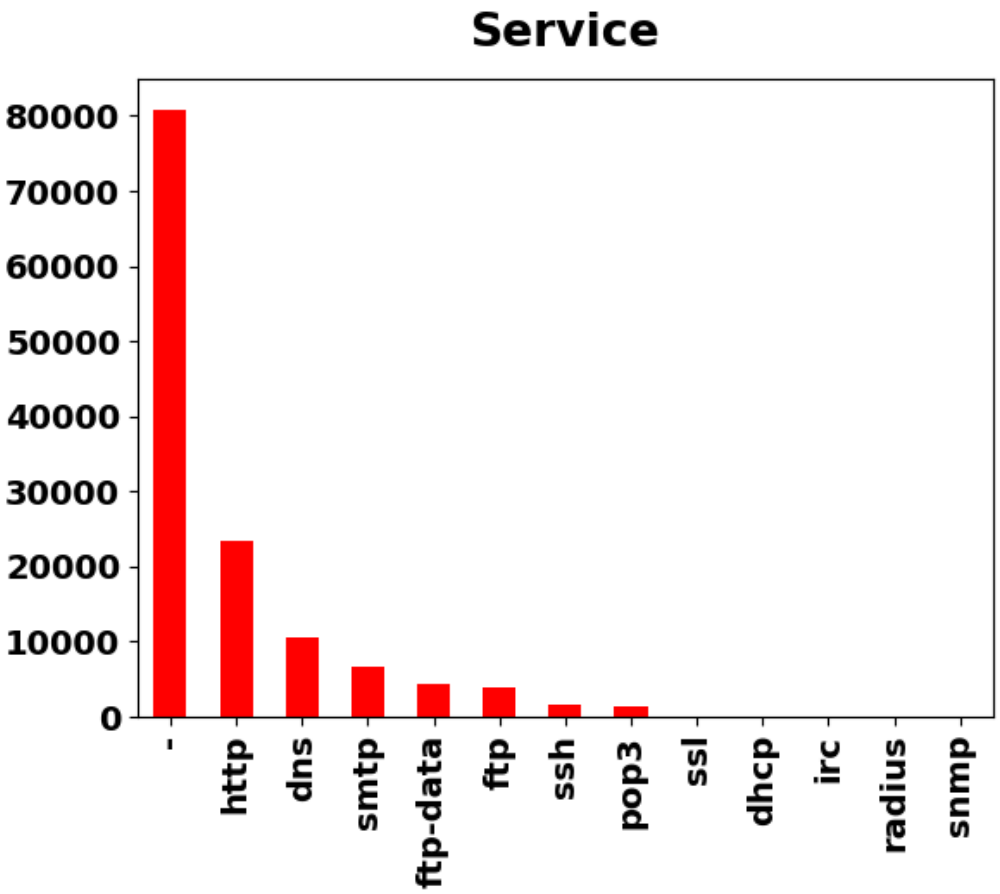

Figure 3. (c). The utilized services and the attacks in the utilized dataset

1.9 DATA PRE-PROCESSING

This crucial step is frequently performed in ML models since it guarantees that the data is in an appropriate format for feeding into the five ML techniques that are being used. Before moving on to the over-sampling stage, the pre-processing stage involved data standardization, cleaning, and encoding; Table 1 provides the details of the normal/attack type class.

Table 1. Features of the utilized dataset

Attack Type	Data Type
Attack - Dur	float32
Attack - Rate	
Attack - Sload	
Attack - Dload	
Attack - sinpkt	
Attack - dinpkt	
Attack - Sjit	
Attack - Djit	
Attack - Tcprtt	
Attack - synack	
Attack - ackdat	
Attack - Spkts	int16
Attack - Dpkts	
Attack - Sloss	

Attack - Dloss	
Attack - smean	
Attack - dmean	
Attack - trans_depth	
Attack - Swin	
Attack - Dwin	
Attack - sbytes	**int32**
Attack - dbytes	
Attack - response_body_len	
Attack – ct – dpor - ltm	**int8**
Attack – ct - spor - ltm	
Attack – is -ftp - login	
Attack – ct – ftp - cmd	
Attack - ct - http - mthd	
Attack – is - ports	
Attack - Stcpb	**int64**
Attack - Dtcpb	
Attack - State	**object**
Attack - Proto	
Attack - Service	**category**

Label encoding was used to transform the category characteristics into integer values once the superfluous label columns were eliminated. As a result, the variables "proto," "service," "state," and "attack_cat" were changed to 133, 13, 11, and 10, respectively. The min-max normalization approach [49] was used to convert the values of numeric columns with disparate feature ranges into a comparable scale, thereby normalizing the data. The Min-max normalization process is shown in Equation (8).

$$Xnew = ,,X - Xmin.-,Xmax - Xmin.. \quad (8)$$

Where

Xnew = the new normalized value,

Xmin = the least value in the X column,

Xmax = the optimal value in the X column.

To prevent significant variations in the convergence issue and to ensure that all of the feature's scale uniformly, the values of the X column are mapped between 0 and 1. Despite the fact that the dataset was rescaled following these procedures, this study only took into account five classes, with Normal, Generic, Exploits, and Fuzzers making up the majority.

1.10 DATA OVER-SAMPLING UTILIZING SMOTE-ENN.

The dataset was unbalanced since there were varying numbers of instances in each of the classes; this necessitates oversampling for the minor classes. In this work, the data were over-sampled using the SMOTE-ENN, which is a hybrid technique that merged the Edited Nearest Neighbors (ENN) under-sampling technique with the SMOTE over-sampling technique [43]. As a result, the minority class is over-sampled, and the majority classes are under-sampled by eliminating noise. Application of SMOTEENN to the dataset altered the number of instances in the classes, as seen in Figure 4.

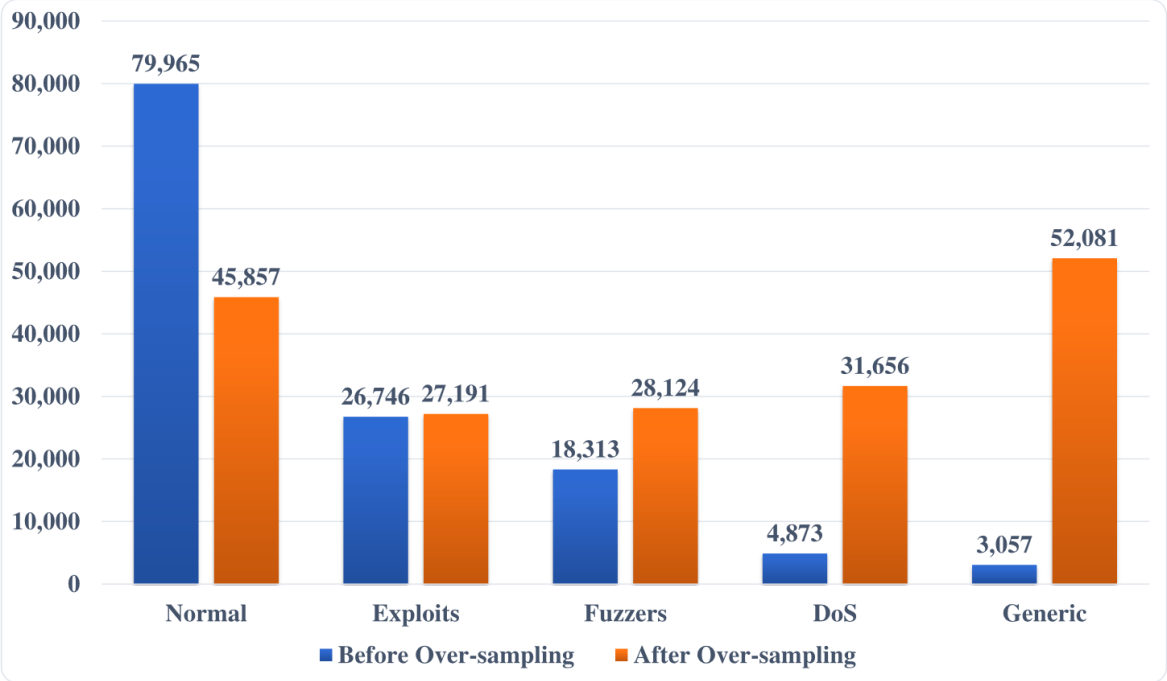

Figure 4. Class instances before/after over-sampling

For network traffic data to be accurately classified, the oversampling phase ensured that each class label had an equal number of instances in the dataset.

