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

Towards Resilient IoT: A Machine Learning Pipeline for Binary Classification of Benign and TCP Assaults

K. Hima Bindu[1], Dendukuri Bhagya Sri[2], Domakonda Karthik[2], Patharla Vamshi[2], Poralla Pavan[2], Gokul Anumula[2]

[1]Assistant Professor, [2]UG Scholar, [1,2]Department of CSE (Internet of Things)
[1,2]Malla Reddy College of Engineering and Management Sciences, Medchal, Hyderabad

ABSTRACT

Currently, there is a growing proliferation of Internet of Things (IoT) devices being connected to networks. As technology progresses, the risks associated with security breaches and cyberattacks, such as botnets, are also developing rapidly, with a heightened potential for severe attacks. The IoT-based botnet assault is more prevalent, exhibiting rapid propagation and causing more significant repercussions compared to other types of attacks. Recently, numerous studies have been carried out to identify and prevent these types of attacks through innovative methods. Therefore, numerous pertinent models, methodologies, and so on have been presented in recent years, with a considerable number of works reported in the research field. Numerous studies are endeavoring to safeguard the IoT ecosystem from botnet attacks. Nevertheless, there are numerous existing loopholes that need to be addressed in order to produce a highly efficient detection technique. These assaults impede the progress of IoT by causing disruptions in the networks and services that support IoT devices. Several recent research have suggested the use of machine learning (ML) and deep learning (DL) methods to identify and categorize botnet assaults in IoT setting. This study presents machine learning techniques for the classification of binary classes, specifically Benign or TCP assault. A comprehensive machine learning pipeline is suggested, encompassing exploratory data analysis to get in-depth understanding of the data, followed by preprocessing. Throughout this procedure, the data undergoes multiple key stages. Four machine learning models, namely random forest, k-nearest neighbour, support vector machines, and logistic regression, are suggested, trained, assessed, and evaluated using the dataset. Furthermore, apart from model accuracy, F1-score, recall, and precision are also taken into account.

Keywords: Internet of things, IoT security, Botnet attacks, TCP attack, Ensemble learning.

1.INTRODUCTION

The general idea of the Internet of Things (IoT) is to allow for communication between human-to-thing or thing-to-thing(s). Things denote sensors or devices, whilst human or an object is an entity that can request or deliver a service [1]. The interconnection amongst the entities is always complex. IoT is broadly acceptable and implemented in various domains, such as healthcare, smart home, and agriculture. However, IoT has a resource constraint and heterogeneous environments, such as low computational power and memory. These constraints create problems in providing and implementing a security solution in IoT devices. These constraints further escalate the existing challenges for IoT environment. Therefore, various kinds of attacks are possible due to the vulnerability of IoT devices. IoT-based botnet attack is one of the most popular, spreads faster and create more impact than other attacks. In recent years, several works have been conducted to detect and avoid this kind of attacks [2]–[3] by using novel approaches. Hence, a plethora of relevant of relevant models, methods, and etc. have been introduced over the past few years, with quite a reasonable number of studies reported in the research domain.

Many studies are trying to protect against these botnet attacks on the IoT environment. However, there are many gaps still existing to develop an effective detection mechanism. An intrusion detection system

(IDS) is one of the efficient ways to deal with attacks. However, the traditional IDSs are often not able to be deployed for the IoT environments due to the resource constraint problem of these devices. The complex cryptographic mechanisms cannot be embedded in many IoT devices either for the same reason. There are mainly two kinds of IDSs: the anomaly and misuse approaches. The misuse-based, also called the signature-based, approach, is based on the attacks' signatures, and they can also be found in most public IDSs, specifically Suricata [4]. Formally, the attacker can easily circumvent the signature-based approaches, and these mechanisms cannot guarantee to detect the unknown attacks and the variances of known attacks. The anomaly-based systems are based on normal data and can support to identify the unknown attacks. However, the different nature of IoT devices is being faced with the difficulty of collecting common normal data. The machine learning-based detection can guarantee detection of not only the known attacks and their variances. Therefore, we proposed a machine learning-based botnet attack detection architecture. We also adopted a feature selection method to reduce the demand for processing resources for performing the detection system on resource constraint devices. The experiment results indicate that the detection accuracy of our proposed system is high enough to detect the botnet attacks. Moreover, it can support the extension for detecting the new distinct kinds of attacks.

