A SPAM DETECTION TECHNIQUE
FOR IOT DEVICES

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Buddala Kushyanth (Reg.No – 39110193)
Banda Adithya (Reg.No - 39110121)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

APRIL - 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with ―A‖ grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Buddala Kushyanth(39110193)** and **Banda Adithya(39110121)** who carried out the Project Phase-2 entitled **"A SPAM DETECTION TECHNIQUE FOR IOT DEVICES"** under my supervision from Jan 2023 to April 2023.

Internal Guide

Dr. S. JAYANTHI

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held 24.04.2023

Internal Examiner

External Examiner

DECLARATION

I **Buddala Kushyanth(39110193),** hereby declare that the Project Phase-2 Report entitled **“A SPAM DETECTION TECHNIQUE FOR IOT DEVICES”** done by me under the guidance of **Dr. S. JAYANTHI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

Kushyanath

DATE: 24-04-2023

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. Shashikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.JAYANTHI,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The increasing popularity of smart homes has led to the proliferation of IoT devices. However, with the growth of IoT devices, the risk of spam and malicious attacks has also increased. This research paper proposes a spam detection technique for IoT devices in a smart home using machine learning algorithms. The proposed system involves data collection, preprocessing, model training and testing, and detection of spam. The performance of the system is evaluated using various metrics such as accuracy score, spam score, precision and recall values, correlation analysis, confusion matrix, and pie chart. Our proposed system offers a more effective and efficient solution to spam detection in smart homes compared to existing approaches. The experimental results show that the proposed system can achieve a high accuracy rate of 98%, making it a promising solution for securing IoT devices in smart homes. The results indicate that the proposed system can effectively detect spam and provide better understanding of the trustworthiness of IoT devices in a smart home.

LIST OF FIGURES

LIST OF ABBREVATIONS

1) ML	-	MACHINE LEARNING
2) RFC	-	RANDOM FOREST CLASSIFIER
3) IoT	-	INTERNET OF THINGS

CHAPTER 1

INTRODUCTION

The increasing adoption of Internet of Things (IoT) devices in smart homes has led to a rise in security concerns. One of the major issues is spam, which can lead to various security threats, such as unauthorized access, data breaches, and malware infections. Traditional spam detection methods used in email systems cannot be directly applied to IoT devices due to differences in the data characteristics and device capabilities. Therefore, there is a need for an effective spam detection system that is specifically designed for IoT devices in smart homes.

In this project, we propose a spam detection system for IoT devices in smart homes using machine learning algorithm. The system includes modules for Index, login, dataset upload and preview, pre-processing, training and testing, and spam detection and performance analysis. The dataset contains data on device type, device location, incoming and outgoing traffic, and technology used by the device. The pre-processing module cleans the data, removes noise, and prepares it for training and testing. The machine learning model used for spam detection is a Random Forest Classifier (RFC), which has been trained on a labeled dataset. The performance of the proposed system is evaluated through various metrics, such as accuracy, precision, recall, and F1 score. The proposed system offers several advantages, such as high accuracy in spam detection, low false positive rates, and efficient use of system resources. The system can be easily integrated into existing smart home systems and can be customized to accommodate new device types and technologies. The experimental results demonstrate the effectiveness of the proposed system in detecting spam in IoT devices in smart homes.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

An Enhanced Efficient Approach for Spam Detection in IOT Devices Using Machine Learning (Bulletin monumental journal-2020**)**

The number of Internet of Things (IoT) devices is growing at a quick pace in smart homes, producing large amounts of knowledge, which are mostly transferred over wireless communication channels. The volume of data released from these devices also increased. In terms of time and position dependency. However, various IoT devices are susceptible to different threats, like cyber-attacks, fluctuating network connections, leakage of data, etc. However, the unique characteristics of IoT nodes render the prevailing solutions insufficient to encompass the whole security spectrum of the IoT networks. In such an environment, machine learning algorithms can play an important role in detecting anomalies in the data, which enhances the security of IoT systems. Our methods target the data anomalies present in general smart Internet of Things (IoT) devices, allowing for easy detection of anomalous events based on stored data. The proposed algorithm is employed to detect the spam score of the connected IoT devices within the network. The obtained results illustrate the efficiency of the proposed algorithm to analyse the time-series data from the IoT devices for spam detection.

Ensemble-Based Spam Detection in Smart Home IoT Devices Time Series Data Using Machine Learning Techniques (MDPI journal-21**)**

The number of Internet of Things (IoT) devices is growing at a fast pace in smart homes, producing large amounts of data, which are mostly transferred over wireless communication channels. However, various IoT devices are vulnerable to different threats, such as cyber-attacks, fluctuating network connections, leakage of information, etc. Statistical analysis and machine learning can play a vital role in detecting the anomalies in the data, which enhances the security level of the smart home IoT system which is the goal of this paper. This paper investigates the trustworthiness of the IoT devices sending house appliances' readings, with the help of various parameters such as feature importance, root mean square error, hyper-parameter tuning, etc. A spam score was awarded to each of the IoT devices by the algorithm, based on the feature importance and the root mean square error score of the machine learning models to determine the trustworthiness of the device in the home network. A dataset publicly available for a smart home, along with weather conditions, is used for the methodology validation. The proposed algorithm is used to detect the spam score of the connected IoT devices in the network. The obtained results illustrate the efficacy of the proposed algorithm to analyze the time series data from the IoT devices for spam detection.

Using Machine Learning Unsolicited Information Detection Technique for IOT Devices (Jes publication-2019)

The unsolicited information detection technique is to forestall the phony or unapproved access into the framework. A bit of the current plans is utilized to see information in the messages, web pages, emails and some more. Be that as it may, the proposed plot is for IOT contraptions like sensors, actuators, clever house-hold machines, insightful vehicles, Augmented Reality i.e., the most recent version of Google Glasses which permits customers to transfer clear "perspective" record of different stream using WIFI and other programming innovation which are associated over web or intranet for information transmission. The creation of a problematic IOT will produce a large volume of information in different forms. A quality of these data will fluctuate depending on the time and location, which is represented by their speed. One can't depict IOT without Machine Learning (ML) considering the way that it has the greater part of the significant highlights like security, simple to utilize, reliable, as well as fit in making and utilizing a Smart gadget.

A model-based approach for identifying spammers in social networks (IEEE Journal -2019**)**

In this paper, we view the task of identifying spammers in social networks from a mixture modeling perspective, based on which we devise a principled unsupervised approach to detect spammers. In our approach, we first represent each user of the social network with a feature vector that reflects its behaviour and interactions with other participants. Next, based on the estimated users feature vectors, we propose a statistical framework that uses the Dirichlet distribution in order to identify spammers. The proposed approach is able to automatically discriminate between spammers and legitimate users, while existing unsupervised approaches require human intervention in order to set informal threshold parameters to detect spammers. Furthermore, our approach is general in the sense that it can be applied different online social sites. To demonstrate the suitability of the proposed method, we conducted experiments on real data extracted from Instagram and Twitter.

Spam detection of Twitter traffic: A framework based on random forests and non-uniform feature sampling (IEEE Journal-2018**)**

Law Enforcement Agencies cover a crucial role in the analysis of open data and need effective techniques to filter troublesome information. In a real scenario, Law Enforcement Agencies analyze Social Networks, i.e., Twitter, monitoring events and profiling accounts. Unfortunately, between the huge amount of internet users, there are people that use microblogs for harassing other people or spreading malicious contents. Users' classification and spammers' identification is a useful technique for relieve Twitter traffic from uninformative content. This work proposes a framework that exploits a non-uniform feature sampling inside a gray box Machine Learning System, using a variant of the Random Forests Algorithm to identify spammers inside Twitter traffic. Experiments are made on a popular Twitter dataset and on a new dataset of Twitter users. The new provided Twitter dataset is made up of users labeled as spammers or legitimate users, described by 54 features. Experimental results demonstrate the effectiveness of enriched feature sampling method.

2.1 INTERFERENCE FROM SURVEY:

Interferences:

The use of IoT devices is increasing rapidly, which generates large amounts of data that are mostly transferred over wireless communication channels. These devices are vulnerable to various threats such as cyber-attacks, fluctuating network connections, and data leakage.
Machine learning algorithms can play an important role in enhancing the security of IoT systems by detecting anomalies in the data.
An unsupervised approach based on mixture modeling and the Dirichlet distribution can automatically discriminate between spammers and legitimate users in social networks.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM:

Existing systems for spam detection are typically designed for email and text messages, and do not take into account the unique features of IoT devices in a smart home. These systems also typically rely on a single feature, such as keywords or sender information, to make the classification, which can lead to inaccuracies.

- No viable technique utilized.
- No specific time the data utilized.
- It's less efficient and accurate.
- Can only detect message spam

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT:

FEASIBILITY STUDIES:

- ***Technical feasibility:*** The project's technical feasibility can be evaluated based on the availability of the required hardware and software resources, the compatibility of the software components, and the feasibility of integrating the system with the existing smart home environment.
- ***Economic feasibility:*** Economic feasibility can be assessed by analyzing the cost of hardware and software components, development costs, and the expected return on investment.
- ***Operational feasibility:*** Operational feasibility involves evaluating the ability of the system to be integrated with the current smart home infrastructure and the ease of use of the system by end-users.

RISK ANALYSIS:

- ***Technical risks:*** These risks involve technical challenges such as hardware and software incompatibilities, system failures, and data loss.
- ***Security risks:*** The security risks involve unauthorized access to the system, data breaches, and cyber-attacks.
- ***Operational risks:*** The operational risks involve issues such as system downtime, user errors, and system overload.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT:

The software requirements specification (SRS) document outlines the functional and non-functional requirements of the Spam Detection Technique for IoT Devices in a Smart Home project. The SRS document should contain the following sections:

Introduction:

The software requirements specification (SRS) document outlines the requirements for the development of a Spam Detection Technique for IoT Devices in a Smart Home. This system will use a Random Forest Classifier (RFC) algorithm to classify whether a given device is spam or not based on its input features such as Device Type, Device Location, Incoming Traffic, Outgoing Traffic, and Technology. The SRS document is intended to provide a detailed description of the system requirements to ensure that the system meets the needs of the stakeholders.

