

An Efficient Spam Detection Technique for IoT Devices using Machine Learning. (Java)

Hrutvij Thorat¹, Meghana Pawar², Sakshi Avachar³, Prof. Salunke N. S.⁴

Students, Department of Computer Engineering¹⁻³

Guide, Department of Computer Engineering⁴

S.P.I.T. Polytechnic, Kurund, Ahmednagar, Maharashtra, India

Abstract: Millions of devices with sensors and actuators linked via wired or wireless channels for transmission of data make up the Internet of Things (IoT). In years to come, the amount of data that these devices release will grow substantially. In such a setting, machine learning algorithms can be important in maintaining biotechnology-based verification and security, as well as in detecting errors to enhance the security and usability of Network of Things systems. Attackers, on the other hand, use machine learning algorithms to take advantage of issues in intelligent IoT-based systems. Inspired by this, in this research, we suggest using machine learning to detect spam so as protect IoT devices. This approach assesses five machine learning models with an array of measured.

Keywords: Internet of Things

I. INTRODUCTION

1.1 Target

The main objectives of this research are to provide an extensive and comprehensive evaluation of the state of the art in the field of review spam detection using a range of machine learning techniques and to establish a methodology for future study.

The Internet of Things (IoT) allows possible for real-world items to implement and merge regardless of their geographical locations. To resolve security threats such intrusions, spoofing attacks, DoS attacks, jamming, eavesdropping, spam, and malware, IoT applications need to protect user privacy.

1.2 SCOPE

This project's primary goals are to provide an extensive and extensive evaluation of the state of the art in the field of review spam detection using a range of machine learning techniques and to establish a methodology for future study. The Internet of Things (IoT) makes it possible for real-world items to implement and combine irrespective of their geographical locations. Privacy and protection measures are essential yet difficult to implement in such a network management and control environment. To resolve security threats such intrusions, spoofing attacks, DoS attacks, jamming, eavesdropping, spam, and malware, IoT applications need to protect user privacy. For instance, wearable technology should protect privacy by collecting and transmitting user health data to a linked smartphone. It has been located.

II. SYSTEM ANALYSIS

2.1 CURRENT SYSTEM

- Denial of service (DDoS) attacks: To prevent IoT devices from accessing different services, the attackers may attack the target database with unexpected queries. DDoS has the ability to consume all of the service provider's resources.
- RFID attacks: These are attacks that target an Internet of Things device's physical layer. Common attacks that can occur at the sensor node include brute-forcing cryptography keys, attacks on availability, attacks on authenticity, and attacks on secret.

- Internet attacks: To access a variety of resources, the IoT device can maintain an Internet connection. Spamming strategies are used by those who wish to obtain information from other systems or to increase the number of visits to their target website. NFC attacks: The primary focus of these attacks is electronic payment fraud. Eavesdropping, tag change, and unencrypted traffic are the potential attacks.
- Conditional privacy protection is the answer to this issue. Thus, the attacker is unable to utilize the user's public key to generate the identical profile. The trusted service manager's random public keys serve as a basis for this concept.

2.1.1 DISADVANTAGES OF THE EXISTING SYSTEM

- The system is less effective in the current work because it does not use a machine learning framework for spam detection in the Internet of Things.
- This system performs worse, making it obvious that supervised machine learning techniques are not present.

2.2 PROBLEM STATEMENT

- In order to address security threats such as malware, spam, eavesdropping, jamming, spoofing, and DoS attacks, IoT applications have to protect user privacy.
- IoT device safety procedures vary depending on the size and kind of enterprise implementing them.

2.3 THE PROPOSED SYSTEM

Smart devices have become essential in the digital world. There should be no spam in the data that is recovered from these devices.

Since the data is gathered from multiple domains, retrieving it from different IoT devices is a significant difficulty.

Because the Internet of Things involves numerous devices, a vast amount of mixed and varied data is generated. Here, spam in specific IoT devices is detected using support vector machines.

This data can be referred to as IoT data. Real-time, multi-source, sparse, and rich data are some of the characteristics of IoT data.

Here, we illustrate the reliability of an IoT device under different conditions using the Randomforest method.

- 1) Five distinct machine learning models are used to validate the suggested spam detection strategy.
- 2) A proposed technique calculates each model's spam city score, which is subsequently utilized for intelligent decision-making and detection.
- 3) The reliability of IoT devices is examined using various evaluation criteria based on the spam city score calculated in the preceding step.

The goal is to fix the problems with the IoT devices that are installed in homes. However, the recommended method takes into account every element of data engineering before confirming it using machine learning models.

2.3.1 PROPOSED SYSTEM ADVANTAGES

Five distinct machine learning models are used to validate the suggested spam detection technique.

- An method is suggested to calculate each model's spam city score, which is thereafter utilized for detection and wise decision-making.
- Using multiple states evaluation measures, the dependability of IoT devices is investigated based on the spam city score calculated in the previous phase.

III. RELATED WORK

IoT systems, which include networks, services, and devices, are exposed to privacy leaks as well as network, physical, and application threats.

- 1) Denial of service (DDoS) attacks: To stop IoT devices from accessing a variety of services, the attackers will bombard the target data with unwanted requests. Typically, these fraudulent queries generated by an IoT device

network are referred to as bots. DDoS will use up every resource the service provider has to provide. In addition to blocking genuine users, it can render network resources unavailable.

2) RFID attacks: These are the attacks that are required at the IoT device's physical layer. The device's integrity is compromised as a result of this attack. Attackers promise to alter the data while it is being transmitted over the network or while it is stored on the node. Common attacks that can be launched against the sensing element node include brute-forcing cryptography keys, attacks on availability, attacks on credibility, and attacks on privacy. Encryption, controlled access management, and parole protection are some of the protections against such attacks.

3) Web attacks: To access a variety of resources, the IoT device will remain linked to the Internet. The United Nations agency employs spamming techniques in order to obtain data from other systems or to make sure that their target website is continuously viewed

[4]. Ad fraud is a typical technique used for something similar. For money-making purposes, it creates the replacement clicks at a specific website. Such a dynamic group is regarded as cybercriminals.

[5] if applicable, close to field communication (NFC). NFC harms: These attacks are mostly connected to unauthorized electronic payments. Eavesdropping, tag modification, and unencrypted traffic are the possible crimes. The conditional privacy protection is the solution to this disadvantage.

As a result, the criminal is unable to create a comparable profile using the user's public key

[6]. This model uses a reliable service manager to generate random public keys.

