International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Securing the Internet of Things: A Random Forest Classifier Approach for Detecting and Mitigating Botnet Attacks on IoT Devices

E. Shravan Kumar[1], Manyam Apoorva Reddy[2], Kale Rakesh[2], Bestha Anusha[2], Naredla Rajkumar[2], Medharametla Gokul[2]

[1]Assistant Professor, [2]UG Scholar, [1,2]Department of Computer Science and Engineering (Cyber Security)
[1,2]Malla Reddy Engineering College and Management Sciences, Kistapur, Medchal, 501401, Hyderabad, Telangana

Abstract

The Internet of Things (IoT) has witnessed significant growth with the proliferation of connected devices in various domains. However, this rapid expansion has also led to an increase in security vulnerabilities, particularly in the form of botnet attacks. Botnets are networks of compromised devices controlled by malicious actors, and they can be leveraged to launch various cyber-attacks. Among the most infamous botnet attacks targeting IoT devices are the Gafgyt and Mirai botnets. These attacks exploit security weaknesses in IoT devices, compromising them and turning them into bots for executing large-scale attacks. The need for a reliable detection mechanism for IoT botnet attacks arises from the increasing frequency and severity of such attacks. As IoT devices continue to grow in number and become more integral to critical infrastructure and everyday life, the potential consequences of botnet attacks become more severe. These attacks can disrupt services, compromise sensitive data, and even pose physical risks. Therefore, it is crucial to develop effective methods to identify and mitigate IoT botnet attacks promptly. One such mechanism is the Random Forest Classifier, which is a machine learning algorithm known for its robustness and accuracy in classification tasks. Random Forest Classifier combines the predictions of multiple decision trees to make accurate predictions about the class labels of input data. This algorithm has been widely used in various domains, including cybersecurity, due to its ability to handle complex and high-dimensional datasets.

Keywords: Internet of Things, Provision PT 37E security camera, Botnet attack, Supervised learning, Random Forest classifier.

1. Introduction

Due to an increase in cybercrime, researchers have been focusing on identifying intrusions in networks [1]. Previously, traditional computer networks and personal computers were the focus of cyberattacks, but now communication infrastructure encompasses the internet of things (IoT), the internet of connected vehicles, the internet of medical things [2] and 5G, these have become the targets of many cyberattacks. Older strategies, such as firewalls and antivirus software, are unable to offer solutions to complex cyberattacks. Machine learning and deep learning algorithms are now used to detect intrusion in networks. Algorithms can also be developed to perform both in group strategies and in combination. Grouping strategies is a machine learning technique to solve intrusion detection and prediction problems. At the present time, big data produced by computers and IoT devices is causing a threat to data traffic in networks and increases the malware that threatens the integrity of data; however, these issues cannot be dealt with by current strategies. A bot in a network is a personal computer containing malware that allows attackers to hack it. A botnet is a spider computer network made up of many hosts, each running independent programs. It runs a bot on many devices connected to the internet to form a botnet network operated by a malicious group. Botnets pose a threat to network security because they are used in cybercrime techniques such as distributed denial of service (DDoS). Machine learning algorithms are used to track such attacks in IoT. A botnet can be used to refuse services directed and

