IoT Malicious Traffic Classification and Detection Using Machine Learning Algorithms

Seyyed Mohammad Ali Abolmaali[a,*] - Reza Mohammadi[a] - Mohammad Nassiri [a]

[a] Department of Computer Engineering, Engineering Faculty, Bu-Ali Sina University, Hamedan, Iran

ABSTRACT

The Internet of Things (IoT) is one of today's most rapidly growing technologies. The exchange of data between IoT devices generates a large amount of information that needs to be shared. There is a potential for security breaches in these communications, which could be deliberately damaging to the connected devices. It is crucial to detect and deal with unauthorized communication and security breaches in order to avoid further harm and repercussions. The goal of this project is to differentiate deliberate communications from unsecure communications among the IoT devices. Different patterns can be observed in intentional communications compared to insecure communications. Machine learning based on artificial intelligence can be used to detect these patterns in intentional and insecure communication.

In this The paper utilizes Random Forest, Decision Tree, and SVM to differentiate between patterns associated with intended and unintended messages. The performance of the machine learning approach proposed was evaluated by utilizing the Aposemat IoT-23 dataset, and it achieved a 99.25% accuracy when compared to the benchmark dataset. It is found that the suggested Random Forest approach performs better than the current ones when there are enough patterns to recognize. A potential solution to be applied on this dataset is also explored and proposed in order to improve the performance of the underperforming classifiers on the imbalanced dataset. Employing machine learning models makes it possible to detect and mitigate IoT malware threats, ultimately safeguarding the integrity and privacy of IoT devices and networks. This paper contributes to the growing body of knowledge in IoT security and provides a foundation for further research in this critical domain.

Keywords: **IoT Devices, Machine learning, Artificial Intelligence, Traffic classification, Malware Analysis, IoT Security**

1. *INTRODUCTION*

Internet of Things (IoT) is a network of physical objects that can be connected to the Internet and have sensors to collect data, identifiers to identify the data's source, and software to analyze the data [1]. Adoption of the IoT has been progressively growing in recent years. Nonetheless, in 2019 there was a 900 percent increase in IoT attacks [2]. Given that IoT devices are more susceptible to hacking than traditional computers, hackers are targeting them. Up to 98% of all IoT device network traffic is not encrypted, in contrast to traditional computers, which come with both a firewall and a virus scanner [3].

Today, Internet of Things (IoT) devices are widely used in critical infrastructure sectors like industrial control systems, power grids, and healthcare. Due to the additional network entry points this integration brought, there is now a greater security risk [25], [26].

An individual compromised device has the potential to gain access to internal networks, expose businesses and infrastructure to significant security breaches, and result in the loss of sensitive and valuable data, such as access credentials and financial records. Ransomware and other more destructive malware can even lead to the failure of military and medical equipment, putting lives in danger or permitting breaches of national security. This presents a significant challenge to the research community and the public sector as a whole [27].

Prior to causing any kind of loss or harm to the organizations, these security threats ought to be recognized. Nonetheless, malevolent actors persistently devise novel tactics to obscure their assaults and evade

identification. This is demonstrated by the rising number of IoT-related attacks that are documented each year [28], [29].

The rapid increase in the use of IoT devices brings many benefits to the digital society, ranging from improved efficiency to higher productivity. However, the limited resources and the open nature of these devices make them vulnerable to various cyber threats. A single compromised device can havean impact on the whole network and lead to major security and physical damages.

The number of commonplace devices that have sensors built into them and are capable of internet-based communication has significantly increased in recent years. IoT Business News reported that the number of devices connected to the Internet of Things is expected to reach 241 billion by 2030, with this number expected to increase daily [14]. By combining digital intelligence with physical devices, the internet of things makes the world a smarter place. The Internet of Things, as defined by the International Telecommunication Union (ITU-T), is a global network of interconnected devices built on information and communication technologies [15]. In the Internet of Things, security is the main concern due to the massive amount of data that is exchanged between connected devices.

IoT devices are vulnerable to various types of cyberattacks since they link objects to the internet and allow them to communicate with one another without the need for human intervention. Early on in the design and implementation of IoT devices, appropriate security requirements should be determined in order to guarantee the security of the network and devices connected to it [4]. Since the Internet of Things is still in its infancy, there is a risk to sensitive data because it does not yet have a strong security infrastructure or mechanism. To keep IoT entities, organizations, and individuals safe, modern security techniques must be implemented on IoT networks.

Furthermore, a majority of IoT devices on the market are constructed with identical low security mechanisms. The major reason for this is that the majority of companies that manufacture IoT devices lack knowledge of IT security. The global economy has been positively impacted by the Internet's advancements and the development of technologies, leading to the integration of the Internet of Things into our daily lives and improving overall quality of life.

Identification of anomaly and malicious traffic in the Internet of things network is essential for IoT security. Tracking and blocking unwanted traffic flows in the IoT network is required to design a framework for the identification of attacks more accurately, quickly, and with less complexity. Many machine learning (ML) algorithms proved their efficiency to detect intrusion in IoT networks.

The network now has nearly 35 billion diverse IoT devices connected, with a record-breaking increase of 5 billion IoT devices per year. IoT links specialized applications with universal services by enabling sensors and actuators to communicate directly for service provision. Identification, sensing and control, communications, computation, services, and semantics are the six elements of the Internet of things [50]. Internet of Things (IoT) sensors use noisy and lossy communication channels to measure and collect data, which is then sent to a database or the cloud for services.

Upon activation, these networked devices run real-time operating systems to process the gathered data. In order to provide the right services, semantics makes sure the collected data is sent to the right resource. Communication in the Internet of Things (IoT) is becoming more popular as various devices are linked to the network. Communication often employs various technologies, such as RFID, WSN using Wi-Fi, along with internet speeds like 3G, 4G, and 5G to support networks of varying sizes [50].

In a survey on the commercial adoption of IoT in Canada, 40% of participants stated that their company uses IoT solutions, and 22% indicated that they plan to integrate IoT solutions [5]. IoT communication is vulnerable to a number of security threats, including replay, eavesdropping, time attacks, Denial of Service (DoS), Man-in-the-Middle, storage space attacks, cross-site scripting, malicious code, and malicious insider attacks. Security solutions such as cryptography hash-based solutions, secure authorization, embedded security, identity-based management, intrusion detection systems, and access control mechanisms are being used to combat these attacks. The findings presented herein contribute to the ongoing efforts to secure the ever-expanding IoT ecosystem and protect the privacy and safety of IoT users worldwide.

