1. INTRODUCTION

1.1. Introduction

IoT is taken into account as an interconnected and distributed network of embedded systems communicating through wired or wireless communication technologies. Massive growth and rapid development in the field of the Internet of Things (IoT), makes the presence of IoT devices prevalent in smart homes and smart cities. It is also defined because the network of physical objects or things empowered with limited computation, storage, and communication capabilities is also embedded with electronics (such as sensors and actuators), software, and network connectivity that permits these objects to gather, sometimes process, and exchange data. The things in IoT ask the objects from our lifestyle starting from smart household devices like a smart bulb, smart adapter, smart meter, smart refrigerator, smart oven, AC, temperature sensor, smoke detector, IP camera, to more sophisticated devices like frequency Identification (RFID) devices, heartbeat detectors, accelerometers, sensors in the parking zone, and a variety of other sensors in automobiles, etc. There are various large amounts of applications and services offered by the IoT ranging from critical infrastructure to agriculture, military, home appliances, and personal health care. As the usage of IoT devices increases the anomalies generated by these devices also grow beyond the count. IoT applications need to ensure information protection to fix security issues like interruptions, spoofing attacks, Dos attacks, jamming, eavesdropping, spam, and malware.

1.2. Brief information about the area of project

The maximum care to be taken is with web-based devices as the maximum number of IoT devices are web-dependent. It is common in the work environment that the IoT devices introduced in an association can be utilized to execute security and protection includes proficiently. For example, wearable devices that collect and send user’s health data to a connected smartphone should prevent leakage of data to ensure privacy. It has been found in the market that 25-30% of working employees connect their Personal IoT devices with the organizational network. The expanding nature of IoT attracts both the audience, i.e., the users and therefore the attackers.

However, with the emergence of ML in various attack scenarios, IoT devices choose a defensive strategy and decide the key parameters in the security protocols for a trade-off between security, privacy, and computation. This work enhances the algorithm to affect the time-series regression model rather than a classification model and may also execute ML models in parallel. This proposed paper focuses on determining the trustworthiness of the IoT device within the smart home network.

1.3. Motivation

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 spamicity score of the connected IoT devices within the network. The obtained results illustrate the efficiency of the proposed algorithm to analyze the time-series data from the IoT devices for spam detection.

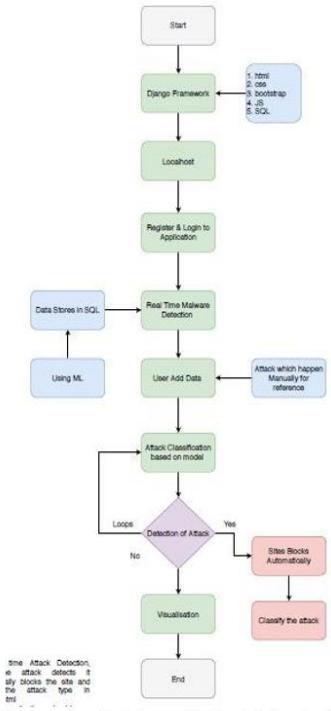


Fig 2.1: Flow Diagram of an efficient spam detection technique for iot devices using machine learning

Fig 1: System Architecture

1.4. Objectives of the Project

- The proposed scheme of spam detection is tested against four different machine learning models.
- An algorithm is designed to calculate the spamicity score of each Machine Learning model.
- Based on this spamicity score the trustworthiness of IoT devices is analyzed.

The use of machine learning models within the IoT has shown promising results for identifying malicious internet traffic using anomaly detection research. Moreover, either detection of anomalies or the employment of a spamicity score to trace the safety of the network components are motivated to possess a safe and secure network infrastructure. Several ML models are utilized for supervised machine learning; however, this paper uses ensemble methods, a group of ML techniques supported by decision trees.

1.5. Organization of the Project

The organization of project report is given below:

Chapter 1

It explains about the introduction to the project, introduction about the area of project, motivation, objectives of the project work and organization of the project report.

Chapter 2

It explains about the Literature survey. It majorly deals with all the findings and observation which is conducted as feasibility study before the actual development of the project.

Chapter 3

It explains about the System Analysis. It includes domain analysis, requirements analysis, and problem statement, problem description, existing system, proposed system and module description.

Chapter 4

It explains about Feasibility Study. It includes economical feasibility, technical feasibility, social feasibility and operational feasibility.

Chapter 5

It explains about Hardware and Software Requirements. It includes hardware components and software components.

Chapter 6

It explains about System Design. It includes input design, output design, code design, Unified Modeling Language and database design.

Chapter 7

It explains about Implementation. It includes implementation criteria, Python, JDBC, Python Server Pages, Servlets and MySQL.

Chapter 8

It explains about Testing. It includes Introduction, Testing Tactics, Test Cases and Testing Strategy.

Chapter 9

It explains about Conclusion and Future Enhancement. It includes project work conclusion and project work future enhancement.

Chapter 10

It explains about Coding. It includes the code of this project.

Chapter 11

It includes Screen Shots of the entire project.

2. LITERATURE SURVEY

A literature survey or literature review means study of references projects and old algorithms that are used for designing the proposed methods. It also helps in reporting summarization of all the old references projects, their drawbacks. The detailed literature survey for the project helps in comparing and contrasting various methods, algorithms in various ways that have implemented in the research.

2.1. RELATED WORK

There are various large amounts of applications and services offered by the IoT ranging from critical infrastructure to agriculture, military, home appliances, and personal health care. As the usage of IoT devices increases the anomalies generated by these devices also grow beyond the count. IoT applications need to ensure information protection to fix security issues like interruptions, spoofing attacks, Dos attacks, jamming, eavesdropping, spam, and malware. The maximum care to be taken is with web-based devices as the maximum number of IoT devices are web-dependent. It is common in the work environment that the IoT devices introduced in an association can be utilized to execute security and protection includes proficiently. For example, wearable devices that collect and send user's health data to a connected smartphone should prevent leakage of data to ensure privacy. It has been found in the market that 25-30% of working employees connect their Personal IoT devices with the organizational network. The expanding nature of IoT attracts both the audience, i.e., the users and therefore the attackers.

However, with the emergence of ML in various attack scenarios, IoT devices choose a defensive strategy and decide the key parameters in the security protocols for a trade-off between security, privacy, and computation. This work enhances the algorithm to affect the time-series regression model rather than a classification model and may also execute ML models in parallel.

This proposed paper focuses on determining the trustworthiness of the IoT device within the smart home network. The algorithm scores an IoT device with a spamicity score to secure smart devices by calculating spam scores using different machine learning models.

It is often used by computer attackers to characterize hosts or networks which they are considering hostile activity against. Thus it is useful for system administrators and other network defenders to detect portscans as possible preliminaries to a more serious attack. It is also widely used by network defenders to understand and find vulnerabilities in their own networks.