IV. METHODOLOGY

This work's major contributions are the collection of SPAM datasets from the Kaggle website and the deployment of deep learning techniques for spam detection in IOT device applications

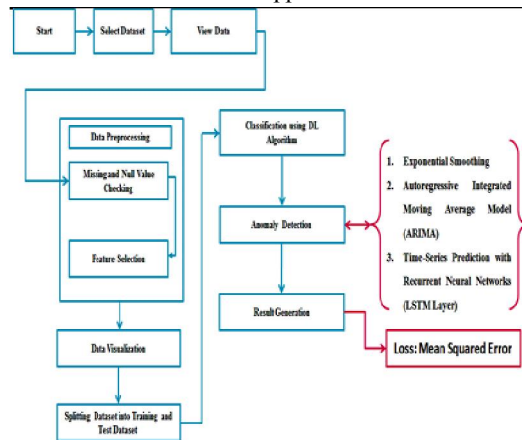


Figure 1: Flow Chart

Steps

1. To begin, download the EEG dataset from the Kaggle website, a major provider of machine learning repositories and datasets for study.
2. Proceed with the data preprocessing, which includes handling missing data and eliminating null values.
3. Next, extract the features of the data and assess the independent and dependent variables.
4. Next, use the deep learning (LSTM) approach-based classification method.
5. Next, create a confusion matrix and display every projected class, including false negative, true positive, and false positive.
6. Next, determine the performance characteristics in terms of accuracy, precision, recall, F_measure, and error rate using the standard formulas.

V. RESULTS OF SIMULATION

The recommended calculation is executed out using Python Spyder 3.7. The tools available in Spyder Climate for different strategies are supported by the sklearn, numpy, pandas, matplotlib, pyplot, seaborn, and os libraries.

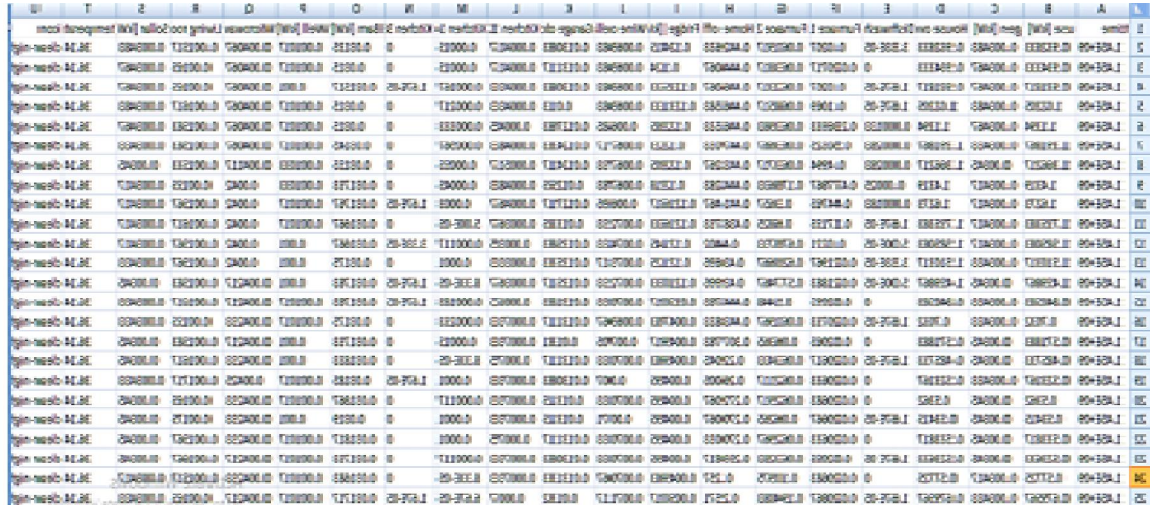


Figure 2: Original dataset in .csv file

The dataset, which was obtained from the Kaggle machine learning platform, is displayed in figure 2.

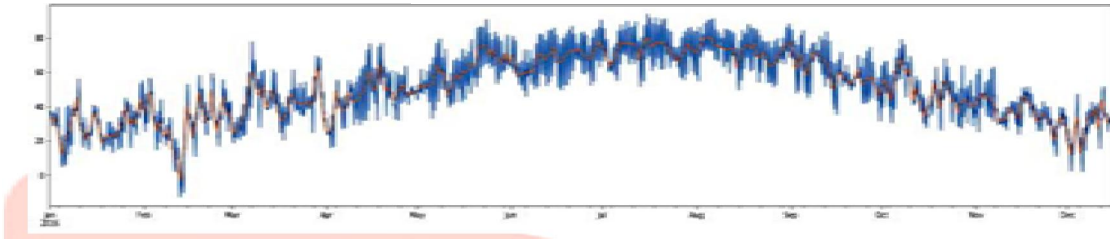


Figure 3: IOT device with month

The total number of IOT devices is shown in Figure 3 in a monthly basis. The months are January through December.

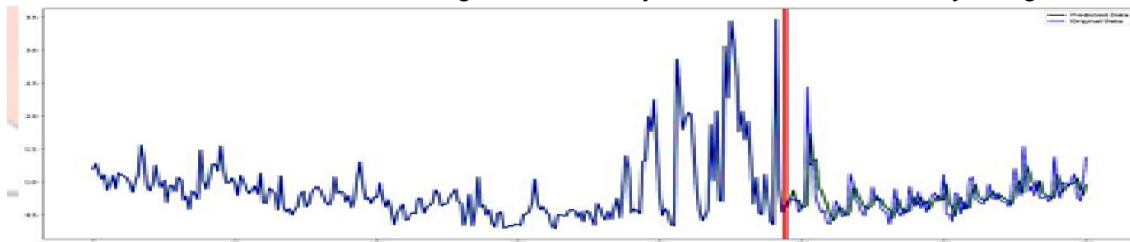


Figure 4: Predicted and actual data

Predicted and actual data from the provided dataset are displayed in Figure 4. The result graph makes it clear that the majority of the data had been expected.

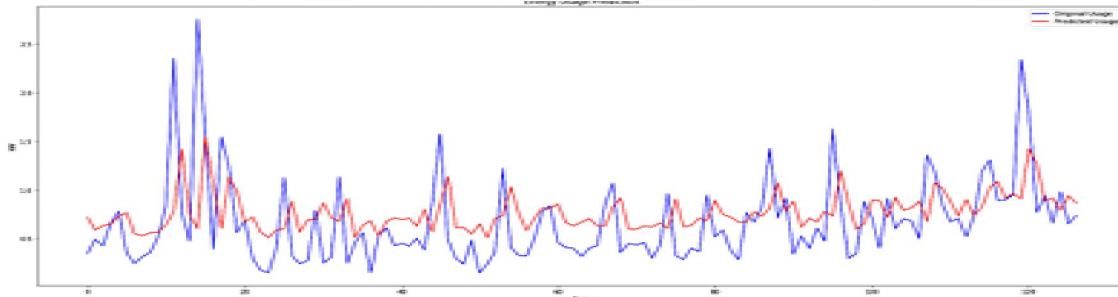


Figure 5: Energy Prediction

Energy prediction is shown in Figure 5. The graphical representation of the initial and expected energy is displayed.

