Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

Network Anomaly Detection in the Internet of Things (IoT)

Benjamin Asubam Weyori[1], Ben Beklisi Kwame Ayawli[2], Samuel Tweneboah-Koduah[1]

[1]Department of Computer and Electrical Engineering, University of Energy and Natural Resources, Sunyani, Ghana

[2]Department of Computer Science, Sunyani Technical University, Sunyani, Ghana

Abstract

This study comprehensively explores developing an efficient network anomaly detection system for the Internet of Things (IoT) through advanced machine learning techniques. The methodology encompasses data collection, preprocessing, feature engineering, and evaluating multiple machine-learning models. Random Forest emerges as the top-performing model, demonstrating impressive accuracy (98.11%), sensitivity (75.86%), specificity (98.71%), and G-Mean (86.53%). Decision Tree and K-Nearest Neighbors also exhibit commendable performances, highlighting the effectiveness of diverse machine-learning approaches in IoT anomaly detection. The proposed ensemble model, integrating Random Forest, XGBoost, and K-Nearest Neighbors, surpasses individual models with an accuracy of 98.14%, sensitivity of 78.75%, specificity of 98.62%, and a G-Mean of 88.12%. Leveraging a complex voting criterion and meticulous optimization through grid search enhances the model's predictive capabilities. Addressing class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE) significantly improves sensitivity, specificity, and G-Mean. Sensitivity increases to 81.25%, specificity improves to 98.96%, and the G-Mean rises to 89.51%, enhancing overall model performance. Future research directions include exploring and optimising more sophisticated ensemble models, real-world deployment of the proposed model in diverse IoT scenarios, investigation of techniques for adapting to dynamic changes in IoT network behaviour, advanced hyperparameter tuning, and addressing potential vulnerabilities and security concerns. The study lays a solid foundation for effective IoT network anomaly detection, providing insights that can contribute to advancing anomaly detection techniques in the ever-evolving landscape of the Internet of Things.

Keywords: Internet of Things (IoT), Network Anomaly Detection, Security Challenges, Anomaly Detection, Machine Learning.

1.0 Introduction

The advent of the Internet has revolutionised communication between humans. Many services are found online using the internet, and this trend is increasing over time (Field, (Maseer et al., 2021). Internet of Things (IoT) devices are reshaping how humans perceive and interact with the physical world (Al-amri et al., 2021; Hasan et al., 2019). Statista estimated a staggering 30.73 billion connected loT devices in 2020, which will double in less than four years (Shaver, 2020; Ullah et al., 2021). By 2025, IoT systems are expected to cross nearly 75 billion connected devices, tripling the global population. Recently, the deployment of the Internet of Things (IoT) has become highly recommended in many applications in different fields (Said & Yahyaoui, 2021). The Internet of Things (IoT) concept was first introduced in 1999 by Kevin Ashton, a member of the Radio Frequency Identification (RFID) development community. Since then, it has emerged rapidly due to the technological advancement of internet-based applications and the exponential growth of intelligent computing devices (Maniriho et al., 2020). An IoT represents a complex and dynamic network that connects endpoints of devices to provide various services (Maseer et al., 2021). The IoT is a network of heterogeneous objects, such as smartphones, laptops, intelligent devices, and sensors, connected to the Internet through various technologies. IoT enables multiple sensors and devices to communicate directly without user interaction and can be applied in

different application domains, such as smart homes, wearable devices, smart cities, healthcare, agriculture, transportation, and industry f(Alghanmi et al., 2021). Methods such as big data analytics, business intelligence, and machine learning algorithms can be used to extract meaningful information from these data to supplement human needs.

Because IoT devices are connected to the global internet with unmatured and vulnerable communication protocols and applications, it is exposed to many potential security threats (Ahmad et al., 2021; Anomaly-based et al., 2022; Materials et al., 2022). Adversaries may exploit these vulnerabilities and inject anomalies that trigger the system to make wrong control decisions in IoT-based applications, causing a catastrophic impact on people's lives, properties, and economics. Therefore, the evolving threats of cyberattacks pose significant challenges to the IoT ecosystem field(Alsoufi, Razak, Siraj, et al., 2021). For example, attackers can always have unauthorised access and harm the personal information of IoT devices without the knowledge of either the owner or administrator (Pathak et al., 2021). Anomalies within the network context are seen as unusually utilizing resources, not abiding by the norm (Stavros, 2019(Stavros, 2019.). Common cyber-attacks involve DDoS (Distributed Denial of Service), ransomware, and botnet attacks, which seek to exploit IoT networks and destroy their computational capabilities (Ullah et al., 2021). Therefore, the need to provide solutions to detect and prevent attacks and intrusions on IoT devices is one of the leading security areas in these networks (Abbasi, 2021). Since intruder nodes can make insider attacks without considering cryptography, a second defence layer is required to provide network security in which system interactions are monitored and relevant alarms are issued upon detecting strange behaviour in the network. Indeed, this defence system not only monitors network behaviour to detect anomalous behaviour of insider attackers but is also applied to detect malicious behaviours of unknown external network attackers (Hoang & Nguyen, 2018). This system can analyse and identify normal behaviour of the IoT network to detect attacks and threats of the insider things, external threats from the Internet and hybrid attacks of these two in interactions of internal things and gateways so that whenever an anomaly is detected, the system is warned so that more damages are prevented. In this regard, system behaviour and network status analysis are the main problems in managing IoT.

The anomaly detection Field is one technique that effectively analyses the collected data stream (Mothukuri et al., 2021). Network intrusion detection systems (IDS) are essential tools for monitoring a network, tracking malicious activities, and identifying intrusions(Maseer et al., 2021). They provide robust defence systems against various threats and cyberattacks. IDS models are classified into anomaly-based and signature-based approaches according to their detection mechanisms. The anomaly-based IDS models rely upon distinguishing abnormal behaviours from normal behaviours (deviations) to detect intrusions; these are effective against polymorphic or unknown attacks. Signature-based IDS models rely on pre-defined patterns of malicious activities and attacks. The large volume and diversity of network traffic as IoT devices, social media applications, and platforms continue to increase [8,9], meaning it is impractical to rely merely on identifying pre-defined attack patterns to detect intrusion of networks. The anomaly-based IDS models are mainly utilised for identifying unknown attacks, such as when pre-defined patterns are absent for IoT networks (Maseer et al., 2021).

For the identification of anomaly detection, one can follow these steps (Mukherjee et al., 2020): F$F \forall sensors\ SR\ in\ N$:

Where,

$$N = \{0, 1, 2, \dots N - 1\} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

$$Anomaly\ =\ (Anticipated\ Outcome\ -\ True\ Outcome) \geq\ threshold \dots\dots\dots\dots\dots (2)$$

Those data can be treated as an anomaly if the difference between the Anticipated and True is more significant than the given threshold. So, if there is an anomaly in our data, someone has interfered with the IoT system. Much recent research has already been done in the field of the Internet of Things for anomaly detection, and machine learning has proven to be extremely useful in this regard. Machine learning enables computers to learn without explicit programming. It is intended to allow a system to learn from the past or the present and to use the knowledge to make future predictions or decisions. Even though "learning" is vital in machine learning, it is not

the objective. The primary aim of machine learning is to create a system that can identify relevant patterns in data automatically and correctly. The most severe issue remains the lack of a dataset. Anomaly detection methods seek to detect unusual behaviours that do not conform to predicted behaviours (Alghanmi et al., 2021), where machine learning algorithms are used to identify the class label or target for the data (average/anomaly). Models are constructed using different keys, e.g. datasets, learning algorithm type, selection of features, and techniques used for evaluation. There are many kinds of attacks possible on IoT systems, which are Denial of Service (Cvitic, 2019), Data Type Probing, Malicious Control, Malicious operation, Scan, Spying, and Wrong set (Mukherjee et al., 2020). The following are some anomalies that can be encountered in a network (Wu et al., 2021): ***Point anomalies*** often refer to an irregularity that happens randomly and may have no particular reason. A ***contextual anomaly*** represents an abnormal behaviour happening within some specific context. Collecting individual data points showing anomalies can be treated as ***collective anomalies***.

