Asia-Pacific Journal of Science and Technology: Volume :30 Issue :06 Article ID :APST-30-06-12. Research Article

APST

Asia-Pacific Journal of Science and Technology

https://www.tci-thaijo.org/index.php/APST/index

Published by Research and Innovation Department,
Khon Kaen University, Thailand

An Intelligent IoT based anomaly detection using predictive analytics

Ranjith Jayaramu, Vergin Sarobin*, Shree Varshan, Briju Demel and Hridhay Vardhan

School of Computer Science & Engineering, Vellore Institute of Technology, Tamil Nadu, Chennai, India
*Corresponding author: verginraja.m@vit.ac.in

Received 22 March 2023
Revised 22 January 2024
Accepted 26 March 2024

Abstract

In this study, we analyze the effectiveness of combining various machine learning approaches in order to detect specific attack classes within the Internet of Things (IoT). Using the IoT23 dataset, the research examines distinct features associated with attack classes derived from the raw data. Multiple algorithms, such as Multi-layer Perceptron, Random Forest, Extreme Gradient Boosting (XGBoost), Decision Tree, K-Nearest Neighbor, Support Vector Machine, Logistic Regression, and Naive Bayes, were thoroughly trained and evaluated. Through the implementation of ensemble learning techniques, the study successfully achieved an elevated detection rate of attack classes and an overall accuracy improvement, maintaining a false alarm rate of up to 15%. The research highlights the importance of using ensemble learning methods to identify and categorize attacks in IoT domains, serving as a valuable resource for further research. The insights revealed in this study provide readers with a compelling reason to read it and should act as a catalyst for further research in similar directions.

Keywords: Anomaly detection, Classification model, IoT-23, Machine learning, Security

1. Introduction

In 2020, Shafique, K. [1] introduced the term Internet of Things (IoT) for the next generation smart system, referring to devices that can gather data from their environment using sensors and execute programming to enhance efficiency and quality of life. IoT devices communicate with each other, and due to their affordability, availability, and numerous applications in various domains like fitness, healthcare, home management, and more, the IoT market has been expanding rapidly, propelled by the fourth industrial revolution 4.0, according to the study by Dwivedi et al. [2].

Although the IoT provides numerous advantages, including enhanced efficiency and convenience, it also presents significant security and privacy risks. One crucial security concern with the IoT is that many devices lack proper security measures and can be easily hacked, enabling attackers to gain access to sensitive information such as personal data or login credentials, and even take control of the device remotely. Furthermore, due to limited processing power and memory, many IoT devices cannot run conventional security software, making it challenging to secure them.

Our research proposes using an intrusion detection system (IDS) to prevent anomaly attacks, which, coupled with machine learning, is capable of detecting anomalies across billions of IoT devices worldwide, as suggested by Xiao et al. [3]. Machine learning-based network intrusion detection has been a research topic for the past two decades. We evaluate the effectiveness of different machine learning classifiers, such as Multi-layer Perceptron (MLP), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR), and Naive Bayes (NB), for detecting anomalies in IoT data. The performance of each classifier is measured using well-known metrics like accuracy, specificity, sensitivity, false positive rate, precision, recall, and F1 Score. Furthermore, we employ random search to fine-tune the parameters of each classifier.

Our study employs statistical analysis to determine whether significant performance differences exist between the various classifiers. We also measure the average response time of the classifiers to evaluate their usability in

our IoT simulation. The performance of machine learning classifiers is assessed using IoT-23 data through repeated holdout and cross-fold validation techniques.

Ullah et al. [4] have proposed a technique for identifying corrupted devices by utilizing the traffic stream generated by IoT devices' temporal periodicity. They employed a Recurrent Neural Network (RNN) to detect deviations from the expected typical behavior of each specific device. Authentication-based methodologies have also been explored for the classification and management of identities in IoT systems. This approach was evaluated, which also introduced the authors' discussion on a survey of the enabling technologies of forthcoming IoT ecosystems by Mukhaini et al. [5]

Rathore et al. in [6] established the foundation for the Enhanced Spatial Fuzzy C-Means (ESFCM) approach, which is a distributed threat detection framework for the Internet of Things. Their fog-based threat detection method, which incorporates both the ESFCM and fog computing paradigms, outperformed existing schemes on various datasets. One significant advantage of the ESFCM technique is that it solves the problem of labeled data, resulting in high detection rates and excellent performance.

In 2019, Liang et al. [7] developed a PD model that employs K-means and perceptron models to identify undesired nodes and calculate IoT node values. Their proposed method addresses challenges by creating a structured multiple-mix-attack framework, streamlining the network path, and building an upgraded perceptron learning process to construct a PDE for increased detection accuracy. The simulation results demonstrate enhanced performance and increased accuracy in detecting node malware using the framework.

Miettinen et al. [8] developed an IoT Sentinel solution to protect IoT networks by using device fingerprinting, which utilizes machine learning to identify IoT devices when they join and register on the network. However, this technique does not take device behavior into account. Another method, proposed by Siby et al. in [9], is called IoT scanner, which examines network traffic flow by analyzing frame headers during a set research period. The disadvantage of this method is that if the traffic capture windows are not synchronized, devices with similar traffic patterns may be mistakenly identified as separate entities. Kawai et al. [10] employed traffic flow characteristics and machine learning techniques to classify IoT devices. Their approach was unique because they exclusively analyzed two types of traffic attributes-packet size and inter-arrival time-without considering other factors. Some techniques incorporate additional traffic features to improve identification accuracy and categorize device traffic at an early stage. Sarker et al. [11] carried out a broad survey work on IoT security intelligence in 2023, where a comprehensive overview of machine learning solutions and research directions is discussed in detail, also highlighting the novel solution approach for securing IoT devices using AI.

The paper 'A survey,' authored by Liao et al. [12], discusses topics related to modeling, detecting, and scheduling large data flows within a software-defined energy cloud environment. Boutaba et al. [13] suggested a composite property to improve the accuracy of early recognition using machine learning, emphasizing the importance of a strong composition relationship when identifying streams of interactive network resources. Dutta et al. [14] developed an ensemble method using deep models and the Stacking methodology to detect network anomalies in cyber-physical systems network traffic. They collected heterogeneous flow-based data, including IoT data, to classify outliers. Thanh and Lang [15] evaluated the performance of several machine learning algorithms, including Bagging, AdaBoost, Stacking, Decorate, Voting, and Random Forest, using a data mining tool to identify the optimal ensemble technique for detecting network intrusions. Among the ensembles, those that utilized Stacking and Decorate procedures to aggregate the results of their base classifiers required longer training and testing times compared to single classifiers. The research work combining multiple machine learning approaches can offer significant benefits. However, there are challenges related to overfitting, complexity, and data biases. Generalizing these strategies for other authentication applications requires a focus on adaptability, robustness, and consistent performance across diverse data types and security measures.

2. Materials and methods

This section presents a summary of commonly employed machine learning (ML) algorithms for developing effective anomaly detection systems and outlines a fundamental approach for AI-based anomaly detection. The ML algorithms are broadly categorized into two types: supervised and unsupervised. The unsupervised algorithms work with unlabeled data to identify anomalies, while supervised algorithms use labeled data to extract crucial information.