Sr. No.	Parameters	Previous Work [1]	Proposed Work
1	Technique	Machine Learning (Linear Model)	Deep Learning (RNN-LSTM)
2	Accuracy	91.8%	95.72 %
3	Classification Error	8.2 %	4.28 %

Table 1: Result Comparison

VI. SYSTEM DEVELOPMENT

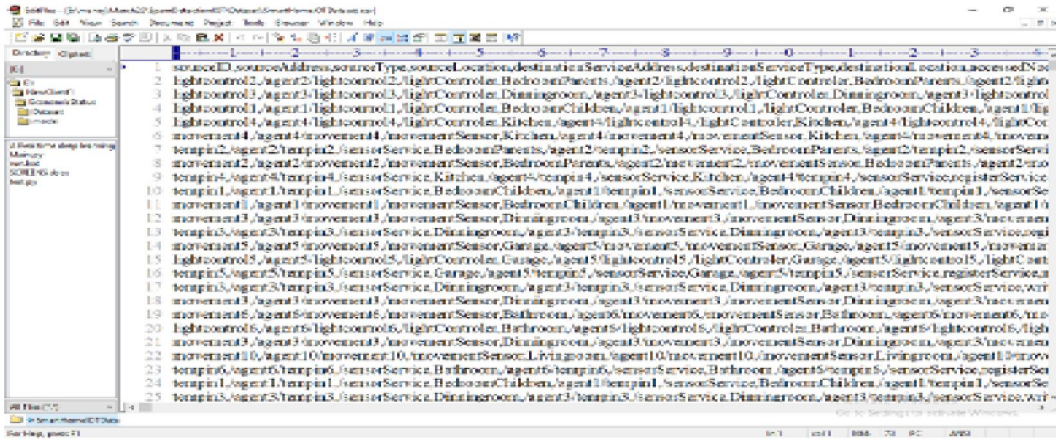
IOT devices are tiny sensors that collect data from the environment and send it to a base station or centralized server. However, some attackers may hack these sensors and then inject false information, which will be sent to the base station and may cause it to make a mistaken decision. For instance, if a health care sensor is attached to a patient's body and sends information about the patient's heart condition to the hospital server, the hospital may prescribe the wrong medication to the patient if the attacker hacks and sends false information.

This sensor could be used for temperature monitoring in agriculture, home monitoring, or anything else. To secure such sensor data, the author is assessing the effectiveness of five machine learning algorithms: Bayesian Generalized Linear Model, Boosted Linear Model, Extreme Gradient Boosting, Bagged Model, and Generalized Linear Model with Stepwise Feature Selection. All four of the first algorithms are being implemented, and the PCA features selection method is being added for the final algorithm.

The REFIT Smart Home dataset, which includes information on IOT signals and certain normal and spam properties, was used by the author to construct this project. We will use this dataset to train all of the algorithms mentioned above before calculating the scores of attack and normal signals

The dataset screen grab is shown below:

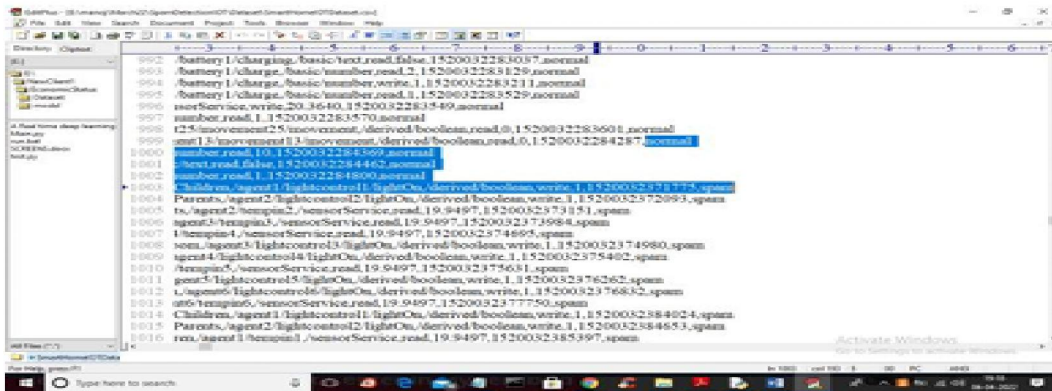
The dataset column names are shown in the first row of the next screen, followed by the dataset values in the following rows, and the class labels Normal or SPAM in the final column for each row.