Related Works

The evolving landscape of the Internet of Things (IoT) has brought anomaly detection to the forefront of cybersecurity concerns, as Field Al-amri et al. (2021) highlighted. This comprehensive review underscores the increasing importance of anomaly detection in the context of IoT-generated data streams. The paper emphasises the role of machine learning and deep learning techniques in addressing challenges such as evolving data streams, feature evolution, and the need for context-specific anomaly detection systems. The authors call for future research to overcome accuracy, scalability, and high dimensionality hurdles in IoT anomaly detection, providing a roadmap for advancements in the field.

(Chatterjee & Ahmed, 2022) contribute to the discussion by advocating for Intrusion Detection Systems (IDS) to combat malicious attacks within the IoT. Their exploration into Anomaly-based IDS categorises detection approaches, emphasising statistical data anomaly detection. The paper highlights challenges in computation overhead, communication, and time complexity in IoT nodes, stressing the necessity for comprehensive anomaly detection systems that consider the heterogeneity inherent in IoT applications. (Hasan et al., 2019) address security threats by comparing machine learning models for predicting attacks in IoT systems. Despite achieving a commendable 99.4% test accuracy with Random Forest, the paper acknowledges the necessity for further research to enhance detection algorithms and address challenges related to microservices' behaviour and model performance in larger datasets.

In the realm of innovative solutions, (Tsogbaatar et al., 2021) propose DeL-IoT, a deep ensemble learning framework designed for IoT anomaly detection. This framework adeptly manages anomalies and flows and forecasts device status, with potential extensions for detecting multiscale attacks. Additionally, (Chatterjee & Ahmed, 2022) contribute to the collective understanding through a survey spanning 2019 to 2021, identifying challenges such as data integration and advocating for unsupervised approaches in IoT anomaly detection. Together, these research endeavours provide a comprehensive exploration of anomaly detection in IoT, encompassing security considerations, machine learning applications, and the promising potential of deep ensemble learning frameworks.

Diverse approaches to anomaly detection within the Industrial Internet of Things (IIoT) and IoT cybersecurity are explored in various studies. (Member et al., 2019) propose the LSTM-Gauss-NBayes method, (Cook et al., 2019) conduct a survey on anomaly detection for IoT time-series data, and (Aversano et al., 2021) focus on IoT network traffic anomaly detection using a deep learning approach, showcasing high accuracy and stability in the presence of noise. (Alrashdi & Alqazzaz, 2019) address cybersecurity challenges in smart cities with AD-IoT, an Anomaly Detection system utilising Random Forest, while (Xu et al., 2023) propose a data-driven approach for IoT cybersecurity, combining various algorithms and achieving a noteworthy 99.7% accuracy in multi-class classification. (Albulayhi & Sheldon, 2021) introduce an Adaptive Anomaly Detection methodology for IoT cybersecurity, emphasising local-global profiling and achieving higher detection accuracy. (Ullah & Mahmoud, 2022) presents an anomaly detection model for IoT networks focusing on flow and control flag features, demonstrating high accuracy and emphasising the creation of new feature sets. (Fahim, 2019) conducts a systematic literature review on anomaly detection techniques in IoT environments, identifying research gaps and

emphasising the need for further development of new methods. (Anomaly-based et al., 2022) introduce a CNN-based model for anomaly-based Intrusion Detection Systems in IoT networks, achieving high accuracy and advocating for ongoing research to enhance threat detection in evolving IoT environments.

(Materials et al., 2022) introduces an intelligent anomaly detection method based on machine learning, focusing on collaborative feature selection and ensemble learning techniques. (Ahmad et al., 2021) contributes to the discussion by presenting an anomaly detection mechanism for IoT security using a deep neural network (DNN) and mutual information. (Alsoufi, Razak, Siraj, et al., 2021) systematic literature review provides a comprehensive overview of anomaly-based intrusion detection systems (IDS) in IoT using deep learning. (Tyagi et al., 2021) focuses on the critical need for an effective Intrusion Detection System (IDS) in IoT networks, emphasising distinguishing between benign and malicious traffic. (Haji & Ameen, 2021) contribute to the broader understanding of IoT security by reviewing the potential of Machine Learning (ML) algorithms. (Brady et al., 2020) evaluates machine learning (ML) techniques for real-time anomaly detection in IoT environments. (Maniriho et al., 2020) proposes an improved anomaly-based Intrusion Detection System (IDS) approach, integrating a hybrid feature selection engine and employing the Random Forest algorithm for classification. (Ullah et al., 2021) and (Ullah et al., 2022) explore deep learning-based models for anomaly detection in IoT networks, leveraging convolutional neural networks (CNNs) and recurrent neural networks (RNNs) with various techniques. (Timčenko & Gajin, 2018) provides a comprehensive overview of evolving security measures in the dynamic landscape of IoT, showcasing the effectiveness of advanced machine learning methodologies. (Khraisat et al., 2019) proposes a Hybrid Intrusion Detection System (HIDS) merging the Signature Intrusion Detection System (SIDS) and an Anomaly-based Intrusion Detection System (AIDS). (Ma et al., 2014) addresses the challenge of anomaly detection in cloud computing systems within the context of IoT, introducing a deep learning algorithm based on Recurrent Neural Networks (RNN). (Stavros, 2019) focuses on robust anomaly detection methods using Graph Neural Networks (GNN) to secure interconnected IoT devices. (Pathak et al., 2021) delves into security threats in IoT systems, proposing machine learning-based anomaly detection methods. Further expanding into the Industrial Internet of Things (IIoT), (Pathak et al., 2021) explores Graph Neural Networks (GNN) for anomaly detection. (Hoang & Nguyen, 2018) uses a PCA-based method to address network traffic anomaly detection in IoT networks. (Shaver, 2020) focuses on anomaly-based network intrusion detection, employing various machine learning algorithms on the IoT Network Intrusion Dataset. (Mishra & Pandya, 2021) presents a survey on anomaly detection for cyberattacks in the IoT environment, proposing a fog layer. (Mukherjee et al., 2020) addresses security threats and anomaly detection in the IoT domain, employing machine learning techniques for anomaly prediction. (Abbasi, 2021) explores the challenge of detecting and preventing intrusions in IoT networks, emphasising the limitations of traditional IDS. (Alsoufi, Razak, & Ali, 2021) delves into the critical role of security in IoT and the challenges of building effective anomaly intrusion detection systems. (Mothukuri et al., 2021) introduces a Federated Learning (FL)-based anomaly detection system for enhancing IoT security against attacks.

The collective body of research on anomaly detection in IoT environments spans various innovative methodologies, each contributing to the evolving landscape of cybersecurity. These studies address the intricate security challenges embedded within the IoT landscape, offering diverse approaches to anomaly detection and intrusion prevention. The combination of research efforts collectively showcases the effectiveness of advanced machine learning methodologies, including CNN, RNN, SVM, and ensemble methods, in enhancing anomaly detection and preventing cyber threats in IoT environments. Challenges such as imbalanced datasets and the need for real-time detection are recognised, setting the stage for future research and development in IoT security.