The Internet of Things (IoT) opens opportunities for wearable devices, home appliances, and software to share and communicate information on the Internet. Given that the shared data contains a large amount of private information, preserving information security on the shared data is an important issue that cannot be neglected. In this paper, we begin with general information security background of IoT and continue on with information security related challenges that IoT will encountered. Finally, we will also point out research directions that could be the future work for the solutions to the security challenges that IoT encounters. The idea of Internet of Things (IoT) is implanting networked heterogeneous detectors into our daily life. It opens extra channels for information submission and remote control to our physical world. A significant feature of an IoT network is that it collects data from network edges. Moreover, human involvement for network and devices maintenance is greatly reduced, which suggests an IoT network need to be highly self-managed and self-secured. For the reason that the use of IoT is growing in many important fields, the security issues of IoT need to be properly addressed.

Among all, Distributed Denial of Service (DDoS) is one of the most notorious attacking behaviors over network which interrupt and block genuine user requests by flooding the host server with huge number of requests using a group of zombie computers via geographically distributed internet connections. DDoS disrupts service by creating network congestion and disabling normal functions of network components, which is even more disruptive for IoT. In this paper, a lightweight defensive algorithm for DDoS attack over IoT network environment is proposed and tested against several scenarios to dissect the interactive communication among different types of network nodes.

As part of the recommended approach, the spammy characteristics are detected. ML models are used in Internet of things. This is the IoT data. it is pre-processed with the aid of pattern development method. By playing around with the structure, each IoT device is rewarded with ML models. The amount of spamming that has been detected As a result, the criteria for success have been refined. IoT equipment operating in a smart house As we go forward, will take into account meteorological conditions as well as the environment IoT devices more secured and reliable.

The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults.

3. SYSTEM ANALYSIS

A software requirement specification (SRS) is a portrayal of the conduct of the framework to be created. SRS is a premise for resulting configuration and usage. It recognizes functional and non- functional requirements and Hardware & Software prerequisites. SRS gives clear and succinct documentation particularly for reference and direction to all gatherings, to guarantee that the right framework is being produced. A reasonable comprehension of how analyzers and all the end clients (incorporates specialist, customers and scientists) collaborate with the framework is the fundamental motivation behind the SRS report at this level. In particular, the primary objective of the SRS is to empower the clients to advance proficiently from their present and insufficient operational idea to the new operational idea.

3.1. FUNCTIONAL REQUIREMENTS

Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified. Functional requirements describe what the system should do. The functional requirements can be further categorized as follows.

- What inputs the system should accept: Our system should accept a cover image and secret message
- What outputs the system should produce: Our proposed system must do embedding of secret into source texture image and decode it
- What data the system must store: The proposed system must store all the user details, images and encryption and decryption details.
- What are the computations to be done: All the pixel computations which involve matrix operations.

3.2. NON-FUNCTIONAL REQUIREMENTS

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy (i.e. how precise are the systems numerical answers). Non-functional requirements are the constraints that must be adhered during development. They limit what resources can be used and set bounds on aspects of the software's quality.

3.2.1. User Interfaces

Graphical User Interface is provided for interacting with the application. The user interface provides options to select the appropriate file for further processing. It also enables the user view the results after the entire processing is done .A good User Interface should be:

Clear

This feature enables the user who is interacting with the system to understand the system easily.A feature not understood is equals to a feature not developed.

Concise

Sometime when above feature is over looked it may lead to different understandings and burden of adding code which is not required for the proper working of the system.So a software should be clear and concise.

Responsive

The software should be responsive so that the user feels that he is answered. Generally responsiveness can be understood in 2 ways.

- Fast Reply
- Interactive Reply

Both are needed for any software.

3.3. EXISTING SYSTEM

Denial of service (DDoS) attacks: The attackers can flood the target database with unwanted requests to stop IoT devices from having access to various services. These malicious requests produced by a network of IoT devices are commonly known as bots. DDoS can exhaust all the resources provided by the service provider. It can block authentic users and can make the network resource unavailable.

RFID attacks: These are the attacks imposed at the physical layer of IoT device. This attack leads to loose the integrity of the device. Attackers attempt to modify the data either at the node storage or while it is in the transmission within network. The common attacks possible at the sensor node are attacks on availability, attacks on authenticity, attacks on confidentiality, Cryptography keys brute-forcing. The countermeasures to ensure prevention of such attacks includes password protection, data encryption and restricted access control.

Internet attacks: The IoT device can stay connected with Internet to access various resources. The spammers who want to steal other systems information or want their target website to be visited continuously, use spamming techniques. The common technique used for the same is Ad fraud. It generates the artificial clicks at a targeted website for monetary profit. Such practicing team is known as cyber criminals.

NFC attacks: These attacks are mainly concerned with electronic payment frauds. The possible attacks are unencrypted traffic, Eavesdropping, and Tag modification. The solution for this problem is the conditional privacy protection. So, the attacker fails to create the same profile with the help of user’s public key. This model is based on random public keys by trusted service manager.

3.3.1. Disadvantages

Drawbacks of Existing system are:

- In the existing work, the system is less effective due to lack of Spam Detection in IoT using Machine Learning framework.
- This system is less performance in which it is clear that Supervised machine learning techniques is absence.

3.4. PROPOSED SYSTEM

The digital world is completely dependent upon the smart devices. The information retrieved from these devices should be spam free. The information retrieval from various IoT devices is a big challenge because it is collected from various domains. As there are multiple devices involved in IoT, so a large volume of data is generated having heterogeneity and variety. We can call this data as IoT data. IoT data has various features such as real-time, multi-source, rich and sparse.

3.4.1. Advantages

- The proposed scheme of spam detection is validated using five different machine learning models.
- An algorithm is proposed to compute the spamicity score of each model which is then used for detection and intelligent decision making.
- Based upon the spamicity score computed in previous step, the reliability of IoT devices is analyzed using different evaluation metrics.

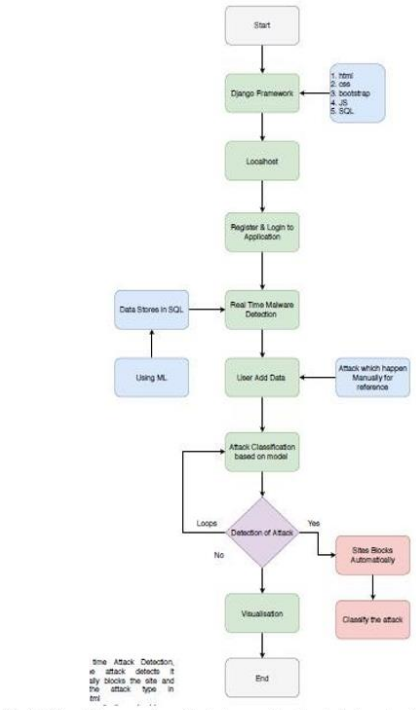


Fig 2.1: Flow Diagram of an efficient spam detection technique for iot devices using machine learning

Fig. 3.1. System Architecture

3.5. MODULES AND ITS DESCRIPTION

Support Vector Machines (SVM)

Support vector machines, also referred to as support vector networks, are a group of related supervised learning methods used for classification and regression. However, it's mostly utilized in classification problems. Within the SVM algorithm, we plot each data item as a degree in n-dimensional space (where n is the number of features you have) with the worth of every feature being the worth of a specific coordinate. Then, we can classify by finding the hyper-plane that differentiates both classes. Hence, we can say that the main objective of SVM is to find a hyperplane in an N- dimensional space that distinctly classifies the data points. SVM can classify both linear and non-linear data. To classify non-linear data it uses a method called the kernel trick to rework your data so it supports these transformations and it also finds an optimal boundary between the possible outputs. A kernel is a function which maps a lower dimensional data into higher dimensional data. Simply put, it does some extremely complex data transformations, then figures out a way to separate your data supporting the labels or outputs you've defined.