sourceID	sourceAddress	sourceType	sourceLocation	destinationAddress	destinationType	destinationLocation	serviceType	predefinedLabel	postLabel	
light control 2	Agent 2	light control 2	light Controller	Bedroom/Parents	Agent 2	light control 2	light Controller	Bedroom/Parents	Agent 2	light control 2
light control 3	Agent 3	light control 3	light Controller	Dinningroom	Agent 3	light control 3	light Controller	Dinningroom	Agent 3	light control 3
light control 4	Agent 4	light control 4	light Controller	Bedroom/Children	Agent 4	light control 4	light Controller	Bedroom/Children	Agent 4	light control 4
light control 5	Agent 5	light control 5	light Controller	Garage	Agent 5	light control 5	light Controller	Garage	Agent 5	light control 5
light control 6	Agent 6	light control 6	light Controller	Bathroom	Agent 6	light control 6	light Controller	Bathroom	Agent 6	light control 6
light control 7	Agent 7	light control 7	light Controller	Bedroom/Children	Agent 7	light control 7	light Controller	Bedroom/Children	Agent 7	light control 7
light control 8	Agent 8	light control 8	light Controller	Bedroom/Parents	Agent 8	light control 8	light Controller	Bedroom/Parents	Agent 8	light control 8
light control 9	Agent 9	light control 9	light Controller	Dinningroom	Agent 9	light control 9	light Controller	Dinningroom	Agent 9	light control 9
light control 10	Agent 10	light control 10	light Controller	Dinningroom	Agent 10	light control 10	light Controller	Dinningroom	Agent 10	light control 10
light control 11	Agent 11	light control 11	light Controller	Garage	Agent 11	light control 11	light Controller	Garage	Agent 11	light control 11
light control 12	Agent 12	light control 12	light Controller	Bathroom	Agent 12	light control 12	light Controller	Bathroom	Agent 12	light control 12
light control 13	Agent 13	light control 13	light Controller	Bedroom/Children	Agent 13	light control 13	light Controller	Bedroom/Children	Agent 13	light control 13
light control 14	Agent 14	light control 14	light Controller	Bedroom/Children	Agent 14	light control 14	light Controller	Bedroom/Children	Agent 14	light control 14
light control 15	Agent 15	light control 15	light Controller	Garage	Agent 15	light control 15	light Controller	Garage	Agent 15	light control 15
light control 16	Agent 16	light control 16	light Controller	Bathroom	Agent 16	light control 16	light Controller	Bathroom	Agent 16	light control 16
light control 17	Agent 17	light control 17	light Controller	Bedroom/Children	Agent 17	light control 17	light Controller	Bedroom/Children	Agent 17	light control 17
light control 18	Agent 18	light control 18	light Controller	Bedroom/Children	Agent 18	light control 18	light Controller	Bedroom/Children	Agent 18	light control 18
light control 19	Agent 19	light control 19	light Controller	Bedroom/Parents	Agent 19	light control 19	light Controller	Bedroom/Parents	Agent 19	light control 19
light control 20	Agent 20	light control 20	light Controller	Dinningroom	Agent 20	light control 20	light Controller	Dinningroom	Agent 20	light control 20
light control 21	Agent 21	light control 21	light Controller	Dinningroom	Agent 21	light control 21	light Controller	Dinningroom	Agent 21	light control 21
light control 22	Agent 22	light control 22	light Controller	Garage	Agent 22	light control 22	light Controller	Garage	Agent 22	light control 22
light control 23	Agent 23	light control 23	light Controller	Bathroom	Agent 23	light control 23	light Controller	Bathroom	Agent 23	light control 23
light control 24	Agent 24	light control 24	light Controller	Bedroom/Children	Agent 24	light control 24	light Controller	Bedroom/Children	Agent 24	light control 24
light control 25	Agent 25	light control 25	light Controller	Bedroom/Children	Agent 25	light control 25	light Controller	Bedroom/Children	Agent 25	light control 25

Figure 6. shows the dataset column names in the first row, the dataset values in the following rows, and the class labels Normal and SPAM in the last column.

All machine learning algorithms are trained on the normal and spam labels in the last column of the screen that follows. The trained model then analyzes fresh test data to determine whether the request is normal or spam, dropping the packet and providing security if it is.



19902	Basic/charge	Basic/number	read	1	1520032284462	normal
19903	Basic/charge	Basic/number	write	2	1520032284462	normal
19904	Basic/charge	Basic/number	write	1	1520032284462	normal
19905	Basic/charge	Basic/number	read	1	1520032284462	normal
19906	Basic/charge	Basic/number	write	1	1520032284462	normal
19907	Basic/charge	Basic/number	write	1	1520032284462	normal
19908	Basic/charge	Basic/number	write	1	1520032284462	normal
19909	Basic/charge	Basic/number	write	1	1520032284462	normal
19910	Basic/charge	Basic/number	write	1	1520032284462	normal
19911	Basic/charge	Basic/number	write	1	1520032284462	normal
19912	Basic/charge	Basic/number	write	1	1520032284462	normal
19913	Basic/charge	Basic/number	write	1	1520032284462	normal
19914	Basic/charge	Basic/number	write	1	1520032284462	normal
19915	Basic/charge	Basic/number	write	1	1520032284462	normal
19916	Basic/charge	Basic/number	write	1	1520032284462	normal
19917	Basic/charge	Basic/number	write	1	1520032284462	normal
19918	Basic/charge	Basic/number	write	1	1520032284462	normal
19919	Basic/charge	Basic/number	write	1	1520032284462	normal
19920	Basic/charge	Basic/number	write	1	1520032284462	normal
19921	Basic/charge	Basic/number	write	1	1520032284462	normal
19922	Basic/charge	Basic/number	write	1	1520032284462	normal
19923	Basic/charge	Basic/number	write	1	1520032284462	normal
19924	Basic/charge	Basic/number	write	1	1520032284462	normal
19925	Basic/charge	Basic/number	write	1	1520032284462	normal
19926	Basic/charge	Basic/number	write	1	1520032284462	normal
19927	Basic/charge	Basic/number	write	1	1520032284462	normal
19928	Basic/charge	Basic/number	write	1	1520032284462	normal
19929	Basic/charge	Basic/number	write	1	1520032284462	normal
19930	Basic/charge	Basic/number	write	1	1520032284462	normal
19931	Basic/charge	Basic/number	write	1	1520032284462	normal
19932	Basic/charge	Basic/number	write	1	1520032284462	normal
19933	Basic/charge	Basic/number	write	1	1520032284462	normal
19934	Basic/charge	Basic/number	write	1	1520032284462	normal
19935	Basic/charge	Basic/number	write	1	1520032284462	normal
19936	Basic/charge	Basic/number	write	1	1520032284462	normal
19937	Basic/charge	Basic/number	write	1	1520032284462	normal
19938	Basic/charge	Basic/number	write	1	1520032284462	normal
19939	Basic/charge	Basic/number	write	1	1520032284462	normal
19940	Basic/charge	Basic/number	write	1	1520032284462	normal
19941	Basic/charge	Basic/number	write	1	1520032284462	normal
19942	Basic/charge	Basic/number	write	1	1520032284462	normal
19943	Basic/charge	Basic/number	write	1	1520032284462	normal
19944	Basic/charge	Basic/number	write	1	1520032284462	normal
19945	Basic/charge	Basic/number	write	1	1520032284462	normal
19946	Basic/charge	Basic/number	write	1	1520032284462	normal
19947	Basic/charge	Basic/number	write	1	1520032284462	normal
19948	Basic/charge	Basic/number	write	1	1520032284462	normal
19949	Basic/charge	Basic/number	write	1	1520032284462	normal
19950	Basic/charge	Basic/number	write	1	1520032284462	normal
19951	Basic/charge	Basic/number	write	1	1520032284462	normal
19952	Basic/charge	Basic/number	write	1	1520032284462	normal
19953	Basic/charge	Basic/number	write	1	1520032284462	normal
19954	Basic/charge	Basic/number	write	1	1520032284462	normal
19955	Basic/charge	Basic/number	write	1	1520032284462	normal
19956	Basic/charge	Basic/number	write	1	1520032284462	normal
19957	Basic/charge	Basic/number	write	1	1520032284462	normal
19958	Basic/charge	Basic/number	write	1	1520032284462	normal
19959	Basic/charge	Basic/number	write	1	1520032284462	normal
19960	Basic/charge	Basic/number	write	1	1520032284462	normal
19961	Basic/charge	Basic/number	write	1	1520032284462	normal
19962	Basic/charge	Basic/number	write	1	1520032284462	normal
19963	Basic/charge	Basic/number	write	1	1520032284462	normal
19964	Basic/charge	Basic/number	write	1	1520032284462	normal
19965	Basic/charge	Basic/number	write	1	1520032284462	normal
19966	Basic/charge	Basic/number	write	1	1520032284462	normal
19967	Basic/charge	Basic/number	write	1	1520032284462	normal
19968	Basic/charge	Basic/number	write	1	1520032284462	normal
19969	Basic/charge	Basic/number	write	1	1520032284462	normal
19970	Basic/charge	Basic/number	write	1	1520032284462	normal
19971	Basic/charge	Basic/number	write	1	1520032284462	normal
19972	Basic/charge	Basic/number	write	1	1520032284462	normal
19973	Basic/charge	Basic/number	write	1	1520032284462	normal
19974	Basic/charge	Basic/number	write	1	1520032284462	normal
19975	Basic/charge	Basic/number	write	1	1520032284462	normal
19976	Basic/charge	Basic/number	write	1	1520032284462	normal
19977	Basic/charge	Basic/number	write	1	1520032284462	normal
19978	Basic/charge	Basic/number	write	1	1520032284462	normal
19979	Basic/charge	Basic/number	write	1	1520032284462	normal
19980	Basic/charge	Basic/number	write	1	1520032284462	normal
19981	Basic/charge	Basic/number	write	1	1520032284462	normal
19982	Basic/charge	Basic/number	write	1	1520032284462	normal
19983	Basic/charge	Basic/number	write	1	1520032284462	normal
19984	Basic/charge	Basic/number	write	1	1520032284462	normal
19985	Basic/charge	Basic/number	write	1	1520032284462	normal
19986	Basic/charge	Basic/number	write	1	1520032284462	normal
19987	Basic/charge	Basic/number	write	1	1520032284462	normal
19988	Basic/charge	Basic/number	write	1	1520032284462	normal
19989	Basic/charge	Basic/number	write	1	1520032284462	normal
19990	Basic/charge	Basic/number	write	1	1520032284462	normal
19991	Basic/charge	Basic/number	write	1	1520032284462	normal
19992	Basic/charge	Basic/number	write	1	1520032284462	normal
19993	Basic/charge	Basic/number	write	1	1520032284462	normal
19994	Basic/charge	Basic/number	write	1	1520032284462	normal
19995	Basic/charge	Basic/number	write	1	1520032284462	normal
19996	Basic/charge	Basic/number	write	1	1520032284462	normal
19997	Basic/charge	Basic/number	write	1	1520032284462	normal
19998	Basic/charge	Basic/number	write	1	1520032284462	normal
19999	Basic/charge	Basic/number	write	1	1520032284462	normal
20000	Basic/charge	Basic/number	write	1	1520032284462	normal