Proposed Method

Building upon the existing body of research on anomaly detection in the Internet of Things (IoT) landscape, a novel methodology can be proposed to address the evolving challenges in ensuring the security and integrity of IoT systems. The study aims to develop an advanced Intrusion Detection System (IDS) that integrates a hybrid approach, combining the strengths of base learners or models. This novel methodology seeks to leverage the capabilities of more than one good-performing model for robust anomaly detection, addressing the need for effective threat identification in diverse IoT environments. Building upon the insights from the reviewed literature,

the proposed method aims to address challenges related to imbalanced datasets and improve the accuracy and robustness of anomaly detection systems in IoT environments. The proposed methodology will involve training several machine learning models on the dataset, and based on their evaluation, the best-performing models shall be combined to achieve an ensemble model for the task. As identified in the literature review, the voting ensemble learning technique will leverage the collective decision-making power of three more robust models, including practical demonstration. This ensemble approach will enable the model to generalise well to different aspects of IoT data, enhancing its adaptability to evolving threats and diverse attack scenarios. Combining these models will provide a more comprehensive and accurate anomaly detection system. Figure 2.1 shows the conceptual view of the proposed ensemble model.

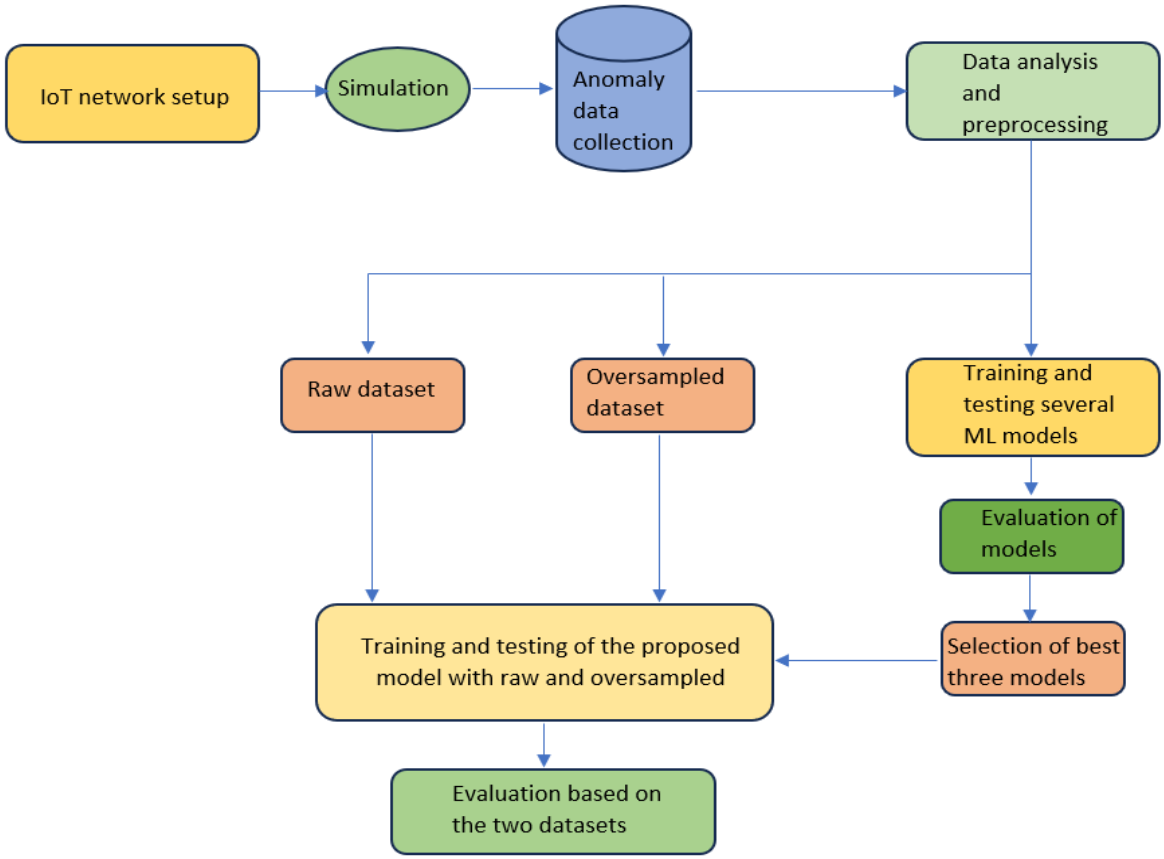

Figure 2.1: Framework of the proposed model

The proposed method will incorporate the SMOTE algorithm for resampling to overcome the challenge of imbalanced datasets inherent in IoT environments. SMOTE will generate synthetic samples for the minority class, balancing the distribution of normal and abnormal instances. This step is crucial for training machine learning models effectively, as imbalanced datasets can lead to biased models that struggle to detect anomalies accurately. Integrating voting ensemble learning and SMOTE aims to create a robust and versatile anomaly detection framework for IoT networks. The ensemble approach ensures diverse perspectives, while SMOTE addresses the data imbalance issue, resulting in a more resilient and accurate model. The proposed method sets the stage for advancing anomaly detection capabilities in IoT's dynamic and heterogeneous landscape, contributing to improved cybersecurity measures and threat prevention. This holistic approach acknowledges the multifaceted challenges discussed in the literature and strives to offer a comprehensive solution for safeguarding IoT networks against evolving security threats.

3.0 Methodology

In this section, we outline the methodology adopted to develop an efficient network anomaly detection system for the Internet of Things (IoT) using machine learning techniques. The goal of our study is to identify relevant patterns in IoT data automatically and accurately. The section covers the data collection method, the machine learning models employed, and the optimisation algorithms used to enhance the best-performing model.

Experimental Set-up and Data Collection

To facilitate the training and evaluation of our network anomaly detection system tailored for the Internet of Things (IoT), we carefully selected a data source that represents real-world IoT network behaviour. The data was acquired from a computer network setup designed to simulate an IoT environment. This network setup comprises various IoT devices connected through a structured network infrastructure, such as sensors, actuators, and communication nodes.

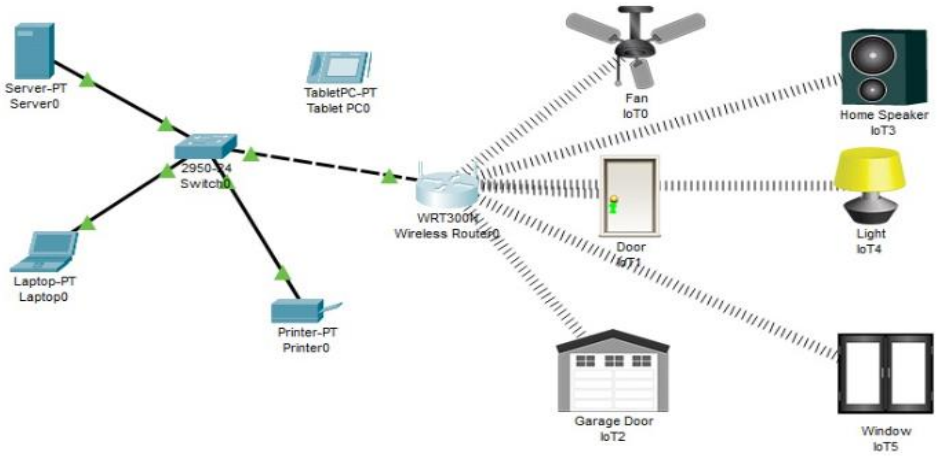

Figure 3: IoT network in Packet Tracer

The data was collected within a simulated network environment using various virtual devices and equipment. As shown in Figure 3, the devices used in the data collection procedure are as follows:

Table 3.0:

S/N	IoT Virtual Network Device	Purpose
1	Server-PT (Server0)	A virtual server utilized to host network services and applications
2	Laptop-PT (Laptop0)	A virtual laptop, which represents a typical user device connected to the network
3	2950-24 (Switch0)	A virtual Cisco switch (2950-24) that plays a pivotal role in network connectivity and management
4	WRT300N (Wireless Router0)	A virtual wireless router used for wireless network connectivity and routing
5	Printer-PT (Printer0)	A virtual printer, emulating a networked printing device
6	Fan (IoT0)	A virtual Internet of Things (IoT) device representing a fan, which is integrated into the network
7	Door (IoT1)	An IoT device simulating a door, contributing to the diversity of network-connected objects

8	Garage Door (IoT2)	Another IoT device, specifically designed to simulate a garage door, further enriching the network environment
9	Home Speaker (IoT3)	An IoT device emulating a home speaker, expanding the range of network-connected devices
10	Light (IoT4)	A virtual IoT light, integrated into the network, representing IoT lighting solutions
11	Window (IoT5)	Another IoT device in the form of a window, adding to the complexity of device interactions

The data collection and simulation activities were carried out in the Asutifi North District Assembly context. This case study setting was chosen to simulate a real-world environment and scenarios typically found in district assemblies and local government offices. It allowed us to evaluate network activities, interactions, and potential anomalies within a practical and relevant context. The data collection procedure involved creating a network infrastructure within the Packet Tracer simulation environment, connecting the devices above, and configuring them to mimic real-world behaviours. The simulation encompassed various aspects of network communication, including data transfers, device interactions, and device-specific activities. Each virtual device was configured to represent its real-world counterpart in the device configuration stage. This included specifying device properties, protocols, and communication settings. The devices were interconnected according to a predefined network topology that emulated a typical district assembly network. Various scenarios and network activities were executed to generate network traffic and device interactions. These activities encompassed routine operations and potential anomaly scenarios. Data logs were developed throughout the simulation to record device interactions, network activities, and communication patterns. The data logs were stored for further analysis and model training. Utilising simulation software and virtual devices provides several advantages, such as precise control over the network environment and scenarios, making it possible to recreate specific conditions for analysis. Furthermore, conducting data collection in a virtual environment ensures the safety of real-world network systems and data. The use of Packet Tracer and the diverse set of virtual devices facilitated realistic data collection in a controlled, secure, and repeatable environment. This dataset forms the foundation for the subsequent stages of our research into network anomaly detection in the context of the Internet of Things.

Data Analysis and Preprocessing

The collected dataset is the fundamental building block for developing our machine learning models. It is essential to provide a diverse and comprehensive range of network activities, device interactions, and communication patterns within the IoT environment. The data encompasses multiple dimensions, including Network Activities, Device Interactions and Communication Patterns. The dataset covers a wide spectrum of network activities, including data transmission, device discovery, command execution, and retrieval. This diversity ensures that the machine learning models can effectively capture various aspects of IoT communication. Different IoT devices, each with their unique communication behaviour, were included in the data collection process. This diversity reflects the reality of an IoT ecosystem, where devices often communicate in distinct ways based on their functions and protocols. The data includes many communication patterns commonly observed in IoT networks. These patterns encompass normal operational behaviour and potential anomalies, ensuring the detection system can distinguish between them effectively. Table 3.1 shows a cross-section of the datatypes captured for the project.

Table 3.1: Cross-Section of the dataset used

Data parameter	Datatype	Number of unique entries	Description

Time	float64	10092	This parameter represents the timestamp or time at which a network event occurred. It is of type float64, indicating a continuous numeric value.
Source address	object	95	This parameter represents the source address of the network communication. It is of type object, suggesting that it is a string (non-numeric format).
Destination address	object	109	Similar to the source address
Protocol	object	18	This parameter indicates the communication protocol used for the network event. It is of type object, and include categorical values such as TLSv1.2, TCP, ARP, DNS, SSDP, etc.
Output	float64	2	This represents the label of the anomaly event. It is a categorical value (0 or 1).
Length	int64	550	This parameter is of type int64, representing an integer. It denotes the length or size of the data associated with the network anomaly event (as the number of bytes transferred).
Information	object	4676	The information parameter is of type object, containing alphanumeric complex information related to the network anomaly event.

The ***Time*** parameter, represented as float64, serves as a continuous timestamp indicating the occurrence of network events. ***Source*** and ***destination*** addresses, both of object type, are non-numeric strings with 95 and 109 unique entries, respectively. The ***Protocol*** parameter, an object type with 18 unique entries, signifies the communication protocol used. The ***Output*** parameter, a float64 with two unique entries (0 or 1), acts as the label for anomaly events. The ***Length*** parameter, an int64 with 550 unique entries, denotes the size of data associated with anomalies. Lastly, the ***Information*** parameter, an object with 4676 unique entries, contains complex alphanumeric details related to the network anomaly events.

After collecting the data, the dataset underwent extensive preprocessing to ensure its quality and suitability for machine learning model development. The preprocessing tasks included data cleaning, data transformation, and feature engineering. During data cleaning, it was found that the Output and info columns contained one and two null values, respectively. By implementing the Python ***drop()*** function, the corresponding rows associated with these null values were removed. With the transformation, all columns with ***object*** datatypes were converted into numerical ones and encoded based on the number of unique entries within the columns. This action makes the dataset suitable for machine learning. Source, Destination, Protocol and Info were the features that underwent numerical encoding. To further enhance the dataset for the machine learning project, feature normalization(B. Li et al., n.d.; Prathyusha, 2021; Umar et al., 2023) was performed after observing the intricate imbalance of standard deviation between individual features. In the Feature engineering stage, the inclusion of features in the machine learning modelling was justified based on the number of unique entries within individual features. Here, since the information column contains 4676 unique entries, this column was not included in the machine learning phase of the project. Hence, the final machine learning model was formulated as:

$$y = f(x_0, x_1, x_2, x_3) \quad \text{..... (2)}$$

Where y represents the Output, x_0 is the Time, x_1 is the Length, x_2 is the Source IP, and x_3 is the Destination IP. The corresponding Python code is:

```
x = data[['Time','Length','SOURCE_IP','DESTINATIONI_IP']]

y = data['Output']
```

the total dataset size, after data cleaning, yielded 16651 rows, of which a splitting ratio of 4:1 was applied using the Python ***test-split ()*** function. The larger portion was used for training, while the smaller was used to validate the machine learning models adopted for the project.

Machine Learning Models

To appreciate and guide the validation of our proposed model, the study first employs a range of machine learning models to identify and classify network anomalies within the IoT dataset. The selected models include Gaussian Naïve Bayes (GNB), Decision Tree (D-Tree), K-Nearest Neighbors (KNN), Random Forest, AdaBoost, XGBoost and Feedforward Neural Network (FFNN). These models offer diverse approaches to anomaly detection, from probabilistic modelling to ensemble learning and neural approach, providing a comprehensive analysis of IoT network behaviour.

Evaluation Metrics

To measure the performance of anomaly detection techniques, the ***'accuracy'*** metric was used (Al-amri et al., 2021). Nevertheless, in the event of imbalanced datasets, the reported accuracy will not accurately represent the technique's efficiency. To measure the performance more accurately, metrics such as True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), precision, recall, and F1 scores are used. TP offers information regarding how many positive cases are accurately detected. TN provides information regarding how many negative cases are accurately labelled as negative cases. FP gives information regarding the falsely labelled cases as positive cases. Similarly, FN offers information regarding positive cases but falsely labelled as negative. Precision is known as the number of class members classified accurately over the total number of cases classified as class members. Recall (Sensitivity) is known as the number of class members classified correctly over the total number of class members. High precision and high recall are needed in anomaly detection to develop a high-quality technique. In such scenarios, the F-measure is applied to provide equal importance to precision and recall. Specificity has recently gained popularity in measuring the performance of the anomaly detection technique. In the context of anomaly detection in IoT (Internet of Things) systems, specificity refers to the ability of the detection system to accurately identify instances that are truly normal or non-anomalous fields (Vartouni, 2018). Specificity is a key performance metric that assesses the system's capability to avoid false positives. In anomaly detection, a false positive occurs when the system incorrectly flags a normal behaviour or event as an anomaly, leading to unnecessary alerts or actions (Schlegl et al., 2019). Furthermore, in anomaly detection problems, the overall accuracy is a metric for performance evaluation; however, it is not appropriate in this case study because an accuracy of 99% is achieved when the imbalance ratio is 1:99 and a stupid classifier discriminates all of the examples as the majority class (Fujiwara et al., 2020). The geometric mean (G-mean) of the sensitivity and the specificity is given by:

$$G_{mean} = \sqrt{sensitivity \times specificity} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

The G-mean measures the classification performance of a classifier for minority class examples as well as majority class examples simultaneously. A low value of the G-mean indicates that the classifier is highly biased toward one class and vice-versa. Thus, the G-mean is an appropriate metric for evaluating the imbalanced data problems (Li et al., 2022).