Given a group of coaching examples, each marked as belonging to 1 of two categories, an SVM training algorithm builds a model that predicts whether a replacement example falls into one category or the opposite. An SVM training algorithm could be a non-probabilistic, binary, linear classifier, although methods like Platt scaling exist to use SVM in a very probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what's called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. SVM also uses another method called Soft Margin which allows SVM to make certain number of mistakes and keep the margin as wide as possible so that other points can be classified correctly.

Random Forest

Random forests (RF) are a bag containing n Decision Trees (DT) having a special set of hyper-parameters and trained on different subsets of information. In machine learning language, Random Forests also are called an ensemble or bagging method. Random forest is one amongst the foremost used algorithms due to its simplicity and stability. Random forests are more stable and reliable than simply a choice tree. Random Forest is a supervised machine learning technique used for both classification and regression. But we'll discuss its use for classification because it’s more intuitive and straightforward to grasp. It's an ensemble of decision trees that helps in reducing the variance in decision trees. It fulfills a balance between high variance and high bias by sampling with each tree fitted and a sample of features at each split, respectively. The performance of random forest relies on the appropriate selection of the number of trees, N. As within the case of bagging, a greater value of N doesn't necessarily overfit the info, and hence, a sufficiently large value of N is chosen.

Decision Tree

A decision tree is a “set of rules” created by learning on a dataset that can be used to make predictions on future data. It employs a top-down approach, by utilizing variance reduction to partition the info into subsets of homogeneous values. It incorporates mixtures of categorical and numerical predictor variables with an integral a part of the procedure to perform internal feature selection. These are the explanations why decision trees have emerged together of the foremost popular data processing learning methods.

Decision trees can create an over-complex tree, which doesn't tend to generalize the information well and might lead to overfitting, while the choice tree doesn't perform as neural networks for nonlinear networks, it's usually prone to noisy data. Decision trees expect visible trends within the data and also perform well on sequential patterns; if this is often not the case, then decision trees must be avoided for statistical applications. Decision trees are used for regression and classification problems. Most often used in classification problems. Decision trees finds the relation between target data and input features with simple decision rules. If the rule is met then the classification of particular node gets break and called terminated leaf. The parent node in decision tree consists of total samples and gets further classified upon decision rules. Edges in decision trees are rules or values that helps to classify the data.

DATA SET

The Data Set used in this paper is REFIT Smart Home Data Set with some modifications. Firstly, the feature reduction is finished. In the IoT dataset used in this proposal, we have 10 features as shown in the above table. After the feature extraction, the feature selection is performed. The features along with their importance score computed by the entropy-based filter are presented in Table I. For better understanding of dataset refer to.

Spamicity Score Extreme Gradient Boosting (XGBOOST) Algorithm

Extreme Gradient Boosting is a well-liked supervised machine learning model with characteristics of distributed and out-of-core computation, efficiency, and parallelization. The parallelization occurs for multiple nodes in a very single tree and not across trees. It's a gradient boosting system that is efficient and scalable. The package includes a good linear model solver and an algorithm for tree learning. It supports various objective functions like regression, grouping, and ranking. It works with numeric vectors. It’s ten times quicker than existing gradient boosting algorithms. The strategy of gradient boosting uses more accurate approximations to seek out the most effective tree model. It uses a variety of clever tricks that make it particularly competitive with structured data normally. The poor learner is made up in each training round and its predictions are matched with the correct outcome.

4. FEASIBILITY STUDY

Introduction

The achievability study is somewhat an estimation and examination of the different potential prerequisites of a momentum evaluated flow improvement which is taking into account wide and broad examination and propelled exploration work to support the procedure of good choice making. Practicality Study is point by point investigation of making examination and social event data for adding to the present advancement. This study gives data with respect to Technical Information, Economical or Cost Information, Procedural Study, Social and different studies which are attainable in meaning the present advancement or not. The significant ranges considered in attainability investigation are as per the following.

- Economic Feasibility
- Technical Feasibility
- Procedural Feasibility

4.1. ECONOMIC FEASIBILITY

The purpose of the economic feasibility appraisal is to determine the positive economic benefits related cost, expenditure and other maintenance to the organization that the projected system will provide. It includes various expenditures and budges related to quantification and identification of all the economic requirements for intending the current development which is expected. This estimation naturally involves a cost benefits analysis.

4.2. TECHNICAL FEASIBILITY

In technical feasibility study we focus on the system requirements for development of the current development. It is technically feasible to intend the current development as the entire modules described in the modules description can be created using Front-End interaction Python Swings and Networking activities using Python RMI. As the current development modules are focused on wireless activities, Python supports J2ME Python Mobile Edition packages for wireless programming, J2EE Python Enterprise Edition packages for Networking programming.

To implement the current development we have selected the given technical environment, we require Pentium/Core-2 Duo Processor with 2 GB Ram and 160 GB Hard disk and Python Programming language.

4.3. PROCEDUREAL FEASIBILITY

Our application provides Graphical Interface for the end user and which very easy and feasible to operate. The front end navigations are created using Python swings which provides very easy to the user to enter the necessary information and get the necessary outputs. The current development is highly user interactive application and network based. The current development is provided with various windows, buttons and other graphical navigations so that the system is fully procedurally feasible.

5. SOFTWARE AND HARDWARE REQUIREMENT SPECIFICATIONS

5.1. SOFTWARE REQUIREMENTS

Operating system	:	Windows 10
Coding Language	:	python
Tool	:	PyCharm
Database	:	MYSQL

5.2. HARDWARE REQUIREMENTS

Processor	:	Pentium Dual Core/ Core 2 Duo/ ICore with Minimum 1.2 GHZ Speed
RAM	:	2 GB
Hard Disk	:	160 GB

6. SYSTEM DESIGN

System design or System planning is the procedure of defining the project Structure, architecture, Planning, components, modules, interfaces, and data elements for a system to satisfy the design requirements and helps to start the work in planned way. Systems design or Planning could be seen as the appliance of systems philosophy and helps to product development in a systematic manner. There is some extensions with the disciplines of systems analysis and planning, systems architecture and development engineering. System Design is broadly divided in two activities.