Results and Analysis

This section summarizes the evaluation results of the developed anomaly-based ML-based IDS for IoT. The results are reported based on actual experimental implementation, including all relevant figures, tables, equations, and visual representations. This section contains information on data processing, model application, and metric-based performance comparison. To assist the intrusion detection model design and validation process, a background analysis was performed alongside several relevant empirical studies. The objective of the analysis was to find similarities in approaches, datasets, and anomaly prediction models in the modern advanced systems based on a robust set of performance metrics. Out of the electronic databases, 157 records were retrieved and selected as part of the project selection process. After the removal of duplicates and title and abstract screening, 73 studies were selected for full analysis. Out of these, 28 studies were selected to establish a comparative benchmark for the proposed model based on the empirical implementation, application of ML algorithms, and the articulated evaluation metrics. The comparative model's performance was used as a benchmark in assessing the model's effectiveness. Below is a summary of the study selection:

i. Records initially identified: 157
ii. After duplicate removal: 143
iii. Screened for relevance: 143
iv. Full texts assessed: 73
v. Excluded (non-empirical or lacking performance results): 45
vi. Studies retained for comparison: 28
vii. Plus: 1 (the proposed model introduced in this study)

1.11 Characteristics of Benchmark Studies

The chosen benchmark studies for this review were diverse with respect to their ML techniques, datasets used, and evaluation methods. Common datasets used include UNSW-NB15, CICIDS2017/2018, KDD Cup 99, and NSL-KDD. The researchers studied a number of ML techniques, including DL models like CNN, DNN, and LSTM, and ensemble models such as Random Forest, XGBoost, and Gradient Boosted Trees. The majority of the studies used the normalization, label encoding, and SMOTE technique for data balancing, which is a standard pipeline for IDS.

1.12 Summary of Performance

Each ML algorithm (DT, DNN, RF, XGBoost, and KNN) was assessed using standard metrics, which are confusion matrices, precision and recall, accuracy, and the F1-

score. Graphs illustrating training/validation accuracy and loss curves (Figure 5 (A and B), confusion matrices (Figures 6, 8, 10, 12, 14), and performance bar charts (Figures 7, 9, 11, 13, 15) are preserved. These results clearly demonstrate that XGBoost consistently outperformed others with an accuracy of 96.37 %.

1.13 Comparative Evaluation

Table 3 illustrates the outcome of the benchmarking exercises, which portrays the performance of the proposed IDS model relative to the rest of the benchmarks. The proposed hybrid approach saw considerable gains in performance relative to the existing models (GAN + DNN = 91 %, CNN-1D = 89.8 %); this confirms the model's capability in generalization as well as its superiority.

1.14 Observed Trends and Findings

It was observed that hybrid models and ensemble learning techniques tend to achieve higher accuracy than single classifiers. Techniques such as SMOTEENN and Min-Max normalization contributed positively to performance. Models that were trained with properly balanced datasets and multiple attack types showed better detection capability. These findings are consistent across the studies included and further validated by the experimental outcomes of the proposed model.

The suggested IDS was assessed using each of the five selected machine learning techniques. For each algorithm, the CM and related performance metrics are shown. Figure 6 (a and b) displays the training and validation accuracy as well as the loss curves; the accuracy grows steadily with the number of epochs, while the training loss reduces as the number of epochs increases (> 20 epochs).