distributed to any system on the internet so that it cannot properly serve its legitimate customers. Currently, DDoS attacks are performed from a botnet platform; despite their simplicity, they are very effective due to the bandwidth of the bots. An intruder can gain illegal access to user data through many attacks. Networks are exposed to a wide variety of attacks, including probing, denial of service, user-to-root and port scanning. To execute these attacks, transport, or protocols such as internet control message protocol, user datagram protocol (UDP), transmission control protocol (TCP) and file transfer protocol may be used. Network-based intrusion detection systems (NIDS) are among the best means to scan networks and identify attacks. Many machine learning algorithms have been used to detect cyberattacks, but these fail the sanctity of big data traffic. Also, they lack the required optimisation. Researchers are now able to develop systems to detect and verify botnet data [3]. However, complex data traffic does not guarantee high accuracy, as the techniques for detecting botnet attacks are constantly changing. Therefore, there is a need for classification techniques based on neural development that can identify the number of layers and neurons when detecting attacks. When training the network and learning weights, hyperparameters must be set. Signature-based and anomaly-based detection are two of the most important methods of intrusion detection. The former uses signature-based detection of a known attack pattern, while anomaly-based detection is used for both known and unknown attack patterns [4]. Traffic is identified by NIDS, which means extracting the most important features from traffic records to classify them as malicious or normal by machine learning algorithms. Network- and host-based systems are two means to classify network traffic records. In the NIDS network, all log traffic is monitored for intrusions such as DoS. When using vast databases for anomaly detection, high detection accuracy and low rate of alerts can be obtained. Training and testing phases are implemented to develop databases for anomaly-based systems; patterns are identified in the training phase and compared during the testing phase. Signature and heuristic methods cannot detect malware or provide an adequate level of detection against new and unknown variants, so machine learning algorithms are used to solve this problem. Machine learning and deep learning techniques detect attacks without requiring advanced security knowledge. The efficiency of intrusion detection systems (IDSs) can be improved through nature-inspired, meta-heuristic data mining, reinforcement learning, grammar-based machine learning and artificial intelligence. IDS performance can also be improved through the artificial bee colony (ABC), grey wolf optimisation and artificial fish swarm (AFS) algorithms. Rajagopal et al. proposed a group model using a meta-classification method based on stacked generalisation and used two data sets, UGR'16 and UNSW NB-15, which were collected from real and emulated network traffic. The proposed system achieved accuracy of 97%, as well as 94% with emulated data sets [5]. The main contribution of this research is to develop an intelligent security system by using advanced machine learning algorithms that detect and classify one of the serious intrusions that threaten IoT platforms. This study presents the random forest model to detect and classify benign, BASHLITE also known as Gafgyt and Mirai attacks of security camera devices connected to IoT applications. It investigates whether the proposed system achieves superior accuracy compared to existing systems.

2. Literature Survey

Alissa et al. [6] proposed machine learning methods for classifying binary classes. This purpose is served by using the publicly available dataset UNSW-NB15. This dataset resolved a class imbalance problem using the SMOTE-Oversampling technique. A complete machine learning pipeline was proposed, including exploratory data analysis, which provided detailed insights into the data, followed by preprocessing. During this process, the data passes through six fundamental steps. A decision tree, an XgBoost model, and a logistic regression model are proposed, trained, tested, and evaluated on the dataset. In addition to model accuracy, F1-score, recall, and precision are also considered. Based on all experiments, it is concluded that the decision tree outperformed with 94% test accuracy.

Kumar et al. [7] presented an IoT botnet detection solution, EDIMA, consisting of a set of lightweight modules designed to be deployed at the edge gateway installed in home networks with the remaining modules expected to be implemented on cloud servers. EDIMA targeted early detection of IoT botnets prior to the launch of an attack and includes a novel two-stage Machine Learning (ML)-based detector developed specifically for IoT bot detection at the edge gateway. The ML-based bot detector first employed supervised ML algorithms for aggregate traffic classification and subsequently Autocorrelation Function (ACF)-based tests to detect individual bots. The EDIMA architecture also comprised a malware traffic database, a policy engine, a feature extractor and a traffic parser.

Alharbi et al. [8] proposed a graph-based ML model for botnet detection that first considers the significance of graph features before developing a generalized model for detecting botnets based on the selected important features. This work explored different feature sets selected using five filter-based feature evaluation measures derived from various theories such as consistency, correlation, and information. Two heterogeneous botnet datasets, CTU-13 and IoT-23 were used to evaluate the effectiveness of the proposed graph-based botnet detection with several supervised ML algorithms. Experiment results showed that using features reduces training time and model complexity and provides high bots detection rate.

Shao et al. [9] explored an adaptive online learning strategy for real-time IoT botnet attack detection. Furthermore, this work operated the proposed adaptive strategy in conjunction with online ensemble learning. To evaluate the proposed strategy, this work used real IoT traffic data, including benign traffic data and botnet traffic data infected by Mirai. In real-time IoT botnet attack detection, experimental results demonstrated that the proposed adaptive online learning strategy achieves remarkable performance.

Alauthman et al. [10] proposed a sophisticated traffic reduction mechanism, integrated with a reinforcement learning technique. This work then evaluated the proposed approach using real-world network traffic and achieved a detection rate of 98.3%. The approach also achieved a relatively low false positive rate (i.e., 0.012%).