1. Logical Design
2. Physical Design

Logical design:

The logical design of a system is concerned to an theoretical representation of the project planning using UML Flows, data flows, inputs and outputs of the system. Logical Design is also called as Graphical Modelling of System planning. In the Logical context of systems design are included.

Physical design:

The physical design and planning relates to the real and actual input and output processes to be given the system. This process is a study of various data inputs and outputs to be processed in the system. Physical Design involves in User Interface Design Front End Screens, Data Design Back end Tables and Process Design Algorithm.

6.1 UML Diagrams

Definitions:

A use-case diagram is a set of use cases, actors and relationships, associations of actors and use-cases and their integration into system as a whole.

A use-case diagram contains

- Use-cases
- Actors
- Association relationship between actors, use-cases
- Generalization between actors

Common uses of use-case diagram:

- Provides high level view of system with respect to users
- To model context of system
- Determine human system interaction

The general components in a use-case diagram are:

- Use-case
- Actor
- Association

Use case:

It functionally provided by system. Use case is represented graphically as ellipse with name inside it.

Actor:

An actor is user of system or database in system. These are represented with stick figure.

Association:

It links actors to use use-case explains in what and how actor interacts with system.

Identifying the actors:

The team actor represents the role user plays with respect to the system. A user may play more than one role. However an actor should represent a single user.

Actors can be identified by following questions:

- Who is utilizing the system or who is affected by the system or which groups need help from the system to carry out a task.
- Which external hardware or other system use the system to execute tasks.
- What problem dies this application solve.

Identifying the use-cases:

1. One of the methods used to identify the use cases is actor based.
    - The actors related to a system are identified.
    - The task, process, functions involved are performed by each actor.
2. The second method used to identify use-cases is event based.
    - The exterior events that a system must react to are acknowledged.
    - Transmit the events to actors and use cases.

Identifying the Relationships:

Communication:

Connecting the actor symbol to the use case symbol with a solid path shows the communication relationship of an actor in a use case. The actor is said to communicate with use case.

Uses:

It occurs when the use case have some sub flows in common. To avoid redundancy in sub flow, the system can have common sub flow and make it a use case of its own.

Extend:

It is used when system has a sub use case, which has specialized features.

Generalization:

A taxonomic relationship between a use case / actor (the child) and the use case /actor (the parent). Sub use cases inherit behavior of parents.

USECASE DIAGRAM

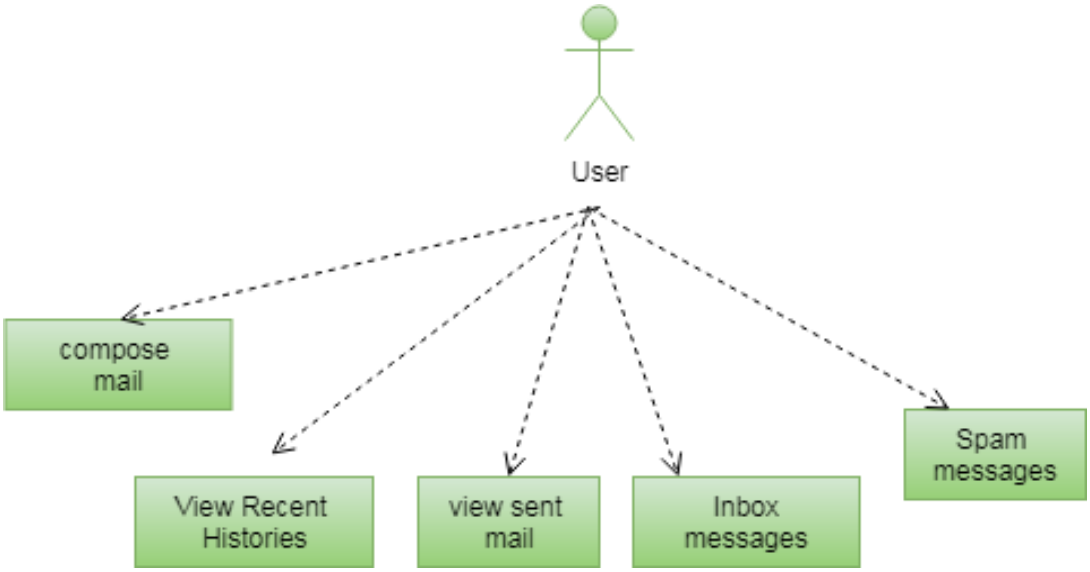


Fig. 6.1.1: use case diagram

CLASS DIAGRAM

The class diagram is used to represent the classes used and their associations. A class diagram stays at middle level between static model and dynamic model. It captures both the static and dynamic part of design. A system can be assumed with a class diagram.

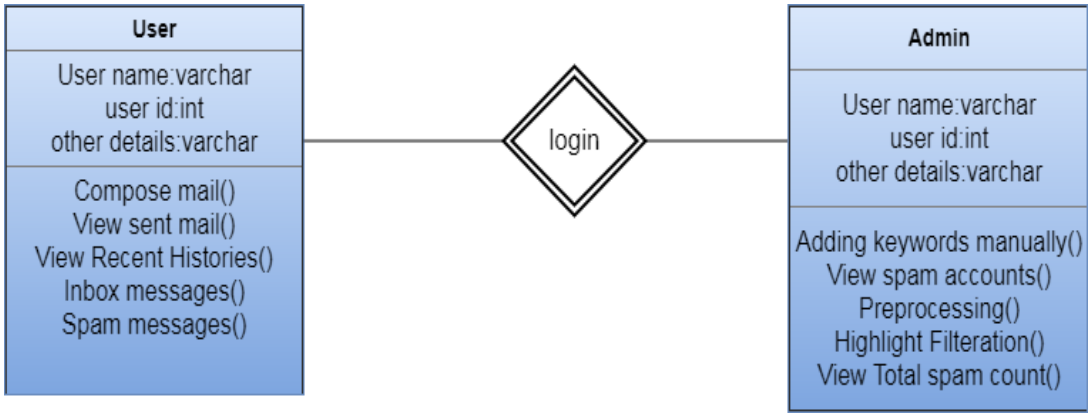


Fig. 6.1.2 : Class diagram

SEQUENCE DIAGRAM

These are used to describe the behavior of the system. These describe states and the movements between to next states on actions. Each diagram usually represents objects of a single class and tracks the different states of its objects through the system.