1.1. Problem statement

Beyond the strategies for securing IoT devices from potential threats, ensuring performance requires the classification of network traffic as a priority. This classification leads to improved security in communication and the protection of confidential data that is shared. This traffic classification is instrumental in smart IoT settings, like smart cities, smart homes, and smart automobiles.

1.2 Advances made in the field's development

1. *This study has introduced a system that can automatically differentiate between benign and malicious IoT communications.*
2. *Researched different machine learning methods to identify the most appropriate algorithm for categorizing IoT traffic.*
3. *In-depth examination of the results produced by the algorithms.*
4. *a solution to the IoT-23 dataset's data imbalance issue was put forth.*

The IoT-23 dataset, a valuable resource for this study, offers a collection of network traffic captures from IoT devices. Developed by Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga in collaboration with Avast Software, this dataset contains real-world instances of both benign and malicious IoT network traffic. It comprises twenty captures of malicious scenarios executed on infected IoT devices, each associated with the specific malware sample employed. Additionally, three captures represent benign IoT devices, which include a Philips HUE smart LED lamp, an Amazon Echo home intelligent personal assistant, and a Somfy smart door lock. These devices are not mere simulations; they are actual hardware, allowing us to examine and analyze real network behavior in controlled environments.

2. *RELATED WORK*

The Internet of Things (IoT) is defined by the Cambridge Dictionary as computing devices that are networked and capable of exchanging data [6]. IoT devices can communicate with humans and other IoT devices via the Internet because they are typically composed of actuators, sensors, and other programmable components [7]. The three primary categories of IoT devices are industrial, enterprise, and consumer [8]. IoT devices are used in industry to monitor manufacturing processes to ensure they are operating efficiently.

IoT devices will identify any errors that occur throughout the process and notify the technician of the issue's root cause. In addition, IoT devices utilized by the company can support meetings. When sensors are installed in conference rooms at large companies, staff members may be able to determine whether the space is available for use and whether it is appropriate for the meetings that are scheduled. Consumers may find IoT devices in smart homes, where they can use data from sensors to change the lighting, temperature, and other aspects of the space. A generic architectural design is used by the majority of IoT devices [9].

There has also been some research on intrusion detection and anomaly detection systems for IoT. A whitelist-based intrusion detection system for IoT devices (Heimdall) has been presented in [30]. The authors in [31] propose an intrusion detection model for IoT backbone networks leveraging two-layer dimension reduction and two-tier classification techniques to detect U2R (User-to-Root) and R2L (Remote-to-Local) attacks.

Malware analysis in the Internet of Things is performed using static, dynamic, and hybrid analysis techniques. The authors in [34] were the first to perform malware analysis based on gray-scale images in 2011. Visual images of malware are created by rewriting the eight-bit code value of executable files to the corresponding gray-scale value. Texture features are extracted from these images. The authors in [35] proposed an approach for analyzing malware using texture images of malware files and machine learning in IoTPOT for Bashlite and Mirai [36]. They provided Haralick texture features from the cooccurrence matrix and used machine learning classifiers.

Identifying network traffic is crucial solely for security reasons and is vital for overseeing, billing, allocating network resources, and ensuring service availability. Generally, traffic classification can be done using various approaches, such as port-based, statistical techniques, behavior-based, and payload-based methods. The authors in [42] The methods mentioned, machine learning based approaches are extensively

utilized. This study utilizes behavioral classification methods to analyze the traffic patterns among IoT devices. DDoS attacks, breaches, and unusual or harmful actions can occur in all IoT-based network communications.

Possible attacks on IoT devices when connected to cloud processors include DOS, Jamming, and Buffer Overflow, posing a risk to their security. The authors in [43] The classification of the assaults is achieved using conventional machine learning techniques like Learning Vector Quantization (LVQ), Radial Basis Function (RBN), and Multilayer Perceptron (MLP). The classifiers identify the type of security breaches that took place in the communications.

Most studies that take advantage of the IoT-23 dataset answer the question: "What are the best machine learning algorithms for detecting [or classifying] anomalies [malicious traffic] generated by IoT devices?" Multiple studies showed high accuracy (95% or greater) with Random Forests. The authors in [32] used this model to classify an imbalanced dataset of approximately 9,300 malware and their variants using a stratified sampling method to prevent overfitting and undergeneralization. They also converted the binaries into 8-bit vectors that were plotted as grayscale images of varying sizes and patterns that were partitioned into a training and testing set using an 80:20 split and fed into their Random Forest model. They used a 10-fold cross-validation to evaluate the training set and train the model. The training and test sets consisted of a 1024 feature vectors and a corresponding label. Their model had a 95% accuracy and a Kappa statistical value of 94%, indicating a strong predictive capability.

Network anomaly detection begins by classifying network traffic. Therefore, there are numerous studies on malware traffic classification. Most studies mainly emphasise on ways to improve the performance of the classifier. The authors in [10] proposed a solution using Artificial Neural Network (ANN) and Principal Component Analysis (PCA). This model is highly efficient in malware attack traffic classification and has 99% accuracy compared to most recent models. The authors in [11] proposed a malicious traffic classification model through stacking Dilated Convolutional Autoencoders (DCAEs). This model is able to learn important features from the unlabelled dataset automatically and produce a low false alarm rate. The authors in [12] suggested a model based on deep belief network. The authors in [13] proposed a model utilizing sparse auto encoder. Both designs function in network-based intrusion detection systems.

To classify the malware traffic in android, The authors in [27] applied Random Forest algorithms and 5-fold cross-validation. This study focuses on testing how different features and the number of trees affect the results of the experiment. Other than the classification of normal malware traffic, research is presented on encrypted traffic.

Six most popular machine learning algorithms are involved in malware traffic detection and Random Forest is found to be the best suit for this problem domain [28]. The authors in [29] proposed a machine learning based hybrid feature selection algorithm to deal with an imbalanced network traffic dataset. Weighted mutual information and area under Receiver Operating Characteristic (ROC) curve metrics are used to choose significant features in network traffic.

The authors in [37] In the realm of IoT, where devices often lack intelligence and resources, the vulnerability to cyber threats looms large, posing risks such as device infections, network disruptions, and service denials to legitimate users. To counter these dangers, advanced artificial intelligence and machine learning techniques are deployed for network security. In this specific research endeavor, a support vector machine (SVM) was employed to discern normal from abnormal network traffic, enabling an in-depth analysis of network data to detect and thwart malicious activities. The study encompassed both static and dynamic malware analysis and utilized a network setup involving a Mininet emulator, VMware Fusion, Ubuntu Linux, and a tree-based network topology. Wireshark was used for scrutinizing network traffic, and the SVM classifier emerged as the top performer, boasting an impressive 99% accuracy rate.