2.1 A general AI-based Anomaly detection methodology

Anomaly detection systems are developed using machine learning, which involves three main steps as shown in Figure 1. Firstly, data is preprocessed by encoding, normalizing, and cleaning it to prepare it for algorithmic processing. This step involves removing duplicate entries and missing data. Next, the preprocessed data is randomly split into two parts: the training dataset, which usually accounts for 80% of the original data, and the testing dataset. In the training phase, the ML algorithm is trained on the training dataset. The duration of this phase

varies depending on the complexity of the proposed model and the size of the dataset. After training the model, it is tested using the dataset, and its performance is assessed based on its predictions. Anomaly detection models classify network traffic instances as benign (normal) or attack.

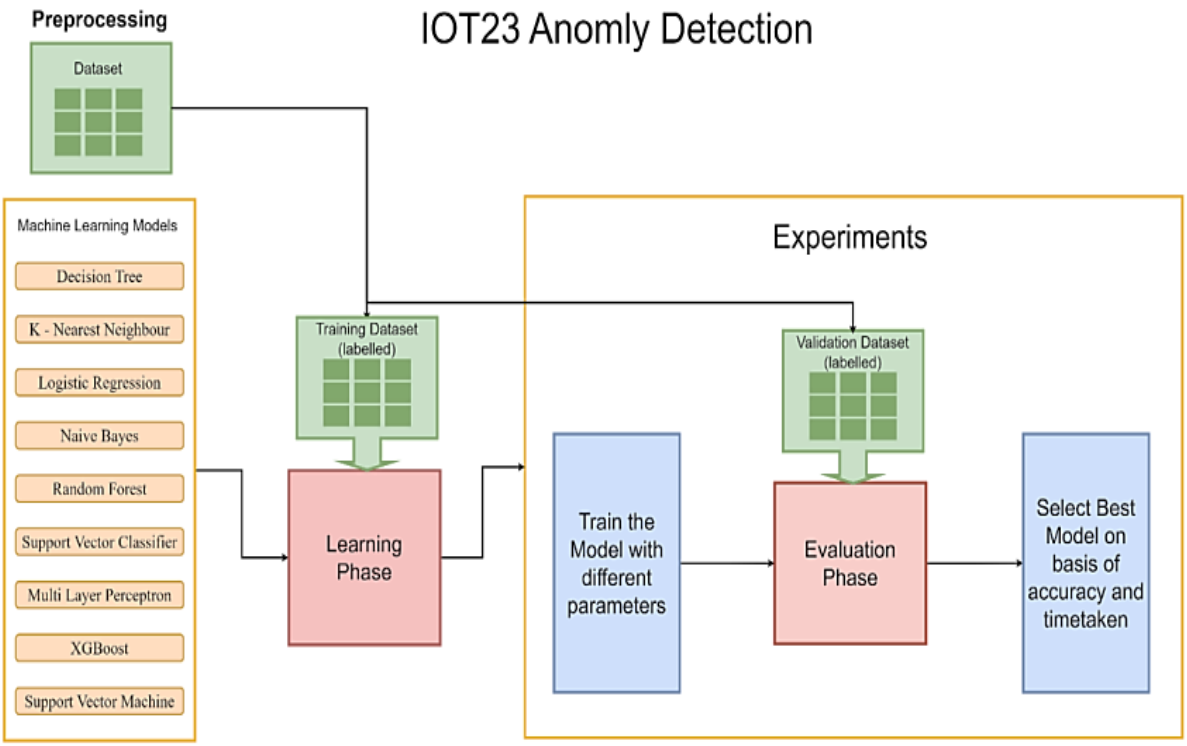

Figure 1 Generalized machine learning-based anomaly detection system methodology.

2.1.1 Decision Tree

The Decision Tree is a supervised learning model that utilizes a series of simple decisions based on information to classify a dataset. This is one of the fundamental techniques in supervised machine learning and can be used for both classification and regression tasks, as explained by Chary et al. [16]. The structure of the model resembles a tree, with nodes, branches, and leaves as depicted in Figure 2. Each node represents an attribute or feature, and each leaf denotes a potential class designation or result. As explained by Sahani et al. [17], each branch of the decision tree represents a rule or decision. To prevent overfitting and reduce extra branches, the DT algorithm automatically selects the most relevant attributes and performs pruning, as demonstrated in the study conducted by Xin et al. [18]. The three most commonly used DT subtypes are CART, C4.5, and ID3. Advanced learning algorithms such as Random Forest (RF) and XGBoost are also based on decision trees.

2.1.2 K – Nearest Neighbors (KNN)

KNN is a supervised learning algorithm that employs the concept of "feature similarity," as described in the broad study conducted by Ahmad et al. [19], to classify data samples into classes similar to the training data based on distance. The algorithm estimates the distance between a given data sample and its neighbors to identify its class. The impact of the KNN algorithm's parameter k on the model's performance is shown in Figure 3. If the value of k is too small, the model may become overfit. However, an excessively large range of k values could result in incorrect classification of the sample case. In their study, Zhang et al. [20] evaluated the performance of several machine learning algorithms using the most recent benchmark dataset, CSE-CIC-IDS 2018. By utilizing the Synthetic Minority Oversampling Technique (SMOTE), they addressed the issue of imbalanced datasets and improved the detection rate for minority class attacks using K-Nearest Neighbors.

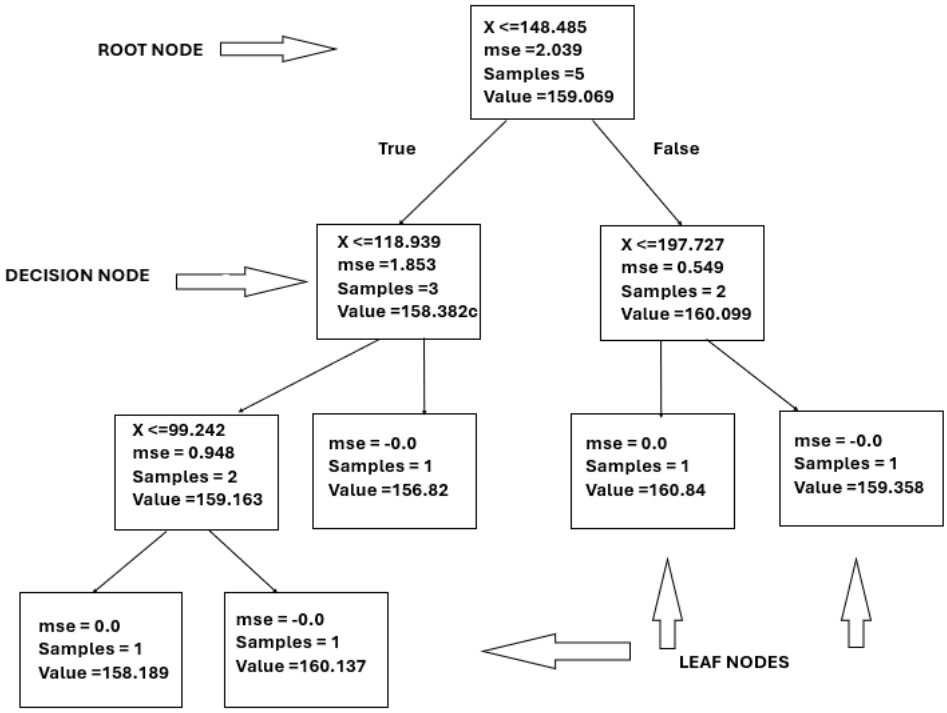

Figure 2 Structure of decision tree classifier for anomaly detection of depth6.