Figure 7. shows the normal and spam labels in the last column of the screen above. All machine learning algorithms are trained using this data, and the trained model then analyzes fresh test data to determine whether the request is normal or SPAM. If it is spam, the packet is dropped and security is offered.

We have created the following modules in order to carry out this project:
 Utilizing this module, we will upload the smart home dataset to the application.
 Preprocess Dataset: this module will read the entire dataset and then clean it up by replacing any missing values with 0.
 Run Features Selection Algorithm: by applying the PCA features selection algorithm to the dataset, this module will choose only the most significant features and exclude the less significant ones, leaving the application with only the most crucial data for ML algorithm training.
 Divide the dataset into train and test, with the application using 80% of the dataset for training and 20% for testing. Run the Bagged Model Algorithm: this module will train the Bagged Model using 80% of the dataset, then apply the trained model to 20% of the dataset to predict a label. The accuracy and spam score will be determined by comparing the predicted label with the original data.
 Using this module, we will train the Bayesian Generalized Linear Model algorithm on 80% of the dataset. Then, we will apply the trained model on 20% of the dataset to predict a label, which will be compared to the original data to determine the accuracy and spam score.

Using this module, we will train a boosted linear model using 80% of the dataset. Then, we will apply the trained model to 20% of the dataset to predict a label, which will be compared to the original data to determine the accuracy and spam score.

Using this module, we will train the Extreme Gradient Boosting algorithm on 80% of the dataset. Then, we will apply the trained model on 20% of the dataset to predict a label, which will be compared to the original data to determine the accuracy and spam score.

Plotting the accuracy of each method will allow us to compare them using the All Algorithms Comparison Graph module.

VII. OUTPUT SCREENS

Double-clicking the "run.bat" file will launch the project and display the screen below. Click the "Upload Smart Home Dataset" button in the screen below to upload the dataset to the application and view the screen below.

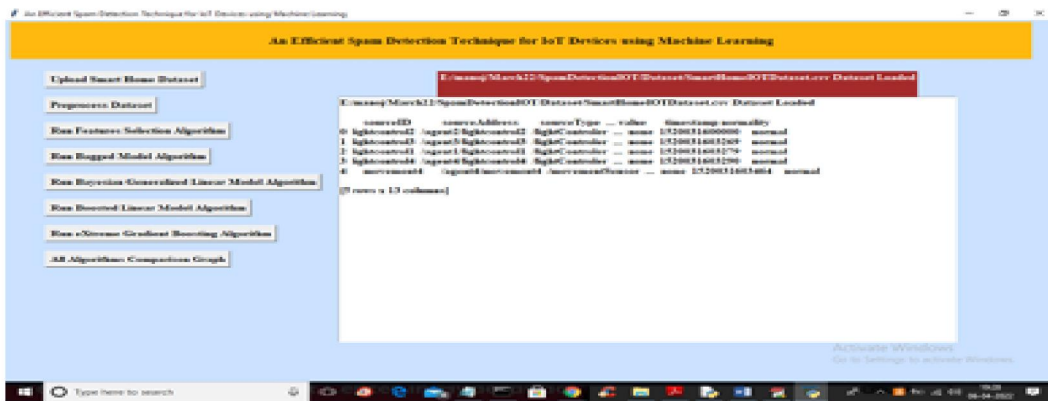


Figure 8: Choose and upload smart home data on the above screen, then click the "Open" button to load the dataset and obtain the output below.