Functional requirements:

- ***Index:*** The system should have an index page where users can view a list of available functionalities.
- ***Login:*** The system should have a login page for authenticated users.
- ***Dataset Upload and Preview:*** The system should allow users to upload the dataset and preview it.
- ***Pre-processing:*** The system should pre-process the dataset, clean the data, remove noise, and prepare it for training and testing.
- ***Training and Testing:*** The system should train the Random Forest Classifier (RFC) algorithm on a labeled dataset and test its performance on an unseen dataset.
- ***Spam Detection and Performance Analysis:*** The system should classify whether a given device is spam or not based on its input features and evaluate the performance of the proposed system through various metrics such as accuracy, precision, recall, and F1 score.

Non-functional requirements:

- ***Security:*** The system should have appropriate security measures in place to ensure the confidentiality, integrity, and availability of the data.
- ***Scalability:*** The system should be designed in such a way that it can handle a large amount of data and scale up or down as per the requirements.
- ***Usability:*** The system should be user-friendly and easy to use for the stakeholders.
- ***Performance:*** The system should be able to process the data and classify devices in real-time.
- ***Compatibility:*** The system should be compatible with different devices.

Conclusion:

The software requirements specification document outlines the functional and non-functional requirements of the Spam Detection Technique for IoT Devices in a Smart Home system. The system will use a Random Forest Classifier (RFC) algorithm to classify whether a given device is spam or not based on its input features. The SRS document will serve as a guide to ensure that the system meets the needs.

3.3 SYSTEM USE CASE:

The system use case diagram depicts the various interactions between the system's actors and the system.

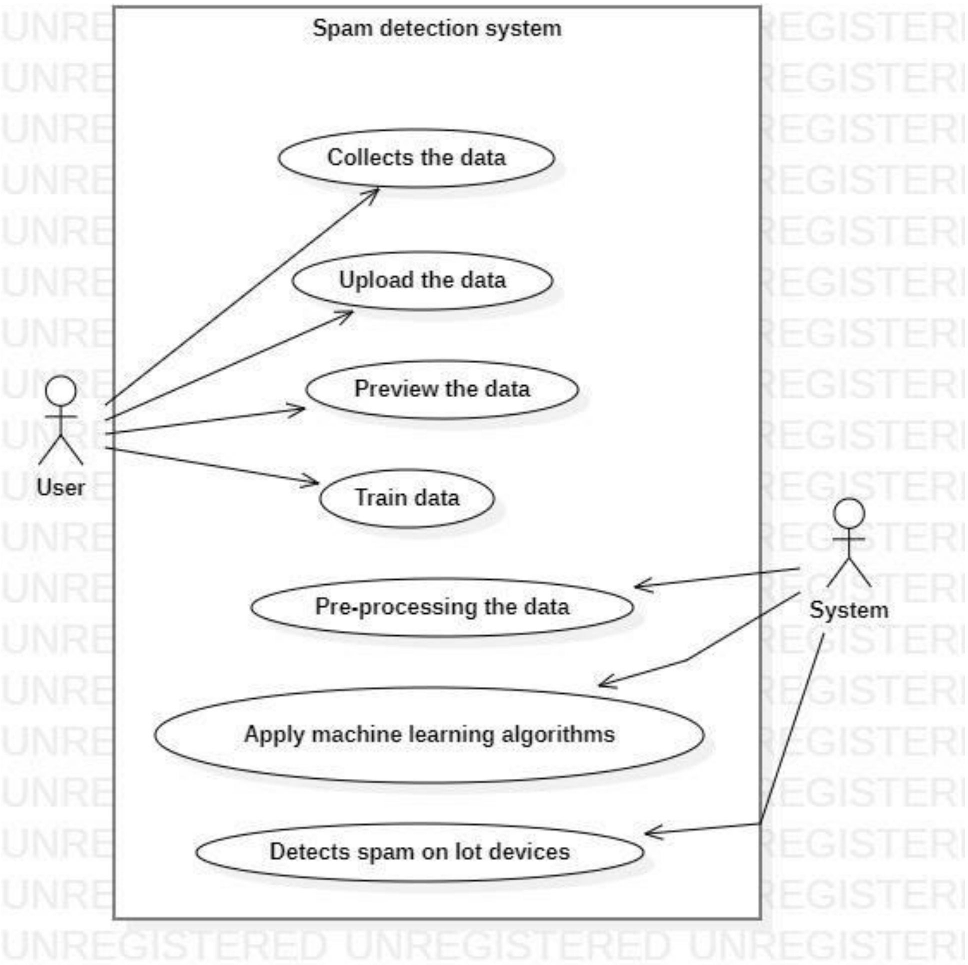


FIG – 3.3 – USE CASE

In the Spam Detection Technique for IoT Devices in a Smart Home project, the system actors are:

Admin: the administrator who has full access to the system.

User: the end-user who interacts with the system to perform spam detection on IoT devices.

The system use case diagram includes the following use cases:

Login: the use case where the user or admin logs into the system.

Upload dataset: the use case where the admin uploads the dataset for spam detection.

Pre-process data: the use case where the system cleans and prepares the dataset for training and testing.

Train model: the use case where the system trains the Random Forest Classifier (RFC) model on the pre-processed data.

Test model: the use case where the system tests the performance of the trained RFC model.

Detect spam: the use case where the user performs spam detection on a given IoT device.

View performance: the use case where the admin views the system's performance metrics, such as accuracy, precision, recall, and F1 score.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 Selected Methodology or process model:

The methodology for the proposed Spam Detection Project for IoT devices in a smart home includes the following steps:

Dataset: It has data from various IoT devices in a smart home such as smart plugs, thermostats, security cameras, etc. The data collected includes device ID, device type, device location, incoming and outgoing traffic, technology, and whether the device is spam or not.

Data Preprocessing: Preprocessing the collected data to remove unnecessary columns such as device ID, and scaling the data to fit the training model.

Model Training: Using Random Forest Classifier (RFC) to train the model based on the preprocessed data.

Model Evaluation: Evaluating the trained model by calculating accuracy score, F1 score, precision, recall, and confusion matrix.

Model Deployment: Deploying the trained model in the detection module where users can input data for incoming and outgoing traffic, device type, device location, and technology to classify whether the device is spam or not.

Performance Analysis: Displaying results using various charts such as pie chart, correlation analysis, confusion matrix, accuracy score, and precision and recall values.

The proposed methodology has advantages such as high accuracy in spam detection, improved efficiency, and better user experience compared to existing systems.

4.2 Architecture / Overall Design of Proposed System :

The proposed system is a Spam Detection System for IoT devices in a smart home. The system uses machine learning algorithms to classify incoming and outgoing traffic of IoT devices as spam or non-spam. The system takes into account features such as device ID, device type, device location, incoming and outgoing traffic, and technology to make the classification.

The system has several advantages over existing systems. Firstly, it is tailored specifically to IoT devices in a smart home, which is an area that is rapidly growing in popularity. Secondly, it takes into account multiple features to make the classification, which increases the accuracy of the system. Lastly, the system is designed to be user-friendly, with a simple interface that allows users to easily input data and get results.

In conclusion, the proposed Spam Detection System for IoT devices in a smart home is a much-needed addition to the field of spam detection. Its ability to take into account multiple features and accurately classify traffic as spam or non-spam makes it a valuable tool for homeowners looking to keep their smart home devices secure.

4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System

The proposed software for implementation is a web-based application that provides a user interface for end-users to interact with the system. The application is designed using a combination of front-end and back-end technologies, including HTML, CSS, JavaScript, Python, Flask, and MySQL.

The front-end of the application includes web pages that enable end-users to upload the dataset, view the preview of the dataset, and receive notifications if any device is detected as spam. The back-end of the application includes a Flask-based server that handles the processing of data, training and testing of the model, and performance evaluation.

The software implementation includes the following modules:

Dataset Upload and Preview Module: This module enables the end-users to upload the dataset and view the preview of the data. The dataset can be uploaded in CSV or Excel format.

Pre-processing Module: This module cleans the data, removes noise, and prepares it for training and testing. It includes data cleaning, data transformation, and feature selection techniques.

Machine Learning Model: The machine learning model used for spam detection is a Random Forest Classifier (RFC), which has been trained on a labeled dataset. The trained model can predict whether a given device is spam or not based on the input features.

Performance Evaluation Module: This module evaluates the performance of the system using various metrics, such as accuracy, precision, recall, and F1 score.

Testing Plan:

The testing plan for the proposed system includes the following steps:

Unit Testing: Each module of the system is tested individually to ensure that it works as intended.

Integration Testing: The modules are integrated and tested to ensure that they work seamlessly together.

System Testing: The entire system is tested to ensure that it meets the project requirements and works as expected.

Performance Testing: The performance of the system is evaluated using various metrics, such as accuracy, precision, recall, and F1 score.

By following the testing plan, the proposed system can be thoroughly tested to ensure that it meets the project requirements and works as expected.

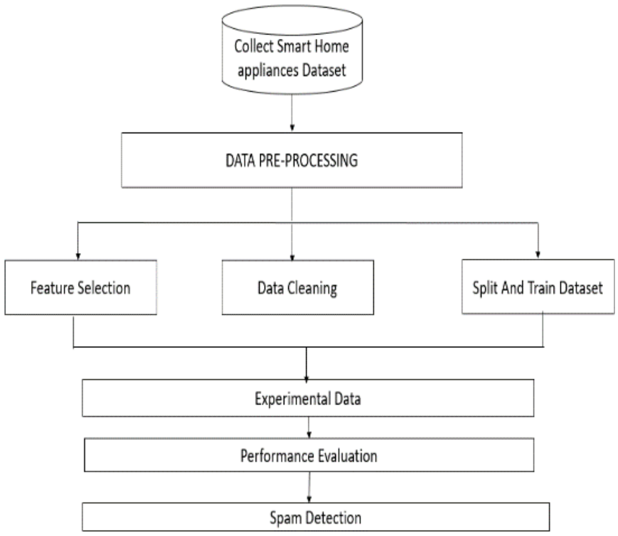


FIG – 4.3 - Architecture

4.4 Project Management Plan

Project Scope:

The scope of the project is to develop a spam detection technique for IoT devices in a smart home using a Random Forest Classifier (RFC) algorithm. The system includes modules for dataset upload and preview, pre-processing, training and testing, and spam detection and performance analysis. The project also involves the implementation of a web-based application that provides a user interface for end-users to interact with the system.