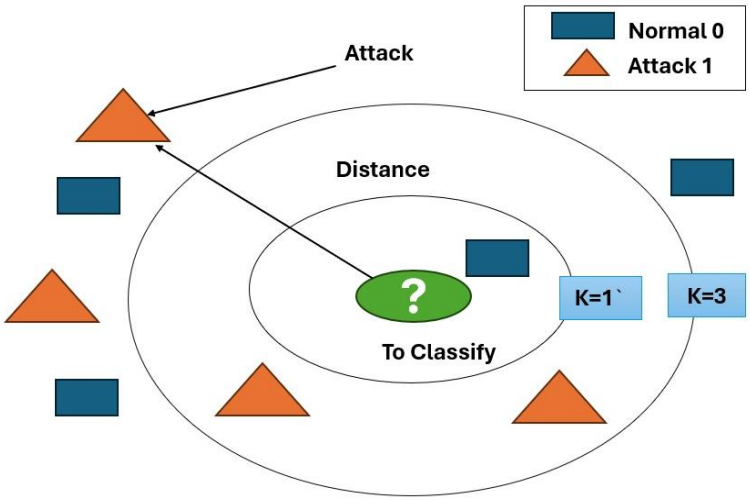

Figure 3 Structure of KNN classifier for anomaly detection.

2.1.3 Logistic Regression

Anomaly detection, also referred to as outlier detection, involves identifying data points that deviate from the expected pattern of the majority of the data. One approach to using logistic regression for anomaly detection is to train a logistic regression model on a dataset that exclusively contains normal instances, and then employ the model to assess the probability that a new instance is normal or anomalous, as indicated in the broad work done by Hasan, M et al. [21]. If the predicted probability falls below a specific threshold, the instance is deemed an anomaly, as depicted in Figure 4. Suppose you wish to detect fraudulent transactions on the IoT23 dataset, which entails detecting abnormal transactions. In that case, you could train a logistic regression model on a dataset of normal transactions from the IoT23 dataset and use the model to predict the likelihood that a new transaction is normal. If the probability is low, it suggests that the transaction may be fraudulent. It's important to note that logistic regression is just one of several techniques that can be employed for anomaly detection, and the choice of method will depend on the unique characteristics of the data and the objectives of the anomaly detection task.

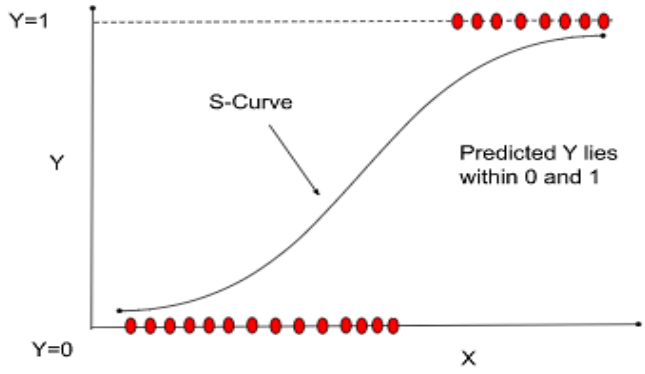

Figure 4 Structure of logistic regression classifier for anomaly detection.

2.1.4 Naive Bayes Classifier

In the context of anomaly detection, Naive Bayes can be used to classify whether an input is an anomalous example or not. To do this, first, we have trained an IoT23 dataset with normal and anomalous samples using the Naïve Bayes method, as studied by Vangipuram et al. [22]. In order to classify fresh inputs as either normal or anomalous, the algorithm learns to recognize patterns within the normal samples. The dataset is initially split into a training and a test set to train the algorithm. Following training on the training set, the algorithm is then used to forecast labels for the test set. Following that, measures like accuracy, precision, and recall are used to evaluate the algorithm's performance, as shown in Figure 5. One advantage of using Naïve Bayes for anomaly detection is that it is relatively simple to implement and fast to run. However, it can be less accurate than some other machine learning algorithms, especially when the data is complex or has a high-dimensional feature space.

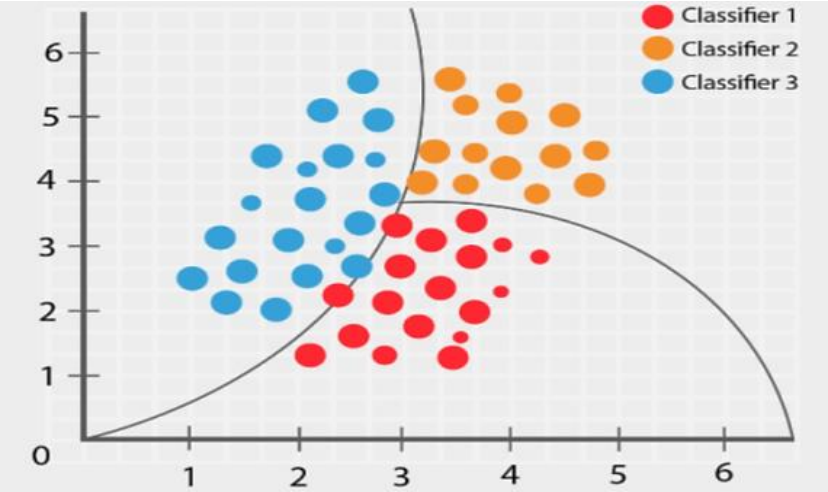

Figure 5 Structure of naïve Bayes classifier for anomaly detection.

2.1.5 Random Forest

The process of utilizing a random forest for anomaly detection involves a similar procedure to that of Naive Bayes. The first step is to prepare the data, which involves scaling the features, addressing missing values, and possibly modifying the data to enhance its compatibility with the model. After data preparation, the dataset should be split into a training set and a test set with a suitable ratio, such as 70/30 or 80/20, as illustrated in Figure 6. Subsequently, a random forest model should be trained on the training set by fitting it to the data and optimizing its hyperparameters to enhance its performance, as conducted by Thakkar et al. [23]. The model is then used to predict the labels for the test set, with the labels indicating whether each example is normal or anomalous in the context of anomaly detection. Model performance can be evaluated using metrics such as accuracy, precision, and recall, and it is recommended to visualize the performance using plots such as a confusion matrix or Russian Olympic Committee (ROC) curve. Such plots can provide insights into the model's strengths and weaknesses,

facilitating identification of areas that need improvement. Despite its simplicity and speed, random forest may be less accurate than other machine learning methods for complex or high-dimensional feature spaces.

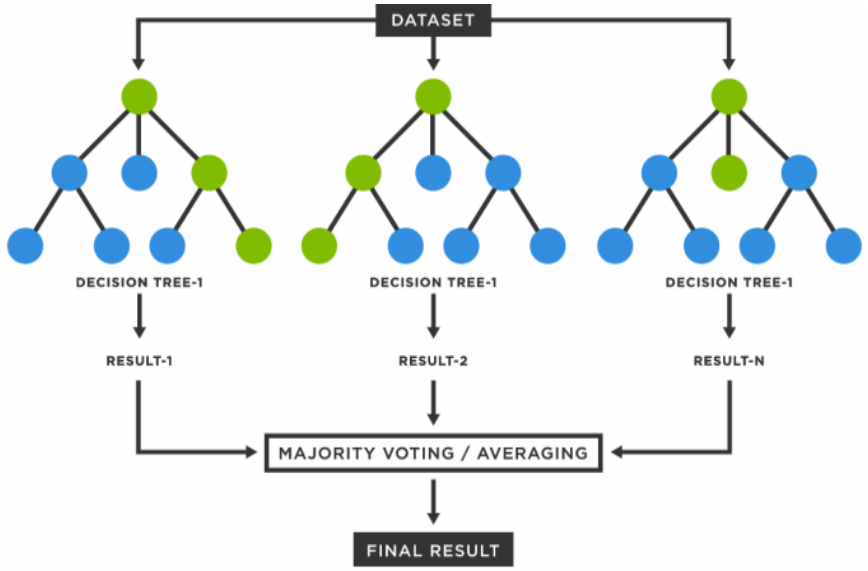

Figure 6 Structure of random forest classifier for anomaly detection.