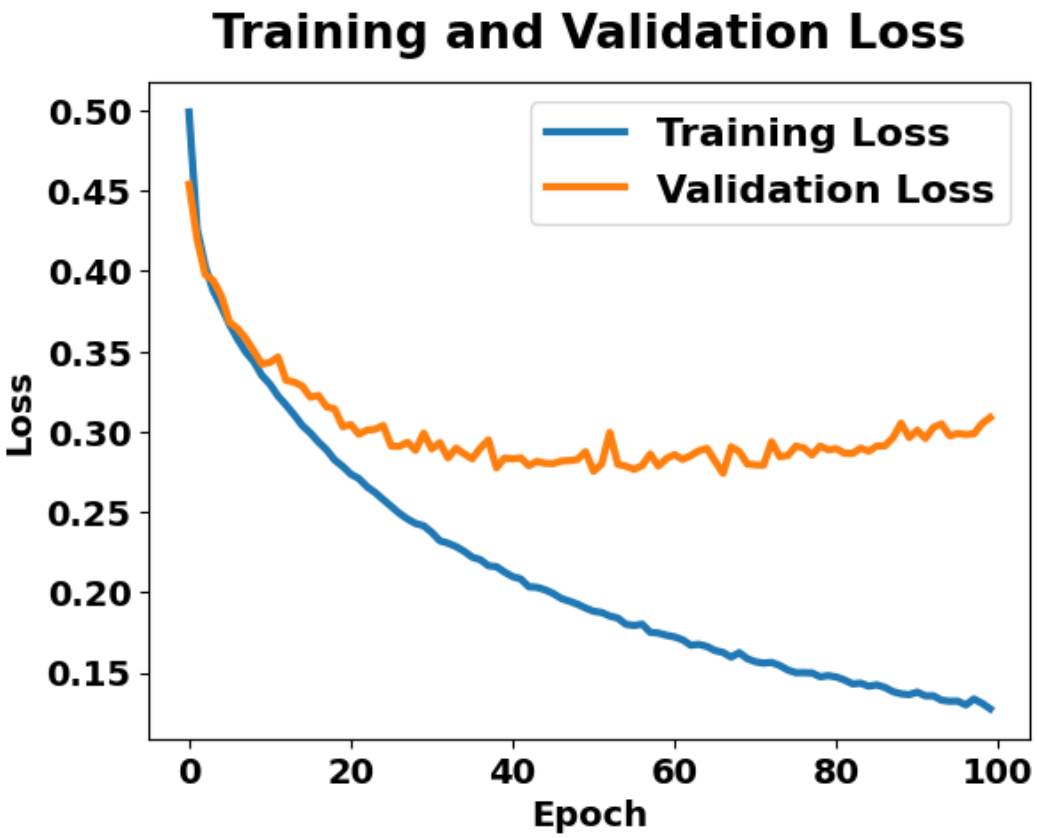

Figure 5 (A). Model's training and validation accuracy and loss

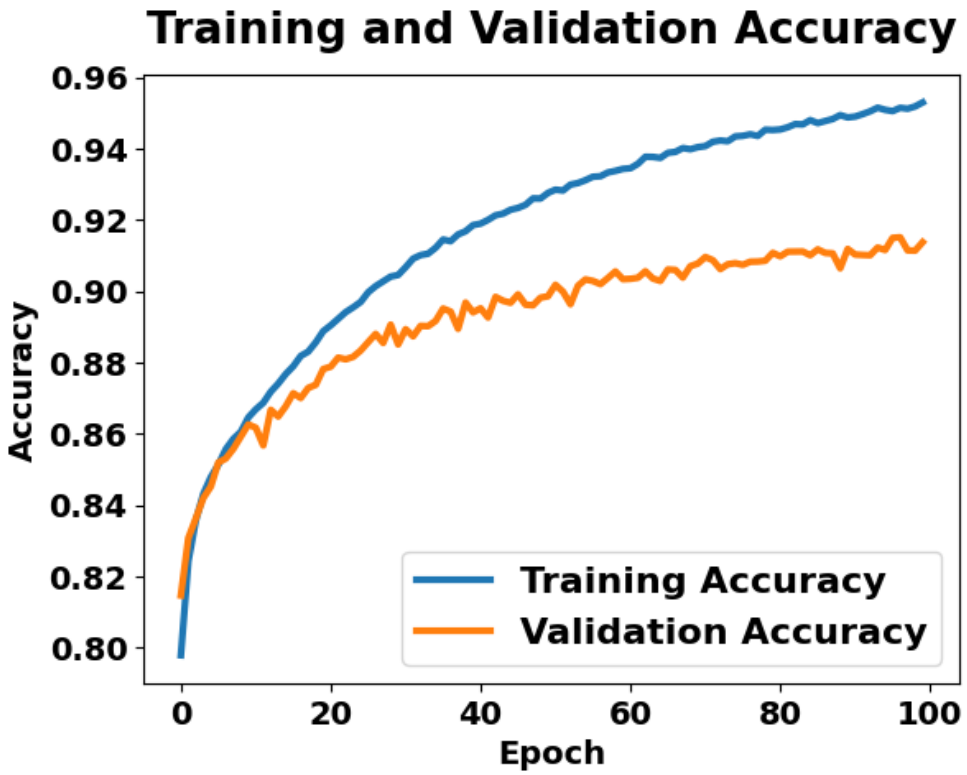

Figure 5. (B). Training and validation accuracy of the model

The XGBoost algorithm successfully classified 96.37 percent of both regular and attack traffic, as illustrated in Figure 6's confusion matrix. In addition, out of the five algorithms tested, the XGBoost method had the best overall accuracy at 96.37 %. Referring to Figure 7, the XGBoost values for recall, precision, and F1 score were 0.9639, 0.9637, and 0.9637, respectively.

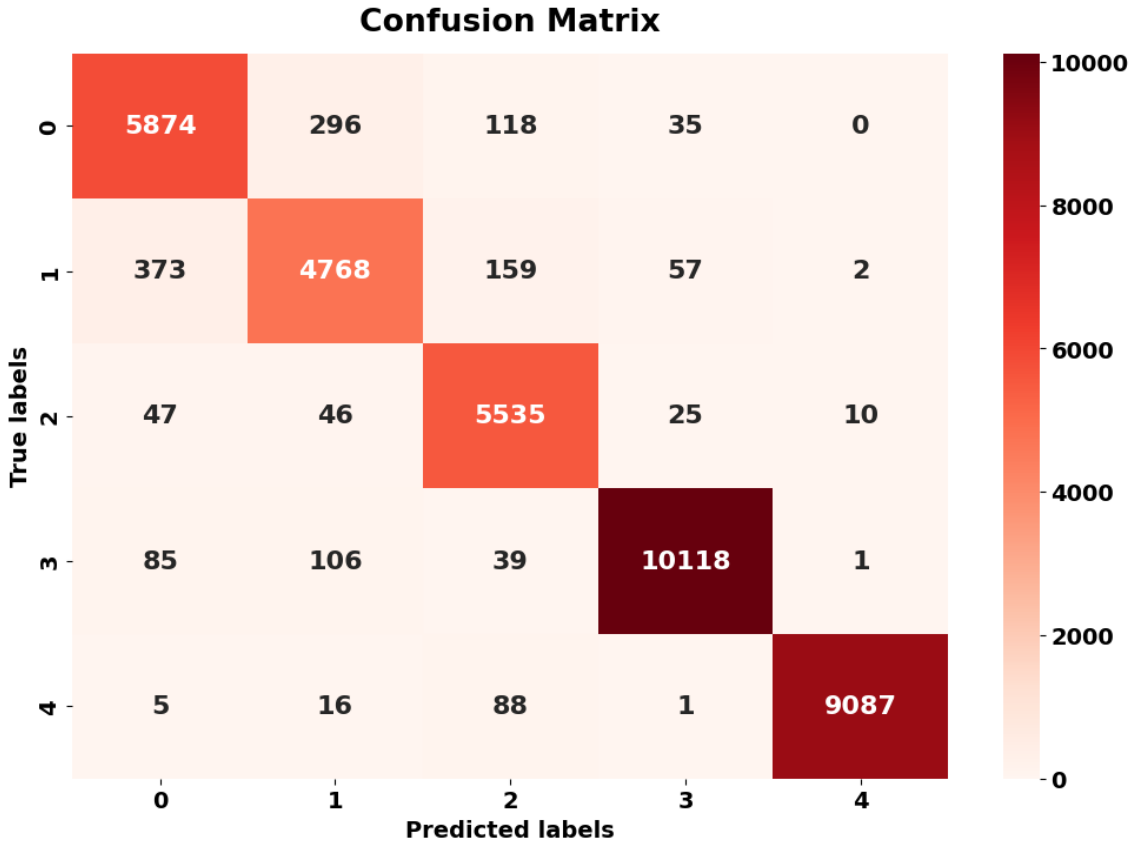

Figure 6. The CM for XGBoost algorithm

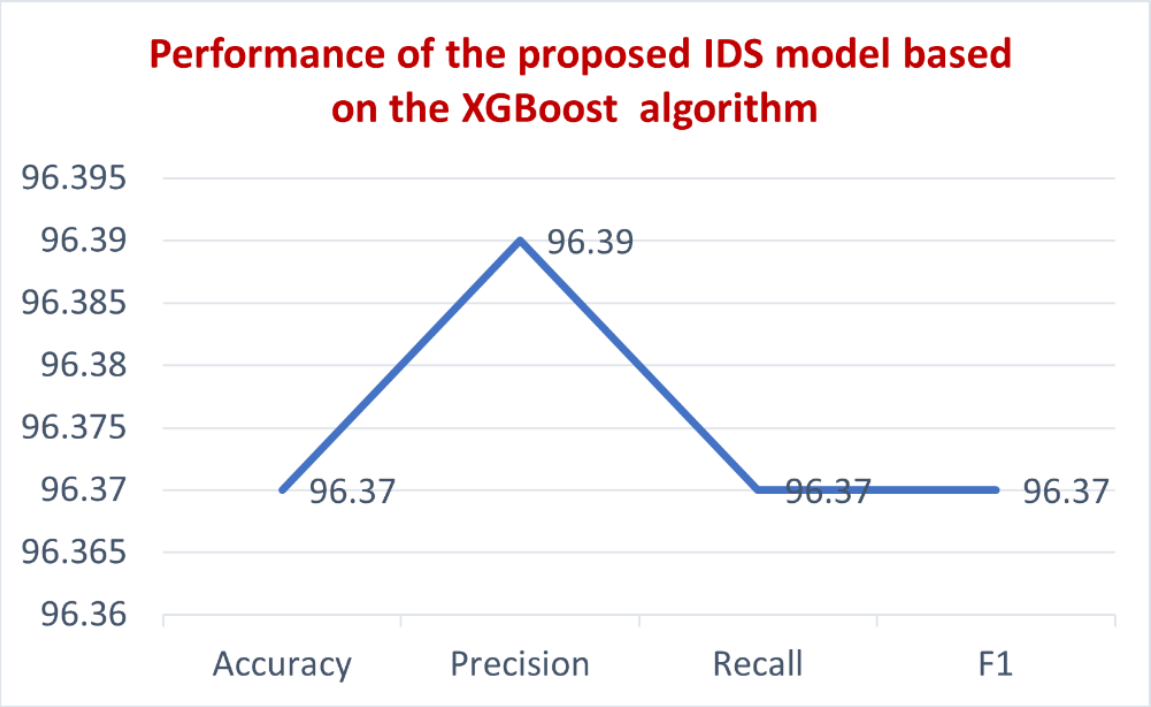

Figure 7. Performance of the suggested IDS model based on the XGBoost algorithm

Figure 8, which shows the decision tree's confusion matrix, displays the outcomes corresponding to that decision tree. The figure demonstrates that DT can accurately classify both attack and normal traffic with an accuracy of 90.67 % each. Furthermore, DT achieved an overall accuracy score of 90.67 %, with a precision of 0.9065, a recall of 0.9067, and an F1-score of 0.9064 (for clarification, see Figure 9).