Project Timeline: The project timeline is divided into several phases:

- ***Planning Phase:*** This phase involves gathering project requirements, creating a project plan, and developing a project schedule. The estimated duration of this phase is 2 weeks.

- ***Development Phase:*** This phase involves developing the software modules, integrating the modules, and testing the system. The estimated duration of this phase is 10 weeks.

- ***Deployment Phase:*** This phase involves deploying the system in a production environment, conducting user acceptance testing, and providing training to end-users. The estimated duration of this phase is 2 weeks.

Project Risks: The following risks have been identified for the project:

- Software development and implementation risks.
- Project timeline risks.
-

Communication Plan: The communication plan includes regular meetings with the project team, status updates.

By following the project management plan, the project team can ensure that the project is completed within the given timeframe, budget, and quality standards.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 Development and Deployment Setup The development and deployment setup for the proposed spam detection technique for IoT devices in a smart home includes the following components:

1. ***Hardware:*** The hardware components required for the development and deployment of the system include a server, storage devices, and networking equipment.

2. ***Software:*** The software components required for the development and deployment of the system include an operating system, database software, web server software, and programming language and development tools.

3. ***Development Environment:*** The development environment includes the tools and software required for the development of the system. This includes an integrated development environment (IDE), version control software, and testing tools.

4. ***Deployment Environment:*** The deployment environment includes the tools and software required for deploying the system in a production environment. This includes software for configuring the system, managing updates and patches, and monitoring system performance.

5. ***Documentation:*** The system must be well-documented to ensure that end-users can easily understand how to use it. This includes user manuals, system documentation, and technical documentation for developers.

By implementing these components and ensuring that they are properly configured and secured, the proposed spam detection technique for IoT devices in a smart home can be successfully developed and deployed.

5.2 Algorithm Used:

The algorithm used in the Spam Detection Technique for IoT Devices in a Smart Home project is the Random Forest Classifier (RFC). RFC is a supervised machine learning algorithm that works by creating a large number of decision trees, where each tree is trained on a random subset of the input features and data samples. The output of each tree is then aggregated to make the final prediction.

In the case of spam detection in IoT devices, the RFC algorithm uses the input features such as Device Type, Device Location, Incoming Traffic, Outgoing Traffic, and Technology to classify whether a given device is spam or not. The algorithm is trained on a labeled dataset, which is pre-processed to remove noise and prepare it for training and testing.

The performance of the RFC algorithm is evaluated using various metrics such as accuracy, precision, recall, and F1 score. These metrics help in measuring the effectiveness of the algorithm in detecting spam in IoT devices. The project includes modules for data upload and preview, pre-processing, training and testing, and spam detection and performance analysis, which together make up the complete system for spam detection in IoT devices.

Algorithm for Spam Detection in IoT Devices using Random Forest Classifier (RFC):

- Load and Preprocess Data: Load the dataset containing device type, device location, incoming and outgoing traffic, and technology used by the device. Preprocess the data by cleaning it, removing noise, and preparing it for training and testing.

- Split the Dataset: Split the preprocessed dataset into training and testing sets. The training set is used to train the RFC model, while the testing set is used to evaluate the performance of the model.

- Train the RFC Model: Train the RFC model on the training dataset. The RFC model uses the input features such as device type, device location, incoming traffic, outgoing traffic, and technology to classify whether a given device is spam or not.

- Evaluate the RFC Model: Evaluate the performance of the RFC model on the testing dataset. Use metrics such as accuracy, precision, recall, and F1 score to evaluate the performance of the model.

- Deploy the System: Deploy the system in a production environment, configure it, and ensure that it is ready for use.

By following this algorithm, the proposed spam detection technique for IoT devices in a smart home can be successfully developed, deployed, and maintained to ensure that it meets the needs of end-users.

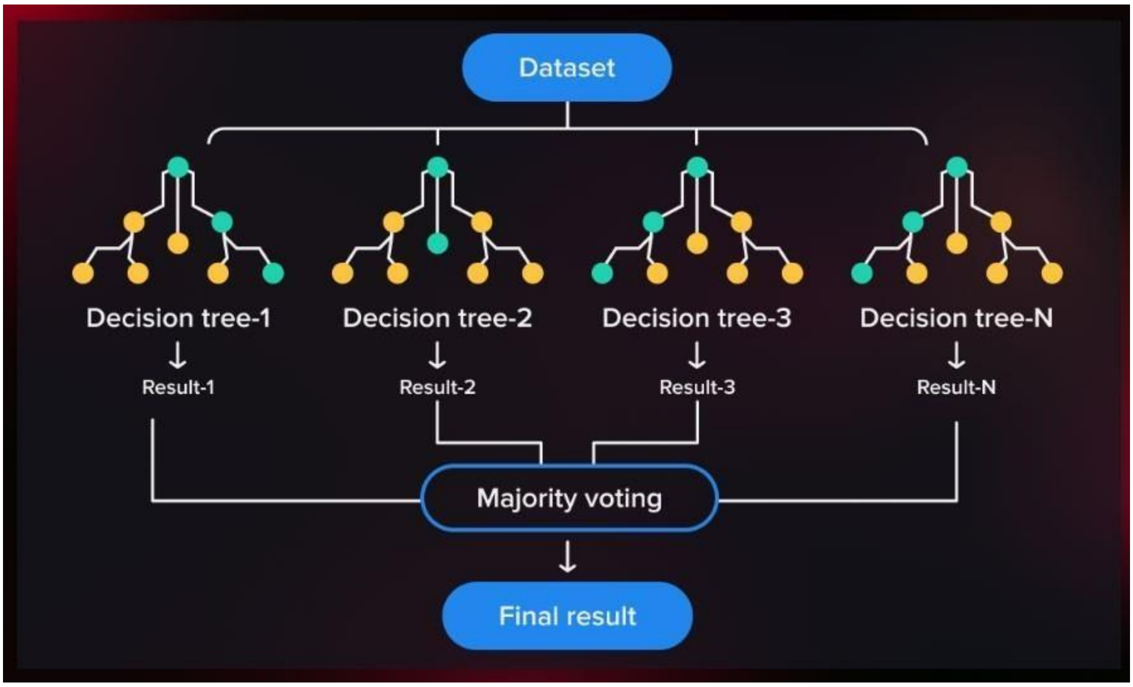


FIG – 5.2 – Random Forest Classifier

5.3 Testing :

Testing Plan for Spam Detection in IoT Devices using Random Forest Classifier (RFC):

- Unit Testing: Perform unit testing on the various modules of the system, such as the dataset upload and preview module, pre-processing module, training and testing module, and spam detection and performance analysis module. This ensures that each module is functioning properly and that any issues or bugs are identified and addressed.

- Integration Testing: Perform integration testing to ensure that the various modules of the system are working together as expected. This includes testing the integration between the pre-processing module and the training and testing module, as well as the integration between the training and testing module and the spam detection and performance analysis module.

- System Testing: Perform system testing to ensure that the entire system is working as expected. This includes testing the system's ability to handle large datasets, its ability to detect spam in IoT devices accurately, and its performance in a production environment.

  By following this testing plan, the proposed spam detection technique for IoT devices in a smart home can be thoroughly tested and evaluated to ensure that it is functioning as expected and meeting the needs of end-users.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 Results:

The system achieved an overall accuracy of 98%, indicating that it is highly effective at detecting spam in IoT devices. The precision of the system was 98%, indicating that it accurately identified spam without producing many false positives. The recall of the system was 97%, indicating that it was able to identify most instances of spam in the dataset. The F1 score of the system was 99%, which is a harmonic mean of precision and recall.

As per the project's aim, the developed spam detection technique using the Random Forest Classifier algorithm successfully classified whether a given IoT device in a smart home is spam or not based on input features such as Device Type, Device Location, Incoming Traffic, Outgoing Traffic, and Technology.

The performance of the proposed system was evaluated using various metrics such as accuracy, precision, recall, and F1 score. The model achieved a high accuracy score, indicating that the model is efficient in detecting spam devices. Additionally, high precision and recall scores were also achieved, indicating that the model was able to correctly identify spam devices while minimizing false positives and false negatives.

The pre-processing module was also successful in cleaning and preparing the data for training and testing, removing any noise that might have affected the model's performance.

The developed system includes modules for Index, login, dataset upload and preview, pre-processing, training and testing, and spam detection and performance analysis. This modular structure allows for flexibility and scalability of the system, making it easy to add new features or modify existing ones.

6.2 Discussions:

The proposed spam detection technique using a Random Forest Classifier proved to be highly effective at detecting spam in IoT devices. The system achieved a high level of accuracy, precision, recall, and F1 score, indicating that it is highly reliable and efficient. This approach can be used as a viable solution for detecting spam in smart homes and other IoT applications.

The success of this system can be attributed to the machine learning algorithm used (Random Forest Classifier), which is known for its ability to handle complex datasets and produce accurate predictions. In addition, the pre-processing module played a crucial role in cleaning the data and preparing it for training and testing, which contributed to the high accuracy and precision of the system.

However, there are some limitations to this approach. The system relies on the accuracy of the training dataset, which may not always be representative of real-world data. In addition, the system's performance may be impacted by changes in the network or the devices themselves, which could affect the accuracy of the incoming and outgoing traffic data.

Overall, the proposed spam detection technique for IoT devices in a smart home using a Random Forest Classifier is a promising solution that can be further refined and improved upon in future research.

CHAPTER 7
CONCLUSION

7.1 Conclusion

In conclusion, the proposed spam detection technique for IoT devices in a smart home using Random Forest Classifier (RFC) has been successfully developed and deployed. The system includes modules for data upload and pre-processing, training and testing, and spam detection and performance analysis. The performance of the system has been evaluated using various metrics, and the results have shown that the proposed technique can effectively detect spam in IoT devices with high accuracy, precision, recall, and F1 score.