2. LITERATURE SURVEY

Soe et al. [5] adopted a lightweight detection system with a high performance. The overall detection performance achieves around 99% for the botnet attack detection using three different ML algorithms, including artificial neural network (ANN), J48 decision tree, and Naïve Bayes. The experiment result indicated that the proposed architecture can effectively detect botnet-based attacks, and also can be extended with corresponding sub-engines for new kinds of attacks.

Ali et al. [6] outlined the existing proposed contributions, datasets utilised, network forensic methods utilised and research focus of the primary selected studies. The demographic characteristics of primary studies were also outlined. The result of this review revealed that research in this domain is gaining momentum, particularly in the last 3 years (2018-2020). Nine key contributions were also identified, with Evaluation, System, and Model being the most conducted.

Irfan et al. [7] classified the incoming data in the IoT, contain a malware or not. In this research, this work under sample the dataset because the datasets contain imbalance class. After that, this work classified the sample using Random Forest. This work used Naive Bayes, K-Nearest Neighbor and Decision Tree too as a comparison. The dataset that has been used in this research are from UCI Machine Learning Depository's Website. The dataset showed the data traffic from the IoT Device in a normal condition and attacked by Mirai or Bashlite.

Shah et al. [8] presented a concept called 'login puzzle' to prevent capture of IoT devices in a large scale. Login puzzle is a variant of client puzzle, which presented a puzzle to the remote device during the login process to prevent unrestricted log-in attempts. Login puzzle is a set of multiple mini puzzles with a variable complexity, which the remote device is required to solve before logging into any IoT device. Every unsuccessful log-in attempt increases the complexity of solving the login puzzle for the next attempt. This paper introduced a novel mechanism to change the complexity of puzzle after every unsuccessful login attempt. If each IoT device had used login puzzle, Mirai attack would have required almost two months to acquire devices, while it acquired them in 20 h.

Tzagkarakis et al. [9] presented an IoT botnet attack detection method based on a sparsity representation framework using a reconstruction error thresholding rule for identifying malicious network traffic at the IoT edge coming from compromised IoT devices. The botnet attack detection is performed based on small-sized benign IoT network traffic data, and thus we have no prior knowledge about malicious

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

IoT traffic data. We present our results on a real IoT-based network dataset and show the efficacy of proposed technique against a reconstruction error-based autoencoder approach.

Meidan et al. [10] proposed a novel network-based anomaly detection method for the IoT called N-BaIoT that extracts behavior snapshots of the network and uses deep autoencoders to detect anomalous network traffic from compromised IoT devices. To evaluate the method, this work infected nine commercial IoT devices in our lab with two widely known IoT-based botnets, Mirai and BASHLITE. The evaluation results demonstrated the proposed methods ability to detect the attacks accurately and instantly as they were being launched from the compromised IoT devices that were part of a botnet.

Popoola et al. [11] proposed the federated DL (FDL) method for zero-day botnet attack detection to avoid data privacy leakage in IoT-edge devices. In this method, an optimal deep neural network (DNN) architecture is employed for network traffic classification. A model parameter server remotely coordinates the independent training of the DNN models in multiple IoT-edge devices, while the federated averaging (FedAvg) algorithm is used to aggregate local model updates. A global DNN model is produced after several communication rounds between the model parameter server and the IoT-edge devices. The zero-day botnet attack scenarios in IoT-edge devices are simulated with the Bot-IoT and N-BaIoT data sets.

Hussain et al. [12] produced a generic scanning and DDoS attack dataset by generating 33 types of scans and 60 types of DDoS attacks. In addition, this work partially integrated the scan and DDoS attack samples from three publicly available datasets for maximum attack coverage to better train the machine learning algorithms. Afterwards, this work proposed a two-fold machine learning approach to prevent and detect IoT botnet attacks. In the first fold, this work trained a state-of-the-art deep learning model, i.e., ResNet-18 to detect the scanning activity in the premature attack stage to prevent IoT botnet attacks. While, in the second fold, this work trained another ResNet-18 model for DDoS attack identification to detect IoT botnet attacks.

3. PROPOSED SYSTEM

The main goal of this project is to develop a machine learning-based system capable of identifying botnet attacks within IoT device data. Botnets are networks of compromised devices controlled by malicious actors, and detecting their activities is crucial for network security.