2.1.6 Support Vector Machine

Support Vector Machines (SVMs) aim to find the optimal boundary or hyperplane that separates different classes in a dataset. In anomaly detection, an SVM model can be trained to separate normal examples from anomalous ones, as illustrated in Figure 7. After training, the model can classify new inputs as normal or anomalous based on which side of the boundary they fall. Before training the model, it is crucial to preprocess the data by handling missing values, scaling features, and ensuring that the model is not influenced by feature scales. After preprocessing, the dataset is split into training and testing sets. According to Yang et al [24], the SVM model is then fitted to the training set using normal examples as the negative class and anomalous examples as the positive class. Choosing the right parameters, such as kernel type, regularization term, and margin width, is essential for the performance of the SVM model, according to the study conducted by Chatterjee et al. in [25].

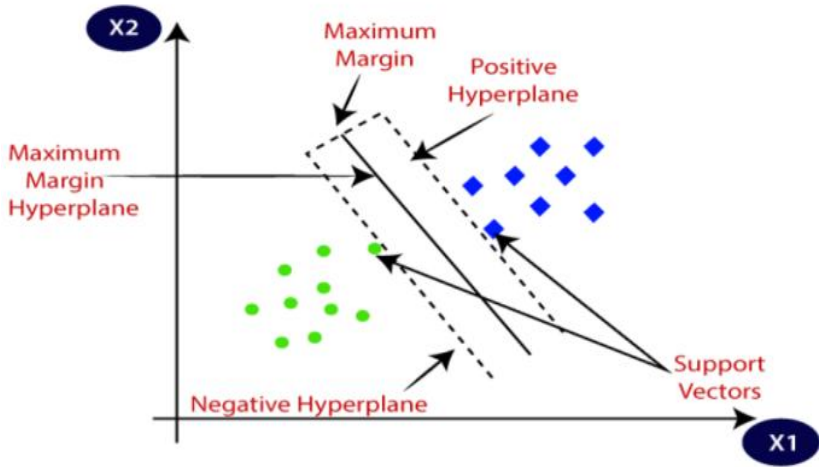

Figure 7 Structure of support vector machine for anomaly detection.

2.1.7 Multi-Layer Perceptron

The configuration of an MLP, including the number of hidden layers and neurons, is a crucial factor that we explore in experiments to attain optimal performance, as stated by Abusitta et al [26]. The training process is iterative and involves selecting an appropriate optimizer and learning rate, with training ending once the model reaches an acceptable level of performance on the training set, as illustrated in Figure 8. After training, we can

use the model to predict the labels of the test set and assess its efficacy using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC. MLPs offer flexibility in maximizing performance based on the specific data and task at hand by allowing modifications to the number of layers and neurons in the network. However, MLPs can be sensitive to the choice of optimizer, the number of hidden layers and neurons, and the learning rate, and may require a large amount of data to achieve good results, as indicated by the studies conducted by Zhang et al. [27] and Alwahedi et al. [28].

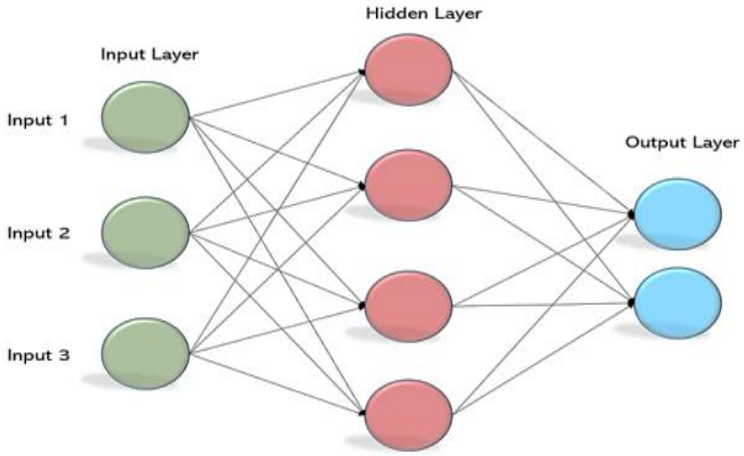

Figure 8 Structure of multi-layer perceptron classifier for anomaly detection.

2.1.8 XGBoost Classifier

The XGBoost classifier is an ensemble learning framework based on decision trees that is highly efficient, flexible, and portable for classifying data, as proposed by Wang et al [29]. In the context of anomaly detection, XGBoost can be used to classify whether an input is an anomalous example or not, as studied and analyzed by Taghavirashidizadeh et al. in [30]. To accomplish this, the data must first be prepared by preprocessing it, handling missing values, and scaling the features. After that, the data can be split into a training set and a test set using a suitable split ratio, such as 70/30 or 80/20. The XGBoost classifier can then be trained using the training set, with the typical cases designated as the positive class and the anomalous examples as the negative class, as shown in Figure 9. Following model training, it is utilized to make predictions on the test set, and its performance is evaluated using measures like precision, recall, accuracy, F1-score, and Area Under the Curve – Receiver Operating Characteristic (AUC ROC). One of the benefits of using XGBoost for anomaly detection is its ability to handle high-dimensional data and automatically learn complex nonlinear relationships within the data.

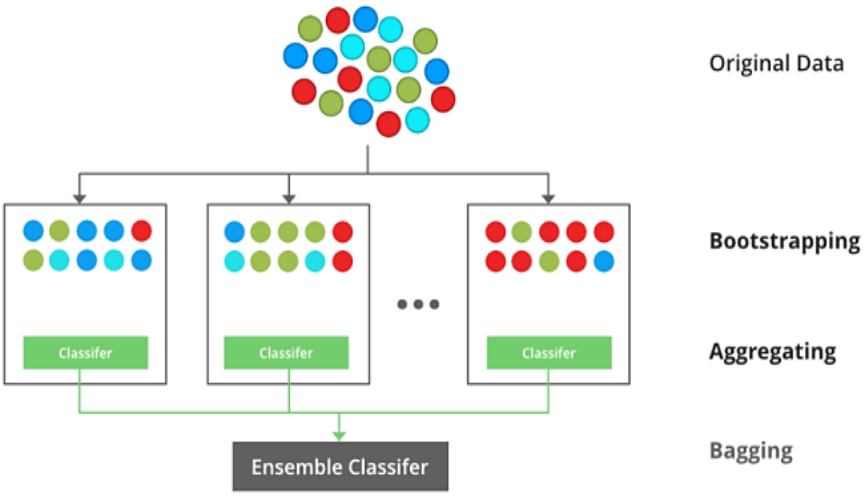

Figure 9 Structure of XGBoost classifier for anomaly detection.