After choosing and uploading smart home data in the screen below, click the "Open" button to load the dataset and see the output below.

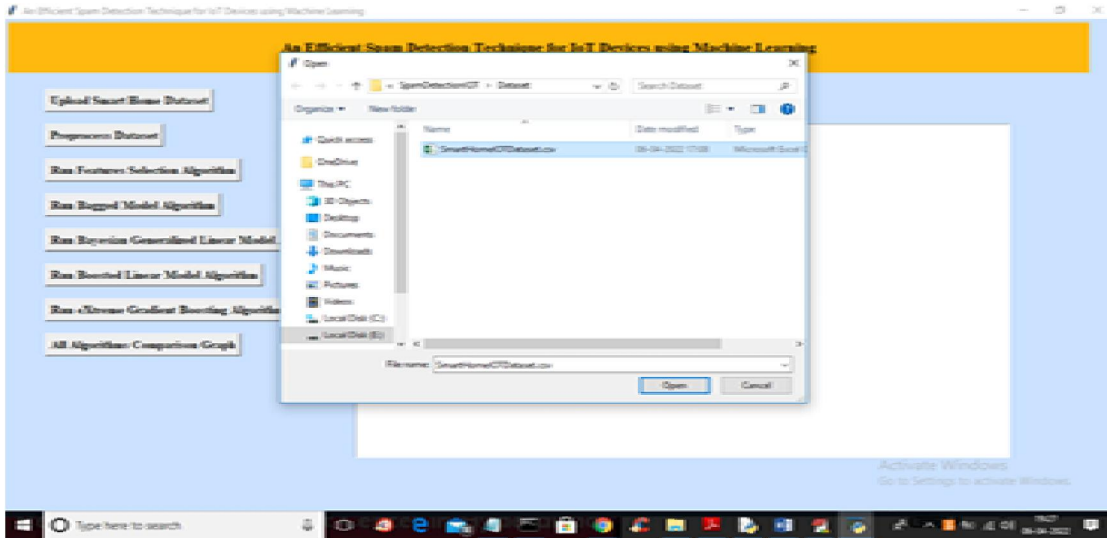


Figure 9: After choosing and uploading smart home data, click the "Open" button to load the dataset and obtain the output shown below.

The dataset loaded in the following screen has some non-numeric data, which ML would not accept. Therefore, we must preprocess the data to assign an integer id to transform the non-numeric data to numeric.

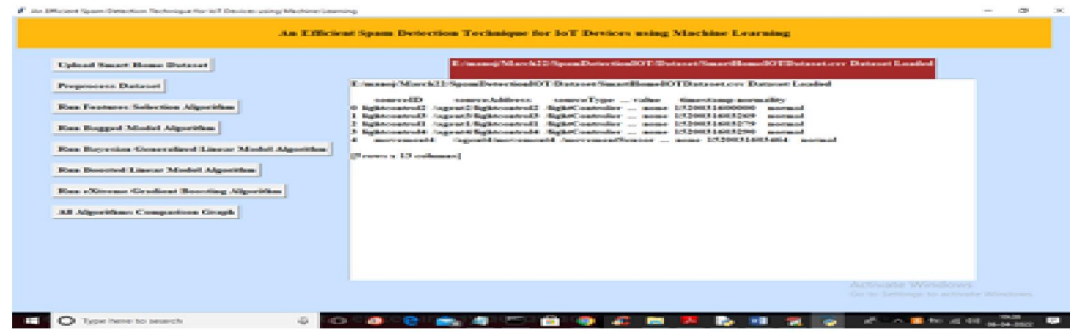


Figure 10. After loading the dataset, we can see that it includes some non-numeric data that ML would not accept. For this reason, we must preprocess the data by assigning an integer id to transform the non-numeric data to numeric. The full dataset has been converted to numeric format in the screen below, and it includes 11 columns and features in total. Click the "Run Features Selection Algorithm" button to apply PCA to the dataset in order to pick key features, and the output is shown below.

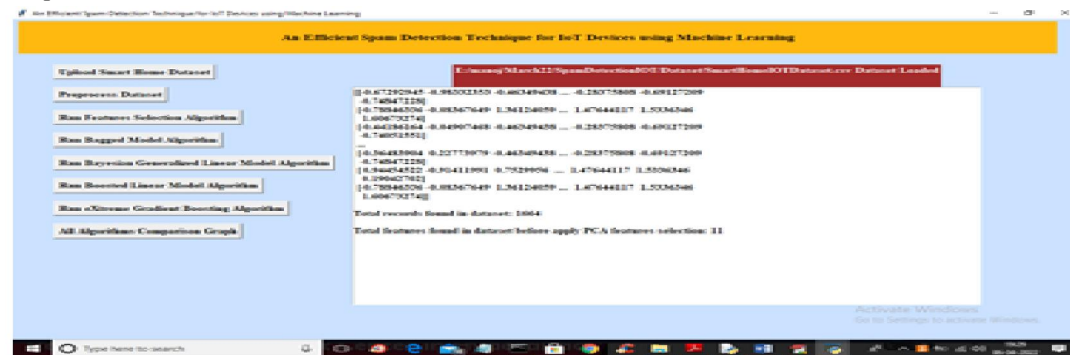


Figure 11 shows the whole dataset converted to numeric format. It has 11 columns and features in total. Click the "Run Features Selection Algorithm" button to apply PCA on the dataset in order to pick key features, and the output is shown below.

The following screen shows that, following PCA, the size of the dataset has been reduced to 10 key features. It comprises 1664 records, of which 80% (1331) are used for training and 20% (333) are used for testing. The train and test data are now ready, and clicking the "Run Bagged Model Algorithm" button will train the Bagged model using the data above and provide a score graph and accuracy values.