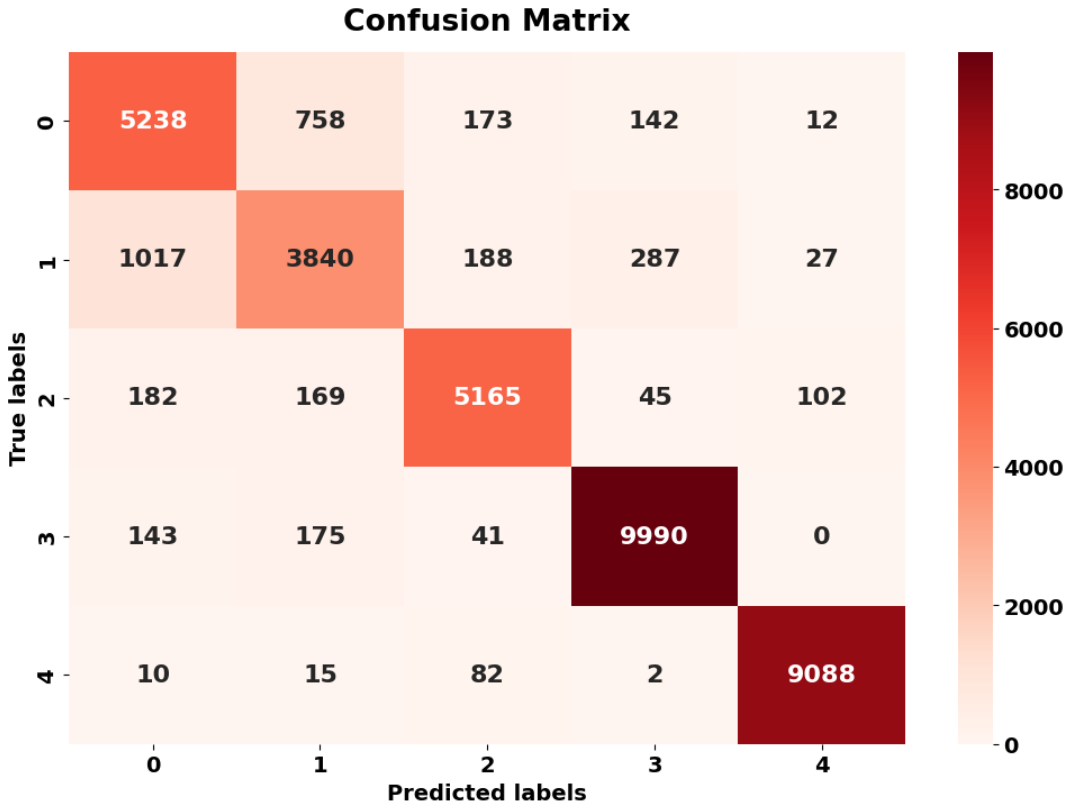

Figure 8. The CM of the DT algorithm

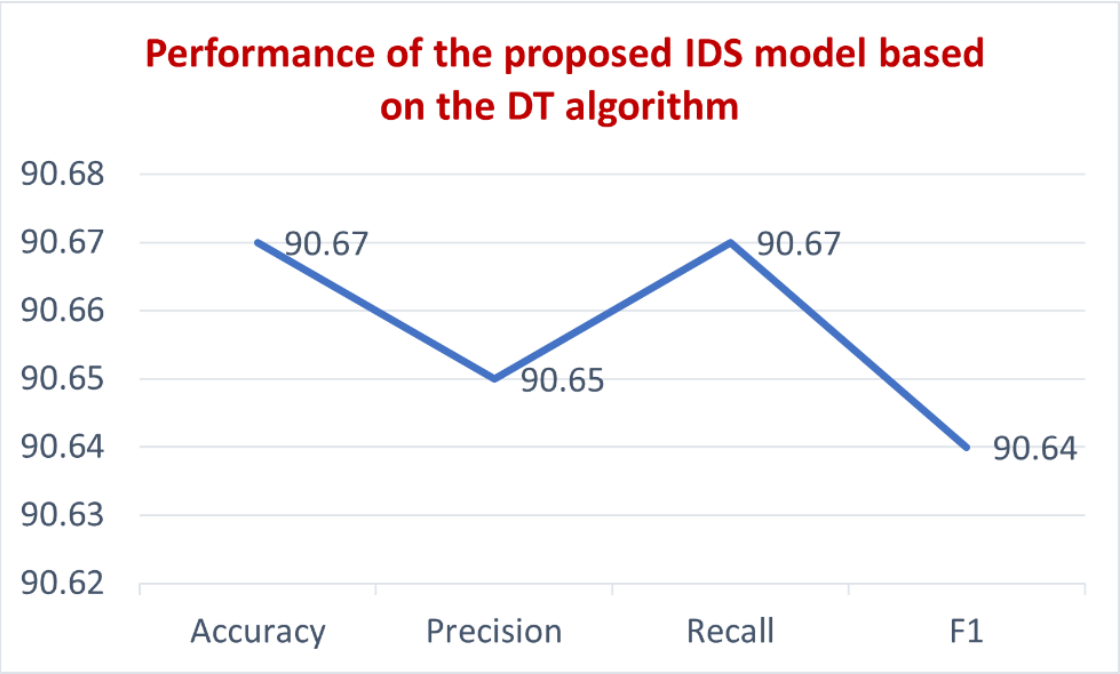

Figure 9. Achievement of the suggested IDS model using DT algorithm

For the K-NN, the CM is shown in Figure 10, which demonstrates that KNN accurately detects 90.29 % of attack traffic and 90.29 % of normal traffic (overall accuracy = 90.29 %). The precision value was 0.9100 higher than the recall 0.9029, while the F1-score was 0.9029 (Refer to Figure 11).