3. Results and discussion

The classification model's accuracy results are limited to two decimal places, suggesting that the actual results may fall between 0.947 and 1.000, as illustrated in Figure 10. XGBoost is the best algorithm, with precision, recall, and F1 scores ranging from 0.86 to 0.87 and 0.83, respectively, as indicated in the confusion matrix in Figures 13, 18, and 19. Considering the study and its results, Decision Trees could be useful to some extent in achieving precision. However, the confusion matrix in Figures 11, 12, and 13 demonstrates that the model tended to favor the most frequent categories and sometimes utilized other categories. It is noteworthy that the algorithm never predicted benign behavior, and thus, benign captures cannot be distinguished from other anomalies by drawing a hyperplane. The ROC curve for the most accurate model is depicted in Figures 14, 15, 16, 17, 18, and 19. The dataset had only 16 instances of a particular category, occurring on average four times per file. Similar issues were encountered in detecting the Torii botnet, but some algorithms were able to produce results. The challenge in identifying smaller attack categories is due to their infrequent occurrence in the dataset. Figure 10 and the table below show that all algorithms achieved at least 69% accuracy in predicting the most common categories. The confusion matrix provides additional information on each algorithm's F1 Score, Precision, Recall, and Accuracy. Researchers can use this information to select the best classification algorithm for their specific research, taking into account the nature of their dataset. The various classification algorithms illustrated in this study identify the best-suited algorithm for anomaly detection. Figure 20 depicts the overall performance of the classification models in terms of accuracy and training time.

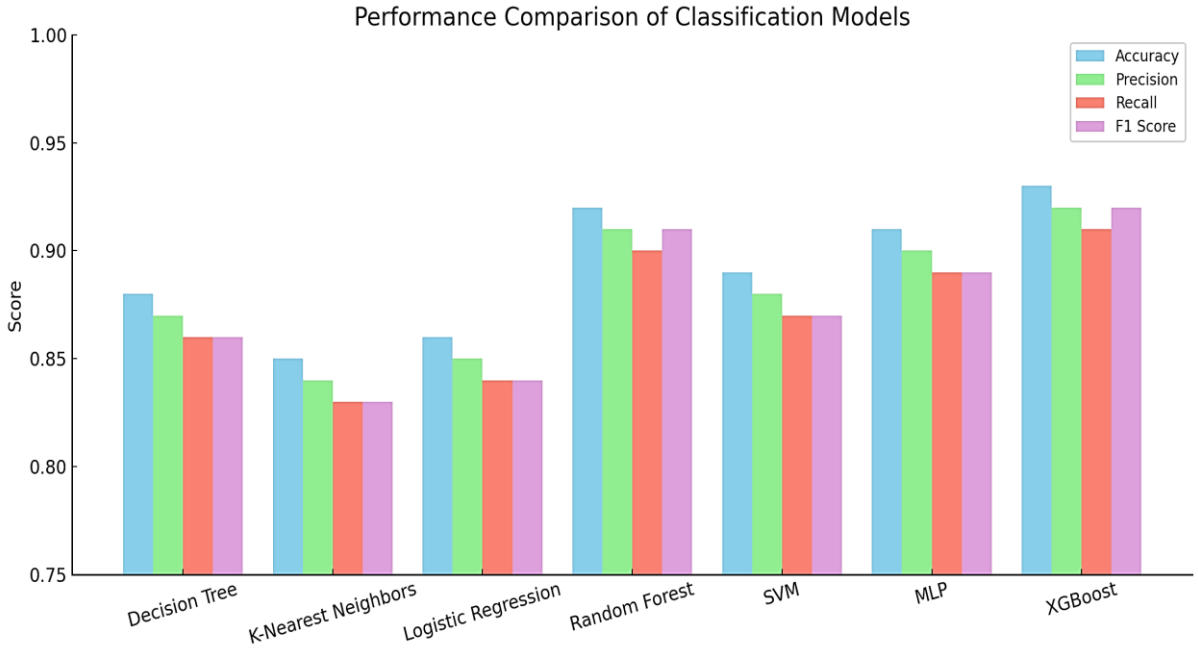

Figure 10 Illustrates various Classification Models for anomaly detection in IoT-23 dataset.

Figures 11 to 13 present the confusion matrices of three prominent classifiers—Random Forest, Support Vector Machine (SVM), and XGBoost used in the evaluation of the proposed IDS framework. These confusion matrices offer a clear visualization of the models' prediction outcomes by detailing the true positives, true negatives, false positives, and false negatives. Figure 11 demonstrates that the Random Forest classifier achieves strong classification performance, with a high number of correctly identified instances and minimal misclassifications. In Figure 12, the SVM classifier also shows competitive results; however, it tends to have a slightly higher false positive rate, which could be attributed to the margin-based decision boundary it employs. Figure 13 depicts the confusion matrix of the XGBoost classifier, which exhibits excellent precision and recall, with very few false negatives, making it highly suitable for real-time intrusion detection in IoT networks. Overall, these results confirm that ensemble learning models, particularly Random Forest and XGBoost, offer superior accuracy and robustness compared to traditional classifiers.

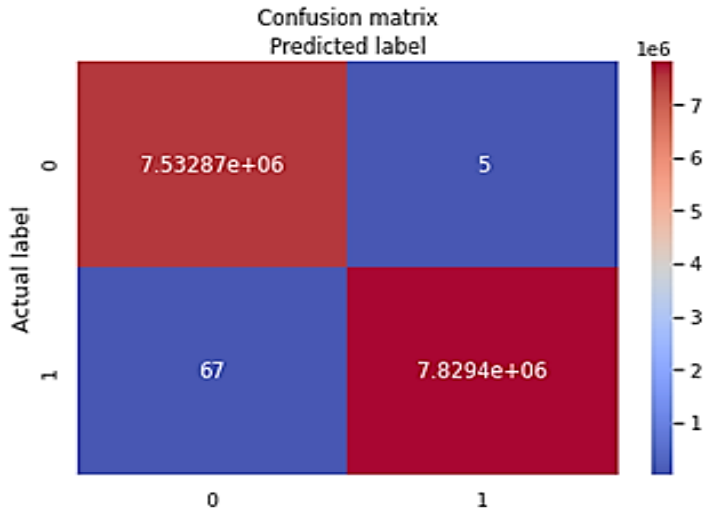

Figure 11 Confusion matrix of Random Forest.

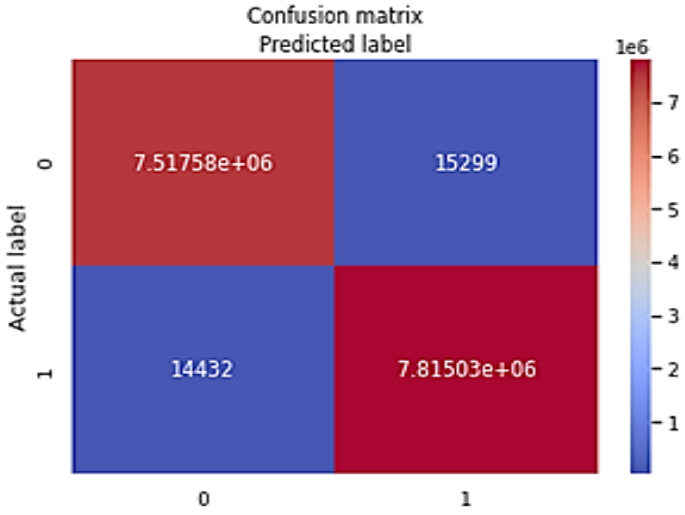

Figure 12 Confusion matrix of Support Vector Machine.