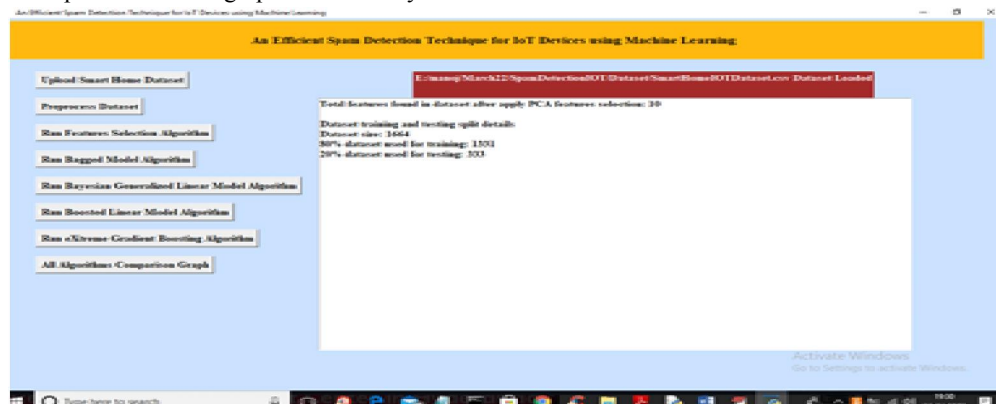


Figure 12. The dataset comprises 1664 records, with 80% (1331) records used for training and 20% (333) records for testing. As shown in the above screen, after applying PCA, the size of the features was reduced to 10 important features. The train and test data are now ready, and clicking the "Run Bagged Model Algorithm" button will train the Bagged model using the above data and provide a score graph and accuracy values. The graph's dashed line denotes the Normal class (0), while the attack class (1), and since the Bagged model can predict attacks from the dataset, both lines overlap. In the screen that follows, we obtained 97% accuracy with the Bagged Model. To train the algorithm and obtain the results below, close the graph above and click the "Run Bayesian Generalized Linear Model" button.

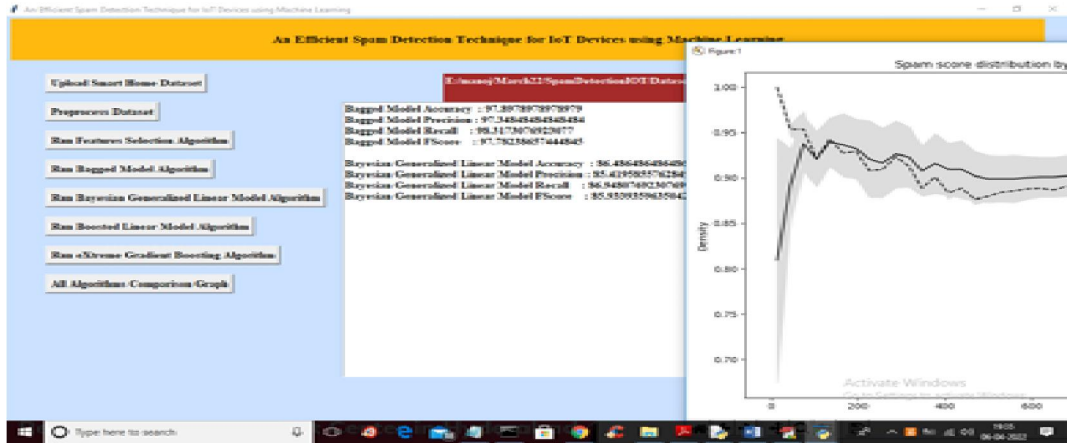


Figure 13. We obtained 97% accuracy with the Bagged Model in the screen above, and the dashed line in the graph denotes Normal classes 0 and 1.

represents the attack type, and both lines overlap since the bagged model can forecast attacks from the dataset. To train the algorithm and obtain the output below, close the graph above and click the "Run Bayesian Generalized Linear Model" button.

We obtained 97% accuracy and a score graph on the following screen with Boosted Linear. Close the graph above and click the "Run Extreme Gradient Boosting" button to train Extreme Boosting and obtain the output below

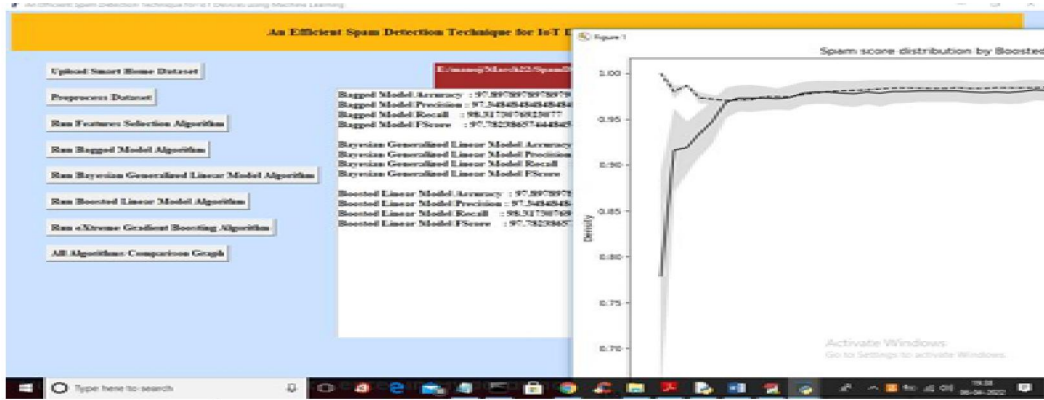


Figure 14. We obtained 86% accuracy and a score graph for the Bayesian algorithm in the screen above. Close the graph above and click the "Run Boosted Linear Model" button to train the boosted algorithm and obtain the output below.

We obtained 97% accuracy and a score graph that predicted both regular and spam in the following screen with Extreme Boosting. Close the graph above and click the "All Algorithms Comparison Graph" button to view the comparison graph below.

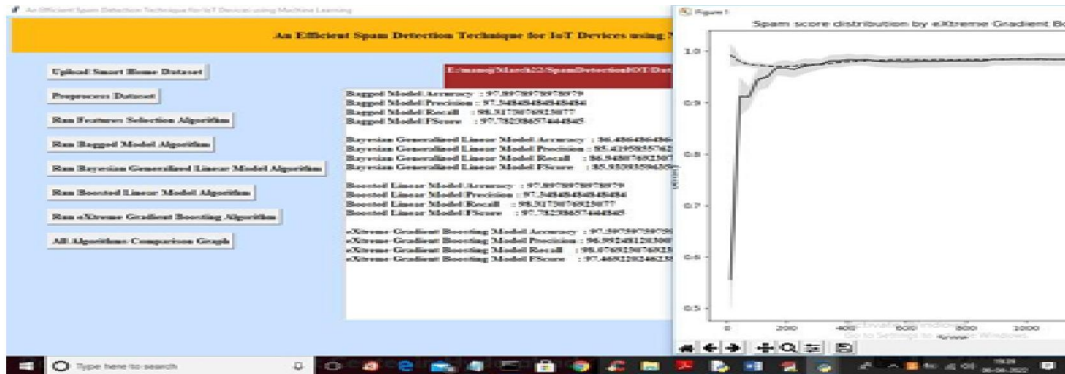


Figure 15. We obtained 97 percent accuracy and a score graph on the above screen with Boosted Linear. Click "Run Extreme Gradient Boosting" to train Extreme Boosting and obtain the output below.