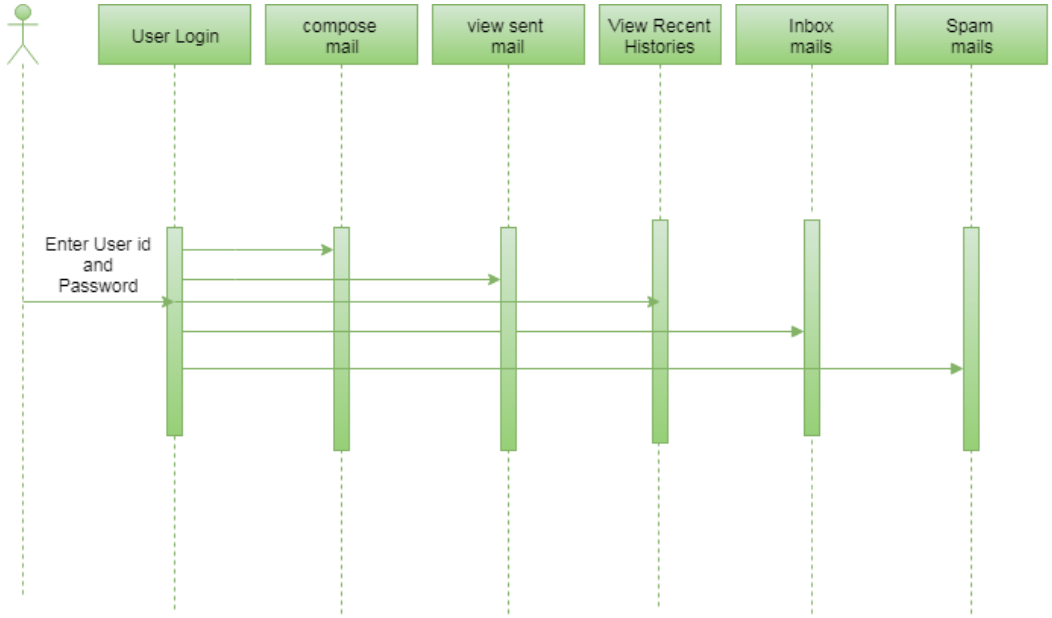


Fig. 6.1.3: Sequence diagram

ACTIVITY DIAGRAM:

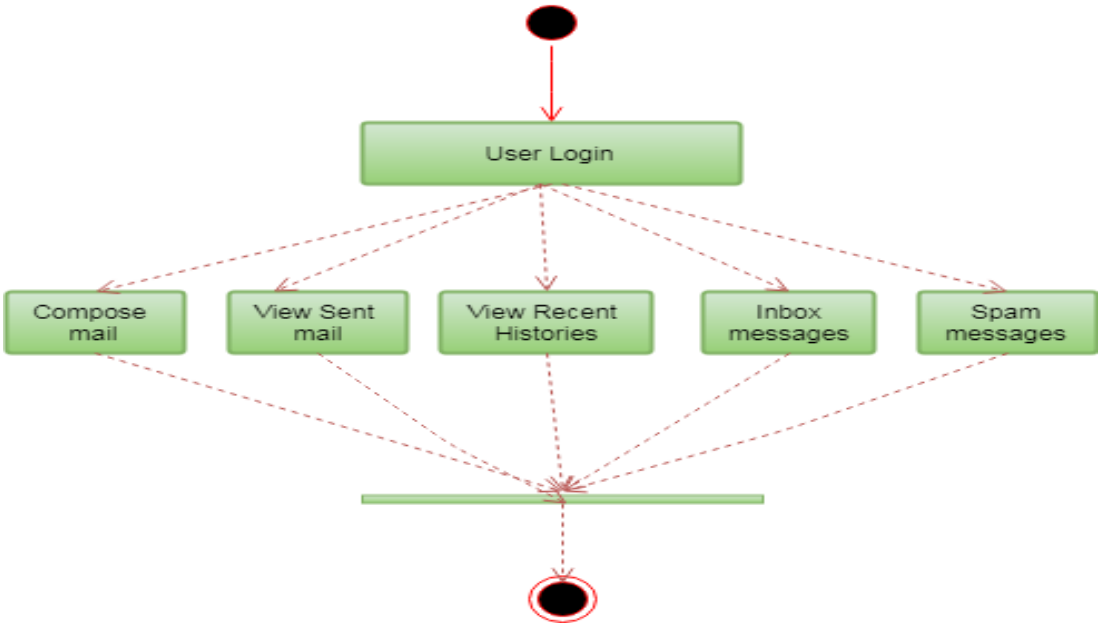


Fig. 6.1.4: Activity diagram

7. IMPLEMENTATION

7.1. Python

Below are some facts about Python. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Python. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python :-

Let’s see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Python and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Python, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Python.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

8. Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

9. Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.
Any doubts till now in the advantages of Python? Mention in the comment section.

8. TESTING

8.1. SOFTWARE TESTING TECHNIQUES

Testing Software is a critical process which includes many activities, elements of software excellence assertion and represents the ultimate review of specification, design and coding, Software Testing presents a wide nature of an interesting variance for the software developers.

8.1.1. Testing Objectives

1. Testing is a series of steps which includes executing a program with various inputs and intent of finding an error from the inputs and making the developer to make corrections on error finding.

2. A good Software test case is one that has a possibility of finding an undiscovered error in the designed program.

3. A successful Software Testing is one that exposes an unknown or undiscovered error.

These above objectives imply a dramatic change in view port.

Software Testing is a series of steps but it cannot show the absence of defects and errors but it can only show various errors that are found software or program.

8.1.2. Test Case Design

Any Software product can be tested in one of two ways:

White Box Testing

White Box Testing is also called as Open or Glass box testing. In White Box Testing, by finding the specified program or function that a software product or a software program has been designed or developed to perform or execute the test can be implemented and conducted for the demonstrates each program or function in a fully operated at the same time finding for errors in each program. It is a glass box or open test case design method that uses the wide control on structure of the procedural program and design to find and drive the test cases. The starting path testing activities is a white box testing.

Black Box Testing

In Black Box testing by understanding and knowing the various program internal operation of a application or product or program, Black Box Testing can be conducted to guarantee that all gears mesh of the internal activities of the product or program or application can be tested. The process provides a internal operation to check the performance and specifications of all the internal mechanism which have been passably exercised. Black Box Testing fundamentally focuses on the functional activities and requirements of the software.

8.2. SOFTWARE TESTING STRATEGIES

Software Testing Strategy integrates the software test cases into a series of well planned steps and series of planned procedures that result in the successful construction, Design and Implementation of a software. Various Software testing Methods are referred for Verification and Validation. Software Verification refers to the set of activities on the designed functions and programs for ensuring that the software or the product correctly implements a specific function or the required output. Software Validation refers to a set of activities that ensure that the software or product or application that has been built for traceable to customer's requirements and providing the customer to input valid data and make Data store free from redundancy.

8.2.1. Unit Testing

In software testing, Unit testing mainly focuses on verification effort on the smallest unit of program or software design that is also called a module. In unit testing the procedural or functional design provides a detailed description as a guide, focal the control paths are tested to uncover errors occurred in the designed software within the boundaries of the module. The unit testing of software is normally white box or open testing oriented and the series of steps can be conducted in corresponding or parallel for multiple modules or functions.