Handling Imbalance Label with SMOTE Resampling Methods

In situations when learning is unbalanced, resampling techniques sometimes entail adding bias to the dataset (Alam et al., 2020). Although classifiers can learn from unbalanced datasets, it is best to correct the imbalance to get more solid and trustworthy results. All of the credit-related datasets in this investigation have a problem with data imbalance. Additionally, the dataset was adjusted using the SMOTE resampling method until the necessary balancing ratio was reached.

Synthetic Minority Oversampling Technique (SMOTE), a powerful oversampling method that Chawla et al. (Chawla et al, 2002) presented, improves the categorization of minority classes in unbalanced datasets. SMOTE makes it possible to under-sample the majority class while oversampling the minority class. SMOTE creates

synthetic minority data as opposed to earlier techniques, which only reproduce minority class samples and may result in overfitting. By choosing the k (e.g. $k = 5$) nearest neighbors for a given minority data sample, calculating the feature differences between it and a randomly selected neighbor, multiplying this difference by a random number ranging from 0 to 1, and then adding it to the feature vector, this method achieves oversampling of the minority class.

$$x_{new} = x_i + (x_i' - x_i) \times \alpha \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where x_i' is one of the k-nearest neighbors of x_i, and $\alpha \in [0, 1]$ is a real random number.

Based on the intended level of oversampling, SMOTE uses an iterative sampling and perturbation procedure to create synthetic minority data samples. For instance, a 200% oversampling would independently disturb a sample along the vectors of two different nearest neighbors to produce two new synthetic minority samples. SMOTE also gives users the option to undersample the majority class, bringing it down to a predetermined portion of the minority class's initial sample size. Depending on the amount of over and under-sampling used, the size of the generated dataset may change, perhaps leading to more or less samples in the minority class than the original data. SMOTE can be modified to work with categorical variables. SMOTE estimates nearest neighbors in situations with mixed categorical and continuous variables, like our datasets, by first figuring out the median of standard deviations for continuous features inside the minority class. The previously determined median is taken into account when calculating the Euclidean distance between samples if there are differences in categorical variables between a sample and its probable nearest neighbors. The majority occurring values among the nearest neighbors are allocated to synthetic categorical features once the k nearest neighbors have been determined, and continuous variables are created similarly to how the original data was calculated. In contrast to the restricted, specialized regions produced by the replication of minority classes, SMOTE develops broader decision regions by establishing synthetic minority classes. It's important to note that this method guarantees that the synthesized data stays within the confines of the extreme values reported in the real data because the perturbation factor ranges from 0 to 1.

4.0 RESULTS AND DISCUSSION

In this section, we present the results of our analysis, evaluating the performance of various machine learning models in the context of our classification task. We have assessed the models based on four key performance metrics: Accuracy, Sensitivity, Specificity, and the Geometric Mean (G-Mean). These metrics provide a comprehensive view of each model's ability to correctly classify and discriminate between classes.

Table 4.1: performance of models in the first

MODEL	ACCURACY	SENSITIVITY	SPECIFICITY	GMEAN
GNB	95.977	3.570	96.761	18.5896
D-TREE	97.5983	71.212	98.131	83.595
KNN	97.688	71.232	98.251	83.671
R-FOREST	98.109	75.862	98.705	86.533
ADABOOST	96.907	66.667	97.044	80.434
XGBOOST	97.928	75.320	98.463	86.120
FFNN	96.997	75.000	97.104	85.330
PROPPOSED MODEL	98.14	78.75	98.62	88.12

Our analysis reveals varying levels of performance across the different machine learning models. Random Forest (R-FOREST) demonstrates the highest accuracy, sensitivity, specificity, and G-Mean, making it the top-performing model for our classification task. Decision Tree (D-TREE) and k-Nearest Neighbors (KNN) also

exhibit strong performance, particularly in sensitivity and specificity. However, the choice of the most suitable model depends on the specific requirements of the task and the trade-offs between these performance metrics. The results presented in table 4.1 as well as figure 4.1 provide valuable insights into the capabilities of each model and serve as a basis for selecting the most appropriate machine learning approach for our proposed new model based on voting ensemble technique.

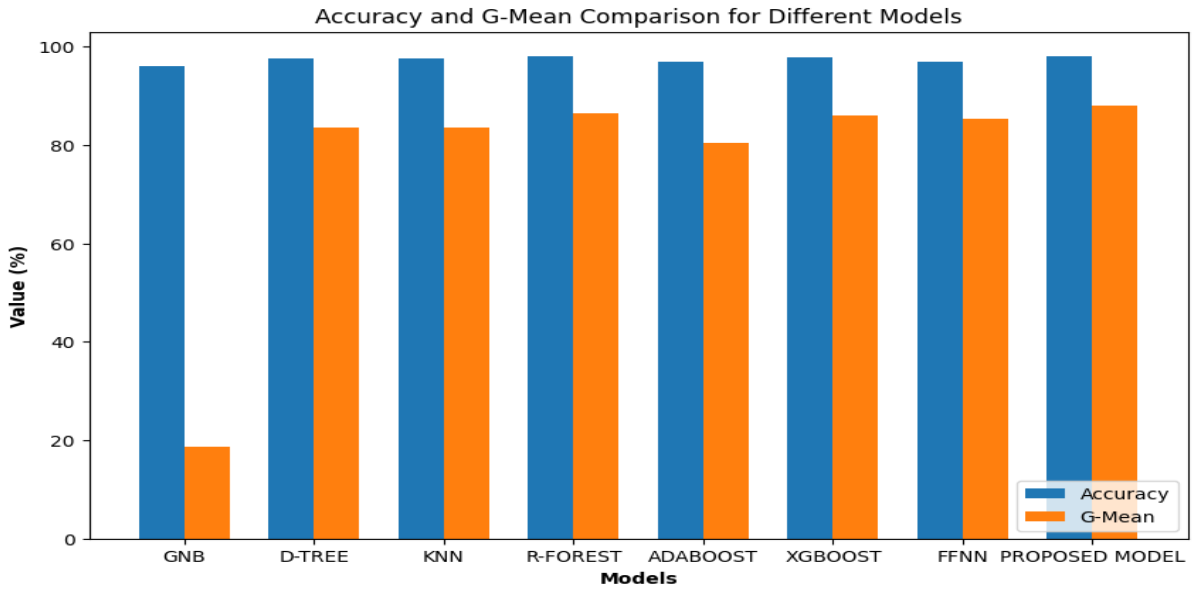

Figure 4.1: performance visualization of the models

Random Forest emerges as the top-performing model with an impressive accuracy of 98.11%, showcasing its ability to correctly classify normal and anomalous network behaviour instances. This high accuracy is complemented by a notable sensitivity of 75.86%, indicating the model's proficiency in identifying actual cases of network anomalies. The specificity of 98.71% further solidifies Random Forest's reliability, as it demonstrates a low rate of false positives. Decision Tree and K-Nearest Neighbors (KNN) models also exhibit commendable performance, with accuracies of 97.60% and 97.69%, respectively. These models strike a good balance between sensitivity and specificity, with Decision Tree achieving a sensitivity of 71.21% and KNN achieving 71.23%. Random Forest surpasses them in both accuracy and sensitivity, making it a more robust choice for IoT network anomaly detection. The G-Mean (Geometric Mean) values provide a consolidated measure of a model's overall performance, considering sensitivity and specificity. Random Forest achieves the highest G-Mean at 86.53%, highlighting its effectiveness in maintaining a harmonious trade-off between correctly identifying anomalies and avoiding false alarms. Overall, these results emphasize the importance of selecting a model with a balanced combination of accuracy, sensitivity, and specificity for effective IoT network anomaly detection.