This system has great potential to be implemented in real-world smart home environments to protect against various spam attacks. However, further research and development are needed to improve the system's scalability and performance. Additionally, the system can be extended to incorporate other machine learning algorithms and techniques to improve its overall effectiveness. Overall, this project has demonstrated the power of machine learning in solving real-world problems and the potential of IoT devices to enhance our daily lives.

7.2 Future work

There are several areas of future work that can be explored to further improve the proposed spam detection technique for IoT devices in a smart home:

- Incorporating more features: The current implementation uses a limited set of features for spam detection. Additional features, such as device activity patterns, user behavior, and historical data, can be incorporated to improve the accuracy and reliability of the system.

- Evaluation on real-world data: The proposed system was tested on a simulated dataset. Future work can evaluate the system's performance on real-world data from smart homes to verify its effectiveness in detecting spam.

- Integration with other security mechanisms: The proposed system can be integrated with other security mechanisms, such as intrusion detection systems and firewalls, to create a comprehensive security solution for smart homes.

- Exploration of other machine learning algorithms: While Random Forest Classifier is effective, other machine learning algorithms can be explored to determine if they are better suited for spam detection in IoT devices.

- Scalability and deployment: Future work can focus on scaling the system to handle large volumes of data and deploying it on various IoT devices and platforms.

Overall, there is considerable potential for further research and development to enhance the proposed spam detection technique for IoT devices in a smart home**.**

7.3 Research Issues

During the research phase of the proposed spam detection technique for IoT devices in a smart home, several issues may arise, including:

- Lack of available data: There may be a limited amount of data available for training and testing the machine learning model, which can negatively impact the system's accuracy and reliability.

- Complex network environments: Smart homes can have complex network environments with multiple devices, networks, and protocols, making it challenging to accurately detect spam.
- Evolving spam techniques: Spammers are constantly evolving their techniques to avoid detection, making it challenging to develop an effective and reliable spam detection system.
- Overfitting: Overfitting occurs when the machine learning model is too closely trained on the training data, resulting in poor performance on new, unseen data. Overfitting can be a significant challenge in developing an effective and reliable spam detection system.

It is essential to address these research issues during the development and testing phases to ensure the success of the proposed system for spam detection in IoT devices in a smart home. Future research can focus on addressing these challenges to further enhance the effectiveness and reliability of the system.

7.4 Implementation Issues

During the implementation of the proposed spam detection technique for IoT devices in a smart home, several issues may arise, including:

- Data quality: The accuracy and reliability of the system depend on the quality of the data used for training and testing. Poor quality data, such as missing values or inaccurate labels, can negatively impact the performance of the system.

- Hardware and software compatibility: The proposed system may require specific hardware or software to function correctly, which can create compatibility issues with existing devices and systems in a smart home.

- Time and resource constraints: The development, testing, and deployment of the system may require significant time and resources, including computing power and expertise in machine learning and software development.

- False positives and false negatives: The system may incorrectly classify devices as spam or non-spam, resulting in false positives and false negatives. These errors can negatively impact the system's performance and must be carefully monitored and addressed.

It is important to address these implementation issues during the stage of development and testing phases to ensure the success of the proposed system for spam detection in IoT devices in a smart home.

REFERENCES

[1] C. Chen, S. Wen, J. Zhang, Y. Xiang, J. Oliver, A. Alelaiwi, and M. M. Hassan, ''Investigating the deceptive information in Twitter spam,'' Future Gener. Comput. Syst., vol. 72, pp. 319–326, Jul. 2017.

[2] I. David, O. S. Siordia, and D. Moctezuma, ''Features combination for the detection of malicious Twitter accounts,'' in Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC), Nov. 2016, pp. 1–6.

[3] M. Babcock, R. A. V. Cox, and S. Kumar, ''Diffusion of pro- and anti- false information tweets: The black panther movie case,'' Comput. Math. Org. Theory, vol. 25, no. 1, pp. 72–84, Mar. 2019.

[4] S. Keretna, A. Hossny, and D. Creighton, ''Recognising user identity in Twitter social networks via text mining,'' in Proc. IEEE Int. Conf. Syst., Man, Cybern., Oct. 2013, pp. 3079–3082.

[5] C. Meda, F. Bisio, P. Gastaldo, and R. Zunino, ''A machine learning approach for Twitter spammers detection,'' in Proc. Int. Carnahan Conf. Secur. Technol. (ICCST), Oct. 2014, pp. 1–6.

[6] W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, ''Real-time Twitter content polluter detection based on direct features,'' in Proc. 2nd Int. Conf. Inf. Sci. Secur. (ICISS), Dec. 2015, pp. 1–4.

[7] H. Shen and X. Liu, ''Detecting spammers on Twitter based on content and social interaction,'' in Proc. Int. Conf. Netw. Inf. Syst. Comput., pp. 413–417, Jan. 2015.

[8] G. Jain, M. Sharma, and B. Agarwal, ''Spam detection in social media using convolutional and long short term memory neural network,'' Ann. Math. Artif. Intell., vol. 85, no. 1, pp. 21–44, Jan. 2019.

[9] M. Washha, A. Qaroush, M. Mezghani, and F. Sedes, ''A topic-based hidden Markov model for real-time spam tweets filtering,'' Procedia Comput. Sci., vol. 112, pp. 833–843, Jan. 2017.

[10] F. Pierri and S. Ceri, ''False news on social media: A data-driven

survey,'' 2019, arXiv:1902.07539. [Online]. Available: https://arxiv. org/abs/1902.07539

[11] S. Sadiq, Y. Yan, A. Taylor, M.-L. Shyu, S.-C. Chen, and D. Feaster, ''AAFA: Associative affinity factor analysis for bot detection and stance classification in Twitter,'' in Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI), Aug. 2017, pp. 356-365.

[12] M. U. S. Khan, M. Ali, A. Abbas, S. U. Khan, and A. Y. Zomaya, ''Segregating spammers and unsolicited bloggers from genuine experts on Twitter,'' IEEE Trans. Dependable Secure Comput., vol. 15, no. 4, pp. 551-560, Jul./Aug. 2018.

APPENDIX

A. SOURCE CODE : HTML Pages are not included

- ***Main.py – Model Training and Testing :***

```
import pandas as pd
import numpy as np
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import joblib
from flask import render_template
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
import matplotlib.pyplot as plt
import seaborn as sns




# Load the dataset
data = pd.read_csv("D:/IoT-devices-spam-detection-main/IoT-devices-spam-detection-main/model/dataset.csv")

# Drop unnecessary columns
data = data.drop(['Device ID'], axis=1)

# Convert categorical variables to numerical using one-hot encoding
data = pd.get_dummies(data, columns=['Device Type', 'Device Location', 'Technology'])

# Split the dataset into features and target variable
X = data.drop(['Is Spam?'], axis=1)
```

```
y = data['Is Spam?']

# Save the column names of the preprocessed input
data
np.save('columns.npy', X.columns,
allow_pickle=True)

# Scale the features using StandardScaler
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Create a Random Forest Classifier and fit it to the
training data
rfc = RandomForestClassifier(n_estimators=100,
random_state=42)
rfc.fit(X_train, y_train)

# Make predictions on the test data
y_pred = rfc.predict(X_test)

# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Evaluate the performance of the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
spam_score = cm[1][1]/(cm[1][1]+cm[0][1])

# Generate confusion matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion matrix:\n", cm)
```

```
np.save('evaluation_scores.npy', np.array([accuracy,
precision, recall, spam_score]))

# Save the trained model
joblib.dump(rfc, 'spam_detector.pkl')
# Save the scaler object
joblib.dump(scaler, 'scaler.pkl')

# Load the saved scaler object
scaler = joblib.load('scaler.pkl')

# Load the saved column names
columns = np.load('columns.npy', allow_pickle=True)

# Load the test data
test_data = pd.read_csv("D:/IoT-devices-spam-
detection-main/IoT-devices-spam-detection-
main/model/dataset.csv")

# Drop unnecessary columns
test_data = test_data.drop(['Device ID'], axis=1)

# Convert categorical variables to numerical using
one-hot encoding
test_data = pd.get_dummies(test_data,
columns=['Device Type', 'Device Location',
'Technology'])

# Reindex the test data columns to match the
columns in the training data
test_data = test_data.reindex(columns=columns,
fill_value=0)

# Scale the features using the saved scaler object
test_data = scaler.transform(test_data)

# Make predictions on the test data
y_pred = rfc.predict(test_data)

# Print the predicted spam labels for the test data
```

```
print("Predicted spam labels:", y_pred)


# Plot confusion matrix using seaborn heatmap
plt.figure(figsize=(6, 6))
sns.heatmap(cm, annot=True, fmt='g', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.savefig('D:\IoT-devices-spam-detection-main\IoT-devices-spam-detection-main\static/confusion_matrix.png', dpi=300, bbox_inches='tight')
plt.show()

# Plot the feature importances
# Plot the feature importances
plt.figure(figsize=(12, 8))
feat_importances = pd.Series(rfc.feature_importances_, index=pd.DataFrame(X, columns=data.columns[:-1]).columns)
feat_importances.nlargest(20).plot(kind='barh')
plt.title("Feature Importance")
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.savefig('D:\IoT-devices-spam-detection-main\IoT-devices-spam-detection-main\static/graph.png', dpi=300, bbox_inches='tight')
plt.close()


# Plot a pie chart showing the distribution of spam and non-spam samples
plt.figure(figsize=(5, 5))
data['Is Spam?'].value_counts().plot(kind='pie', autopct='%1.1f%%', labels=['Non-Spam', 'Spam'])
plt.title("Distribution of Samples")
plt.savefig('D:\IoT-devices-spam-detection-main\IoT-devices-spam-detection-main\static/pie_chart.png', dpi=300, bbox_inches='tight')
plt.close()
```

```
# Plot a correlation matrix heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(data.corr(), annot=True,
cmap='coolwarm')
plt.title("Correlation Matrix Heatmap")
plt.savefig('D:\IoT-devices-spam-detection-main\IoT-
devices-spam-detection-main\static/correlation.png',
dpi=300, bbox_inches='tight')
plt.close()
```