The authors in [38] state that in order to prevent cyberattacks originating from Internet of Things (IoT) devices, strong cybersecurity measures are becoming more and more important. IoT device malware prevention is crucial, but it can be difficult because of the intricacy of infiltration methods and the constrained processing power of security apps on these devices. Consequently, identifying malware infections to stop their spread is also crucial. Machine learning and other advanced anomaly detection technologies are crucial due to the increasing variety of malware and IoT device types. However, IoT devices frequently lack the computational power to use machine learning for self-analysis, necessitating the execution of such analysis at internet-connected gateway devices. Instead of examining the complete packet content, the architecture presented in this paper uses summarized statistical data from packets to detect malware activity. This method saves storage space and allows for the analysis of a large number of IoT devices with little computational

resources by using only the traffic volume and destination addresses for each IoT device. Using the suggested architecture, the study carried out malware traffic detection using machine learning algorithms like K-means clustering and Isolation Forest, and it was shown that high accuracy could be attained with the condensed statistical data.

The authors in [39] emphasize the critical need of effective data collection and lightweight threat detection in tackling the security issues brought on by the wide and varied array of IoT devices. This paper presents an architecture for malware detection, minimizing data transmission between servers within this framework and presenting techniques to identify malware through flow information analysis. The study evaluates the efficiency of malware detection as well as the decrease in data volume attained by using these techniques. Notably, the study shows that malware detection performance consistently stays strong even with the drop in data volume.

The authors in [40] state that with the development of contemporary network architectures that enable reliable computing and communication at the network's edge, enabling extensive Internet of Things applications, anomaly detection is showing promise as a technique for guaranteeing quality control in wireless and telecommunication networks. As IoT technologies become more widely used, they become more appealing targets for threats such as Denial of Service, Data Type Probing, Malicious Control, Malicious Operation, Scanning, Spying, and Wrong Setup.

As a result, it is critical to detect attacks in IoT infrastructure. Given the continued proliferation of internet applications, the paper acknowledges the growing necessity for information network security. The significance of dynamic adaptability in anomaly detection systems is emphasized, considering how network operations are constantly changing. In order to ascertain whether an IoT sensor network is functioning normally or exhibiting anomalies, the paper's main goal is to detect attacks on these networks and create a generalized anomaly detection model using machine learning techniques. Levenberg-Marquardt optimization was used to train the model, which showed excellent performance on the Kaggle and NSL-KDD datasets used in this investigation. Furthermore, for convenience, the paper develops a menu-based environment in Jupyter.

The authors in [41] In this paper, the focus is on addressing the critical security challenges in Internet of Things (IoT) networks, particularly the vulnerability to Distributed Denial of Service (DDoS) attacks, which can severely disrupt IoT services. The paper introduces a lightweight defensive algorithm designed to mitigate DDoS attacks in IoT environments. The proposed algorithm is tested across various scenarios to analyze communication patterns among different network nodes. Overall, the paper highlights the importance of securing IoT networks against DDoS attacks and presents a practical solution to enhance their resilience in the face of such threats.

The Authors in [18] examined TCP SYN network attacks and Authors in [19] introduced deep neural networks for attack detection in IoT systems. The self-adaptive identification method of the security index of the network was studied, performed risk assessment was conducted, and the system was mapped. Authors in [20] developed network NIDS based on the conception of DL. For attack detection, they implemented network intrusion detection system on fog node. Authors in [21] used a novel method that combines isolation forest and One Class Support Vector Machine (OCSVM) with an active learning method to detect attacks with no prior information. Authors in [22] used a two-stage approach combining a fast preprocessing or filtering method with a variation auto encoder using reconstruction probability. Authors in [23] performed a Distributed Denial of Service (DDoS) attack using the ping of death technique and detected it using RF algorithm by using the WEKA tool with classification accuracy of 99.76%. Authors in [24] proposed the detection of network dictionary attacks using a data set collected as flows based on a clustered graph. The results of the mentioned methods on the CAIDA 2007 data set give high accuracy for the model.

The Authors in [16] employed algorithms such as RF, NB, SVM, and DT to detect anomalies in IoT networks. To perform their experiments, they exploited the IoT-23 [17], which offers a large dataset consisting of twenty-three captures of different IoT network traffic. These scenarios were divided into twenty network captures from infected IoT devices and three network captures of real IoT devices network traffic. Malware captures are executed for long periods, performing diversity attack scenarios, including Mirai, Torii, and Gagfyt.

The next section explores the different loss functions that impact the effectiveness of machine learning methods.

2.1. Loss functions

The effectiveness of a machine learning algorithm relies on the loss function that aids in adjusting the weights and parameters. Log loss, hinge loss, Exponential loss, Mean square error, Mean absolute error, and Huber loss are loss functions commonly utilized in binary classification and regression tasks [44].

Supervised learning involves mapping input space I to output space J using a function f_L for every input sample, where the set of labels L is $\{L_1,L_2,...,L_n\}$ which is a finite subset of J. In binary classification, the labels $L = \{L_1,L_2\}$ will be a subset of L, with $L_1 = 0$ and $L_2 = 1$ representing the output space.For each new input Inew, the classifier uses the function $f_L(I, J)$ to anticipate a new output J_{new}, which equals 1 if $f_L(I, J)$ is greater than 0 or 0 if $f_L(I, J)$ is lesser or equal to 0 and is represented as $f_L(I, J) \in RdX \{1,0\}$ or $f_L: I \rightarrow J$. If D consists of n samples in the dataset and is denoted by $D = \{(I, J)\}n$. Assuming κ represents the other parameters of the classifier, the mapping function can be denoted as $f_L(I : \kappa)$.

The classifier anticipates the result J_{new}, expecting it to be similar to the real J; the difference between J_{new} and J will be the error. It is a metric that indicates the level of understanding the model has gained from the information. Loss is determined by a loss function or error function, which is limited to a single training sample. The cost function is defined as the mean of the losses computed across all samples in the dataset. During training, an optimization function will be used by all supervised learning algorithms to reduce this cost function. When working with multiple classifiers in machine learning, it is necessary to choose a loss function based on the specific classifier, data type, presence of outliers, and gradient computation ease. Classification loss and regression loss are the two distinct types of loss classifications

2.2. *Loss functions in binary classification*

2.2.1. Binary Cross-Entropy / log loss

This loss function frequently used for binary classification tasks calculates the difference between two probability distributions. When there is a small disparity, the prediction is close to the actual truth; otherwise, they are not similar. If the estimated probability does not match the true probability, the Binary Cross-Entropy will rise, where the optimal value for cross entropy of a perfect model will be 0.

$$H(p) = -\frac{1}{N}\sum_{m=1}^{N} J\log\big(p(J)\big) + \big(1-J\big)\log\big(1-p(J)\big) \quad (1)$$

J is the label with predicated probability p(J) of being L1 for all N samples, with L1 = 1 and L2 = 0 as the labels. In accordance with Equation 1, when L1 = 1 in a sample, the loss increases by log(p(J)), and when L2 = 0 in a sample, log(1 - p(J)) is added to the loss. This implies that the likelihood of the anticipated outcome will be penalized based on how far it is from the current outcome. By computing the cross-entropy using the formula, incorrect guesses are penalized and the likelihood of the accurate guess being 1 results in a loss of 0, and vice versa.