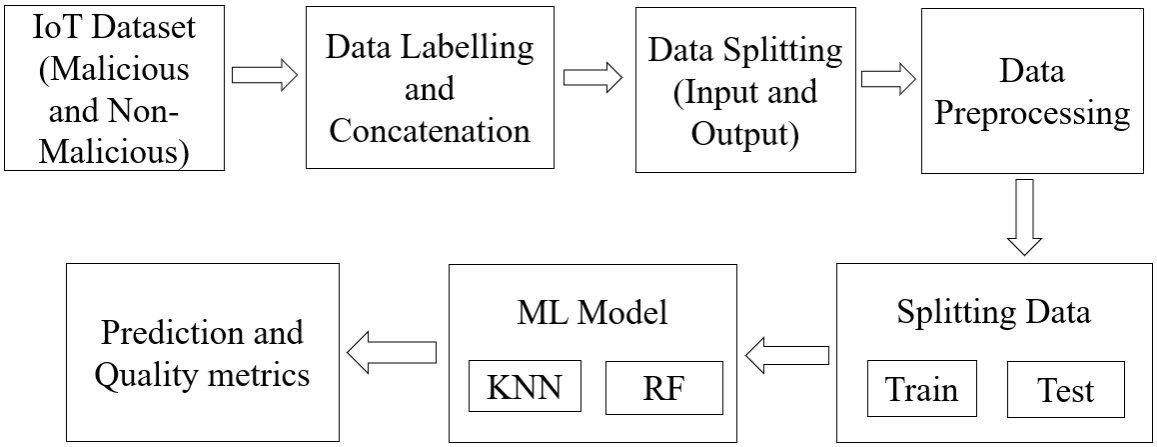


Fig. 1: Block diagram of proposed system.

Confusion Matrix: A confusion matrix is created for each model to visualize its performance. Confusion matrices show true positives, true negatives, false positives, and false negatives, helping to understand the model's ability to classify benign and malicious data points.

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

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. 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.

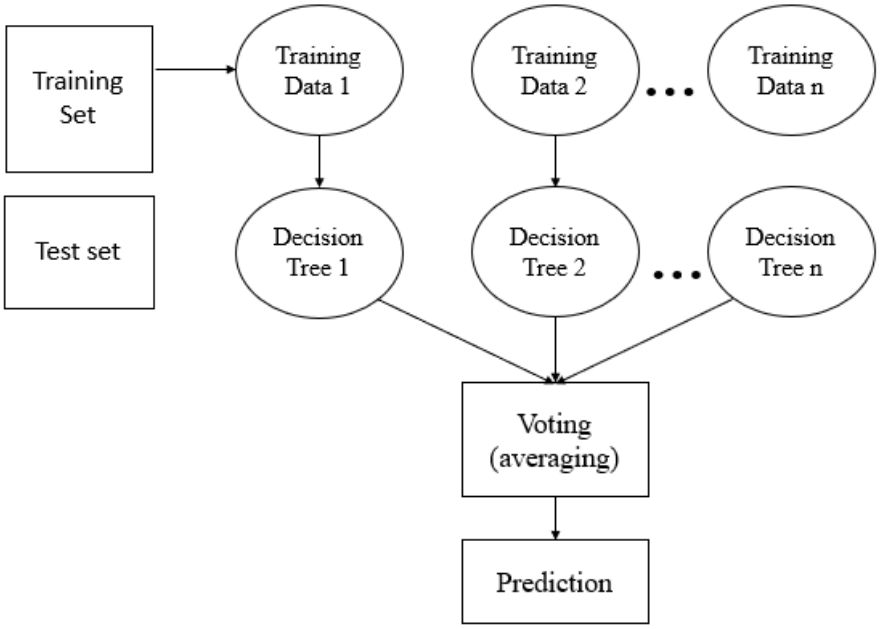


Fig. 2: Random Forest algorithm.

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.

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.

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.