8.2.2. Integration Testing

Integration testing is another Testing for systematic technique and product module integrating which constructs the program structure and makes the data flow between the modules, while conducting Integration Testing it requires to uncover errors associated with various interfaces. The main objective is to take unit tested methods and activities to build a program structure that have been dictated by design.

Top-Down Integration

The next Testing process is top down integrations is an sequence approach for construction and testing of a program structure. In a Software or product or application various modules are integrated with each other by moving downward through the systematic control hierarchy between the modules, beginning with the main control or home control or index program. Various activities or modules connected to the main program are included in the structure of the project or either in the breath first or depth first manner.

Bottom-up Integration

The next testing method as the name suggests, which begins in construction and testing with various atomic modules of the product i.e., modules or functions at the lowest level. Because the all the functions or modules are having integration between bottom up manner in which the processing is required for the modules having connection to a given level is always available and the need for remnant is eliminated.

8.2.3. Validation Testing

The Validation Testing is integration testing for software which is completely assembled as a package. The Validation testing is the next stage in Testing Activities, which can be defined as successful testing process for the software functions in the manner reasonably expected by the customer. The validation Testing is mainly performed at the end approach of the user needs in testing the information inputed to the product and information contained in those sections are to validated through various testing approaches.

The sensible prospect is defined in the software development with a requirement specification, and a document that gives de tailed information of all uservisible attributes of the software methodologies. The document of specification contains a section titled Validation Criteria in which the end user should follow various indications in give the inputs.

1. Test 2. C. No.	3. Input	4. Expected Behavior	5. Observed behavior	6. Status 7. P=Passed 8. F = Failed
9. 1	10. Input Userid, 11. Password for login	12. Home page of User or 13. Administrator show display	14. -do-	15. P
16. 2	17. Comment control	18. Displays and controls Comments	19. -do-	20. p

TEST CASES

EXECUTION STEPS:

How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

Download the Correct version into the system

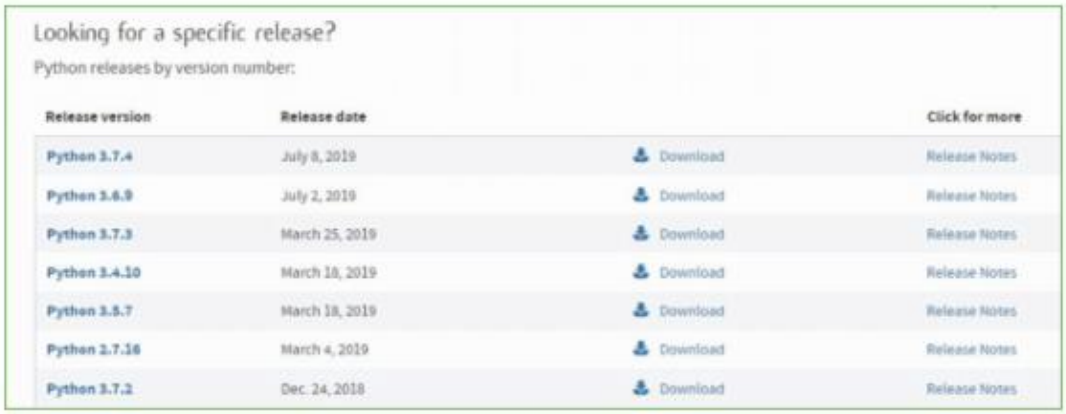
Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: **https://www.python.org**

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Step 4: Scroll down the page until you find the Files option.

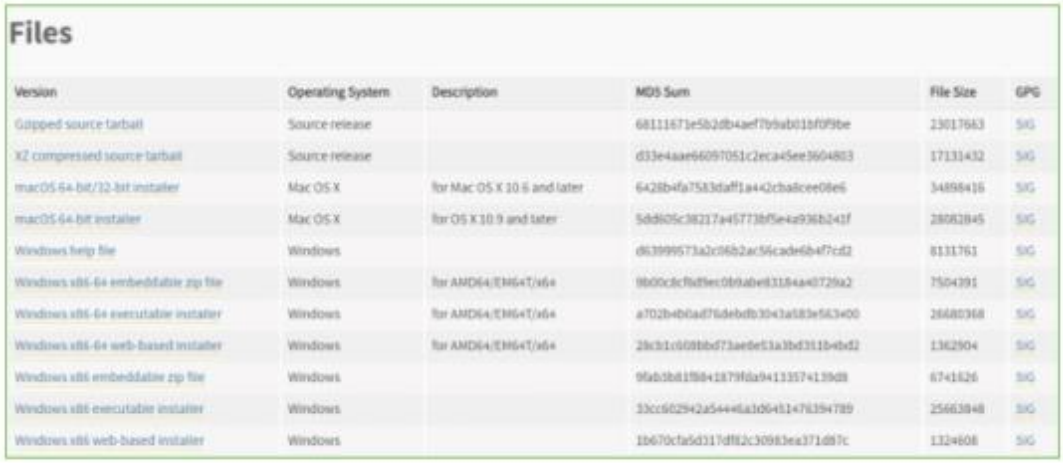


Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 5: Here you see a different version of python along with the operating system.