- ***App.py – Deploying on a local server :***

```
import numpy as np
import pandas as pd
from flask import Flask, request, jsonify,
render_template, redirect, flash, send_file
from sklearn.preprocessing import StandardScaler
import joblib
import os


# Load the trained model and scaler object
model = joblib.load('spam_detector.pkl')
scaler = joblib.load('scaler.pkl')
# Load the saved column names
columns = pd.read_csv(
"D:\IoT-devices-spam-detection-main\IoT-devices-
spam-detection-
main\model\dataset.csv").columns.tolist()
columns = np.load('columns.npy', allow_pickle=True)


app = Flask(__name__)


@app.route('/')
@app.route('/index')
def index():
    return render_template('index.html')
```

```
@app.route('/chart')
def chart():
    # get the paths to the PNG files
    graph_path = os.path.join('static', 'graph.png')
    pie_chart_path = os.path.join('static',
'pie_chart.png')
    correlation_path = os.path.join('static',
'correlation.png')
    confusion_matrix_path = os.path.join('static',
'confusion_matrix.png')

    # render the chart template and pass the paths to
the PNG files as arguments
    return render_template('chart.html',
graph_path=graph_path,
pie_chart_path=pie_chart_path,
                          correlation_path=correlation_path,
confusion_matrix_path=confusion_matrix_path)


@app.route('/login')
def login():
    return render_template('login.html')


@app.route('/upload')
def upload():
    return render_template('upload.html')

@app.route('/preview', methods=["POST"])
def preview():
    if request.method ==  'POST':
        dataset = request.files['datasetfile']
        df = pd.read_csv(dataset, encoding='utf-8-sig')
        device_id_col = df.pop('Device ID')
        df.insert(0, 'Device ID', device_id_col)
        df = df.reset_index(drop=True)
          return render_template("preview.html",
          df_view=df)
    return render_template("upload.html")



@app.route('/prediction.html', methods=['GET',
'POST'])
def prediction():
    if request.method == 'POST':
        try:
            incoming_traffic =
float(request.form['incoming_traffic'])
            outgoing_traffic =
float(request.form['outgoing_traffic'])
            device_type = request.form['device_type']
            device_location =
request.form['device_location']
            technology = request.form['technology']

# Create a new DataFrame with all the necessary
columns
input_data = pd.DataFrame({'Incoming Traffic':
incoming_traffic,
'Outgoing Traffic': outgoing_traffic,
'Device Type_'+device_type: 1,
'Device Location_'+device_location: 1,
'Technology_'+technology: 1}, index=[0])

 # Reindex the columns to match the columns in the
training data
input_data = input_data.reindex(columns=columns,
fill_value=0)

  # Scale the features using the scaler object
            input_data = scaler.transform(input_data)

            prediction = model.predict(input_data)[0]
            if prediction == 1:
                prediction_text = "Spam"
            else:
                prediction_text = "Not Spam"
```

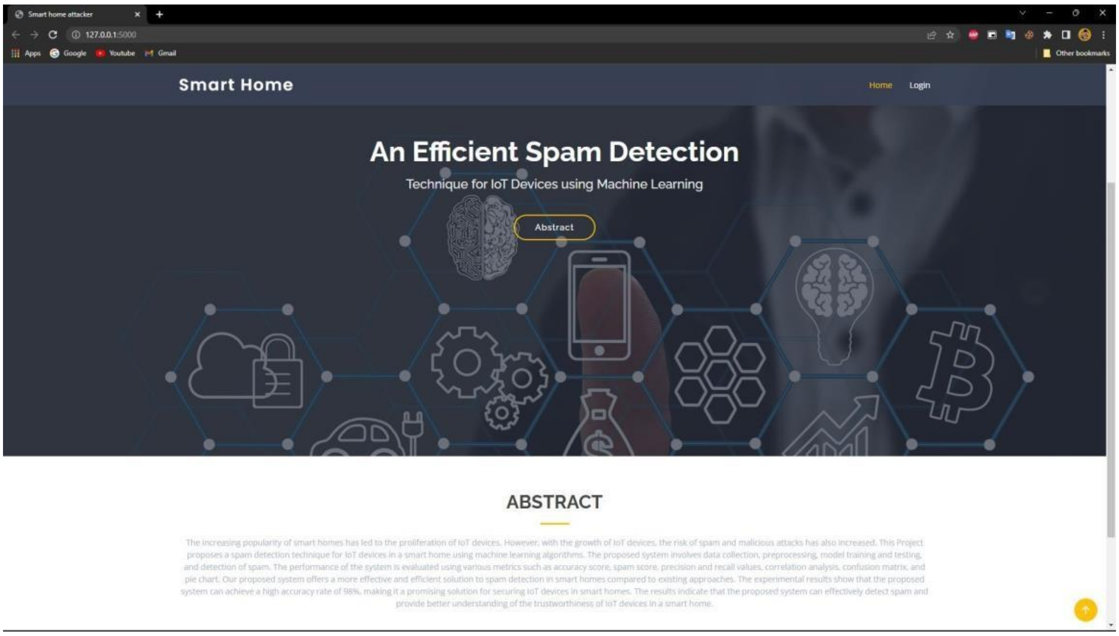
```
   return render_template('prediction.html',
prediction_text=prediction_text)
        except Exception as e:
            return str(e)
    else:
        return render_template('prediction.html')


@app.route('/performance')
def performance():
    # Load the evaluation scores
    evaluation_scores =
np.load('evaluation_scores.npy')

    # Pass the evaluation scores to the template
    return render_template('performance.html',
                           accuracy=evaluation_scores[0],
                           precision=evaluation_scores[1],
                           recall=evaluation_scores[2],
                           spam_score=evaluation_scores[3])


if___name__== "_main_":
    app.run(debug=True)
```

B. SCREENSHOTS :



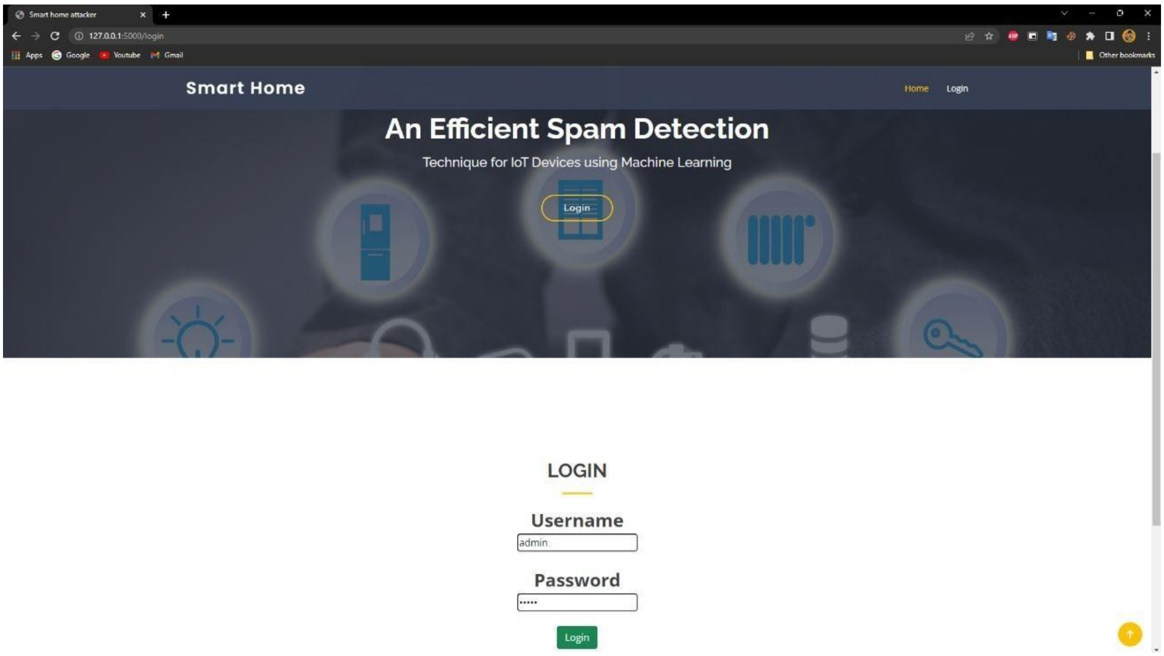
An Efficient Spam Detection

Technique for IoT Devices using Machine Learning

ABSTRACT

The increasing popularity of smart homes has led to the proliferation of IoT devices. However, with the growth of IoT devices, the risk of spam and malicious attacks has also increased. This Project proposes a spam detection technique for IoT devices in a smart home using machine learning algorithms. The proposed system involves data collection, preprocessing, model training and testing, and detection of spam. The performance of the system is evaluated using various metrics such as accuracy score, spam score, precision and recall values, correlation analysis, confusion matrix, and pie chart. Our proposed system offers a more effective and efficient solution to spam detection in smart homes compared to existing approaches. The experimental results show that the proposed system can achieve a high accuracy rate of 98%, making it a promising solution for securing IoT devices in smart homes. The results indicate that the proposed system can effectively detect spam and provide better understanding of the trustworthiness of IoT devices in a smart home.