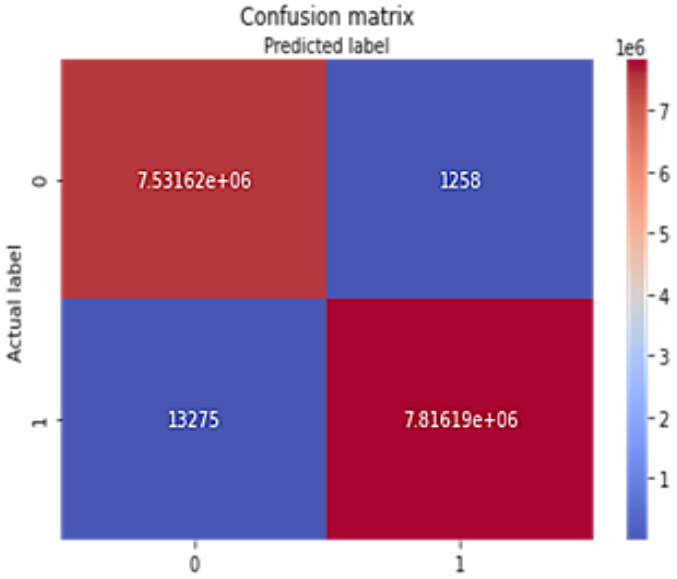

Figure 13 Confusion matrix of XGBoost.

Figures 14 to 19 illustrate the ROC curves and Precision vs. Recall curves for the Random Forest, SVM, and XGBoost classifiers. These performance curves provide valuable insights into the trade-offs between classification sensitivity and specificity across different threshold settings. Figure 14 shows the ROC curve of the Random Forest classifier, which achieves an AUC close to 1, indicating excellent discriminatory power. Complementing this, Figure 15 presents the Precision vs. Recall curve of Random Forest, which maintains high precision and recall even at varying thresholds, reflecting its robustness in detecting intrusions. Similarly, Figures 16 and 17 display the ROC and Precision vs. Recall curves for the SVM classifier. While the ROC curve in Figure 16 indicates good classification capability, it may slightly underperform compared to ensemble methods due to sensitivity to feature scaling and non-linearity. The Precision vs. Recall curve in Figure 17 further confirms SVM's stable, though comparatively moderate, performance. Figures 18 and 19 correspond to the XGBoost classifier, where the ROC curve (Figure 18) demonstrates near-perfect classification with an AUC close to 1, showcasing XGBoost's strength in handling complex decision boundaries. The Precision vs. Recall curve in Figure 19 supports this observation, as XGBoost maintains high precision and recall even in imbalanced data scenarios. Collectively, these figures highlight the superiority of ensemble-based classifiers like Random Forest and XGBoost in achieving both high detection accuracy and reliable performance across varying thresholds.

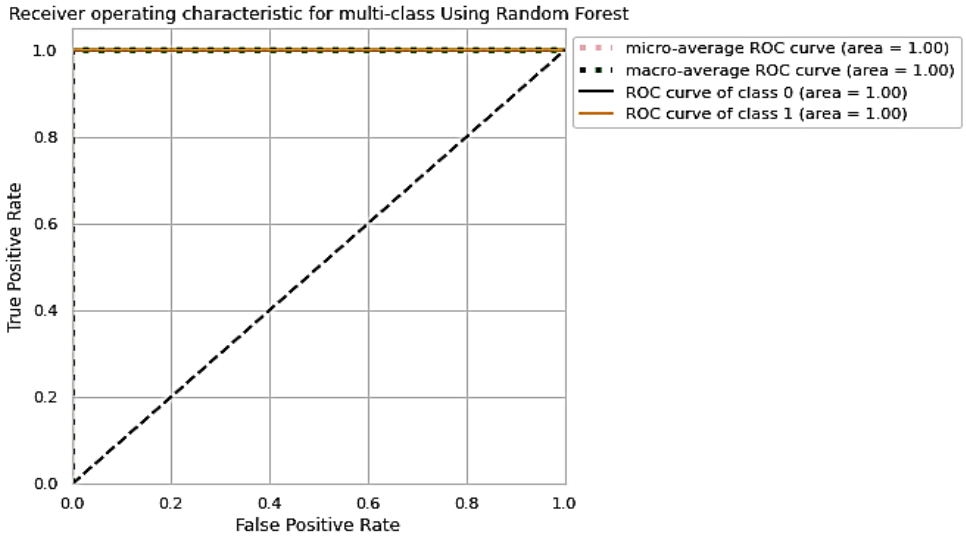

Figure 14 ROC curve of Random Forest

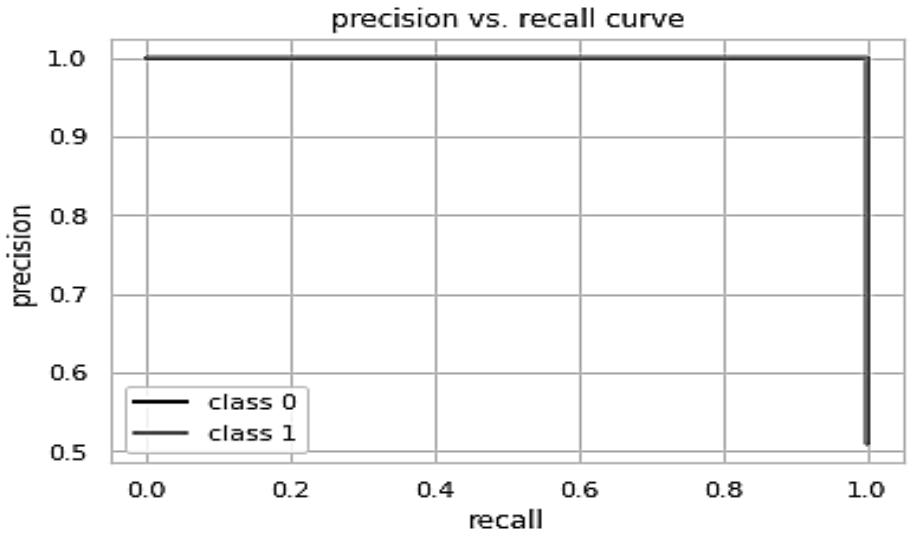

Figure 15 Precision V/S Recall Curve of Random. Forest.

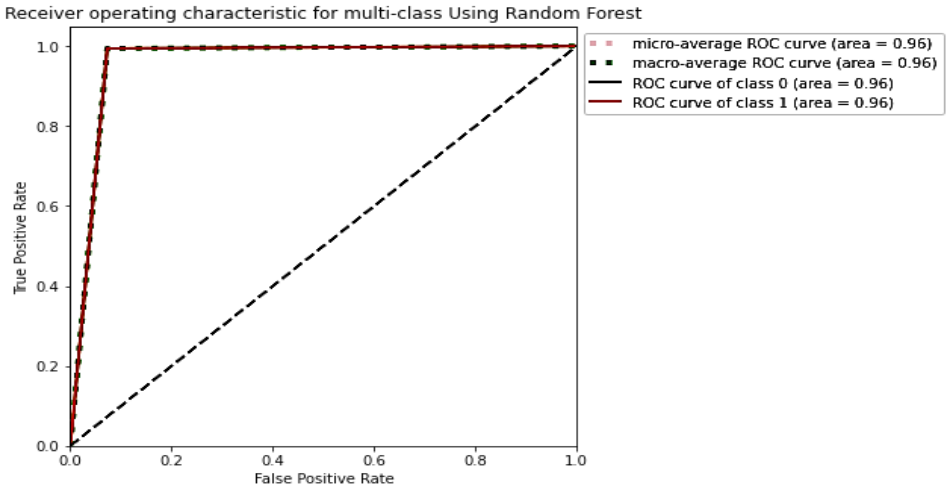

Figure 16 ROC curve of Support Vector Machine.