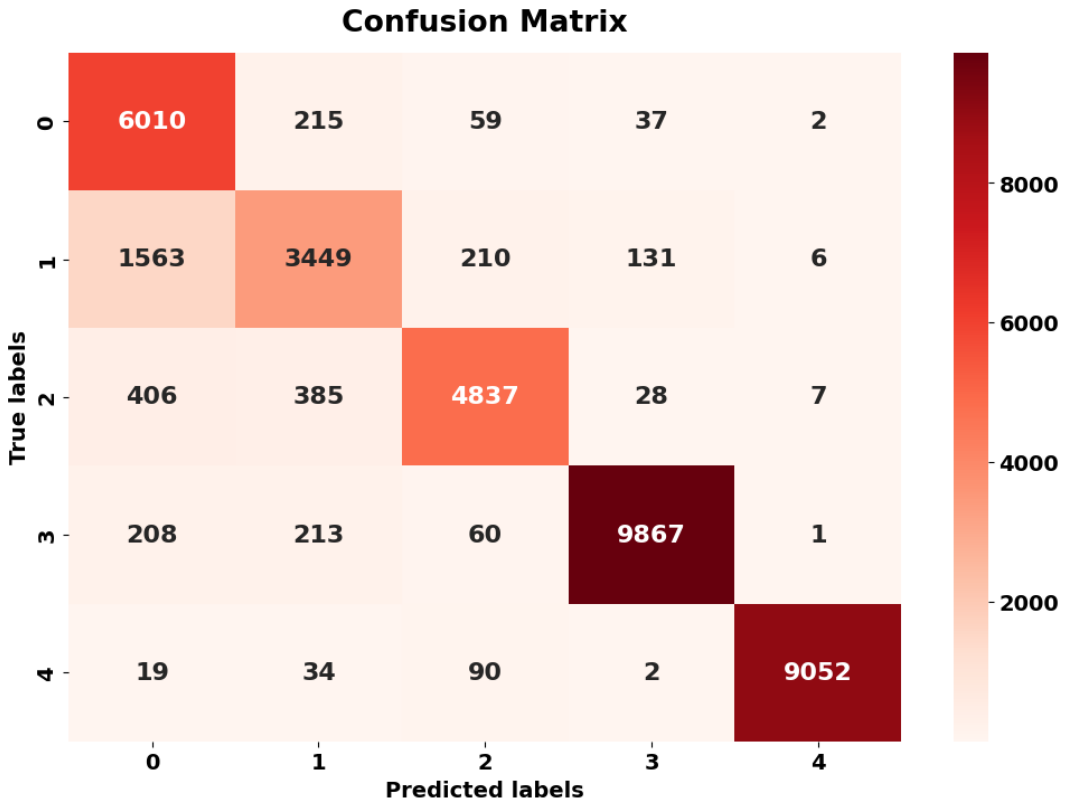

Figure 10. The CM for the KNN algorithm

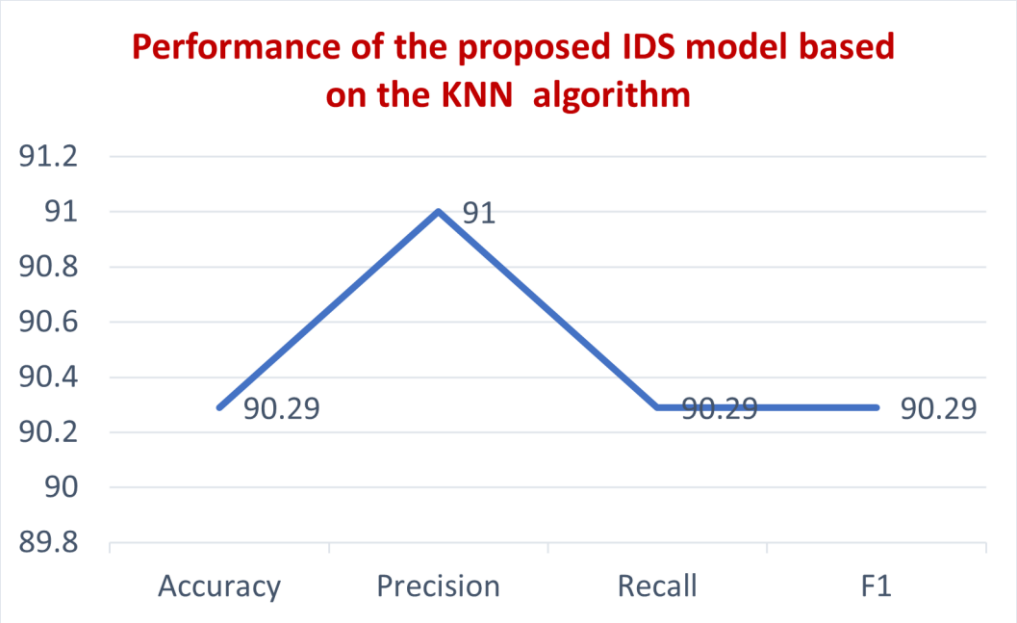

Figure 11. Achievement of the suggested IDS model using the KNN algorithm

For the RF, the CM is shown in Figure 12; the model recorded 94.52 % accuracy in normal and attack traffic classification. The accuracy attained by RF was second only to XGBoost, at 94.52 % while the precision, recall, and F1-score were 0.9456, 0.9452, and 0.9452, respectively (see Figure 13).