2.2.2. Hinge log loss

Especially in SVM classifier for binary classification, hinge loss is the second loss that is frequently employed in classification. The idea of maximum margin serves as its foundation.

$$H\big(f_L(I,J)\big) = \max\{0,1 - J f_L(I,J)\} \quad (2)$$

$f_L(I,J)$ represents the forecast generated by the classification algorithm. Hinge loss is applicable when the target values are either 1 or -1, with 0 being mapped to -1. Additional mistakes will occur if there is a discrepancy between the signs of the predicted and actual values.

2.3. *Loss functions in regression problems*

2.3.1. Mean Square Error loss (MSE)

MSE is commonly utilized when the predicted or dependent variable follows a Gaussian distribution in its probability distribution. The cost function is calculated as the square of the discrepancy between predicted and observed values.

$$H(f_L(I,J)) = \frac{1}{N}\sum_{1}^{N}(J - f_L(I,J))^2 \quad (3)$$

N represents the batch's size. The squaring operation upholds the positive result. This penalty will result in the classifier being punished more for bigger mistakes, placing more emphasis on outliers. Because of the quadratic nature of the loss function, it possesses a single global minimum and no local minimums.

An alternative form of this function is Mean Squared Logarithmic Error (MSLE), designed to prevent harshly penalizing the classifier for a significant error. This function computes the squared difference of the logarithmic values of the predicted and actual values for each sample.

$$H(f_L(I,J)) = \frac{1}{N}\sum_{1}^{N}\left(\log(J+1) - \log(f\_L(I,J)+1)\right)^2 \quad (4)$$

2.3.2. Mean Absolute Error loss (MAE)

A different loss function commonly employed in regression tasks in MAE is one that measures the absolute discrepancy between predicted and observed values.

$$H(f_L(I,J)) = \frac{1}{N}\sum_{1}^{N}\left|J - f\_L(I,J)\right|^2 \quad (5)$$

MAE outweighs MSE in terms of power and effectiveness, but it still faces the challenge of local minima and maintaining a high gradient during training. This prevents convergence, so a dynamic learning rate may be necessary.

2.3.3. Huber loss (Smooth Mean Absolute Error)

The MSE and MAE calculations are combined to create the huber loss. Whether to use MAE or MSE is determined by a threshold or limit value. MSE will be applied if the error is less than the threshold E
In addition to lacking local minima, Huber loss is not affected by outliers. But it needs the additional parameter ϵ to be optimized.

$$H(f_L(I,J)) = \begin{cases} \frac{1}{N}(J - f_L(I,J))^2 & for\ |J - f_L(I,J)| \le \varepsilon \\ otherwise & \\ \varepsilon\left(|J - f_L(I,J)| - \frac{1}{2}\varepsilon\right)^2 & \end{cases} \quad (6)$$

3. *METHODOLOGY*

The system in question aims to differentiate between benign and malicious communications among IoT devices. Traffic packets containing the required data will be utilized as attributes for pattern recognition and classification purposes. This study employed machine-learning classifiers for IoT traffic classification. Decision tree, Random Forest and SVM are implemented as binary classifiers in order to forecast communication. IoT data is pre-processed in different stages to create and organize important feature vectors. The optimal classifier for recognizing secure IoT communications is determined once the models are assessed for performance utilizing evaluation metrics. The proposed system's overview can be observed in Figure 1.

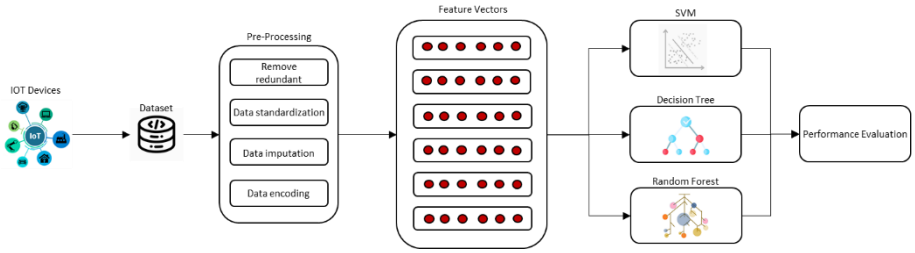

Fig. 1. *Overview of proposed system*

3.1. Pre-processing of the inputs

Into a single common data frame, the CSV files were combined. We used preliminary data-processing techniques to examine the data frame. We removed unnecessary rows and columns. Converting all non-values to NaN helped standardize the data frame. Extrapolation from known data was used to fill in the missing values. In the 'duration' column, for instance, missing values were substituted with the most frequently reported time, and in the'service' column, new categories were used in place of missing values. Both "Malicious" and "Benign" were added to the labels. Using the get dummies function, columns containing categorical data were changed to dummy codes. In the end, the Label Encoder function in Python was used to encode the non-numeric data.

By employing the statistical sampling technique known as stratified sampling, a representative and objective sample is drawn from a larger population. Utilizing a particular set of attributes that are critical to the current research or study, this method divides the population into strata, or subgroups. Each stratum is treated as a separate and distinct population, and a random or systematic sampling technique is applied to select a sample. This approach is particularly beneficial in cases where there is a high degree of population heterogeneity because it ensures that every subgroup will have adequate representation in the final sample. An imbalance in the proportion of malicious and benign data was noted. Consequently, in this work, stratified sampling was used.

For the purpose of training and testing the model, 70% and 30% of the total data were used, respectively. Based on their track record in comparable classification tasks, their versatility in learning techniques, their suitability for high-dimensional and possibly non-linear data, and their capacity for pattern recognition, Random Forest, Decision Tree, and SVM were selected. Each of these approaches is used to categorize IoT communications due to its unique advantages. While Random Forest and Decision Trees provide robustness and interpretability, SVM efficiently manages high-dimensional and non-linear data. The specifics of the Internet of Things dataset and the trade-offs between interpretability, computational complexity, and accuracy have also impacted the algorithm selection.