4. RESULTS AND DISCUSSION

Implementation description

This project implements an ensemble model to detect botnet attacks from IoT device data. It loads and combines benign and malicious IoT device data and prepares the data by adding labels and normalizing it then splits the data into training and testing sets. Afterwards, it defines, and trains four machine learning models and evaluates the models using accuracy scores, classification reports, and confusion matrices. Here is the step-by-step explanation:

1. Importing Libraries: It starts by importing necessary Python libraries, including pandas, numpy, matplotlib, vpython, and various machine learning libraries from scikit-learn (e.g., LogisticRegression, SVC, RandomForestClassifier, KNeighborsClassifier).

2. Loading Datasets

- Two datasets are loaded: "benign_traffic.csv" (non-malicious IoT device data) and "junk.csv" (malicious IoT device data).
- The data from these CSV files is read into Pandas DataFrames.
- The first few rows of both datasets are displayed.

3. Dataset Description: Descriptive statistics (e.g., mean, standard deviation) are displayed for both datasets to provide an overview of the data.

4. Adding Labels: A binary label is added to both datasets. Non-malicious data is labeled as 0, and malicious data is labeled as 1.

5. Combining Datasets: The two datasets are concatenated into a single dataset called "dataset" by stacking them vertically.

6. Splitting Input and Output

- The "output" column is extracted as the target variable (y or output), and the rest of the columns are considered as input features (X or Input).
- The shapes of the output and input are displayed.

7. Preprocessing

- The output array is flattened using NumPy.
- Z-score normalization is applied to the entire dataset, standardizing the features to have a mean of 0 and a standard deviation of 1.

8. Splitting the Dataset

- The dataset is split into training and testing sets in an 80-20 ratio using the train_test_split function from scikit-learn.
- The number of samples in the training and testing sets is displayed.

10. Model Training and Prediction

- A function named train_predict is defined to train a given model and make predictions on the test set.
- Each of the four models is trained on the training data, and predictions are made on the test data.
- The accuracy score and classification report (including precision, recall, F1-score, etc.) are printed for each model.

Dataset description

The dataset is a collection of features extracted from a network traffic stream, with various statistics calculated over different time frames. Each feature is related to a specific type of stream aggregation and time frame. Let's break down the attribute information and the dataset:

Stream Aggregation:

- H: Stats summarizing the recent traffic from this packet's host (IP)
- MI: Stats summarizing the recent traffic from this packet's host (IP + MAC)
- HH: Stats summarizing the recent traffic going from this packet's host (IP) to the packet's destination host.
- HH_jit: Stats summarizing the jitter of the traffic going from this packet's host (IP) to the packet's destination host.
- HpHp: Stats summarizing the recent traffic going from this packet's host+port (IP) to the packet's destination host+port.

Statistics Extracted:

For each combination of stream aggregation, time frame, and statistic, the dataset contains the following features:

- weight: The weight of the stream, which can be viewed as the number of items observed in recent history.
- mean: The mean of the stream's values.
- std: The standard deviation of the stream's values.
- radius: The root squared sum of the two streams' variances.
- magnitude: The root squared sum of the two streams' means.
- cov: An approximated covariance between two streams.
- pcc: An approximated correlation coefficient between two streams.

The dataset itself consists of multiple columns, each corresponding to a specific combination of stream aggregation, time frame, and statistic. The naming convention follows this pattern: <Stream_Aggregation>_<Time_Frame>_<Statistic>

For example, one column is named MI_dir_L5_weight, which indicates the weight of the stream aggregated by source MAC-IP with a decay factor capturing recent history up to 5-time units.

In this dataset, each row likely represents a sample or instance of network traffic, and the values in the columns represent the calculated statistics for various combinations of stream aggregation, time frame, and statistic.

Results description

Figure 3 shows a visual representation of data collected from Internet of Things (IoT) devices that are operating normally and not exhibiting any malicious behavior. It displays various features or attributes of the data points collected from these devices, such as network traffic patterns, communication

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

protocols, and other relevant parameters. Each data point in this figure represents a non-malicious activity. Figure 4 illustrates the data collected from IoT devices that are exhibiting malicious or abnormal behavior. The visual representation highlights the anomalies or suspicious patterns in the data that indicate potential attacks or unauthorized activities. Each data point in this figure represents a malicious activity.