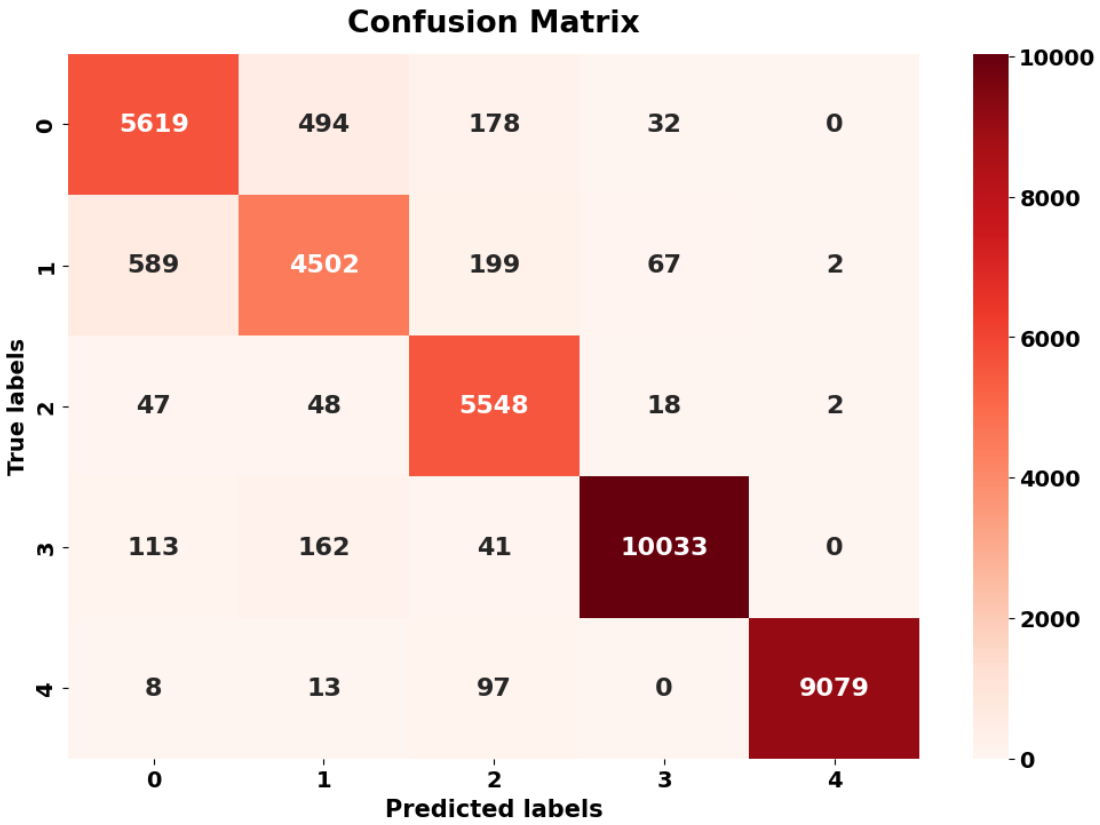

Figure 12. The CM of the RF algorithm

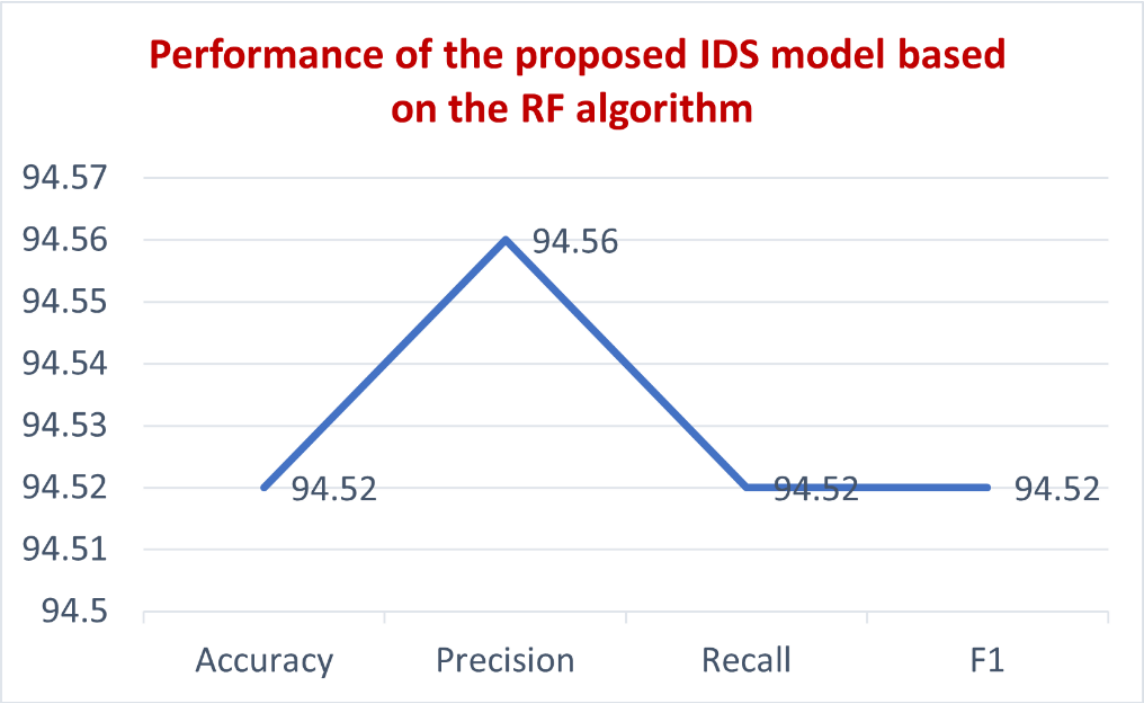

Figure 13. Achievement of the suggested IDS model using the RF algorithm

Lastly, Figure 14 shows the DNN's confusion matrix, which shows that 91 % of both normal and malicious traffic were correctly identified by the algorithm. Additionally, 91 % accuracy was reached, and the F1 score was 0.91; the precision and recall scores were both 0.91 (see Figure 15).

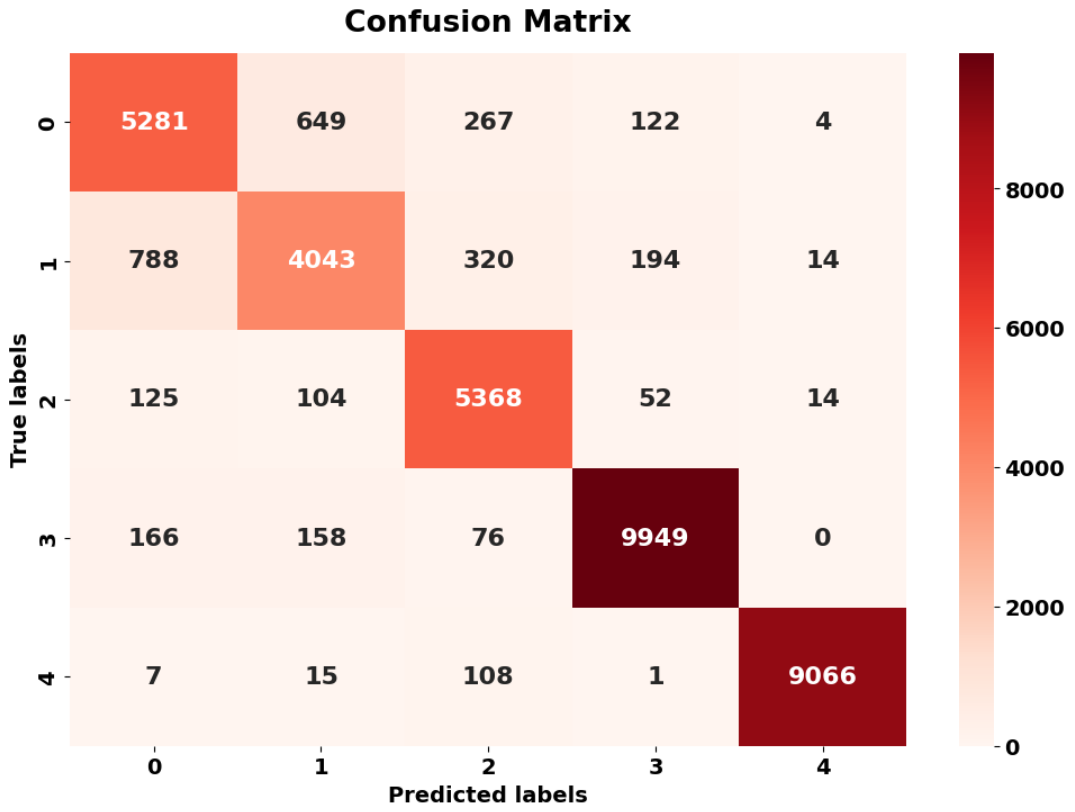

Figure 14. The CM of the DNN algorithm

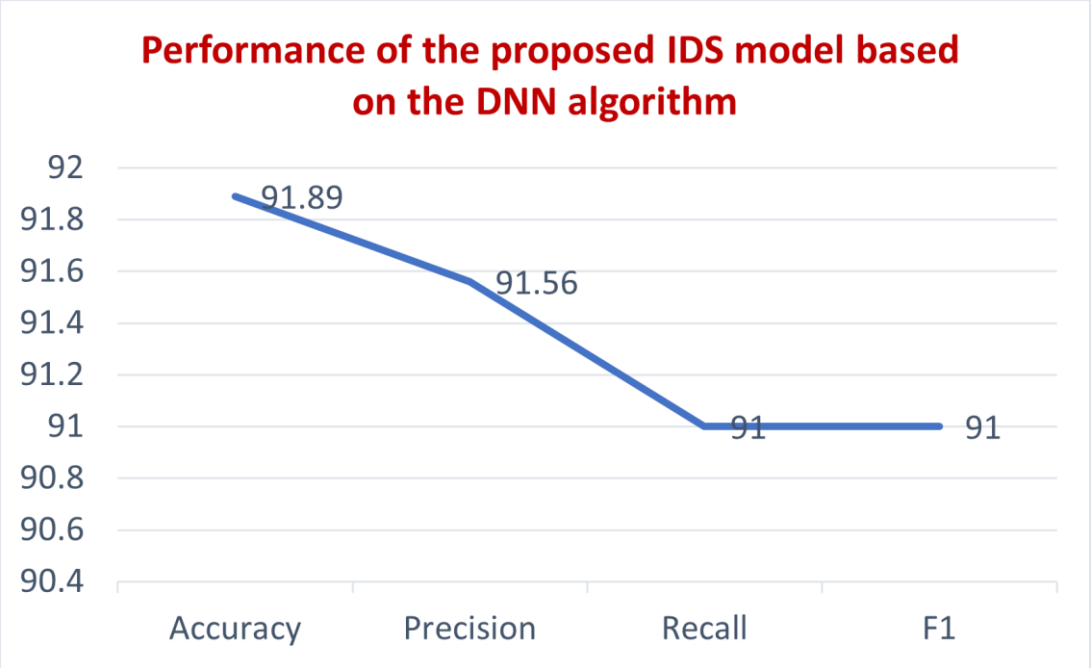

Figure 15. Achievement of the suggested IDS model using the DNN algorithm

The comparison of the performances of the five distinct algorithms is illustrated in Figure 16, which clearly indicates that the XGBoost classifier exhibits the best scores for accuracy, F1-score, precision, and recall, followed by the RF. The five classifiers performed well in network assault identification within the IoT networks.