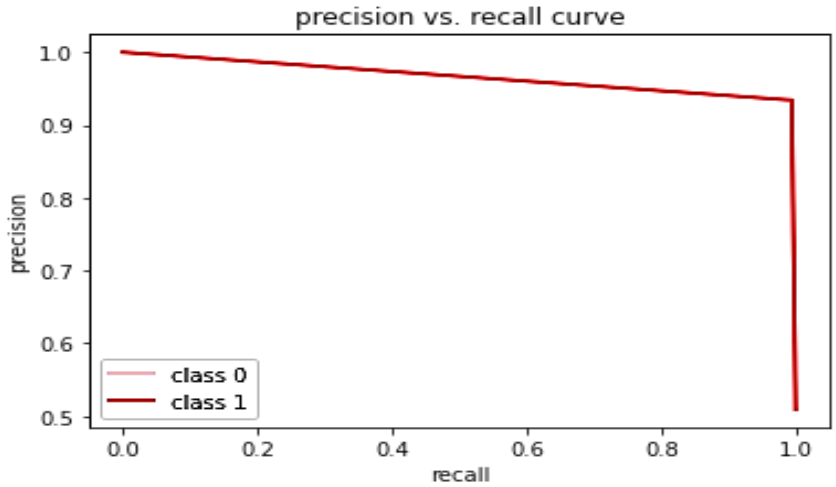

Figure 17 Precision V/S Recall Curve of Support Vector Machine.

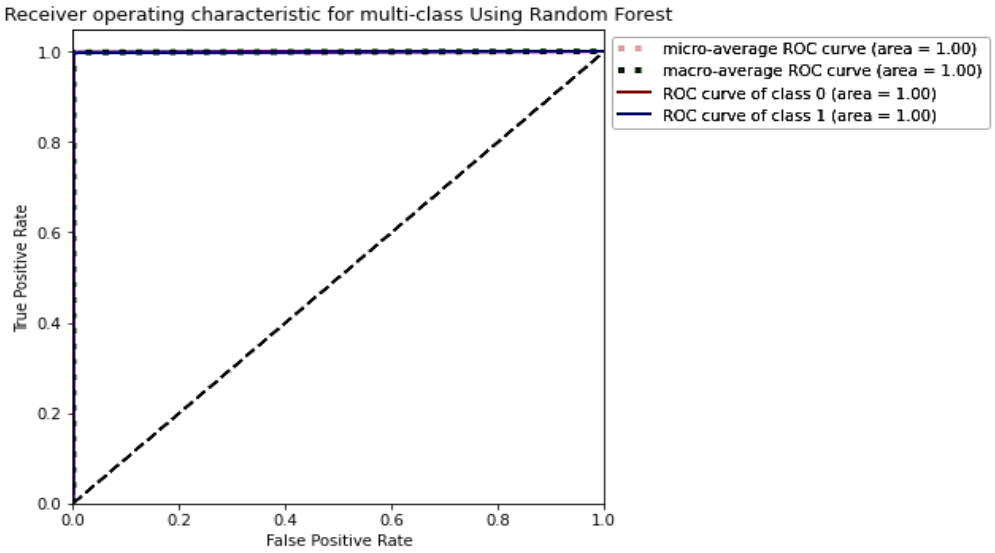

Figure 18 ROC curve XGBoost.

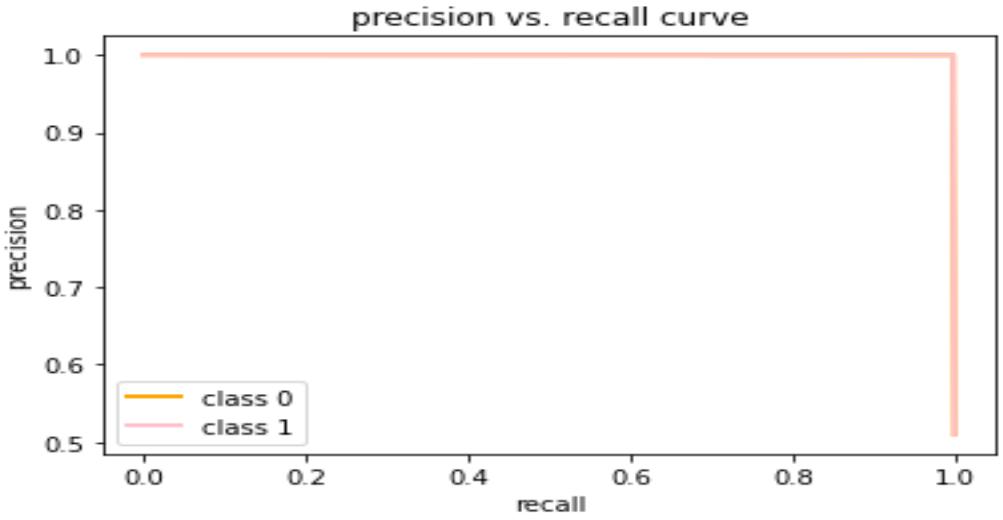

Figure 19 Precision V/S Recall Curve of XGBoost.

Figure 20 illustrates the comparative performance of various machine learning models by plotting their accuracy against the model names alongside the time taken for training or inference. This multi-dimensional visualization highlights the trade-offs between model accuracy and computational efficiency. Models such as Random Forest, Support Vector Machine, and XGBoost are compared, showing that while some models achieve higher accuracy, they may require longer processing times. Conversely, models with faster execution times sometimes deliver slightly lower accuracy. This figure provides valuable insights for selecting the optimal model based on the desired balance between predictive performance and resource consumption in the given application context.

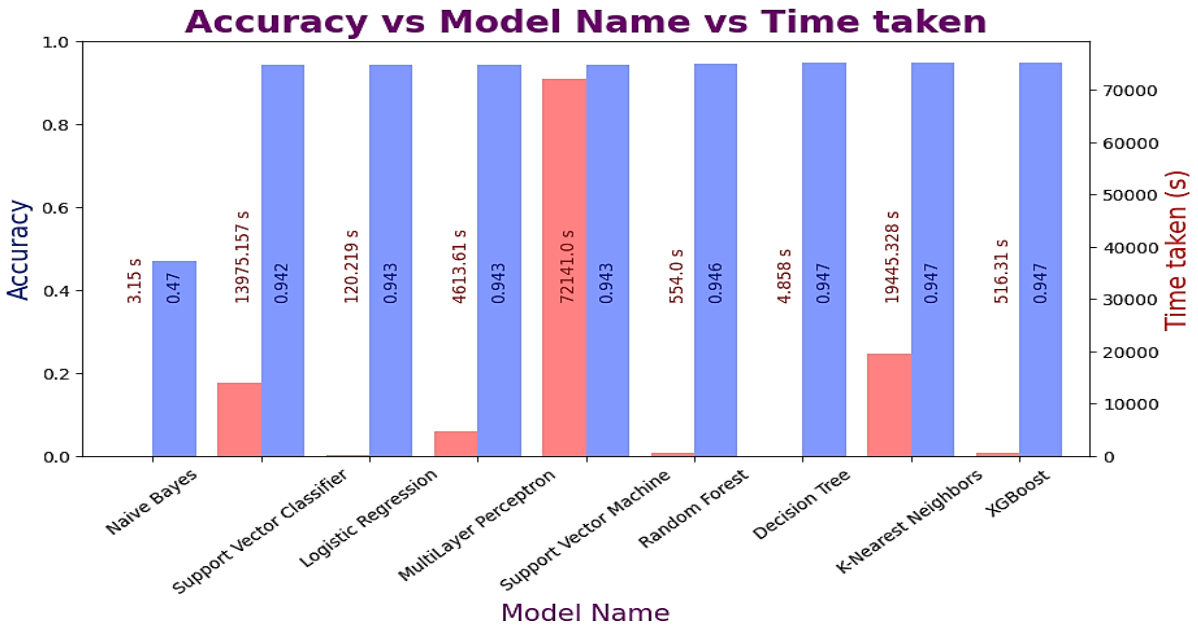

Figure 20 Accuracy V/S Model Name V/S time taken.