4.1 Performance of the proposed model

An innovative approach to anomaly detection has been proposed through a sophisticated ensemble model. The model integrates the top three powerful algorithms, namely Random Forest, XGBoost, and K-Nearest Neighbors (KNN), from the first training and evaluation, forming a robust defence mechanism against network intrusions and irregularities. The foundation of this model lies in the collective decision-making prowess of these algorithms, orchestrated by a hard voting criterion. To refine the performance of each algorithm, a meticulous optimization strategy is employed, harnessing the grid search algorithm for hyperparameter tuning. For instance, hyperparameters, such as the number of estimators and maximum depth for Random Forest or the learning rate for XGBoost, are systematically tuned to enhance the individual models' predictive capabilities. Results from extensive evaluations showcase the remarkable efficacy of this ensemble model in IoT network anomaly

detection. The accuracy of 98.14% underscores its proficiency in discerning between normal and anomalous network behaviours. Moreover, the model demonstrates a commendable sensitivity of 78.75%, ensuring it can effectively identify instances of network anomalies. Equally important is its high specificity, measured at 98.62%, signifying a low rate of false positives. The ensemble's balanced performance is encapsulated by a G-Mean of 88.12%, affirming its prowess in maintaining equilibrium between sensitivity and specificity, making it a formidable asset in fortifying IoT networks against potential security threats.

4.2 Effect of data imbalance and resampling

The effect of resampling on the Internet of Things (IoT) dataset significantly impacted the performance of the newly proposed machine learning model. Before resampling the dataset exhibits an imbalance ratio of 3.1: 96.9 for the output label as shown in figure 4.2. Hence, the newly proposed model demonstrates a sensitivity of 78.75%, specificity of 98.62%, and a G-means score of 88.12. These metrics represent the model's ability to accurately identify positive instances, correctly classify negative instances, and overall balance between sensitivity and specificity.

Table 4.2: performance of the newly proposed model before and after resampling

Before Resampling			
Accuracy	**Sensitivity**	**Specificity**	**G-Means**
98.14	78.75	98.62	88.12
After Resampling with SMOTE			
Accuracy	**Sensitivity**	**Specificity**	**G-Means**
98.26	81.25	98.96	89.51

After applying the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance in the dataset, notable improvements were observed. The sensitivity increased to 81.25%, indicating a better ability to capture true positive instances. The specificity also improved to 98.96%, reflecting enhanced accuracy in identifying true negative instances. Consequently, the G-means score rose to 89.51, indicating an overall improvement in the model's performance, particularly in maintaining a balance between sensitivity and specificity.

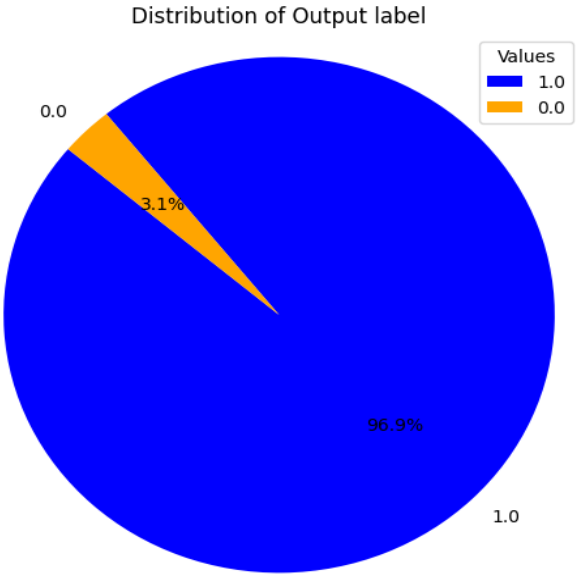

Figure 4.2: Visualization of the output label before resampling

This positive impact suggests that addressing class imbalance through resampling techniques like SMOTE contributes to a more robust and accurate machine learning model for IoT anomaly detection. The model becomes more adept at handling positive and negative instances, crucial for ensuring reliable and effective anomaly detection in IoT systems.

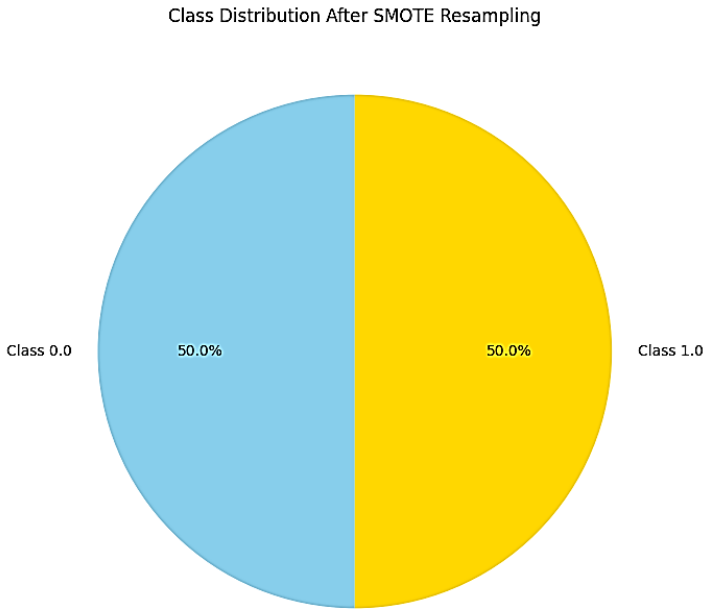

Figure 4.3: Visualization of the output label after SMOTE resampling

4.3 Discussion

One of the key observations from our findings is the substantial performance variation among the individual machine learning models employed. Random Forest emerged as the top performer, showcasing the highest accuracy, sensitivity, specificity, and G-Mean. This reinforces the notion that ensemble methods, specifically Random Forest, excel in capturing complex patterns within IoT network data. The decision tree-based nature of Random Forest makes it adept at handling diverse communication patterns exhibited by different IoT devices in the simulated environment. The proposed ensemble model, a combination of Random Forest, XGBoost, and K-Nearest Neighbors, demonstrated a synergistic effect, surpassing the individual models in various performance metrics. This indicates the value of integrating diverse machine-learning approaches to achieve a more robust and adaptable anomaly detection system. The ensemble model's high accuracy and balanced sensitivity and specificity suggest its potential for real-world deployment in securing IoT networks effectively. The impact of addressing class imbalance through the Synthetic Minority Over-sampling Technique (SMOTE) was evident in the results. The model's performance significantly improved after resampling, particularly in terms of sensitivity and specificity. This underscores the importance of handling class imbalance to ensure a more reliable and accurate anomaly detection system for IoT environments. Furthermore, the study's simulated environment, mimicking a district assembly network, adds a layer of real-world relevance to the findings. The diverse set of virtual IoT devices and network activities captured in the dataset reflects the complexity and heterogeneity of actual IoT ecosystems. This lends credibility to the generalizability of the developed anomaly detection system across different IoT scenarios. Looking forward, the proposed model presents avenues for further exploration and optimization. Future research could focus on enhancing the ensemble model by incorporating additional state-of-the-art machine learning algorithms. Furthermore, efforts should be directed toward real-world deployments and validations to assess the model's adaptability to diverse IoT environments and its scalability. Additionally, dynamic anomaly detection mechanisms must be explored to ensure the model's resilience to changes in IoT

network behaviour over time. Continuous monitoring and adaptation will be crucial for maintaining the effectiveness of the anomaly detection system in the face of evolving threats and network dynamics. In conclusion, the discussion chapter emphasizes the significance of the study's findings, highlighting the superiority of the ensemble model, the impact of addressing class imbalance, and the real-world relevance of the simulated environment. It sets the stage for future research endeavours, guiding the field toward more sophisticated, adaptive, and deployable anomaly detection solutions for the ever-expanding realm of the Internet of Things.