Kaur et al. [11] presented the efforts to catalogue and compare attacks, datasets and machine learning algorithms and architectures for intrusion detection systems for IoT devices. This work classified attacks aimed at IoT devices at different layers and protocols. This work also highlighted potential features that can be used by machine learning-based intrusion detection systems to detect different types of attacks. This work provided a comparative study of IoT datasets used for model training and identify key properties which helps in assessing their suitability in particular scenarios. Finally, this work discussed observations and proposed the research directions for building a robust IoT intrusion detection system.

Alzahrani et al. [12] propose a robust system specifically to help detect botnet attacks of IoT devices. This was done by innovatively combining the model of a convolutional neural network with a long short-term memory (CNN-LSTM) algorithm mechanism to detect two common and serious IoT attacks (BASHLITE and Mirai) on four types of security camera. The data sets, which contained normal malicious network packets, were collected from real-time lab-connected camera devices in IoT environments.

Sajjad et al. [13] used a collaborative trust relationship-based threat intelligence-sharing mechanism to prevent other IoT devices from being compromised by the detected botnet. The researchers have evaluated the collaborative threat intelligence sharing mechanism using Ethereum Virtual Machine and Hyperledger. The performance of proposed system can detect 97% of the Mirai botnet attack activities.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Furthermore, collaborative threat intelligence sharing mechanism based on the Ethereum Virtual Machine showed more scalability.

Azhari et al. [14] deemed necessary to have a system to detect Mirai botnet attack using the Support Vector Machine (SVM) model. This model becomes a solution to detect Mirai botnet attacks for having good generalization performance when the parameter is properly configured in modeling the training set (training dataset) and getting good classification results without a lot of training data. The results of the analysis from the use of the SVM model showed the accuracy of 92.91% with a linear kernel and max_iter 10000.

3. Proposed methodology

The Internet of Things (IoT) has witnessed significant growth with the proliferation of connected devices in various domains. However, this rapid expansion has also led to an increase in security vulnerabilities, particularly in the form of botnet attacks. Botnets are networks of compromised devices controlled by malicious actors, and they can be leveraged to launch various cyber-attacks. Among the most infamous botnet attacks targeting IoT devices are the Gafgyt and Mirai botnets. These attacks exploit security weaknesses in IoT devices, compromising them and turning them into bots for executing large-scale attacks.

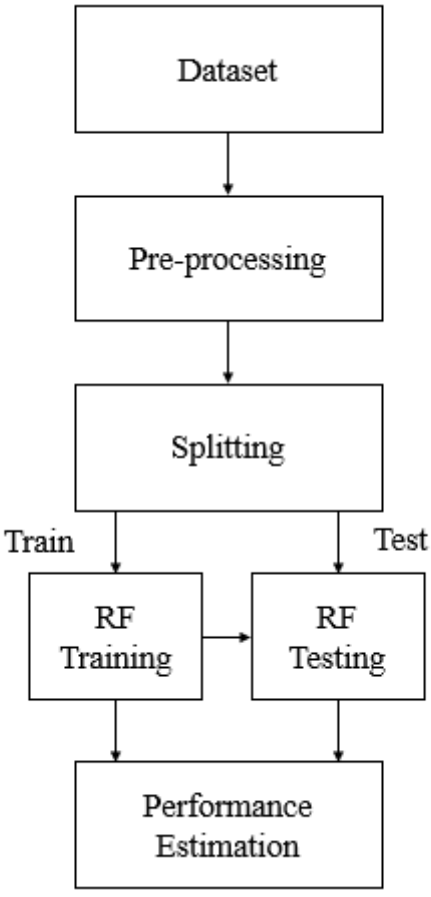

Fig. 1: Block diagram of proposed system.

The Random Forest Classifier offers several advantages for the detection of IoT botnet attacks. Firstly, it can handle large-scale datasets with high dimensionality, which is often the case when dealing with IoT data. The Provision PT 37E Security Camera dataset, in this context, can contain a vast amount of information related to network traffic, device behavior, and communication patterns, which can be

leveraged for attack detection. Secondly, the Random Forest Classifier is robust to noise and outliers, which are common in real-world IoT data. IoT environments can exhibit variations in network conditions, device configurations, and user behavior, leading to data inconsistencies. The Random Forest Classifier can handle such variations and still provide accurate predictions. Furthermore, the ensemble nature of the Random Forest Classifier allows it to effectively detect different types of IoT botnet attacks, including benign, Gafgyt, and Mirai attacks. By leveraging a combination of decision trees, the classifier can capture various attack patterns and generalize well to new, unseen attacks. This versatility is crucial as the landscape of IoT botnet attacks evolves rapidly, with attackers employing new techniques and strategies.