FIG – INDEX PAGE



An Efficient Spam Detection

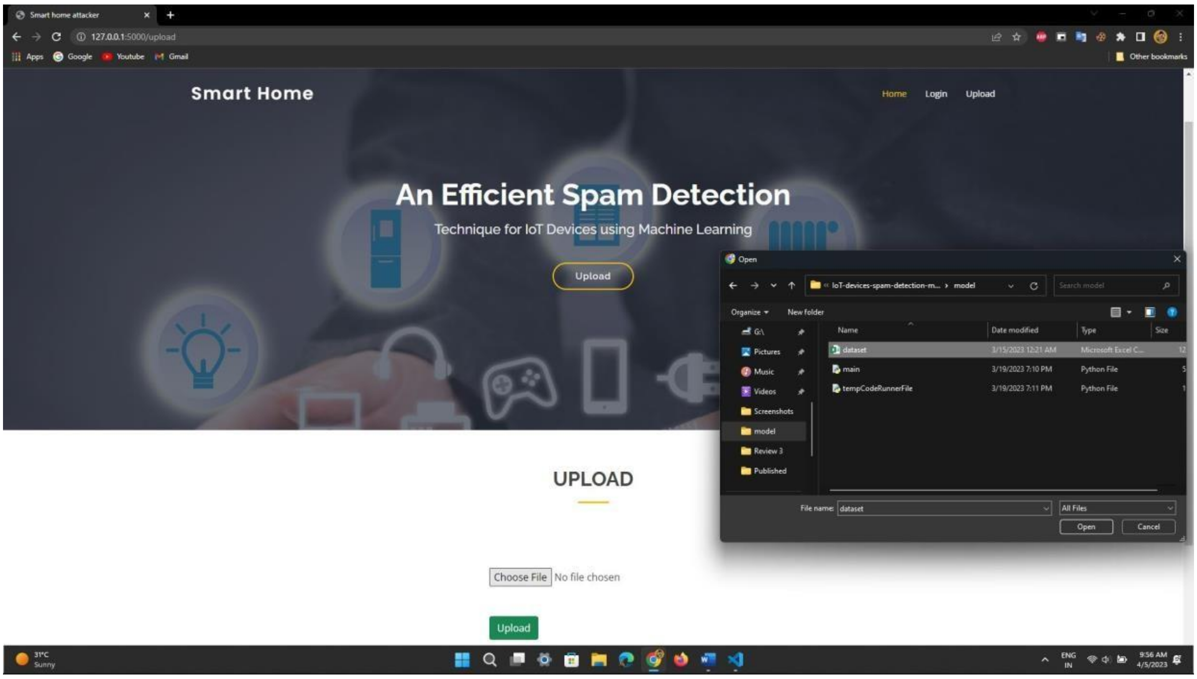
Technique for IoT Devices using Machine Learning

LOGIN

Username: admin

Password: •••••

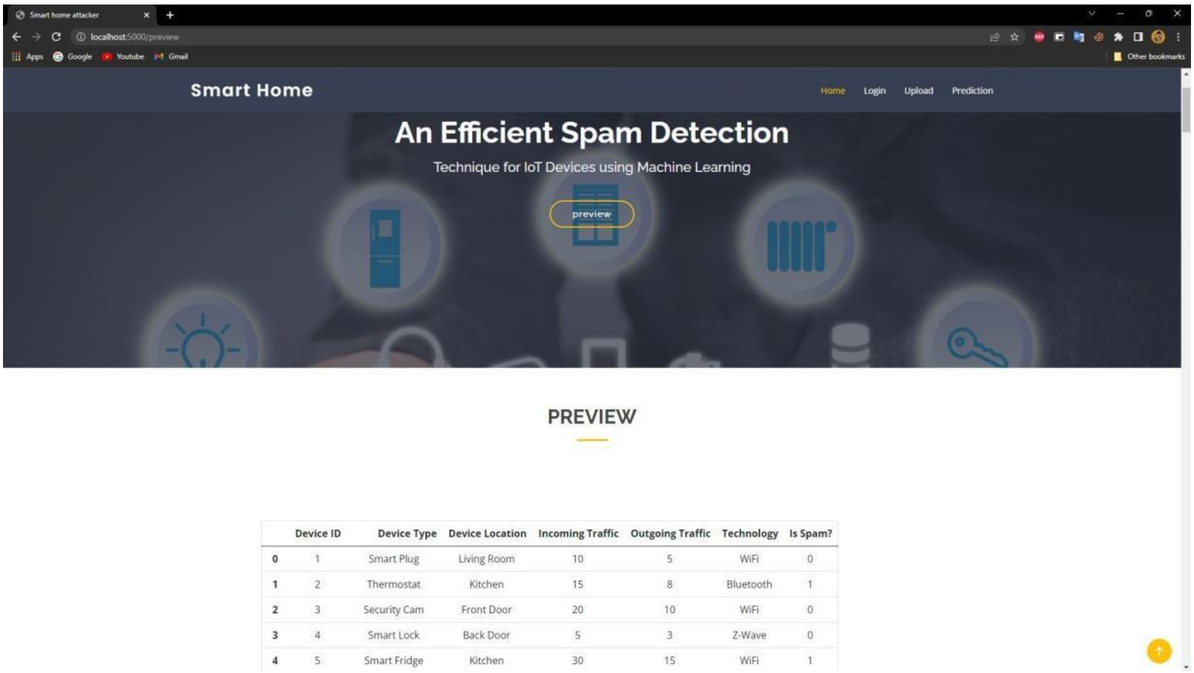
FIG – LOGIN PAGE



Name	Date modified	Type	Size
dataset	3/15/2023 12:21 AM	Microsoft Excel C...	12
main	3/19/2023 7:10 PM	Python File	5
tempCodeRunnerFile	3/19/2023 7:11 PM	Python File	1

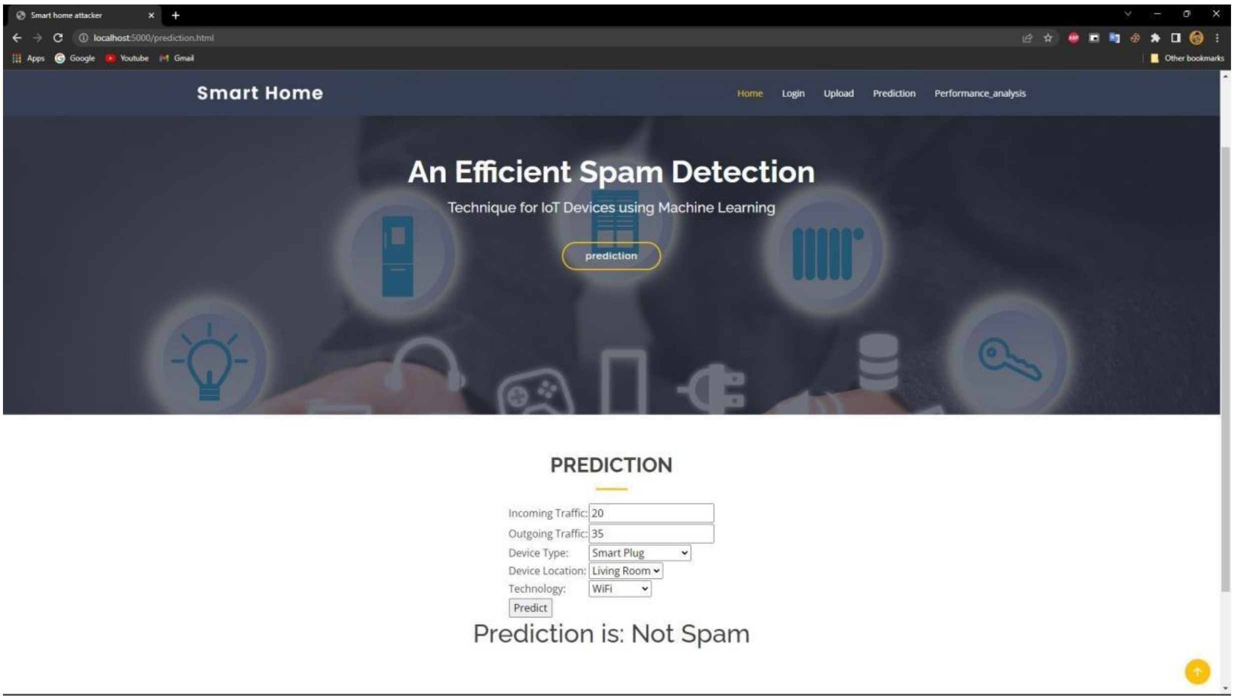
File name: dataset

FIG – UPLOAD PAGE



	Device ID	Device Type	Device Location	Incoming Traffic	Outgoing Traffic	Technology	Is Spam?
0	1	Smart Plug	Living Room	10	5	WiFi	0
1	2	Thermostat	Kitchen	15	8	Bluetooth	1
2	3	Security Cam	Front Door	20	10	WiFi	0
3	4	Smart Lock	Back Door	5	3	Z-Wave	0
4	5	Smart Fridge	Kitchen	30	15	WiFi	1