3.2. Decision tree

The decision tree algorithm is a tree structured classifier because its leaf nodes are classifier nodes and its interior nodes are decision nodes. The decision rules are represented as branches and outcomes are represented as each leaf node. The instances' possible classes are represented by the edges descending from the tree's nodes, and each node acts as a test case for a particular attribute. This recursive process is repeated for each subtree that is placed at the new node. In essence, the algorithm learns how to split data efficiently so that the leaf nodes have very little impurity, i.e., minimal entropy. Small amounts of impurities are crucial for the leaf nodes, though, as they could indicate that the model is overfitting the dataset. Pruning the tree when the threshold impurity is reached can help solve the over-fitting issue.

3.2.1. Training the decision tree classifier

The training dataset's imbalance was taken into consideration through the use of stratified sampling. For the purpose of training and testing the model, 70% and 30% of the total data were used, respectively. The model obtained an F1-score of 99.24 percent and an accuracy of 98.79 percent.

3.3. Random Forest

Using a set of decision trees that were usually trained using the bagging technique, Random Forest, a supervised learning technique, creates a forest. The bagging method relies on the integration of learning models to improve the final product. The final inference of the majority of decision trees is used by the random forest algorithm to classify an entity into a specific category. Every decision tree has internal nodes known as decision nodes and leaf nodes known as classification nodes. The branches of the trees in the Random Forest are arranged so as to maximize information gained or minimize entropy decrease. The precision and accuracy of the model are 99.3% and 98.8%, respectively.

3.4. Support Vector Machine (SVM)

*Support Vector Machine (SVM), the most popular and extensively used machine learning algorithm, uses a hyperplane as a decision boundary in an N-dimensional feature space to categorize data into classes. A hyper plane can be created to divide the feature points by defining their similarity and transforming them using the various SVM kernel functions. When there are numerous hyperplanes, the SVM selects the one with the largest margin. To examine the approach's performance on the dataset, SVM is applied with a variety of kernels, including poly, linear, sigmoid, and rbf. With the kernel coefficient gamma set to 1 / (n features * X), the regularization parameter is fixed at 1. As a tolerance to end the SVM iterations without restricting the maximum number of iterations, the other parameter is 1e−3.*

4. IMPLEMENTATION

4.1. Dataset

The dataset used in this research is the Aposemat IoT-23 dataset [5], published in January 2020. that compiles network traffic from different Internet of Things devices. There are twenty-three traffic captures, or scenarios, available from the Internet of Things communication. These are real-time traffic communications classified as malicious or benign. Twenty of the twenty-three captures come from IoT devices infected with malware, and the three remaining captures come from devices that are benign. A total of 760 million packets were captured in the lab during 2018 and 2019, encompassing 500 hours of traffic with 325 million annotated flows. The malware types that are listed in Table 1 are among the 20 malicious scenario.

Table 1. *Different types of malicious software groups found in IoT-23 database*

Malware	Description
Hajime	Targets multiple CPU architectures by building peer-to-peer botnet
Okiru	Uses ARC-processing embedded devices to its advantage
Hide & seek	It works similarly to worms and gives victims random IP address
Hakai	Malware known as DDOS that targets routers
Kenjiro	A Hakai malware variation
Gagfyt	Takes advantage of shell weaknesses and targets embedded systems
Mirai	Converts Linux based system into bots, killing processes to TCP
IRC Bot	Trojan that can access contacts on MSN Messenger and use IRC server
Torii	An unanticipatedly complex attack on IoT devices
Muhstik	DDOs are malicious programs that mine cryptocurrency
Hajime	Targets multiple CPU architectures by building peer-to-peer botnets

There are 21 features (columns) in the dataset; 4 are numerical types and the remaining 17 are categorical in nature. The characteristics include the date and time of capture, the identity of the captured data, the IP address of the compromised devices, the port number, the IP address of the device that originated the data, the length of the attack, the application, the network protocol linked to the attack, the amount of data sent and received in terms of packets and bytes, the state of the connection, its history, the origin of the data and the response, the bytes missing from the packets, and the classification of the scenario as benign or malicious. If the scenario is malicious, its type is present in that learning instance.

Exploring the IoT-23 dataset provides insights into the communication feature relationships through data analysis. The correlation heap map in Figure 2 indicates a weak correlation among features such as id orig ip, id resp ip, missed bytes, and orig pkts in the dataset. They are visible in a darker hue on the correlation map, showing very minimal to no correlation with each other. There is a strong correlation between response packets and response IP bytes, both of which are visible in lighter shades.

These traits of the attributes will assist in determining which attributes will or will not impact the decision-making process of the classifiers.

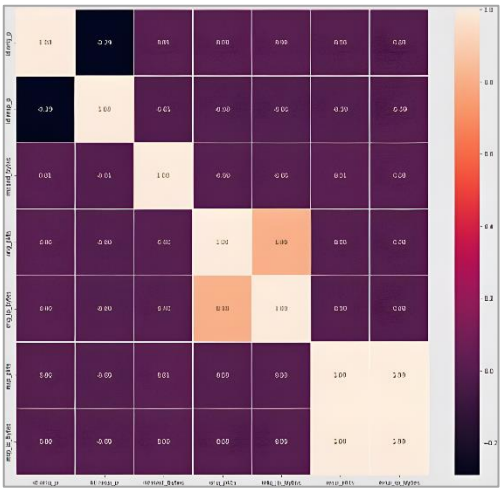

Fig. 2. *Heat map of correlation between the features in the IoT-23 dataset*

In Figure 3, the benign and malicious class labels samples histogram is displayed. Malicious samples are classified as class A, and benign samples are classified as class B. The class samples do not appear to be balanced, which could have an impact on how well the models that classify them perform.

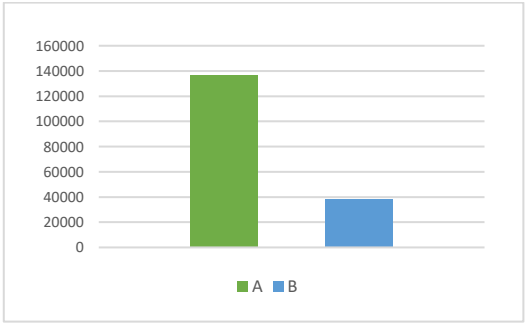

Fig. 3. *IoT-23 dataset's benign and malicious class histogram*

4.2. Performance Metrics

The same set of examples is used to train each algorithm in this work, and evaluation metrics such as accuracy, recall, precision, and F1 score as well as False Positive Rate (FPR), False Negative Rate (FNR), and False Discovery Rate (FDR)are used to assess each algorithm's performance. From the F1 scores obtained from the three classifiers, it can be observed that Random Forest has outperformed well compared to the other 2 classifiers. The second-best F1 score belongs to Decision Tree.