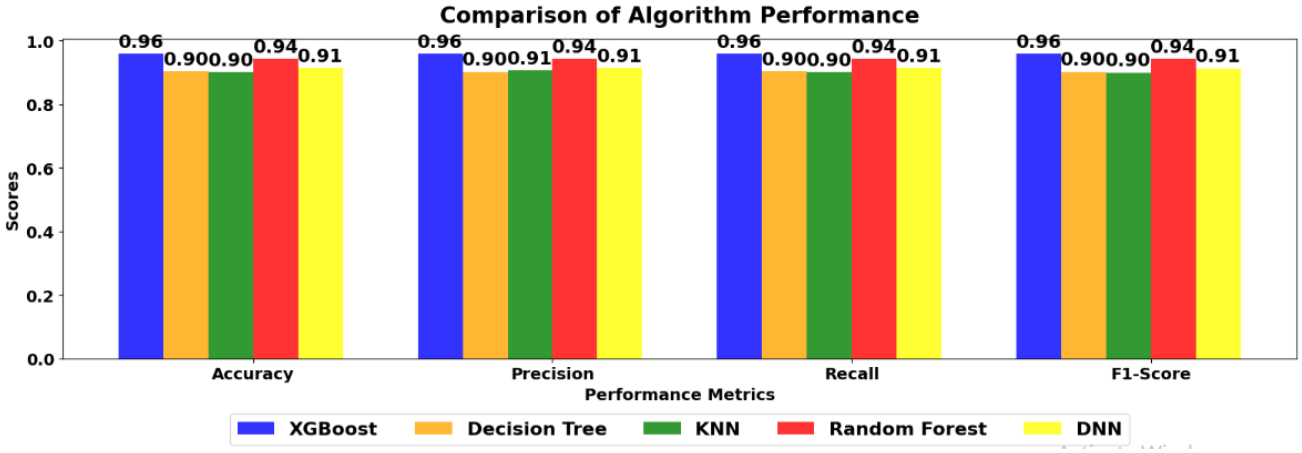

Figure 16. Achievement Comparison of all the studied algorithms

Research Benchmarking

It was determined that five distinct machine learning algorithms were good choices for building an IDS to protect IoT networks. Among the criteria utilized to assess the five algorithms was accuracy; XGBoost was the most accurate algorithm, with an accuracy of 0.9637. A DNN approach has earlier been used [26] for building IDS within the IoT network following model training on the original samples; the study also used a GAN to augment DOS attacks. The DNN model initially achieved 84.4 % accuracy in identifying both normal and attack classes; after the installation of GAN, the model attained an accuracy of 91 %. A separate study conducted by employed various ML techniques for both binary and multi-class classification of occurrences. The authors reported an accuracy of 89.8 % using their CNN-1D algorithm. Table 3 displays a comparison of the suggested IDS model with the existing methods. The suggested IDS model has achieved the targeted objectives of this study by performing better than most of the existing models in the applied instances.

Table 3. Research benchmarking

Reference	Classifier	Accuracy (%)
[26]	GAN and DNN	0.910
[50]	CNN and 1D	0.898
	GBT	0.931
Results achieved	XGBoost	0.964
	RF	0.945
	KNN	0.912
	DT	0.893
	DNN	0.910

This study indicates that machine learning-based anomaly detection methods may hold promise for IoT intrusion detection, but still have limitations. According to the experimental results, the accuracy of ML algorithms reveals that ensemble and boosting techniques (XGBoost and Random Forest) offer high accuracy (the accuracy of XGBoost is 96.37 %). The viability of ML in IDS can be supported when appropriate structured preprocessing and feature engineering are used. Moreover, there are obstacles despite these positive results; when models are used in traffic outside of benchmarks, their performance can be poor; therefore, there is a need for adaptable and robust models. The findings show that the shift towards ensemble learning is useful, but at the same time, computational costs and interpretability need to be addressed, especially in resource-constrained IoT environments.

Future Research Directions

This research identified key methodological gaps in the literature, which informed the experimentation framework. The inconsistencies in the evaluation metrics, non-standardization of datasets, and lack of external validation are some of these gaps. This study eliminates these shortcomings by proposing a single framework using the UNSW-NB15 dataset, uniform preprocessing steps, and performance evaluation through cross-validation. A key constraint in the wider field is the absence of realistic, up-to-date, and open datasets relating to IoT networks. The UNSW-NB15 dataset is useful for experimentation, but it may not encompass all contemporary attack vectors. In the future, datasets need to be more diverse and adaptable for real-time applications for a more dynamic IoT ecosystem. This study highlights the importance of XAI methods like SHAP and LIME to bolster the adequacy and clarity of the alerts; this is critical for IoT scenarios like healthcare and industrial IoT, where the decisions made by the IDS models need to be substantiated and rationalized.

Conclusion

This elaborative experimental study aims to build and assess an improved anomaly-based intrusion detection mechanism model in IoT environments through machine learning methods. This study focused on the performance of machine learning algorithms in IoT settings using the UNSW-NB15 dataset. Of all the algorithms, the model that showed the most accuracy was XGBoost, with a value of 96.37, followed by the Random Forest, thus reinforcing the fact that ensemble and boosting methods are used with greater effectiveness in IoT security. To validate the model's robustness and reproducibility, a controlled dataset was implemented; the evaluation parameters were controlled, and an accuracy assessment was performed independently. These findings can confidently support the integration of these models in intrusion detection systems and can also serve as a basis for future development. This study aims to not only model comparisons but also offer a practical, authenticated model, which can be used in smart environments that face anomaly-based threats.

Furthermore, the existing issues of lack of reliable literature, absence of standard evaluations, and the

outdated dataset used can hamper progress towards real-world implementation of these models; hence, this work, through implementing more realistic datasets, embracing reproducible research, and integrating interpretability standards in IDS design, seeks to address these problems in the field.

In conclusion, applying ML techniques to ensure IoT cybersecurity is possible, but extensive evaluations are needed to satisfy the fundamental requirements. This research will be of immense value to both scholars and professionals aiming to construct reliable, interpretable, and effective IDS for the evolving world of IoT.