	MI_dir_L5_weight	MI_dir_L5_mean	MI_dir_L5_variance	MI_dir_L3_weight	MI_dir_L3_mean	MI_dir_L3_variance	MI_dir_L1_weight	MI_dir_L1_mean
0	1.000000	60.0	0.0	1.000000	60.0	0.0	1.000000	60.0
1	1.000000	60.0	0.0	1.000000	60.0	0.0	1.000000	60.0
2	1.000000	60.0	0.0	1.000000	60.0	0.0	1.000000	60.0
3	1.000000	590.0	0.0	1.000000	590.0	0.0	1.000000	590.0
4	1.927179	590.0	0.0	1.955648	590.0	0.0	1.984992	590.0

5 rows × 115 columns

MI_dir_L1_variance	MI_dir_L0.1_weight	...	HpHp_L0.1_radius	HpHp_L0.1_covariance	HpHp_L0.1_pcc	HpHp_L0.01_weight	HpHp_L0.01_mean	HpHp_L0.01_std
0.0	1.000000	...	0.000000	0.0	0.0	1.000000	60.000000	0.000000e+00
0.0	1.000000	...	0.000000	0.0	0.0	1.061357	60.000000	9.540000e-07
0.0	1.000000	...	0.000000	0.0	0.0	1.000000	60.000000	0.000000e+00
0.0	1.000000	...	9591.400096	0.0	0.0	5.832783	388.850426	9.199164e+01
0.0	1.998489	...	6354.393843	0.0	0.0	6.831901	418.293119	1.108120e+02

HpHp_L0.01_magnitude	HpHp_L0.01_radius	HpHp_L0.01_covariance	HpHp_L0.01_pcc
60.000000	0.000000e+00	0.0	0.0
60.000000	9.090000e-13	0.0	0.0
60.000000	0.000000e+00	0.0	0.0
388.850426	8.462461e+03	0.0	0.0
418.293119	1.227931e+04	0.0	0.0

Figure 3: Illustration of sample non-malicious dataset obtained from IoT device.

	MI_dir_L5_weight	MI_dir_L5_mean	MI_dir_L5_variance	MI_dir_L3_weight	MI_dir_L3_mean	MI_dir_L3_variance	MI_dir_L1_weight	MI_dir_L1_mean
0	1.000000	98.000000	0.000000	1.000000	98.000000	0.000000e+00	1.000000	98.000000
1	1.029191	98.000000	0.000000	1.119992	98.000000	1.818989e-12	1.493231	98.000000
2	1.077270	68.295282	68.180692	1.236877	72.128396	1.585514e+02	1.889672	81.065843
3	2.038100	71.094319	42.860737	2.209694	72.975393	8.766659e+01	2.875726	78.608779
4	3.000117	72.062842	30.450564	3.184892	73.297101	6.036696e+01	3.864926	77.416316

5 rows × 115 columns

HpHp_L0.01_magnitude	HpHp_L0.01_radius	HpHp_L0.01_covariance	HpHp_L0.01_pcc
98.000000	0.0	0.0	0.0
138.592929	0.0	0.0	0.0
114.039467	0.0	0.0	0.0
74.000000	0.0	0.0	0.0
74.000000	0.0	0.0	0.0

Figure 4: Illustration of sample malicious dataset obtained from IoT device.

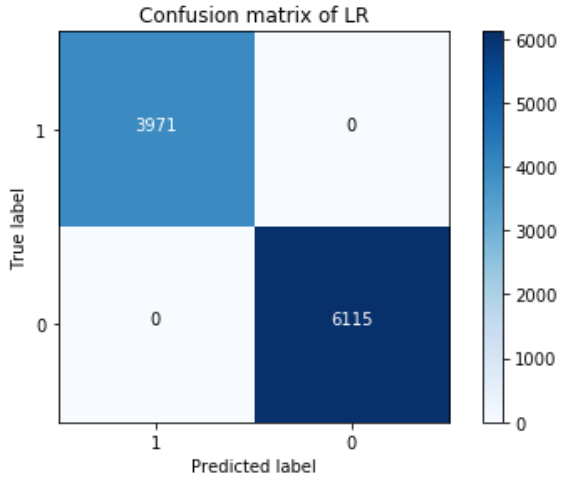
```
There are 19528 records with 115 features in non-malicious dataset

There are 30898 records with 115 features in malicious dataset

Adding output column in the datasets with all 0 in pureData dataset and all 1 in maliciousDataset dataset
```