Because of its intrinsic properties, Random Forest performs well with any kind of feature categorical, numerical, or binary features without the need for any pre-processing. The features are not scaled or altered in any way. It has also dealt with the outliers in the IoT-23 dataset. The random forest's design has addressed the issue of class imbalance and includes built-in methods to lower the overall error rate. It has also lessened the issue of overfitting because it averages across multiple trees. These have demonstrated that Random Forest is a more reliable model than decision trees for dividing IoT communications into malicious and benign categories. The class imbalance issue and the existence of outliers, as demonstrated by Table 2, cause SVM to perform poorly.

4.2.1. Performance Parameters. the proposed work was evaluated on the basis of the following parameters:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} * 100,$$

$$\text{Precision} = \frac{TP}{(TP + FP)} * 100,$$

$$\text{Recall} = \frac{TP}{(TP + FN)} * 100,$$

$$\text{F\_Measure} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}, \quad (7)$$

We need the confusion matrix of each machine learning classifier to calculate the performance mentioned above parameters. Further, we also need to define the following terms:

- *True Positive (TP) : The ML model truly predicted the attack flow as an attack.*
- *True Negative (TN) : The ML model truly predicted the normal flow as normal.*
- *False Positive (FP) : The ML model wrongly predicted the normal flow as an attack.*
- *False Negative (FN) : The ML model wrongly predicted the attack flow as normal.*

where TP stands for true positive that means if actual and predicted data samples are an anomaly in nature, then TP is evaluated, TN stands for true negative that means if actual and predicted data samples are not an anomaly in nature, then TN is evaluated, FP stands for false positive that means if actual and predicted data samples are normal and anomaly in nature, respectively, then FP is evaluated, and FN stands for false negative that means if actual and predicted data samples are an anomaly and normal, respectively, then FN is evaluated.

Table 2. *Performance Metrics*

Classifier	Precision	Recall	F1	Accuracy
Random Forest	0.9936	0.9926	0.9931	0.9891
SVM	0.7888	1.0	0.8819	0.7887
Decision Tree	0.9894	0.9953	0.9924	0.9879

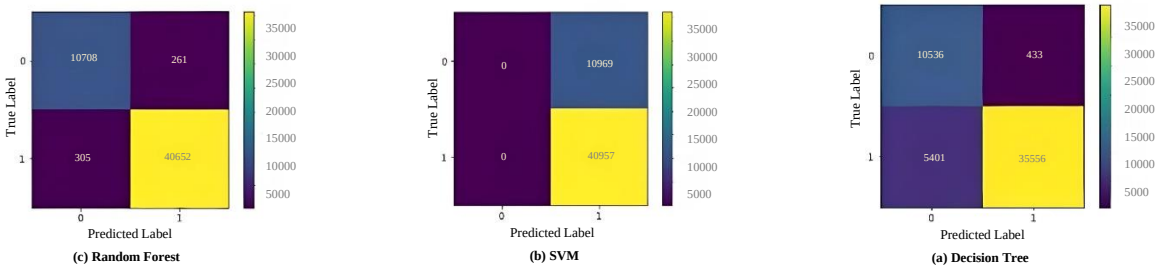

Fig. 4. *All classifiers' confusion matrix*

The confusion matrix for each of the three models is displayed in Figure 4. SVM can be observed with poor classification, and the random forest's confusion matrix demonstrates that the quantity of false positives and false negatives is extremely low in comparison to the other classifiers. The Random Forest classifier correctly classified 0.6 percent of malicious traffic as benign, according to Table 3's lowest false discovery rate (FDR) of 0.0063, The classifier with the lowest false positive rate (FPR) among all others has also demonstrated this. The performance metrics are represented graphically in Figure 5, which demonstrates the classifiers' nearly equal performance in identifying IoT communications.

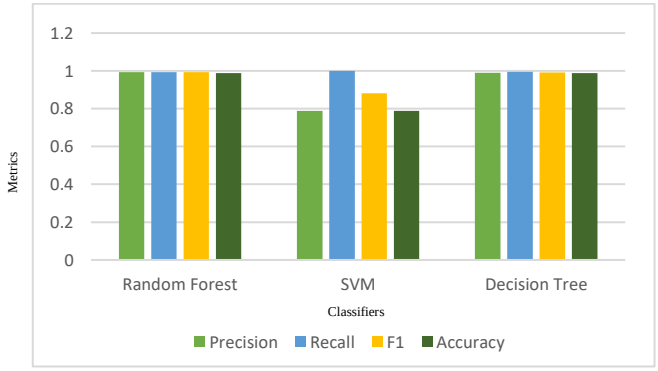

Fig. 5. *Metrics of all classifiers*

Table 3. *Additional relevant metrics*

Classifier	FDR	MCC	FPR	FNR
Random Forest	0.0063	0.9673	0.0237	0.0074
SVM	0.2112	0.0	1.0	0.0
Decision Tree	0.0120	0.7333	0.0395	0.1318

Every binary classifier that measures the correlation coefficient between the predicted and the true class equations also computes the Matthews Correlation Coefficient (MCC).The stronger the coefficient, the stronger the correlation and, consequently, the classifier's prediction.

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) + (TP + FN) + (TN + FP) + (TN + FN)}} \quad (8)$$

4.3. Comparison of results from similar studies

Table 4 presents a comparison of the proposed traffic classification system with related works on the same IoT-23 dataset. Nearly perfect performance on the F1 score has been reported in almost all of the works on this dataset. The authors in [45]only employed a portion of the IoT-23 and obtained a perfect F1 score, and The authors in [46]obtained 97.3% using Random Forest and 92.3% using Linear SVM, and it was noted that no preprocessing of the dataset's features was carried out. Comparable to this research, SVM has shown a lower performance compared to other models.

Table 4. *Comparison of results from similar studies*

Study	Algorithm	F1 Score
[46]	Linear SVM	92.35%
[46]	Random forest	97.3%
[45]	XGBoost	100%
Proposed Method	Random forest	99.31%

4.4. Notes and recommendations for addressing classifier underperformance

These behaviors were noted in the machine learning algorithms that were used to categorize IoT traffic for communications that were malicious or benign.

1. *Random Forest handled the data with or without data pre-processing thanks to its adaptable algorithm design. Random forests, which function as ensemble learners, combine the salient characteristics of multiple weak learners into decision trees. The model's performance is therefore unaffected by the class imbalance.*

2. *After the Random Forest classifier approach, the Decision Tree algorithm was found to be the second-best method for classifying the communications.*

3. *With multiple outliers, a low correlation coefficient between the features, and an imbalanced dataset, SVM may not be appropriate for the IoT-23 dataset. Additionally, more false positives and false negatives were observed.*

When a classifier is trained on an unbalanced dataset, it is unable to learn the data from classes with smaller sample sizes. When tested, this type of model will only predict that the sample belongs to one of the majority classes because it has learned that the minor class is noise. This could result in a higher number of false positives and false negatives [47, 48, 49]. In real-time applications, this type of behavior is not acceptable.