Table 4.3: Comparison of the study to the literature

Citation	Best Model	Accuracy
(Materials et al., 2022)	XGboost and LSTM	99.984%
(Maniriho et al., 2020)	Random Forest	99.97%
(Anomaly-based et al., 2022)	CNN	92.85%
(Mukherjee et al., 2020)	Random Fores	99.4%
(Ahmad et al., 2021)	DNN	99.01%

As shown in Table 4.3, though most of the models spanning from ensemble learners to CNN performed well in terms of accuracy, they were not evaluated against sensitivity, specificity and g-means to clarify the robustness of the models on anomaly detection.

5.0 Conclusion And Future Works

In conclusion, this study has comprehensively explored the development of an efficient network anomaly detection system for the Internet of Things (IoT) using advanced machine learning techniques. The methodology employed in this research encompassed various crucial aspects, including data collection, preprocessing, feature engineering, and the evaluation of multiple machine learning models. The findings shed light on the performance of individual models and the proposed ensemble model, providing valuable insights into their efficacy for IoT network anomaly detection. The experimental results revealed that Random Forest emerged as the top-performing model among the individual machine learning algorithms. With an impressive accuracy of 98.11%, high sensitivity (75.86%), specificity (98.71%), and G-Mean (86.53%), Random Forest demonstrated its ability to accurately classify instances of both normal and anomalous network behaviour. Decision Tree and K-Nearest Neighbors (KNN) also showcased commendable performances, indicating that various machine-learning approaches can contribute effectively to IoT anomaly detection. The proposed ensemble model, a strategic amalgamation of Random Forest, XGBoost, and K-Nearest Neighbors, surpassed the individual models, showcasing an accuracy of 98.14%, sensitivity of 78.75%, specificity of 98.62%, and a G-Mean of 88.12%. This ensemble model harnessed the collective decision-making capabilities of diverse algorithms, orchestrating them through a hard voting criterion. The meticulous optimization strategy employed, utilizing grid search for hyperparameter tuning, further enhanced the predictive capabilities of each algorithm. Moreover, this study addressed the critical issue of class imbalance in the dataset by leveraging the Synthetic Minority Over-sampling Technique (SMOTE). The results after resampling demonstrated a notable improvement in sensitivity, specificity, and G-Mean. Sensitivity increased to 81.25%, showcasing the model's enhanced capability to capture true positive instances, while specificity improved to 98.96%, reflecting heightened accuracy in identifying true negative instances. The G-Mean score rose to 89.51%, indicating an overall improvement in the model's performance, particularly in maintaining a balance between sensitivity and specificity.

The following can be addressed in the aspect of future works.

1. Future research could delve into the exploration and optimization of more sophisticated ensemble models. Integrating additional machine learning algorithms and refining voting mechanisms could potentially lead to even higher performance levels.

2. Taking the proposed model from simulated environments to real-world IoT setups is crucial for validating its effectiveness. Deployment in diverse IoT scenarios will assess adaptability, scalability, and generalizability.
3. Investigating techniques for adapting the anomaly detection system to dynamic changes in IoT network behaviour is essential. This ensures the model's ongoing effectiveness in evolving and dynamic environments.
4. Advanced hyperparameter tuning and optimization strategies can be explored further to fine-tune individual machine-learning models within the ensemble. This can potentially maximize their predictive capabilities.
5. Addressing potential vulnerabilities and security concerns associated with anomaly detection systems is paramount. Ensuring the robustness of the proposed model against adversarial attacks will be crucial for real-world applications.

References

[1] Abbasi, F. (2021). *Anomaly detection in Internet of Things using feature selection and classification based on Logistic Regression and Artificial Neural Network on N-BaIoT dataset*. *IoT*.

[2] Ahmad, Z., Khan, A. S., Nisar, K., Haider, I., Hassan, R., Haque, M. R., Tarmizi, S., & Rodrigues, J. J. P. C. (2021). *applied sciences Anomaly Detection Using Deep Neural Network for IoT Architecture*.

[3] Al-amri, R., Murugesan, R. K., Man, M., & Abdulateef, A. F. (2021). *applied sciences A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data*.

[4] Alam, T. M., Shaukat, K., Hameed, I. A., Member, S., Luo, S., Sarwar, M. U., Shabbir, S., Li, J., & Khushi, M. (2020). *An Investigation of Credit Card Default Prediction in the Imbalanced Datasets*. *8*, 201173–201198. https://doi.org/10.1109/ACCESS.2020.3033784

[5] Albulayhi, K., & Sheldon, F. T. (2021). *An Adaptive Deep-Ensemble Anomaly-Based Intrusion Detection System for the Internet of Things*. *November 2022*. https://doi.org/10.1109/AIIoT52608.2021.9454168

[6] Alghanmi, N., Alotaibi, R., & Buhari, S. M. (2021). Machine Learning Approaches for Anomaly Detection in IoT : An Overview and Future Research Directions. *Wireless Personal Communications*, *0123456789*. https://doi.org/10.1007/s11277-021-08994-z

[7] Alrashdi, I., & Alqazzaz, A. (2019). *AD-IoT : Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning*. 1–6.

[8] Alsoufi, M. A., Razak, S. A., & Ali, A. (2021). *Anomaly Intrusion Detection Systems in IoT Using Deep Learning Techniques : Anomaly Intrusion Detection Systems in IoT Using Deep Learning Techniques : A Survey*. *May*. https://doi.org/10.1007/978-3-030-70713-2

[9] Alsoufi, M. A., Razak, S., Siraj, M., Nafea, I., & Ghaleb, F. A. (2021). *applied sciences Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning : A Systematic Literature Review*.

[10] Anomaly-based, S. A., Citation, A., & Engineering, E. (2022). *LJMU Research Online Anomaly-based Intrusion Detection System for IoT Networks through Deep Learning Model*.

[11] Aversano, L., Bernardi, M. L., Cimitile, M., Pecori, R., & Veltri, L. (2021). *Effective Anomaly Detection Using Deep Learning in IoT Systems*. *2021*(i).

[12] Brady, S., Magoni, D., Murphy, J., Assem, H., Omar, A., Portillo-dominguez, O., Brady, S., Magoni, D., Murphy, J., Assem, H., Portillo-dominguez, A. O. O., Brady, S., Magoni, D., Murphy, J., Assem, H., & Portillo-dominguez, A. O. (2020). *Analysis of Machine Learning Techniques for Anomaly Detection in the Internet of Things To cite this version : HAL Id : hal-02493464 Analysis of Machine Learning Techniques for Anomaly Detection in the Internet of Things*.

[13] Chatterjee, A., & Ahmed, B. S. (2022). *Internet of Things IoT anomaly detection methods and applications : A survey*. *19*(July), 1–17.