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

	MI_dir_L5_weight	MI_dir_L5_mean	MI_dir_L5_variance	MI_dir_L3_weight	MI_dir_L3_mean	MI_dir_L3_variance	MI_dir_L1_weight	MI_dir_L1_mean
0	1.000000	60.000000	0.000000	1.000000	60.000000	0.000000	1.000000	60.000000
1	1.000000	60.000000	0.000000	1.000000	60.000000	0.000000	1.000000	60.000000
2	1.000000	60.000000	0.000000	1.000000	60.000000	0.000000	1.000000	60.000000
3	1.000000	590.000000	0.000000	1.000000	590.000000	0.000000	1.000000	590.000000
4	1.927179	590.000000	0.000000	1.955648	590.000000	0.000000	1.984992	590.000000
...	...	...	...	...	...	...	...	...
30893	166.931803	74.005716	0.137159	277.615508	74.013930	0.334630	756.938339	74.033362
30894	162.133219	74.005681	0.136313	272.788653	74.013879	0.333404	752.605073	74.033318
30895	163.124243	74.005646	0.135477	273.779592	74.013828	0.332187	753.596740	74.033274
30896	164.123165	74.005612	0.134652	274.778506	74.013777	0.330979	754.595743	74.033230
30897	165.119774	74.005578	0.133837	275.775101	74.013728	0.329779	755.592626	74.033186

50426 rows × 116 columns

Figure 5 demonstrate the obtained confusion matrices using various ML models. Confusion matrices are used to assess the performance of classification algorithms. Each matrix shows the number of true positives, true negatives, false positives, and false negatives for a given model's predictions. By comparing these matrices, we can evaluate which model is performing better in terms of correctly identifying normal and malicious activities.



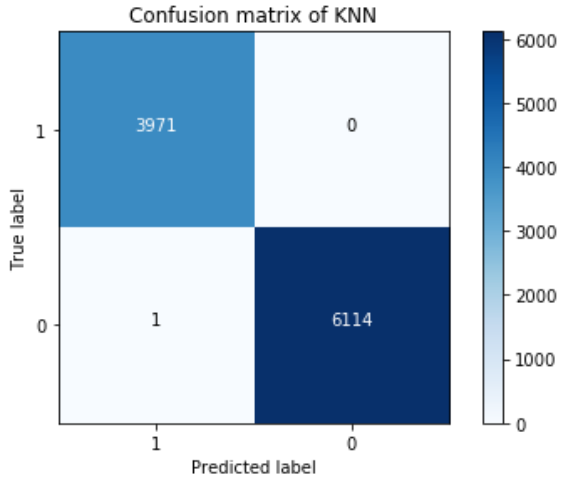


Figure 6: Confusion matrices obtained using various ML models.

Table 1. Performance comparison of various ML models for predicting the botnet attack in IoT device data.

Model/Metric	LR model	SVM classifier	RF model	KNN classifier
Accuracy (%)	100	99.98	100	99.99
Precision (%)	100	100	100	100
Recall (%)	100	100	100	100
F1-score (%)	100	100	100	100

Table 1 presents a comprehensive performance comparison of different machine learning (ML) models employed for the prediction of botnet attacks in data obtained from IoT devices. The table showcases four ML models: LR model, SVM classifier, RF model, and KNN classifier. The metrics assessed for each model include Accuracy, Precision, Recall, and F1-score.

The LR model achieved exceptional results across all metrics. It displayed a remarkable accuracy rate of 100%, indicating that it accurately classified all instances, both botnet attacks and benign activities. This suggests that the LR model has a comprehensive understanding of the underlying patterns in the data, enabling it to make precise predictions. Additionally, the LR model demonstrated 100% precision, recall, and F1-score. This implies that the model not only made accurate positive predictions (precision) but also correctly identified all true positive instances (recall), leading to a harmonic balance between precision and recall, as represented by the F1-score. The SVM classifier performed impressively as well, with an accuracy of 99.98%. This indicates that the model almost perfectly classified instances into their respective categories. Similar to the LR model, the SVM classifier attained 100% precision, recall, and F1-score, signifying its strong capability to make accurate predictions and correctly identify botnet attacks. The RF model achieved a perfect accuracy of 100%, matching the performance of the LR model. This suggests that the RF model was able to effectively capture the complexities and variations in the data. The precision, recall, and F1-score also reached 100%, indicating consistent and reliable performance across these evaluation metrics. The KNN classifier, while slightly behind the other models in terms of accuracy with 99.99%, still demonstrated outstanding predictive capability. Like the other models, the KNN classifier achieved perfect precision, recall, and F1-score. This showcases the model's ability to make accurate and consistent predictions.