In order to address the imbalance data, the following solutions are suggested:

1. *Data level techniques*
2. *Methods at the algorithmic level*
3. *Hybrid strategies*

The IoT-23 dataset will be better suited for data-level approaches. One specific kind of data level technique that can be used to address the class imbalance issue in the IoT-23 dataset is random under-sampling, which lowers the number of samples from the majority class. The alternative data level approach, random over-sampling, which replicates data from the minority class, might not be the best way to boost classifier performance because it could result in overfitting of the model and poor generalization.

5. CONCLUSIONS

Due to its revolutionary effect on human life, the Internet of Things (IoT) has become a topic of great interest for both the scientific community and the business community. The idea of smart gadgets, smart healthcare, smart industry, smart city, smart grid, and other concepts have been introduced by the Internet of Things' explosive growth, completely changing human life. The security of IoT devices is becoming a major concern these days, particularly for the healthcare industry where recent attacks have shown dangerous IoT security flaws. Conventional approaches to network security are well-known. However, the current security mechanisms cannot be directly implemented to secure the Internet of Things devices and network from cyberattacks because of the resource-constrained nature of IoT devices and the unique behavior of IoT protocols. Researchers require IoT-specific tools, methods, and datasets in order to improve the security of the Internet of Things.

The use of machine learning techniques to identify communications between Internet of Things devices and categorize them as secure or malicious traffic was investigated in this paper. Following preprocessing, SVM, random forest, and decision tree classifiers are used to model the IoT-23 dataset's learning samples. When comparing the 99.31 percent F1 score of Random Forest to that of SVM and decision trees, Random Forest performs better in identifying secure and insecure traffic. Because SVM has poor feature correlation, is sensitive to outliers, and primarily suffers from class imbalance, it was found to be the least effective classifier. In order to improve the performance of the other classifiers, an extension of this work that addresses the issue of class imbalance has also been proposed. To develop a model that separates the malicious from the secure communications of Internet of Things devices, unsupervised representational learning can also be tried.

6. FUTURE WORKS

As the IoT landscape evolves, future research in IoT security should adopt a multi-faceted approach. This includes refining machine learning models, exploring ensemble techniques, and focusing on real time threat detection. Collaborative data sharing efforts and compliance with emerging regulations are crucial. Additionally, advanced behavioral analysis, blockchain integration, quantum resistant security, and user awareness initiatives are key areas for development. A holistic approach that combines technological innovation, regulatory compliance, ethical considerations, and user education will ensure a secure and resilient IoT ecosystem.

REFERENCES

[1] *Rayes, A. and S. Salam. (2019). Internet of things (IoT) overview. Internet of Things from hype to reality. Springer. 1-35.*

[2] *WatchGuard Threat Lab. (2020). Internet Security Report: Q4 202.*

[3] *Palo Alto Networks. (2020). Unit 42 IoT Threat Report.*

[4] *Eclipse Foundation. (2020). The Eclipse Foundation Releases IoT Commercial Adoption Survey Results. 2020 March 10; Available from:* https://www.eclipse.org/org/press-release/20200310-iot-commercial-adoption-survey-2019.php.

[5] *Garcia, S., A. Parmisano, and M. J. Erquiaga. (2020). IoT-23:A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0). 2020: Zenodo.*

[6] *The Internet of Things. (2022). Cambridge Dictionary.*

[7] *IEEE. (2015). Internet of Things (IoT) Ecosystem Study.*

[8] *Posey, B. (2022). IoT devices (internet of things devices).*

[9] *Muhammad, F., W. Anjum, and K. S. Mazhar. (20150. A critical analysis on the security concerns of internet of things (IoT). International Journal of Computer Applications, 111(7),1-6.*

[10] *Arivudainambi, D., V. K. K. A, and P. Visu. 2019. Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance. Computer Communications, 147, 50-57.*

[11] *Yu, Y., J. Long, and Z. Cai. (2017). Network intrusion detection through stacking dilated convolutional autoencoders. Security and Communication Networks.*

[12] *Gao, N., et al. (2014). An intrusion detection model based on deep belief networks. 2014 Second International Conference on Advanced Cloud and Big Data. IEEE.*

[13] *Javaid, A., et al. (2016). A deep learning approach for network intrusion detection system. Eai Endorsed Transactions on Security and Safety, 3(9), p. e2.*

[14] *Parker2005, "The IoT in 2030: 24 billion connected things generating $1.5 trillion,"*

[15] *"Internet of Things Global Standards Initiative."*

[16] *Thamaraiselvi, D.; Mary, S. Attack and Anomaly Detection in IoT Networks using Machine Learning. Int. J. Comput. Sci. Mob.Comput. 2020, 9, 95–103.*

[17] *Parmisano, A.; Garcia, S.; Erquiaga, M.J. Stratosphere Laboratory. A Labeled Dataset with Malicious and Benign IoT Network Traffic. Available online:* https://www.stratosphereips.org/datasets-iot23

[18] *B. K. Mohanta, U. Satapathy, and D. Jena, "Addressing Security and Computation Challenges in IoT Using Machine Learning," in Advances in Distributed Computing and Machine Learning, Singapore, Asia,2021, pp. 67–74,* https://doi.org/10.1007/978-981-15-4218-3_7.

[19] *J. Li and B. Sun, "A Network Attack Detection Method Using SDA and Deep Neural Network Based on Internet of Things," International Journal of Wireless Information Networks, vol. 27, no. 2, pp. 209–214,Jun. 2020,* https://doi.org/10.1007/s10776-019-00462-7.

[20] *N. Sahar, R. Mishra, and S. Kalam, "Deep Learning Approach-Based Network Intrusion Detection System for Fog-Assisted IoT," in Proceedings of International Conference on Big Data, Machine Learning and their Applications, Singapore, 2021, pp. 39–50,* https://doi.org/10.1007/978-981-15-8377-3_4.

[21] *S. Kavitha, U. Maheswari, and R. Venkatesh, "Network Anomaly Detection for NSL-KDD Dataset Using Deep Learning," Information Technology in Industry, vol. 9, no. 2, pp. 821–827, Mar. 2021,* https://doi.org/10.17762/itii.v9i2.419.