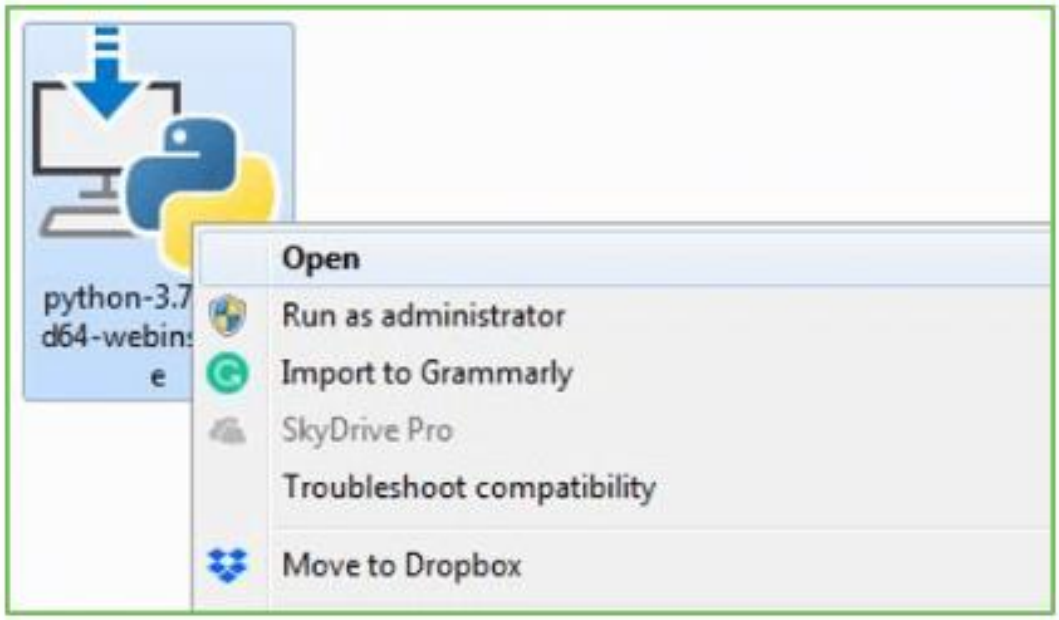


Files

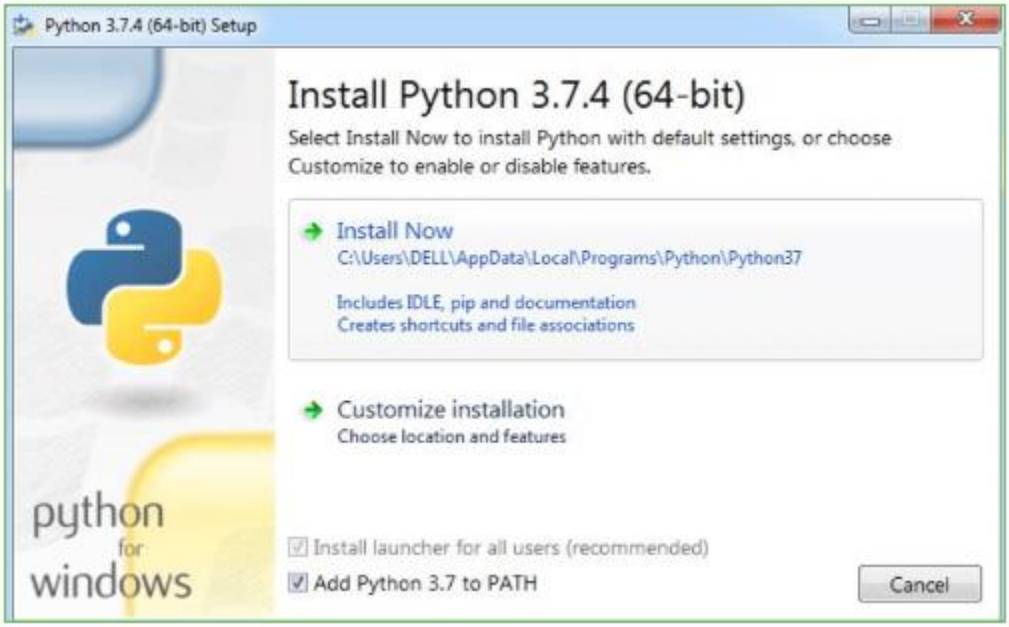
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab018f0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a457738f5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c86b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cfbdfec0b9abe83184a4072fa2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0adf76debdb30e3a583e563e0	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c3c1c6088bd73ae8e53a3bd3118b4bd2	1362504	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda9413357413bd8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a5444a6a1d6451e76394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa6d317dff82c30983ea371d87c	1324608	SIG

Installation of Python

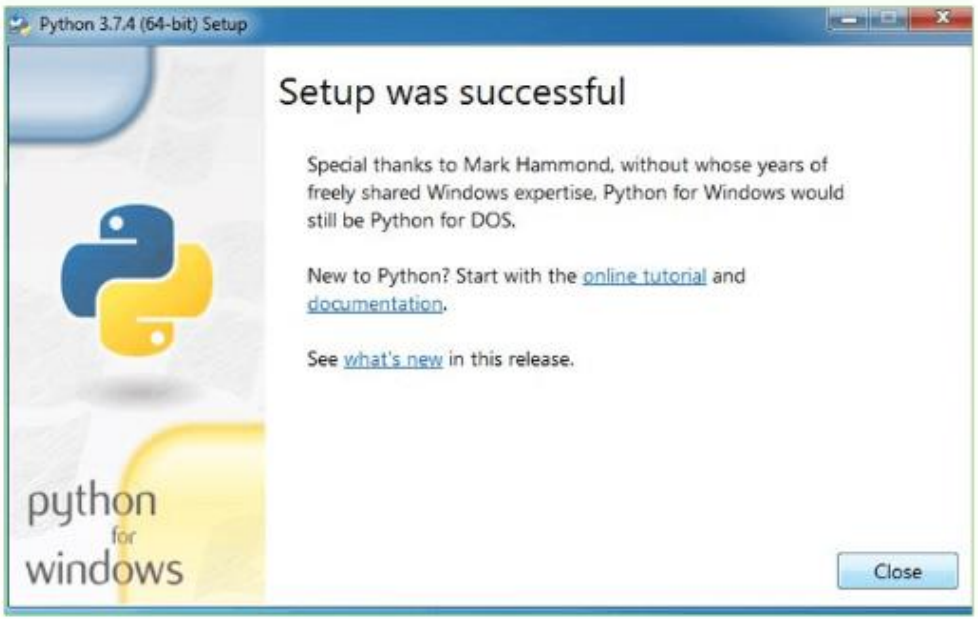
Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



Step 3: Click on Install NOW After the installation is successful. Click on Close.



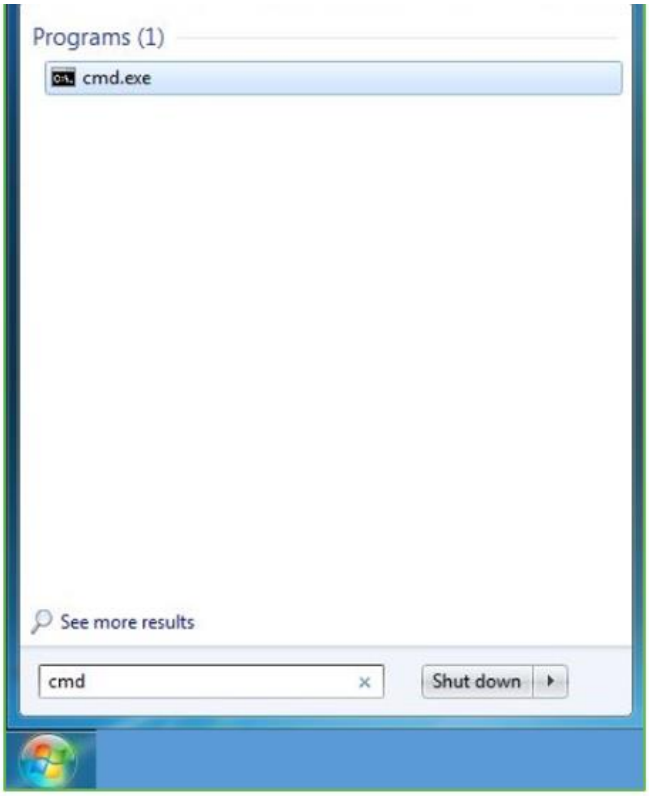
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