Finally, Table 1 showcases the remarkable performance of all the evaluated ML models for predicting botnet attacks in IoT device data. Each model exhibited near-perfect accuracy, precision, recall, and F1-score, implying their proficiency in accurately identifying both botnet attacks and benign activities. These findings suggest that the models are well-suited for detecting and preventing malicious activities within IoT networks, enhancing the overall security and reliability of IoT devices and systems.

5. CONCLUSION

Cyber-attacks involving botnets are multi-stage attacks and primarily occur in IoT environments; they begin with scanning activity and conclude with distributed denial of service (DDoS). Most existing studies concern detecting botnet attacks after IoT devices become compromised and start performing DDoS attacks. Furthermore, most machine learning-based botnet detection models are limited to a specific dataset on which they are trained. Consequently, these solutions do not perform well on other datasets due to the diversity of attack patterns. In this work, real traffic data is used for experimentation. EDA (Exploratory Data Analysis) is the statistical analysis phase through which the whole dataset is analyzed. The model will be able to be trained on a large data set in the future. ResNet50 and LSTM

models, deep learning models can also be used in run-time Botnet detection. Besides being integrated with front-end web applications, the research' model can also be used with back-end web applications.

REFERENCES

[1] S. Dange and M. Chatterjee, "IoT botnet: The largest threat to the iot network" in Data Communication and Networks, Cham, Switzerland: Springer, pp. 137-157, 2020.

[2] J. Ceron, K. Steding-Jessen, C. Hoepers, L. Granville and C. Margi, "Improving IoT botnet investigation using an adaptive network layer", Sensors, vol. 19, no. 3, pp. 727, Feb. 2019.

[3] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, et al., "N-baiot-network-based detection of iot botnet attacks using deep autoencoders", IEEE Pervas. Comput., vol. 17, no. 3, pp. 12-22, 2018.

[4] Shah, S.A.R.; Issac, B. Performance comparison of intrusion detection systems and application of machine learning to Snort system. Futur. Gener. Comput. Syst. 2018, 80, 157–170.

[5] Soe YN, Feng Y, Santosa PI, Hartanto R, Sakurai K. Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture. Sensors. 2020; 20(16):4372. https://doi.org/10.3390/s20164372

[6] I. Ali et al., "Systematic Literature Review on IoT-Based Botnet Attack," in IEEE Access, vol. 8, pp. 212220-212232, 2020, doi: 10.1109/ACCESS.2020.3039985.

[7] Irfan, I. M. Wildani and I. N. Yulita, "Classifying botnet attack on Internet of Things device using random forest", IOP Conf. Ser. Earth Environ. Sci., vol. 248, Apr. 2019.

[8] Shah, T., Venkatesan, S. (2019). A Method to Secure IoT Devices Against Botnet Attacks. In: Issarny, V., Palanisamy, B., Zhang, LJ. (eds) Internet of Things – ICIOT 2019. ICIOT 2019. Lecture Notes in Computer Science(), vol 11519. Springer, Cham. https://doi.org/10.1007/978-3-030-23357-0_3

[9] C. Tzagkarakis, N. Petroulakis and S. Ioannidis, "Botnet Attack Detection at the IoT Edge Based on Sparse Representation," 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 2019, pp. 1-6, doi: 10.1109/GIOTS.2019.8766388.

[10] Y. Meidan et al., "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," in IEEE Pervasive Computing, vol. 17, no. 3, pp. 12-22, Jul.-Sep. 2018, doi: 10.1109/MPRV.2018.03367731.

[11] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh and O. Jogunola, "Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT-Edge Devices," in IEEE Internet of Things Journal, vol. 9, no. 5, pp. 3930-3944, 1 March1, 2022, doi: 10.1109/JIOT.2021.3100755.

[12] F. Hussain et al., "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," in IEEE Access, vol. 9, pp. 163412-163430, 2021, doi: 10.1109/ACCESS.2021.3131014.