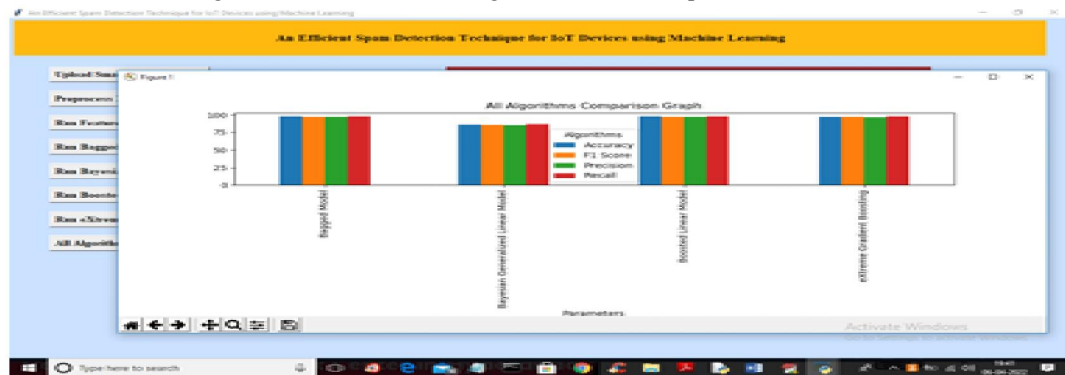


Figure 16. We obtained 97% accuracy and a score graph that predicted both regular and spam on the above screen with Extreme Boosting. Close the above graph and click the "All Algorithms Comparison Graph" button to view the comparison graph below.

The x-axis in the graph below represents the names of the algorithms, the y-axis accuracy and other metrics values, and each different color bar represents a different metric. As we can see in the screen above, three of the four algorithms provide more than 95% accuracy, meaning that we can secure the data of IoT devices by using machine learning algorithms

VIII. CONCLUSION

This study uses deep learning to propose an effective spam detection method for IOT devices. The Python spyder environment is used to conduct the simulation, and the findings indicate that the overall accuracy attained by the suggested work is 95.72%, compared to the prior 91.8 %. The suggested technique's error rate is 4.28 percent, whereas the previous work's is 8.2 percent. Thus, it is evident from the simulation results that the suggested work outperformed the current effort by a wide margin.

REFERENCES

[1] Asst. Prof. Sulthana A.S. Rilyas Abdi Mohamed Department of Computer Science, Rathinam College of Arts and Science, Coimbatore, Tamil Nadu, India -641021 An Efficient Spam Detection Technique for IoT Devices using Machine Learning International Journal of Scientific Research & Engineering Trends Volume 8, Issue 3, May-Jun-2022, ISSN (Online): 2395-566X

[2] Sarfaraj Alam1, Ms. Sonal Chaudhary2 M.Tech Scholar1, Associate Professor2 Department of Computer Science & Engineering All Saints' College of Technology, Bhopal, Madhya Pradesh, India An Efficient Deep Learning Technique for Detecting of Spam in IOT network © 2022 IJCR | Volume 10, Issue 8 August 2022 | ISSN: 2320-2882

- [3] Debasish Nath¹, M S Sowmya² ¹Student, Dept. of MCA, Bangalore Institute of Technology, Bengaluru, India
²Asst. Professor, Dept. of MCA, Bangalore Institute of Technology, Bengaluru, India
An Efficient Spam Detection Technique for IOT Devices using Machine Learning
ISSN (O) 2393-8021, ISSN (P) 2394-1588
- [4] Sidharth S^{*1}, Vasantha S^{*2} ^{*1}Dept. of MCA, Sir M Visvesvaraya Institute of Technology, Bengaluru, India.
^{*2}Dept. of MCA, Faculty of MCA, Sir M Visvesvaraya Institute of Technology, Bengaluru, India. AN EFFICIENT SPAM DETECTION TECHNIQUE FOR IOT DEVICES USING MACHINE LEARNING
e-ISSN: 2582-5208
- [5] D. Shine Rajesh, C. Sindhu, Ch. Nandini, Ch. Rajnandini ¹. Assistant Professor, Department of Information Technology, Malla Reddy Engineering College For Women (UGC-Autonomous), Hyderabad, India.
An Efficient Spam Detection Technique For IOT devices Using Machine Learning
International Journal of Engineering and Techniques - Volume 8 Issue 5, October 2022
- [6] Shi, Y., Wang, Z., Wang, X., & Zhang, S. (2015). Internet of things application to monitoring plant disease and insect pests. In 2015 International conference on Applied Science and Engineering Innovation. Atlantis Press, 31-34.
- [7] Lakshmi, K., & Gayathri, S. (2017). Implementation of IoT with image processing in plant growth monitoring system. Journal of Scientific and Innovative Research, 6(2), 80-83.
- [8] Sai, V. P., Ponnammal, T. A., & Rithvikikaran, (2017). Arduino Based Pest Control Using Real Time Environmental Monitoring Sensors. International Journal of Advanced Scientific Research and Management, 2(4), 6-9.
- [9] A. Makkar, S. Garg, N. Kumar, M. S. Hossain, A. Ghoneim and M. Alrashoud, "An Efficient Spam Detection Technique for IoT Devices Using Machine Learning," in IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 903-912, Feb. 2021, doi: 10.1109/TII.2020.2968927.
- [10] F. Hossain, M. N. Uddin and R. K. Halder, "Analysis of Optimized Machine Learning and Deep Learning Techniques for Spam Detection," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2021, pp. 1-7.
- [11] A. Makkar, U. Ghosh, P. K. Sharma and A. Javed, "A Fuzzy-based approach to Enhance Cyber Defence Security for Next-generation IoT," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2021.3053326.
- [12] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "IoT security: ongoing challenges and research opportunities," in 2014 IEEE 7th international conference on service-oriented computing and applications. IEEE, 2014, pp. 230-234.
- [13] C. Zhang and R. Green, "Communication security in internet of thing: preventive measure and avoid ddos attack over iot network," in Proceedings of the 18th Symposium on Communications & Networking. Society for Computer Simulation International, 2015, pp. 8-15.
- [14] W. Kim, O.-R. Jeong, C. Kim, and J. So, "The dark side of the internet: Attacks, costs and responses," Information systems, vol. 36, no. 3, pp. 675-705, 2011.
- [15] Nurul Fitriah Rusland, Norfaradilla Wahid, Shahreen Kasim, Hanayanti Hafit, "Analysis of Naive Bayes Algorithm for Email Spam Filtering across Multiple Datasets".