4. Conclusions

This paper aims to enhance clarity and resonance in its final remarks. The principal contributions, underscored by numerical improvement percentages, are key focal points for reader engagement. XGBoost emerges as the standout anomaly detection and classification solution within the IoT-23 dataset, demonstrating superior performance across all metrics. To elucidate XGBoost's success, future investigations will delve into the underlying mechanisms, addressing the current ambiguity surrounding its superiority and evaluating with deep

learning model & hybrid model to evaluate their effectiveness in anomaly detection by considering accuracy as a parameter. The transparency now features numerical insights, providing readers with a tangible grasp of the research's impact. This conclusion not only invites researchers to build upon our findings but also suggests avenues for open research. The integration of deep learning & hybrid models becomes a strategic direction to overcome identified challenges, promising enhanced adaptability and performance in the realm of anomaly detection. The incorporation of numerical data, coupled with an enticing glimpse into ongoing and potential research directions, sets the stage for future investigations into the mechanisms behind XGBoost's success and the potential of deep learning in mitigating identified shortcomings. This ongoing research endeavors to advance anomaly detection capabilities and contribute to the ongoing evolution of effective and robust systems in the field of IoT.

5. Acknowledgements

The first author gratefully acknowledges the financial support provided by a Ph.D. scholarship from Vellore Institute of Technology, India.

6. References

[1] Shafique K, Khawaja BA, Sabir F, Qazi S, Mustaqim M. Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. IEEE Access. 2020;8:23022-23040.

[2] Dwivedi YK, Hughes L, Baabdullah AM, Ribeiro-Navarrete S, Giannakis M, Al-Debei MM, Dennehy D, Metri B, Buhalis D, Cheung CM, Conboy K. Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. Int J Inf Manage. 2022;66:102542.

[3] Xiao L, Wan X, Lu X, Zhang Y, Wu D. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security. IEEE Signal Process Mag. 2018;35(5):41-49.

[4] Ullah I, Mahmoud QH. Design and development of RNN anomaly detection model for IoT networks. IEEE Access. 2022;10:62722-62750.

[5] Mukhaini GA, Anbar M, Manickam S, Al-Amiedy TA, Al Momani A. A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in Internet of Things networks. J King Saud Univ Comput Inf Sci. 2023:101866.

[6] Rathore S, Park JH. Semi-supervised learning based distributed attack detection framework for IoT. Appl Soft Comput. 2018;72:79-89.

[7] Liu L, Ma Z, Meng W. Detection of multiple-mix-attack malicious nodes using perceptron-based trust in IoT networks. Future Gener Comput Syst. 2019;101:865-879.

[8] Miettinen M, Marchal S, Hafeez I, Asokan N, Sadeghi AR, Tarkoma S. Iot sentinel: Automated device-type identification for security enforcement in iot. IEEE 37th International Conference on Distributed Computing Systems (ICDCS). 2017;5: 2177-2184.

[9] Siby S, Maiti RR, Tippenhauer NO. IoT Scanner: Detecting privacy threats in IoT neighborhoods. In poceedings of the 3rd ACM international workshop on IoT privacy, trust, and security. 2017;2:23-30.

[10] Kawai H, Ata S, Nakamura N, Oka I. Identification of communication devices from analysis of traffic patterns. 13th International Conference on Network and Service Management (CNSM). 2017;26:1-5.

[11] Sarker IH, Khan AI, Abushark YB, Alsolami F. Internet of things security intelligence: a comprehensive overview, machine learning solutions and research directions. Mob Netw Appl. 2023;28(1):296-312.

[12] Liao LX, Chao HC, Chen MY. Intelligently modeling, detecting, and scheduling elephant flows in software defined energy cloud: A survey. J Parallel Distrib Comput. 2020;146:64-78.

[13] Boutaba R, Salahuddin MA, Limam N, Ayoubi S, Shahriar N, Estrada-Solano F, Caicedo OM. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. J Internet Serv Appl. 2018;9(1):1-99.

[14] Dutta V, Choraś M, Pawlicki M, Kozik R. A deep learning ensemble for network anomaly and cyber-attack detection. Sensors. 2020;20(16):4583.

[15] Thanh HN, Van Lang T. Use the ensemble methods when detecting DoS attacks in network intrusion detection systems. EAI Endorsed Trans Context-aware Syst Appl. 2019;6(19):e163484.

[16] Chary SN, Rama B. A survey on comparative analysis of decision tree algorithms in data mining. Int J Adv Sci Technol Eng Manag Sci. 2017;3(1):91-95.

[17] Sahani R, Shatabdinalini, Rout C, Chandrakanta Badajena J, Jena AK, Das H. Classification of intrusion detection using data mining techniques. Prog Comput Anal Netw Proc ICCAN. 2017;18:753-764.

[18] Xin Y, Kong L, Liu Z, Chen Y, Li Y, Zhu H, Gao M, Hou H, Wang C. Machine learning and deep learning methods for cybersecurity. IEEE Access. 2018;6:35365-35381.

[19] Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerg Telecommun Technol. 2021;32(1):e4150.
[20] Yang K, Kpotufe S, Feamster N. An efficient one-class SVM for anomaly detection in the internet of things. arXiv. 2104;1:11146.
[21] Hasan M, Islam MM, Zarif MI, Hashem MM. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things. 2019;7:100059.
[22] Vangipuram R, Gunupudi RK, Puligadda VK, Vinjamuri J. A machine learning approach for imputation and anomaly detection in IoT environment. Expert Syst. 2020;37(5):e12556.
[23] Thakkar A, Lohiya R. A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artif Intell Rev. 2022;55(1):453-563.
[24] Yang K, Kpotufe S, Feamster N. An efficient one-class SVM for anomaly detection in the internet of things. arXiv. 2104;1:11146.
[25] Chatterjee A, Ahmed BS. IoT anomaly detection methods and applications: A survey. Internet Things. 2022;19:100568.
[26] Abusitta A, de Carvalho GH, Wahab OA, Halabi T, Fung BC, Al Mamoori S. Deep learning-enabled anomaly detection for IoT systems. Internet Things. 2023;21:100656.
[27] Zhang H, Xie R, Li K, Huang W, Yang C, Liu J. Anomaly detection based on deep learning: insights and opportunities. IEEE 10th international conference on cyber security and cloud computing (CSCloud). IEEE 9th international conference on edge computing and scalable cloud (EdgeCom). 2023;1:30-36.
[28] Alwahedi F, Aldhaheri A, Ferrag MA, Battah A, Tihanyi N. Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. Internet Things Cyber Phys Syst. 2024;3:1-5.
[29] Wang X, Lu X. A host-based anomaly detection framework using XGBoost and LSTM for IoT devices. Wirel Commun Mob Comput. 2020;20:1-3.
[30] Taghavirashidizadeh A, Zavvar M, Moghadaspour M, Jafari M, Garoosi H, Zavvar MH. Anomaly Detection In IoT Networks Using Hybrid Method Based On PCA-XGBoost. 8th iranian conference on signal processing and intelligent systems (ICSPIS). 2022;28:1-5.