3.1 Pre-processing

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

Why do we need Data Pre-processing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

3.1.1 Splitting the Dataset into the Training set and Test set

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model.

Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

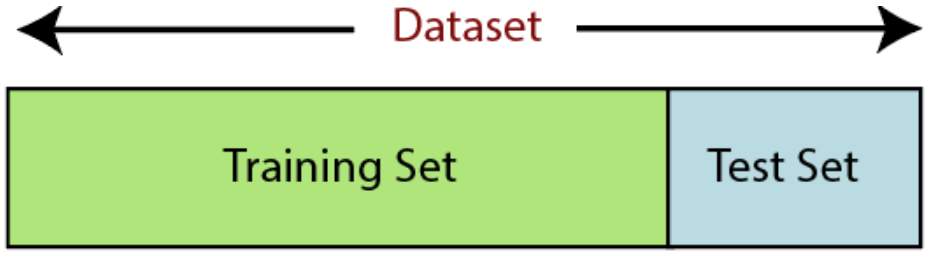

Fig 2: Dataset splitting.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

4.2 Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

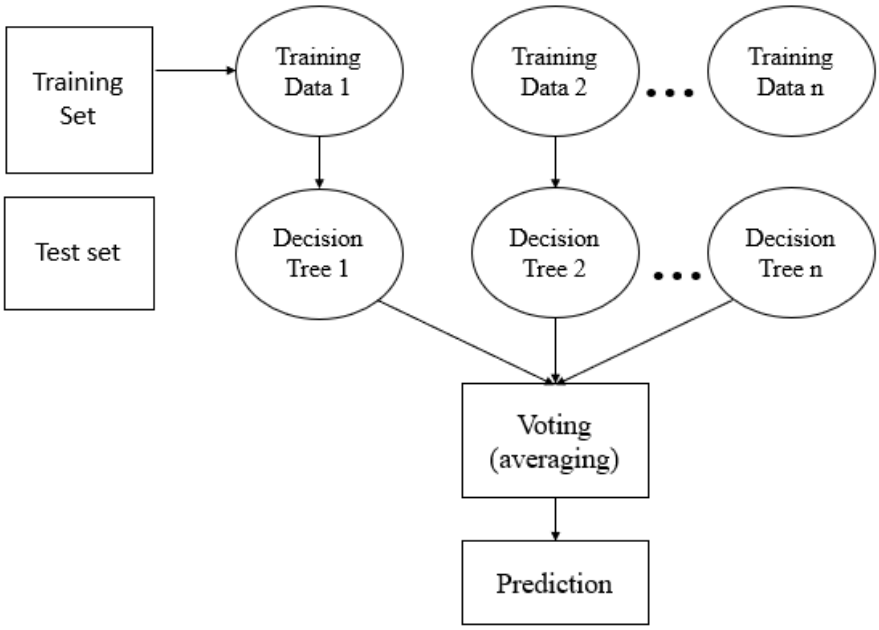

Fig. 3: Random Forest algorithm.

Random Forest algorithm

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

Important Features of Random Forest

- **Diversity**- Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- **Immune to the curse of dimensionality**- Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization**-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
- **Train-Test split**- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability**- Stability arises because the result is based on majority voting/ averaging.

Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Below are some points that explain why we should use the Random Forest algorithm

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Types of Ensembles

Before understanding the working of the random forest, we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model. Ensemble uses two types of methods:

Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest. Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

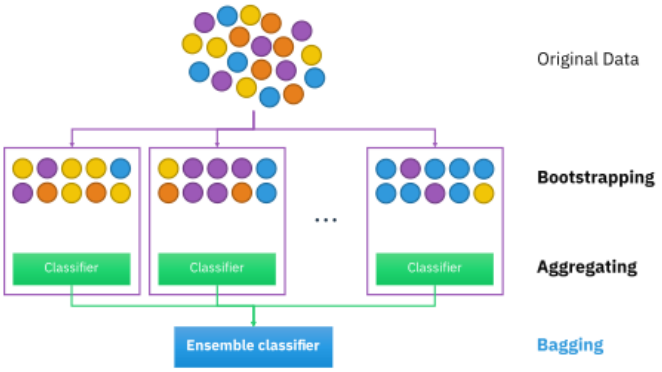

Fig. 4: RF classifier analysis.

Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.

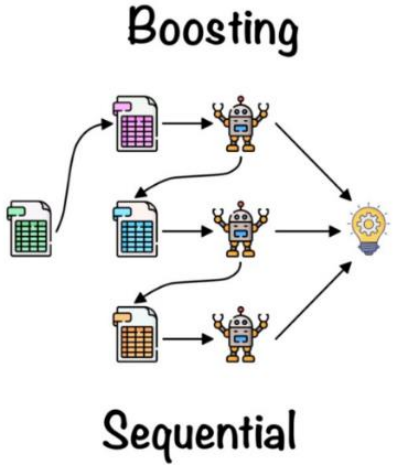

Fig. 5: Boosting RF classifier.

Applications of Random Forest: There are mainly four sectors where Random Forest mostly used:

- Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
- Medicine: With the help of this algorithm, disease trends and risks of the disease scan be identified.
- Land Use: We can identify the areas of similar land use by this algorithm.
- Marketing: Marketing trends can be identified using this algorithm.

Advantages of Random Forest

- It can be used in classification and regression problems.
- It solves the problem of overfitting as output is based on majority voting or averaging.
- It performs well even if the data contains null/missing values.
- Each decision tree created is independent of the other thus it shows the property of parallelization.
- It is highly stable as the average answers given by a large number of trees are taken.
- It maintains diversity as all the attributes are not considered while making each decision tree though it is not true in all cases.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

- It is immune to the curse of dimensionality. Since each tree does not consider all the attributes, feature space is reduced.