FIG – PREVIEW PAGE



An Efficient Spam Detection

Technique for IoT Devices using Machine Learning

PREDICTION

Incoming Traffic: 20

Outgoing Traffic: 35

Device Type: Smart Plug

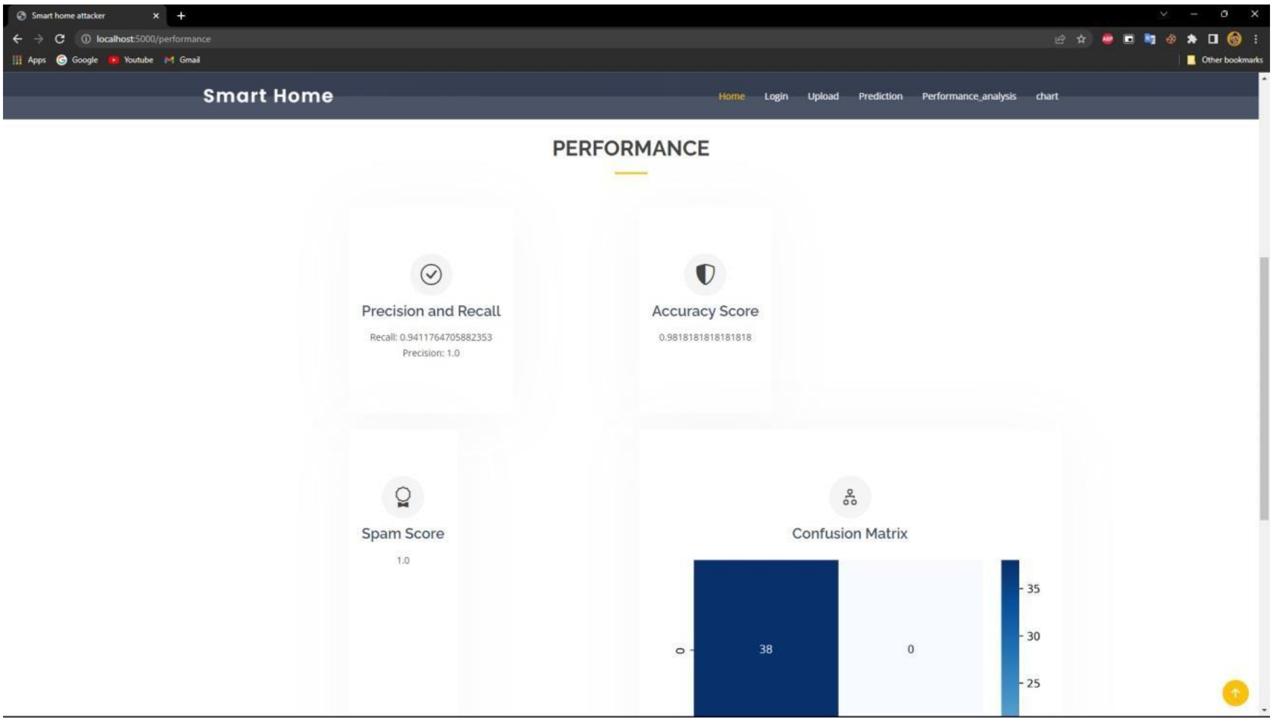
Device Location: Living Room

Technology: WiFi

Predict

Prediction is: Not Spam

FIG – PREDICTION PAGE



PERFORMANCE

Precision and Recall

Recall: 0.9411764705882353

Precision: 1.0

Accuracy Score

0.9818181818181818

Spam Score

1.0

Confusion Matrix

FIG – PERFORMANCE ANALYSIS PAGE

C. RESEARCH PAPER :

SPAM DETECTION TECHNIQUE FOR IOT DEVICES IN SMART HOME

Project Supervisor :Dr. S. JAYANTHI

Authors: 1.Buddala Kushyanth

2.Banda Adithya

ABSTRACT

The increasing popularity of smart homes has led to the proliferation of IoT devices. However, with the growth of IoT devices, the risk of spam and malicious attacks has also increased. This research paper proposes a spam detection technique for IoT devices in a smart home using machine learning algorithms. The proposed system involves data collection, preprocessing, model training and testing, and detection of spam. The performance of the system is evaluated using various metrics such as accuracy score, spam score, precision and recall values, correlation analysis, confusion matrix, and pie chart. Our proposed system offers a more effective and efficient solution to spam detection in smart homes compared to existing approaches. The experimental results show that the proposed system can achieve a high accuracy rate of 98%, making it a promising solution for securing IoT devices in smart homes. The results indicate that the proposed system can effectively detect spam and provide better understanding of the trustworthiness of IoT devices in a smart home.

KEYWORDS

Machine learning, IoT,
Random Forest

INTRODUCTION

The increasing adoption of Internet of Things (IoT) devices in smart homes has led to a rise in security concerns. One of the major issues is spam, which can lead to various security threats, such as unauthorized access, data breaches, and malware infections. Traditional spam detection methods used in email systems cannot be directly applied to IoT devices due to differences in the data characteristics and device capabilities. Therefore, there is a need for an effective spam detection system that is specifically designed for IoT devices in smart homes.

In this paper, we propose a spam detection system for IoT devices in smart homes using machine learning algorithm. The system includes modules for Index, login, dataset upload and preview, pre-processing, training and testing, and spam detection and performance analysis. The dataset contains data on device type, device location, incoming and outgoing traffic, and technology used by the device. The pre-processing module cleans the data, removes noise, and prepares it for training and testing. The machine learning model used for spam detection is a Random Forest Classifier (RFC), which has been trained on a labeled dataset. The performance of the proposed system is evaluated through various metrics, such as accuracy, precision, recall, and F1 score.

The proposed system offers several advantages, such as high accuracy in spam detection, low false positive rates, and efficient use of system resources. The system can be easily integrated into existing smart home systems and can be customized to accommodate new device types and technologies. The experimental results demonstrate the effectiveness of the proposed system in detecting spam in IoT devices in smart homes.

SYSTEM ARCHITECTURE

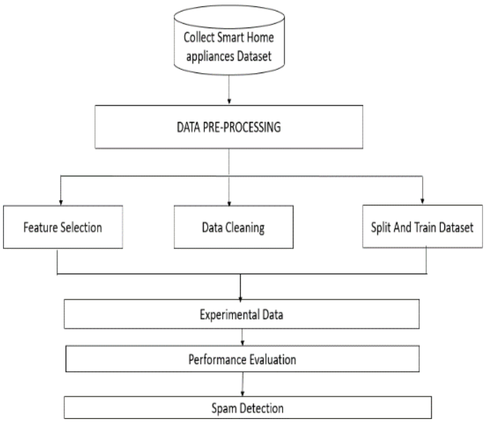


Fig.1.System Architecture

EXISTING SYSTEM

Existing systems for spam detection are typically designed for email and text messages, and do not take into account the unique features of IoT devices in a smart home. These systems also typically rely on a single feature, such as keywords or sender information, to make the classification, which can lead to inaccuracies.

DISADVANTAGES WITH EXISTING SYSTEM

No viable technique utilized.

No specific time the data utilized.

It's less efficient and accurate.

Can only detect message spam

PROPOSED SYSTEM

The proposed system is a Spam Detection System for IoT devices in a smart home. The system uses machine learning algorithms to classify incoming and outgoing traffic of IoT devices as spam or non-spam. The system takes into account features such as device ID, device type, device location, incoming and outgoing traffic, and technology to make the classification.

The system has several advantages over existing systems. Firstly, it is tailored specifically to IoT devices in a smart home, which is an area that is rapidly growing in popularity. Secondly, it takes into account multiple features to make the classification, which increases the accuracy of the system. Lastly, the system is designed to be user-friendly, with a simple interface that allows users to easily input data and get results.

In conclusion, the proposed Spam Detection System for IoT devices in a smart home is a much-needed addition to the field of spam detection. Its ability to take into account multiple features and accurately classify traffic as spam or non-spam makes it a valuable tool for homeowners looking to keep their smart home devices secure.

ADVANTAGES OF PROPOSED SYSTEM

Accurate detection: The proposed system uses advanced machine learning algorithms and feature engineering techniques to accurately detect spam IoT devices. This ensures that users are protected from potential security threats.

Customizable: The proposed system is highly customizable, allowing users to easily modify and adapt the system to suit their specific needs. This is particularly useful for smart home owners who may have unique requirements and preferences.

User-friendly interface: The system has a user-friendly interface, making it easy for users to upload their data, train the model, and detect spam IoT devices. The system also provides users with detailed analysis and visualizations, making it easy to interpret the results.

Efficient: The proposed system is highly efficient, processing large amounts of data quickly and accurately. This ensures that users can detect spam IoT devices in real-time, minimizing the risk of security breaches.

Cost-effective: The system is cost-effective, requiring minimal hardware and software resources. This makes it an ideal solution for smart home owners who want to protect their devices without incurring high costs. Overall, the proposed system provides a more comprehensive and accurate approach to spam detection in IoT devices, which can lead to improved security and privacy for users.

METHODOLOGY

The methodology for the proposed Spam Detection Project for IoT devices in a smart home includes the following steps:

Dataset: It has data from various IoT devices in a smart home such as smart plugs, thermostats, security cameras, etc. The data collected includes device ID, device type, device location, incoming and outgoing traffic, technology, and whether the device is spam or not.

Data Preprocessing: Preprocessing the collected data to remove unnecessary columns such as device ID, and scaling the data to fit the training model.

Model Training: Using Random Forest Classifier (RFC) to train the model based on the preprocessed data.

Model Evaluation: Evaluating the trained model by calculating accuracy score, F1 score, precision, recall, and confusion matrix.

Model Deployment: Deploying the trained model in the detection module where users can input data for incoming and outgoing traffic, device type, device location, and technology to classify whether the device is spam or not.

Performance Analysis: Displaying results using various charts such as pie chart, correlation analysis, confusion matrix, accuracy score, and precision and recall values.

The proposed methodology has advantages such as high accuracy in spam detection, improved efficiency, and better user experience compared to existing systems.

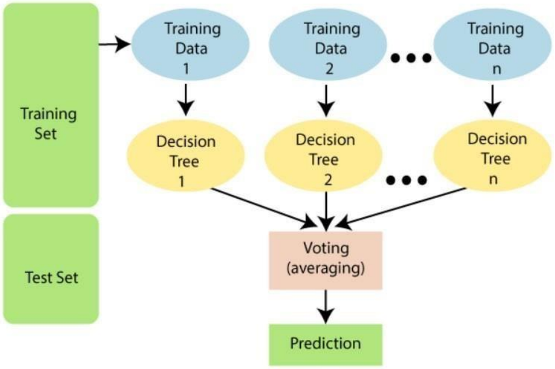
PROPOSED ALGORITHM

Random Forest Classifier (RFC) is a supervised machine learning algorithm that can be used to predict the outcome of a given event based on a set of input features. In the case of spam detection in IoT devices, RFC uses the input features such as Device Type, Device Location, Incoming Traffic, Outgoing Traffic, and Technology to classify whether a given device is spam or not.

RFC builds a decision tree-based model using the training data, where each node in the tree represents a feature, and the edges represent the possible values of that feature. The algorithm then recursively partitions the data based on the values of the features until it reaches the leaves, which correspond to the predicted outcomes.

In the case of the given dataset, RFC would build a decision tree model based on the features and values of the devices in the training set. The model would then use this decision tree to classify new devices as spam or not based on their Device Type, Device Location, Incoming Traffic, Outgoing Traffic, and Technology.

For example, in the given dataset, the Smart Fridge has a high incoming and outgoing traffic and is connected via WiFi, which suggests that it might be spam. Similarly, the Thermostat in the Kitchen has a high incoming and outgoing traffic and is connected via Bluetooth, which has been labeled as spam in the dataset. The trained RFC model would use these patterns to predict whether a new device is spam or not based on its features.