[14] Cook, A., Fan, Z., & Member, S. (2019). *Anomaly Detection for IoT Time-Series Data : A Survey*. https://doi.org/10.1109/JIOT.2019.2958185

[15] Cvitic, I. (2019). *Novel approach for detection of IoT generated DDoS traffic*. *1*. https://doi.org/10.1007/s11276-019-02043-1

[16] Fahim, M. (2019). Anomaly Detection , Analysis and Prediction Techniques in IoT Environment : A Systematic Literature Review. *IEEE Access*, *7*, 81664–81681. https://doi.org/10.1109/ACCESS.2019.2921912

[17] Fujiwara, K., Huang, Y., Hori, K., Nishioji, K., & Kobayashi, M. (2020). *Over- and Under-sampling Approach for Extremely Imbalanced and Small Minority Data Problem in Health Record Analysis*. *8*(May), 1–15. https://doi.org/10.3389/fpubh.2020.00178

[18] Haji, S. H., & Ameen, S. Y. (2021). *Attack and Anomaly Detection in IoT Networks using Machine Learning Techniques : A Review Attack and Anomaly Detection in IoT Networks using Machine Learning Techniques : A Review*. *June*. https://doi.org/10.9734/ajrcos/2021/v9i230218

[19] Hasan, M., Islam, M., Zarif, I. I., & Hashem, M. M. A. (2019). Internet of Things Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, *7*, 100059. https://doi.org/10.1016/j.iot.2019.100059

[20] Hoang, D. H., & Nguyen, D. H. (2018). *A PCA-based Method for IoT Network Traffic Anomaly Detection*. *February*. https://doi.org/10.23919/ICACT.2018.8323766

[21] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). *A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks*.

[22] Li, B., Wu, F., Serge, S. L., & Tech, C. (n.d.). *On Feature Normalization and Data Augmentation*. 12383–12392.

[23] Li, T., Kou, G., Peng, Y., Yu, P. S., & Fellow, L. (2022). An Integrated Cluster Detection , Optimization , and Interpretation Approach for Financial Data. *IEEE Transactions on Cybernetics*, *52*(2), 13848–13861. https://doi.org/10.1109/TCYB.2021.3109066

[24] Ma, B., Charkowski, A. O., Glasner, J. D., & Perna, N. T. (2014). *Identification of host-microbe interaction factors in the genomes of soft rot-associated pathogens Dickeya dadantii 3937 and Pectobacterium carotovorum WPP14 with supervised machine learning*. 1–18.

[25] Maniriho, P., Niyigaba, E., Bizimana, Z., Twiringiyimana, V., Mahoro, L. J., Ahmad, T., & Africa, S. (2020). *Anomaly-based Intrusion Detection Approach for IoT Networks Using Machine Learning*. *Cenim*.

[26] Maseer, Z. K., Yusof, R., Mostafa, S. A., Bahaman, N., Musa, O., & Al-rimy, B. A. S. (2021). *DeepIoT . IDS : Hybrid Deep Learning for Enhancing IoT Network Intrusion Detection*. https://doi.org/10.32604/cmc.2021.016074

[27] Materials, C., Alanazi, M., Corporation, V. T., & Aljuhani, A. (2022). *Anomaly Detection for Internet of Things Cyberattacks Anomaly Detection for Internet of Things Cyberattacks*. *May*. https://doi.org/10.32604/cmc.2022.024496

[28] Member, D. W., Jiang, Z., Xie, X., & Member, X. W. (2019). *LSTM Learning with Bayesian and Gaussian Processing for Anomaly Detection in Industrial IoT*. *3203*(c). https://doi.org/10.1109/TII.2019.2952917

[29] Mishra, N., & Pandya, S. (2021). Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review. *IEEE Access*, *9*, 59353–59377. https://doi.org/10.1109/ACCESS.2021.3073408

[30] Mothukuri, V., Khare, P., Parizi, R. M., Member, S., Pouriyeh, S., & Member, A. (2021). *Federated Learning-based Anomaly Detection for IoT Security Attacks*. *4662*(c), 1–10. https://doi.org/10.1109/JIOT.2021.3077803

[31] Mukherjee, I., Sahu, N. K., & Sahana, S. K. (2020). Machine Learning based anomaly detection for IoT Network : (Anomaly detection in IoT Network) Simulation and Modeling for Anomaly Detection in IoT Network Using Machine Learning. *International Journal of Wireless Information Networks*, *January 2022*. https://doi.org/10.1007/s10776-021-00542-7

[32] N. V. Chawla, K.W. Bowyer, L. O. Hall, andW. P. K. (2002). SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, *16*, 321–357.

[33] Pathak, A. K., Saguna, S., Mitra, K., & Ahlund, C. (2021). *Anomaly Detection using Machine Learning to Discover Sensor Tampering in IoT Systems*.

[34] Prathyusha, K. S. (2021). *Normalization Methods for Multiple Sources of Data*. *Iciccs*, 1013–1019.

[35] Said, A. M., & Yahyaoui, A. (2021). *Efficient Anomaly Detection for Smart Hospital IoT Systems*. 1–24.

[36] Schlegl, T., Waldstein, S. M., & Langs, G. (2019). *f-AnoGAN : Fast Unsupervised Anomaly Detection with Generative Adversarial Networks f-AnoGAN : Fast unsupervised anomaly detection with generative adversarial networks*. *January*. https://doi.org/10.1016/j.media.2019.01.010

[37] Shaver, A. (2020). *Anomaly Detection on loT Network Intrusion Using*.

[38] Stavros, A. P. (n.d.). *A Graph Neural Network Method for Distributed Anomaly Detection in IoT*.

[39] Timčenko, V., & Gajin, S. (2018). *Machine Learning based Network Anomaly Detection for IoT environments*.

[40] Tsogbaatar, E., Bhuyan, M. H., Taenaka, Y., & Fall, D. (2021). *DeL-IoT : A Deep Ensemble Learning Approach to Uncover Anomalies in IoT*.

[41] Tyagi, H., Islamia, J. M., Kumar, R., & Islamia, J. M. (2021). *Attack and Anomaly Detection in IoT Networks Using Supervised Machine Learning Approaches Revue d ' Intelligence Artificielle Attack and Anomaly Detection in IoT Networks Using Supervised Machine Learning Approaches*. *June*, 10–21. https://doi.org/10.18280/ria.350102

[42] Ullah, I., & Mahmoud, Q. H. (2022). *An Anomaly Detection Model for IoT Networks based on Flow and Flag Features using a Feed- Forward Neural Network*. *November*. https://doi.org/10.1109/CCNC49033.2022.9700597

[43] Ullah, I., Mahmoud, Q. H., & Member, S. (2021). *Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks*. *9*.

[44] Ullah, I., Mahmoud, Q. H., & Member, S. (2022). Design and Development of RNN Anomaly Detection Model for IoT Networks. *IEEE Access*, *10*, 62722–62750. https://doi.org/10.1109/ACCESS.2022.3176317

[45] Umar, M. A., Zhanfang, C., Umar, M. A., & Zhanfang, C. (2023). *Effects of Feature Selection and Normalization on Network Intrusion Detection Effects of Feature Selection and Normalization on Network Intrusion Detection*.

[46] Vartouni, M. (2018). *An Anomaly Detection Method to Detect Web Attacks Using Stacked Auto-Encoder*. 131–134.

[47] Wu, Y., Member, S., Dai, H., Member, S., & Tang, H. (2021). *Graph Neural Networks for Anomaly Detection in Industrial Internet of Things Graph Neural Networks for Anomaly Detection in Industrial Internet of Things*.

[48] Xu, H., Sun, Z., Cao, Y., & Bilal, H. (2023). A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Computing*, *4*(July). https://doi.org/10.1007/s00500-023-09037-4