4. Results

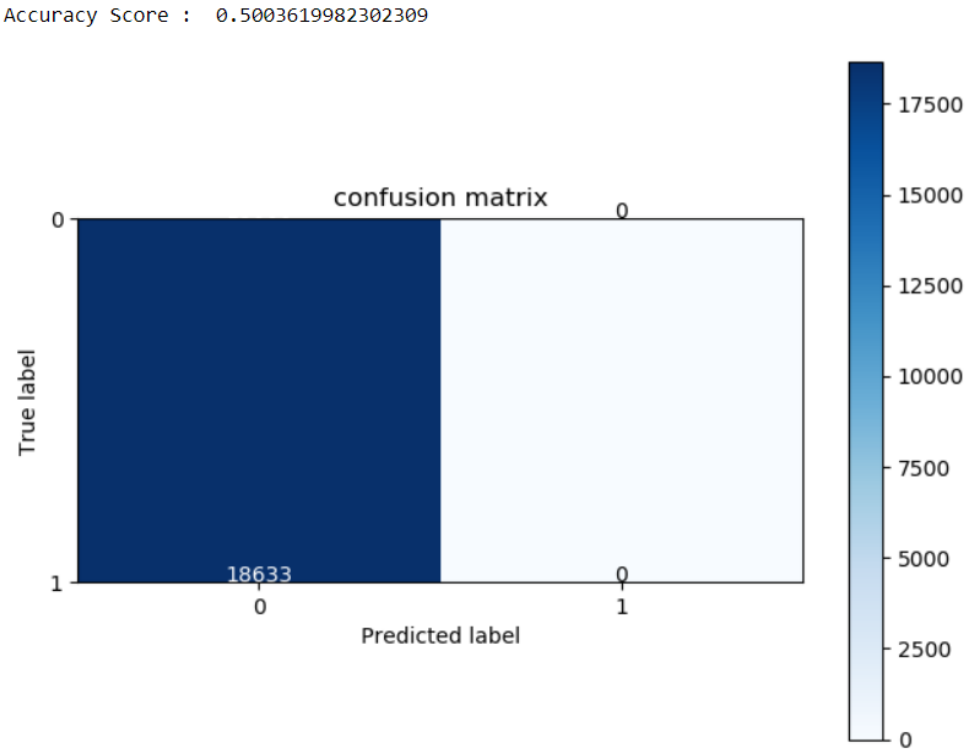

Figure 6: Accuracy and confusion matrix of Logistic Regression

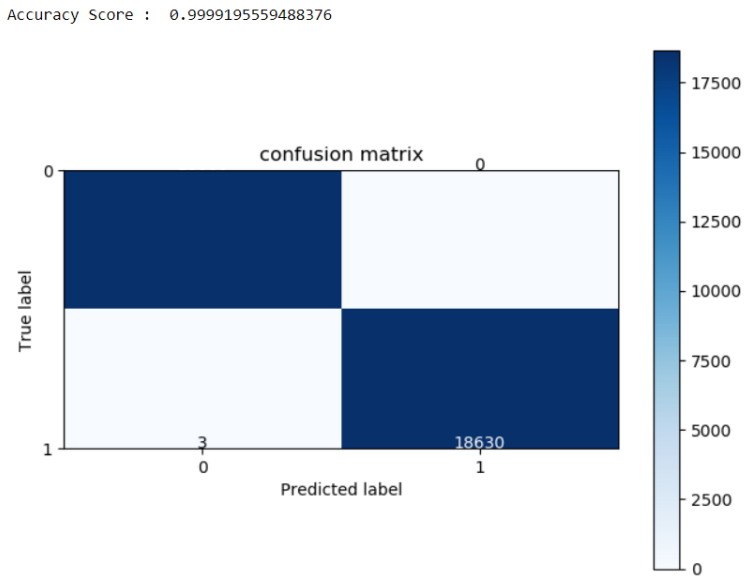

Figure 7: Accuracy score and confusion matrix of Random Forest classifier

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

*Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	18660
1	1.00	1.00	1.00	18633
accuracy			1.00	37293
macro avg	1.00	1.00	1.00	37293
weighted avg	1.00	1.00	1.00	37293

Figure 8: Classification Report of Random Forest classifier

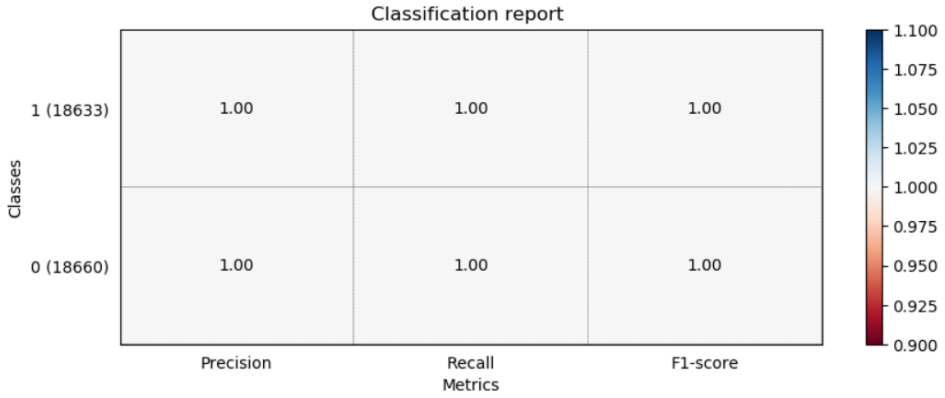

Figure 9: Heatmap of Random Forest Classifier Classification report

5. Conclusion

The Internet of Things (IoT) has grown rapidly as more and more devices become connected in various areas of our lives. However, this expansion has also brought about an increase in security vulnerabilities, specifically through botnet attacks. Botnets are networks of compromised devices that are controlled by malicious individuals. These attackers can use these botnets to launch cyber-attacks on a large scale. Two well-known botnet attacks that target IoT devices are the Gafgyt and Mirai botnets. These attacks take advantage of security weaknesses in IoT devices, compromising them and using them as bots to carry out their malicious activities. As these attacks become more frequent and severe, it is crucial to have a reliable way to detect and prevent IoT botnet attacks. The number of IoT devices is continually growing, and they are becoming increasingly integrated into critical infrastructure and our everyday lives. The consequences of botnet attacks can be very serious, disrupting services, compromising sensitive data, and even posing physical risks. To address this challenge, it is important to develop effective methods to identify and mitigate IoT botnet attacks promptly. One promising approach is the use of the Random Forest Classifier, which is a machine learning algorithm known for its accuracy and robustness in classification tasks. The Random Forest Classifier combines the predictions of multiple decision trees to accurately determine the class labels of input data. It has been widely used in various fields, including cybersecurity, due to its ability to handle complex and large datasets.