References

1. Bajaber, F., *Adaptive Density Control Based on Random Sensing Range for Energy Efficiency in IoT Sensor Networks.* Pertanika Journal of Science Technology, 2023. **31**(4).
2. Javadi, A., et al., *Secure and Efficient Lightweight Authentication Protocol (SELAP) for multi-sector IoT applications.* Internet of Things, 2025: p. 101499.
3. Singh, S., et al., *Analysis of Soil Viability Monitoring System for In-House Plantation Growth Using an Internet of Things Approach.* Pertanika Journal of Science Technology, 2024. **32**(6).
4. Altulaihan, E., M.A. Almaiah, and A. Aljughaiman, *Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms.* Sensors, 2024. **24**(2): p. 713.
5. Al-Majdi, K., et al., *MLCM: An efficient image encryption technique for IoT application based on multi-layer chaotic maps.* International Journal of Nonlinear Analysis Applications, 2022. **13**(2): p. 1591-1615.
6. Li, C., et al., *A review of IoT applications in healthcare.* Neurocomputing, 2024. **565**: p. 127017.
7. Hmissi, F. and S. Ouni. *A review of application protocol enhancements for Internet of things*. in *UBICOMM 2021: The Fifteenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. 2021.
8. Mishra, N. and S. Pandya, *Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review.* IEEE Access, 2021. **9**: p. 59353-59377.
9. Hajiheidari, S., et al., *Intrusion detection systems in the Internet of things: A comprehensive investigation.* Computer Networks, 2019. **160**: p. 165-191.
10. Yaseen, N.A., A.A.-A. Hadad, and M.S. Taha. *An anomaly detection model using principal component analysis technique for medical wireless sensor networks*. in *2021 International Conference on Data Science and Its Applications (ICoDSA)*. 2021. IEEE.
11. Heidari, A. and M.A. Jabraeil Jamali, *Internet of Things intrusion detection systems: a comprehensive review and future directions.* Cluster Computing, 2023. **26**(6): p. 3753-3780.
12. Asharf, J., et al., *A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions.* Electronics, 2020. **9**(7): p. 1177.
13. Li, J., et al., *Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning.* Journal of Big Datas, 2024. **11**(1): p. 36.
14. Mendialdua, I., et al., *Classifier Subset Selection to construct multi-classifiers by means of estimation of distribution algorithms.* Neurocomputing, 2015. **157**: p. 46-60.
15. Choobdar, P., M. Naderan, and M. Naderan, *Detection and multi-class classification of intrusion in software defined networks using stacked auto-encoders and CICIDS2017 dataset.* Wireless Personal Communications, 2022. **123**(1): p. 437-471.
16. Hadad, A.A.-A., et al., *A robust color image watermarking scheme based on discrete wavelet transform domain and discrete slantlet transform technique.* Ingenierie des Systemes d'Information, 2022. **27**(2): p. 313.
17. Rahman, R.U. and D.S. Tomar, *Security attacks on wireless networks and their detection techniques*, in *Emerging wireless communication and network technologies: principle, paradigm and performance*. 2018, Springer. p. 241-270.
18. Diro, A., et al., *Anomaly detection for space information networks: A survey of challenges, techniques, and future directions.* Computers Security, 2024. **139**: p. 103705.
19. Alsoufi, M.A., et al., *Anomaly-based intrusion detection systems in iot using deep learning: A*

systematic literature review. Applied sciences, 2021. **11**(18): p. 8383.

20. Martins, I., et al., *Host-based IDS: A review and open issues of an anomaly detection system in IoT.* Future Generation Computer Systems, 2022. **133**: p. 95-113.
21. Saba, T., et al., *Anomaly-based intrusion detection system for IoT networks through deep learning model.* Computers Electrical Engineering, 2022. **99**: p. 107810.
22. Natarajan, Y., et al., *Enhancing building energy efficiency with iot-driven hybrid deep learning models for accurate energy consumption prediction.* Sustainability, 2024. **16**(5): p. 1925.
23. Kasongo, S.M. and Y. Sun, *A deep learning method with filter based feature engineering for wireless intrusion detection system.* IEEE access, 2019. **7**: p. 38597-38607.
24. Bae, G., et al. *Autoencoder-based on anomaly detection with intrusion scoring for smart factory environments*. in *International conference on parallel and distributed computing: Applications and technologies*. 2018. Springer.
25. Ferrag, M.A., et al., *Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study.* Journal of Information Security Applications, 2020. **50**: p. 102419.
26. Sharma, B., et al., *Anomaly based network intrusion detection for IoT attacks using deep learning technique.* Computers Electrical Engineering, 2023. **107**: p. 108626.
27. Thamilarasu, G. and S. Chawla, *Towards deep-learning-driven intrusion detection for the internet of things.* Sensors, 2019. **19**(9): p. 1977.
28. Ge, M., et al., *Towards a deep learning-driven intrusion detection approach for Internet of Things.* Computer Networks, 2021. **186**: p. 107784.
29. Nagisetty, A. and G.P. Gupta. *Framework for detection of malicious activities in IoT networks using keras deep learning library*. in *2019 3rd international conference on computing methodologies and communication (ICCMC)*. 2019. IEEE.
30. Sharma, B., et al., *Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach.* Expert Systems with Applications, 2024. **238**: p. 121751.
31. Qiu, H., et al., *Adversarial attacks against network intrusion detection in IoT systems.* IEEE Internet of Things Journal, 2020. **8**(13): p. 10327-10335.
32. Zhao, F., et al., *Application of deep learning-based intrusion detection system (IDS) in network anomaly traffic detection.* 2024.
33. Sun, P., et al., *DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System.* Security communication networks, 2020. **2020**(1): p. 8890306.
34. Aswad, F.M., et al., *Deep learning in distributed denial-of-service attacks detection method for Internet of Things networks.* Journal of Intelligent Systems, 2023. **32**(1): p. 20220155.
35. Osa, E., P.E. Orukpe, and U. Iruansi, *Design and implementation of a deep neural network approach for intrusion detection systems.* e-Prime-Advances in Electrical Engineering, Electronics Energy, 2024. **7**: p. 100434.
36. Fenanir, S., F. Semchedine, and A. Baadache, *A Machine Learning-Based Lightweight Intrusion Detection System for the Internet of Things.* Revue d'Intelligence Artificielle, 2019. **33**(3).
37. Almaiah, M.A., et al., *A novel hybrid trustworthy decentralized authentication and data preservation model for digital healthcare IoT based CPS.* Sensors, 2022. **22**(4): p. 1448.
38. Siam, A.I., et al., *Secure health monitoring communication systems based on IoT and cloud computing for medical emergency applications.* Computational Intelligence Neuroscience, 2021. **2021**(1): p. 8016525.
39. Ali, A., et al., *An industrial IoT-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network.* Sensors, 2022. **22**(2): p. 572.
40. Al Hwaitat, A.K., et al., *Improved security particle swarm optimization (PSO) algorithm to detect radio jamming attacks in mobile networks.* International Journal of Advanced Computer Science Applications, 2020. **11**(4).
41. Disha, R.A. and S. Waheed, *Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted*

Random Forest (GIWRF) feature selection technique. Cybersecurity, 2022. **5**(1): p. 1.

42. Dhal, P. and C. Azad, *A comprehensive survey on feature selection in the various fields of machine learning.* Applied intelligence, 2022. **52**(4): p. 4543-4581.
43. Zhang, H., et al. *Research on network intrusion detection based on SMOTEENN and improved CatBoost algorithm*. in *Sixth International Conference on Computer Information Science and Application Technology (CISAT 2023)*. 2023. SPIE.
44. Meenal, R., et al., *Weather prediction using random forest machine learning model.* Indonesian Journal of Electrical Engineering Computer Science, 2021. **22**(2): p. 1208-1215.
45. Charbuty, B. and A. Abdulazeez, *Classification based on decision tree algorithm for machine learning.* Journal of applied science technology trends, 2021. **2**(01): p. 20-28.
46. Uddin, S., et al., *Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction.* Scientific Reports, 2022. **12**(1): p. 6256.
47. Li, J., et al., *Application of XGBoost algorithm in the optimization of pollutant concentration.* Atmospheric Research, 2022. **276**: p. 106238.
48. Hussain, H., P. Tamizharasan, and C. Rahul, *Design possibilities and challenges of DNN models: a review on the perspective of end devices.* Artificial Intelligence Review, 2022. **55**(7): p. 5109-5167.
49. Mazziotta, M. and A. Pareto, *Normalization methods for spatio-temporal analysis of environmental performance: Revisiting the Min–Max method.* Environmetrics, 2022. **33**(5): p. e2730.
50. Zhou, Y., et al. *Deep learning approach for cyberattack detection*. in *IEEE infocom 2018-ieee conference on computer communications workshops (INFOCOM WKSHPS)*. 2018. IEEE.