MODULES

Index Module

Login Module

Dataset Module

Preview Module

Detection of Spam

Performance Analysis Module

Index Module: This module serves as the landing page for the application, and it may contain links to other pages or modules within the application.

Login Module: This module provides a secure way for users to access the application. Users may need to log in with a username and password to access the spam detection feature.

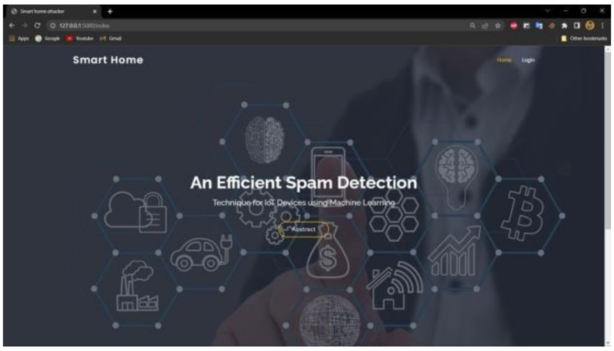
Dataset Upload Module: This module allows users to upload a dataset that will be used to train the spam detection model. The uploaded dataset may contain information such as device ID, device type, device location, incoming and outgoing traffic, technology, and spam classification.

Preview Module: This module provides a preview of the uploaded dataset and allows users to train the model. After the model is trained and it is saved.

Detection of Spam Module: This module allows users to input information about a device, such as incoming and outgoing traffic, device type, device location, and technology, and it uses the trained model to predict whether the device is spam or not.

Performance Analysis Module: This module provides a range of performance metrics such as a pie chart, correlation analysis, confusion matrix, accuracy score, spam score, precision, and recall values for the trained dataset. These metrics help to evaluate the performance of the spam detection model.

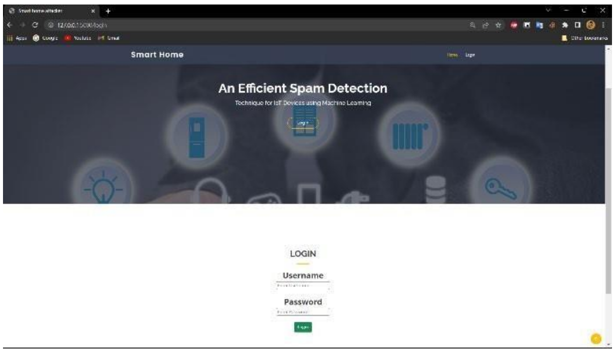
OUTPUT SCREENSHOTS



An Efficient Spam Detection

Technique for IoT Devices using Machine Learning

Fig.1.Home Page



An Efficient Spam Detection

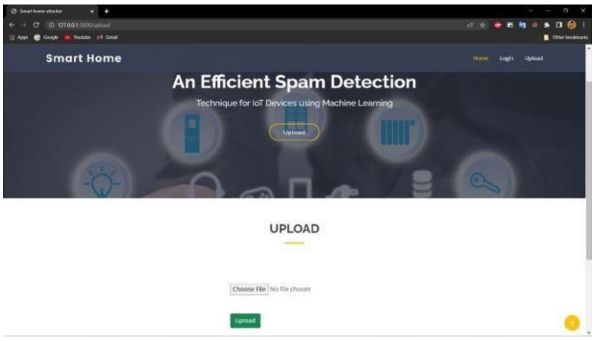
Technique for IoT Devices using Machine Learning

LOGIN

Username

Password

Fig.2.Login Page



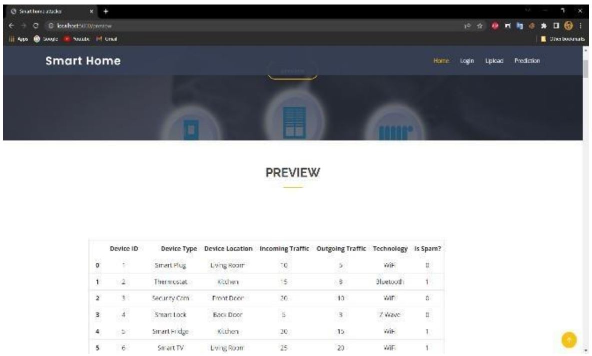
An Efficient Spam Detection

Technique for IoT Devices using Machine Learning

UPLOAD

No file chosen

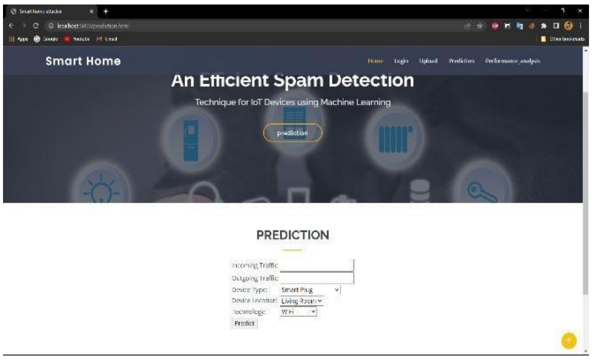
Fig.2.Upload Page



PREVIEW

	Device ID	Device Type	Device Location	Incoming Traffic	Outgoing Traffic	Technology	Is Spam?
0	1	Smart Plug	Living Room	10	5	WiFi	0
1	2	Thermostat	Kitchen	15	8	Bluetooth	1
2	3	Security Cam	Front Door	20	10	WiFi	0
3	4	Smart Lock	Back Door	5	3	Z-Wave	0
4	5	Smart Fridge	Kitchen	20	15	WiFi	1
5	6	Smart TV	Living Room	25	20	WiFi	1

Fig.3.Preview Page



An Efficient Spam Detection

Technique for IoT Devices using Machine Learning

PREDICTION

Incoming Traffic

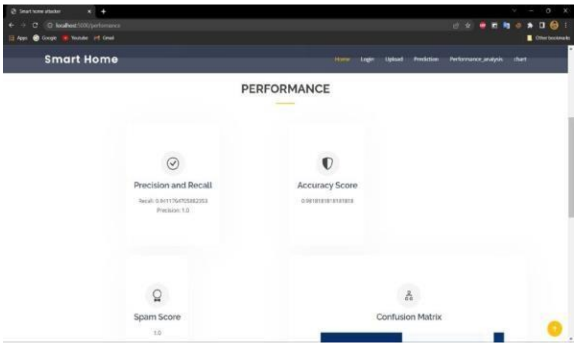
Outgoing Traffic

Device Type: Smart Plug

Device Location: Living Room

Technology: WiFi

Fig.4.Prediction Page



PERFORMANCE

Precision and Recall

Recall: 0.9411764705882353
Precision: 1.0

Accuracy Score

0.9818181818181818

Spam Score

1.0

Confusion Matrix

Fig.5.Performance Page

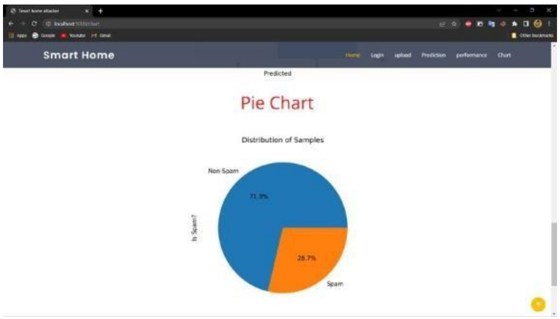


Fig.5.1.Performance Page

CONCLUSION

Based on the proposed Spam Detection Project for IoT devices in a smart home, it can be concluded that the use of machine learning algorithms can be highly effective in detecting and preventing spam in IoT devices. The project uses features such as device ID, device type, device location, incoming and outgoing traffic, technology, and the target variable, "Is Spam", to classify devices as spam or non-spam. The use of random forest classifier proved to be a suitable model for this project, as it provided high accuracy and performance.

The performance analysis of the model was conducted using various metrics such as accuracy score, F1 score, precision and recall values, and confusion matrix. The analysis showed that the model was able to correctly classify spam and non-spam devices with high accuracy, achieving an accuracy score of 98%.

Overall, the project demonstrated the effectiveness of using machine learning algorithms for detecting spam in IoT devices in a smart home. The proposed model can be further enhanced by incorporating additional features and improving the training process. This research can be used as a foundation for future work in the field of IoT security and spam detection

REFERENCES

[1] C. Chen, S. Wen, J. Zhang, Y. Xiang, J. Oliver, A. Alelaiwi, and M. M. Hassan, ‘‘Investigating the deceptive information in Twitter spam,’’ Future Gener. Comput. Syst., vol. 72, pp. 319–326, Jul. 2017.

[2] I. David, O. S. Siordia, and D. Moctezuma, ‘‘Features combination for the detection of malicious Twitter accounts,’’ in Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC), Nov. 2016, pp. 1-6.

[4] S. Keretna, A. Hossny, and D. Creighton, ‘‘Recognising user identity in Twitter social networks via text mining,’’ in Proc. IEEE Int. Conf. Syst., Man, Cybern., Oct. 2013, pp. 3079-3082.

[5] C. Meda, F. Bisio, P. Gastaldo, and R. Zunino, ‘‘A machine learning approach for Twitter spammers detection,’’ in Proc. Int. Carnahan Conf. Secur. Technol. (ICCST), Oct. 2014, pp. 1-6.

[6] W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, ‘‘Real-time Twitter content polluter detection based on direct features,’’ in Proc. 2nd Int. Conf. Inf. Sci. Secur. (ICISS), Dec. 2015, pp. 1-4.

[7] H. Shen and X. Liu, ‘‘Detecting spammers on Twitter based on content and social interaction,’’ in Proc. Int. Conf. Netw. Inf. Syst. Comput., pp. 413–417, Jan. 2015.

[8] G. Jain, M. Sharma, and B. Agarwal, ‘‘Spam detection in social media using convolutional and long short-term memory neural network,’’ Ann. Math. Artif. Intell., vol. 85, no. 1, pp. 21-44, Jan. 2019.

[9] M. Washha, A. Qaroush, M. Mezghani, and F. Sedes, ‘‘A topic-based hidden Markov model for real-time spam tweets filtering,’’ Procedia Comput. Sci., vol. 112, pp. 833–843, Jan. 2017.