In the future, further research and development efforts should focus on improving methods for promptly detecting and mitigating IoT botnet attacks. This may involve enhancing the capabilities of the Random Forest Classifier to adapt to evolving attack techniques and incorporating other machine learning and data analysis techniques to improve detection accuracy. Additionally, it is important to strengthen the security measures of IoT devices themselves to minimize vulnerabilities and prevent them from being

infected by botnets in the first place. The ultimate goal is to ensure the security and reliability of IoT systems as they continue to grow and play a more significant role in different sectors of our lives.

References

[1] Hussain, B.; Du, Q.; Sun, B.; Han, Z. Deep Learning-Based DDoS-Attack Detection for Cyber-Physical System over 5G Network. IEEE Trans. Ind. Inform. 2021, 17, 860–870.

[2] Manimurugan, S.; Al-Mutairi, S.; Aborokbah, M.M.; Chilamkurti, N.; Ganesan, S.; Patan, R. Effective attack detection in internet of medical things smart environment using a deep belief neural network. IEEE Access 2020, 8, 77396–77404

[3] Tuan, T.A.; Long, H.V.; Son, L.H.; Kumar, R.; Priyadarshini, I.; Son, N.T.K. Performance evaluation of Botnet DDoS attack detection using machine learning. Evol. Intell. 2020, 13, 283–294

[4] Ullah, I.; Mahmoud, Q.H. A two-level flow-based anomalous activity detection system for IoT networks. Electronics 2020, 9, 530

[5] Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A stacking ensemble for network intrusion detection using heterogeneous datasets. Secur. Commun. Netw. 2020, 2020, 4586875.

[6] K. Alissa, T. Alyas, K. Zafar, Q. Abbas, Nadia Tabassum, Shadman Sakib, "Botnet Attack Detection in IoT Using Machine Learning", Computational Intelligence and Neuroscience, vol. 2022, Article ID 4515642, 14 pages, 2022. https://doi.org/10.1155/2022/4515642

[7] A. Kumar, M. Shridhar, S. Swaminathan, T. J. Lim, Machine learning-based early detection of IoT botnets using network-edge traffic, Computers & Security, Volume 117, 2022, 102693, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2022.102693.

[8] A. Alharbi and K. Alsubhi, "Botnet Detection Approach Using Graph-Based Machine Learning," in IEEE Access, vol. 9, pp. 99166-99180, 2021, doi: 10.1109/ACCESS.2021.3094183.

[9] Z. Shao, Sha Yuan, Yongli Wang, Adaptive online learning for IoT botnet detection, Information Sciences, Volume 574, 2021, Pages 84-95, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2021.05.076.

[10] M. Alauthman, N. Aslam, M. Al-kasassbeh, S. Khan, A. Al-Qerem, K. K. R. Choo, An efficient reinforcement learning-based Botnet detection approach, Journal of Network and Computer Applications, Volume 150, 2020, 102479, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2019.102479.

[11] B. Kaur, S. Dadkhah, F. Shoeleh, E. C. P. Neto, P. Xiong, S. Iqbal, P. Lamontagne, S. Ray, Ali A. Ghorbani, Internet of Things (IoT) security dataset evolution: Challenges and future directions, Internet of Things, Volume 22, 2023, 100780, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2023.100780.

[12] Alzahrani, M.Y., Bamhdi, A.M. Hybrid deep-learning model to detect botnet attacks over internet of things environments. Soft Comput 26, 7721–7735 (2022). https://doi.org/10.1007/s00500-022-06750-4

[13] S. M. Sajjad, M. R. Mufti, M. Yousaf, W. Aslam, R. Alshahrani, N. Nemri, H. Afzal, M. A. Khan, C Chen, "Detection and Blockchain-Based Collaborative Mitigation of Internet of Things Botnets", Wireless Communications and Mobile Computing, vol. 2022, Article ID 1194899, 26 pages, 2022. https://doi.org/10.1155/2022/1194899

[14] R. G. Azhari, V. Suryani, R. R. Pahlevi and A. A. Wardana, "The Detection of Mirai Botnet Attack on the Internet of Things (IoT) Device Using Support Vector Machine (SVM) Model," 2022 10th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2022, pp. 397-401, doi: 10.1109/ICoICT55009.2022.9914830.