[22] *H. Neuschmied, M. Winter, K. Hofer-Schmitz, and B. Stojanovic, "Two Stage Anomaly Detection for Network Intrusion Detection," in 7th International Conference on Information Systems Security and Privacy, Vienna, Austria, Feb. 2021, pp. 450–457.*

[23] *S. Pande, A. Khamparia, D. Gupta, and D. N. H. Thanh, "DDOS Detection Using Machine Learning Technique," in Recent Studies on Computational Intelligence: Doctoral Symposium on Computational Intelligence, A. Khanna, A. K. Singh, and A. Swaroop, Eds. Singapore, Asia: Springer, 2021, pp. 59–68.*

[24] *A. T. Siahmarzkooh, J. Karimpour, and S. Lotfi, "A Cluster-based Approach Towards Detecting and Modeling Network Dictionary Attacks," Engineering, Technology & Applied Science Research, vol. 6,*

no. 6, pp. 1227–1234, Dec. 2016, https://doi.org/10.48084/etasr.937.

[25] *Shire, Robert and Shiaeles, Stavros and Bendiab, Keltoum and Ghita, Bogdan and Kolokotronis, Nicholas, "Malware squid: a novel iot mal- ware traffic analysis framework using convolutional neural network and binary visualisation," in Internet of Things, Smart Spaces, and Next Generation Networks and Systems. Springer, 2019, pp. 65–76.*

[26] *Verma, Abhishek and Ranga, Virender, "Machine learning based intrusion detection systems for iot applications," Wireless Personal Communications, vol. 111, no. 4, pp. 2287–2310, 2020.*

[27] *Singh, Raman and Kumar, Harish and Singla, RK, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," Expert Systems with Applications, vol. 42, no. 22, pp. 8609–8624, 2015.*

[28] *Barcena, Mario Ballano and Wueest, Candid, "Insecurity in the internet of things," Security response, symantec, 2015.*

[29] *Bendiab, Gueltoum and Shiaeles, Stavros and Alruban, Abdulrahman and Kolokotronis, Nicholas, "Iot malware network traffic classification using visual representation and deep learning," in 2020 6th IEEE Conference on Network Softwarization (NetSoft). IEEE, 2020, pp. 444– 449.*

[30] *J. Habibi, D. Midi, A. Mudgerikar, and E. Bertino, "Heimdall:Mitigating the Internet of Insecure Things," IEEE Internet of Things Journal, vol. 4, no. 4, pp. 968–978, Aug 2017.*

[31] *H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K. K. R. Choo, "A Two-layer Dimension Reduction and Two-tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks," IEEE Transactions on Emerging Topics in Computing, vol. PP, no. 99, pp. 1–1, 2016.*

[32] *Felan Carlo C Garcia and Felix P Muga Ii. "Random Forest for Malware Classification". In: arXiv preprint arXiv:1609.07770 (), p. 4.*

[33] *Nicolas-Alin Stoian. "Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set". In: Bachelor's thesis, University of Twente (), p. 10.*

[34] *Karthikeyan, L., Jacob, G. & Manjunath, B., 2011. Malware images: Visualization and automatic classification.*

[35] *Karanja, E., Masupe, S. & Jeffrey, M., 2020. Analysis of internet of things malware using image texture features and machine learning techniques.*

[36] *Pa, Y. et al., 2015. IoTPOT: Analysing the Rise of IoT Compromises.*

[37] *Mishra, S. (2021). Network Traffic Analysis Using Machine Learning Techniques in IoT Networks. International Journal of Software Innovation (IJSI), 9(4), 107-123.*

[38] *Nakahara, M., Okui, N., Kobayashi, Y., & Miyake, Y. (2020). Machine Learning based Malware Traffic Detection on IoT Devices using Summarized Packet Data. In IoTBDS (pp. 78-87).*

[39] *Nakahara, M., Okui, N., Kobayashi, Y., Miyake, Y., & Kubota, A. (2022). Malware detection for IoT devices using hybrid system of whitelist and machine learning based on lightweight flow data. Enterprise Information Systems, 2142854.*

[40] *Mithran, K., & Gopi, C. (2022, June). Anomaly Detection in IoT Sensor Networks using Machine Learning. In 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) (pp. 1-7). IEEE.*

[41] *Zhang, C., & Green, R. (2015, April). Communication security in internet of thing: preventive measure and avoid DDoS attack over IoT network. In Proceedings of the 18th symposium on communications & networking (pp. 8-15).*

[42] *Liu, D., Xu, X., Liu, M. & Liu, Y. (2021a) Dynamic traffic classification algorithm and simulation of energy Internet of things based on machine learning. Neural Computing and Applications. 33, 3967-3976. doi:10.1007/s00521-020-05457-7.*

[43] *Fatayer, T. S. & Azara, M. N. (2019) IoT secure communication using ANN classification algorithms. In 2019 International Conference on Promising Electronic Technologies (ICPET), 23-24 October 2019, Gaza, Palestine, IEEE. pp. 142-146. doi: 10.1109/ICPET.2019.00033.*

[44] *Wang, Q., Ma, Y., Zhao, K. & Tian, Y. (2022) A Comprehensive Survey of Loss Functions in Machine Learning. Annals of Data Science. 9, 187-212.doi:10.1007/s40745-020-00253-5.*

[45] *Bansal, A. & Mahapatra, S. (2017) A comparative analysis of machine learning techniques for botnet detection. In Proceedings of the 10th International Conference on Security of information and Networks. pp. 91-98. doi:10.1145/3136825.3136874.*

[46] *Austin, M. (2021) IoT Malicious Traffic Classification Using Machine Learning. Graduate Theses,*

Dissertations, and Problem Reports. 8024.West Virginia University.

[47] *Hasanin, T., Khoshgoftaar, T. M., Leevy, J. L. & Bauder, R. A. (2019a) Severely imbalanced big data challenges: investigating data sampling approaches. Journal of Big Data. 6(1), 107. doi:10.1186/s40537-019-0274-4.*

[48] *Hasanin, T., Khoshgoftaar, T. M., Leevy, J. L. & Seliya, N. (2019b) Investigating random undersampling and feature selection on bioinformatics big data. In 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (Big Data Service), 04-09 April 2019, Newark, CA, USA, IEEE. pp. 346-356. doi: 10.1109/BigDataService.2019.00063.*

[49] *Hasanin, T., Khoshgoftaar, T. M., Leevy, J. L. & Bauder, R. A. (2020) Investigating class rarity in big data. Journal of Big Data. 7(1), 1-17. doi:10.1186/s40537-020-00301-0.*

[50] *Hemalatha, D. and Afreen, B.E. (2015) Development in RFID (Radio Frequency Identification) Technology in Internet of Things (IOT). International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 4.*