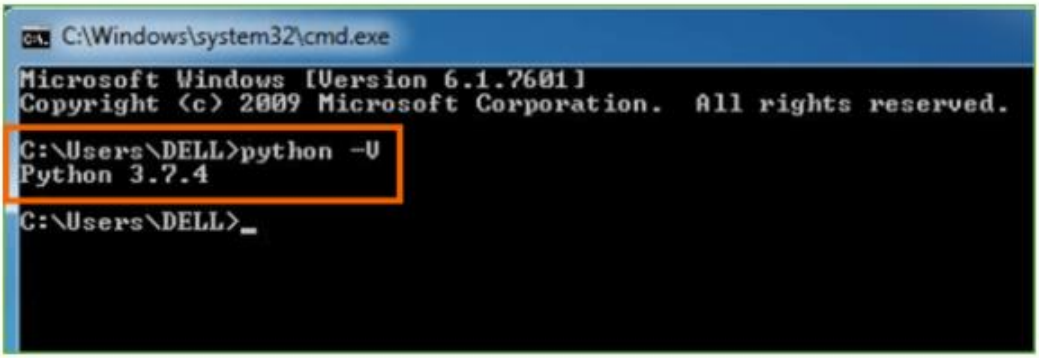
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



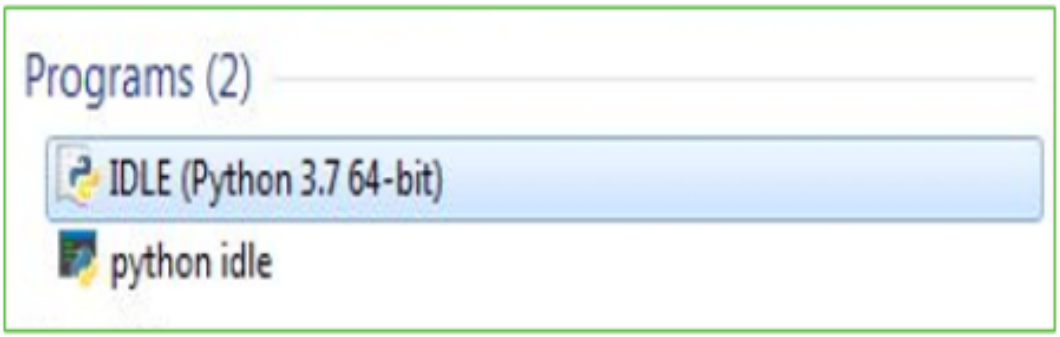
Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

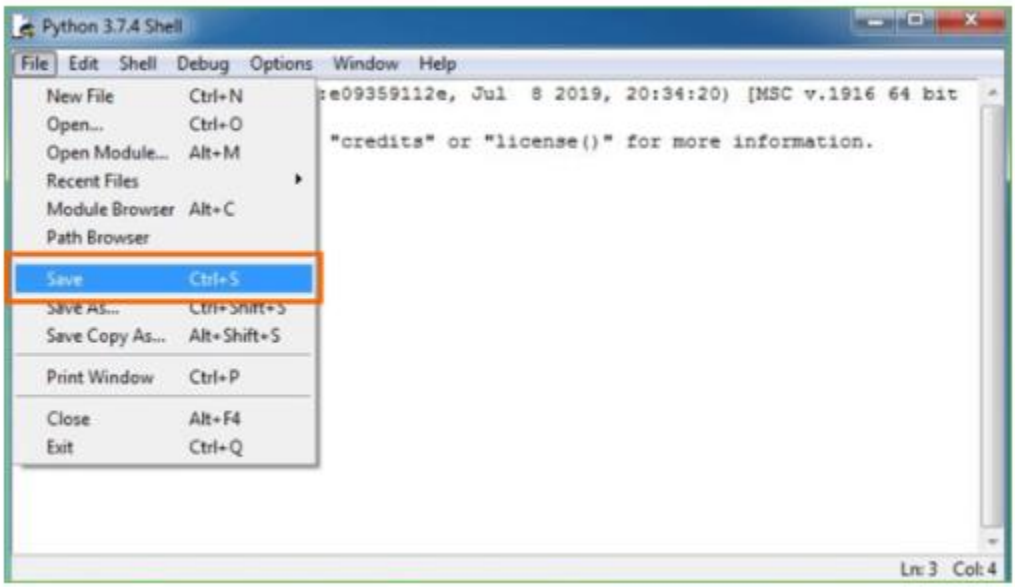
Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. **enter print**

9. CONCLUSIONS AND FUTURE ENHANCEMENTS

9.1 CONCLUSION

The proposed framework, detects the spam parameters of IoT devices using machine learning models. The IoT dataset used for experiments, is pre-processed by using feature engineering procedure. By experimenting the framework with machine learning models, each IoT appliance is awarded with a spam score.

9.2 FUTURE ENHANCEMENTS

This refines the conditions to be taken for successful working of IoT devices in a smart home. In future, we are planning to consider the climatic and surrounding features of IoT device to make them more secure and trustworthy. n

SAMPLE CODE

```
from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

import pymysql

from django.http import HttpResponse

from django.conf import settings

from django.core.files.storage import FileSystemStorage

import datetime

import matplotlib.pyplot as plt

import re

import numpy as np

from sklearn import svm

import pandas as pd

from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split

import nltk

from nltk.corpus import stopwords

from sklearn.metrics import classification_report

from sklearn.metrics import confusion_matrix

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive_bayes import MultinomialNB

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import BaggingClassifier
```

```
from sklearn_extensions.extreme_learning_machines.elm import
GenELMClassifier
from sklearn_extensions.extreme_learning_machines.random_layer import
RBFRandomLayer, MLPRandomLayer
from sklearn.linear_model import LogisticRegression
from django.core.files.storage import FileSystemStorage
import datetime
from numpy.linalg import norm
from numpy import dot

global classifier
global label_count
global X
global Y
corpus = []

def index(request):
    if request.method == 'GET':
        return render(request, 'index.html', {})

def SendPost(request):
    if request.method == 'GET':
        return render(request, 'SendPost.html', {})

def Register(request):
```

```
    if request.method == 'GET':
        return render(request, 'Register.html', {})

def Admin(request):
    if request.method == 'GET':
        return render(request, 'Admin.html', {})

def Login(request):
    if request.method == 'GET':
        return render(request, 'Login.html', {})

def AddCyberMessages(request):
    if request.method == 'GET':
        return render(request, 'AddCyberMessages.html', {})

def RunAlgorithms(request):
    if request.method == 'GET':
        return render(request, 'RunAlgorithms.html', {})

def MonitorPost(request):
    if request.method == 'GET':
        strdata = '<table border=1 align=center width=100%><tr><th>Sender
Name</th><th>File Name</th><th>Message</th><th>Post Time</th>
<th>Status</th></tr><tr>'
```

```
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password =
'root', database = 'cyber',charset='utf8')
        with con:
          cur = con.cursor()
          cur.execute("select * FROM posts")
          rows = cur.fetchall()
          for row in rows:
            strdata+='<td>'+str(row[0])+'</td><td><img
src=/static/photo/'+str(row[1])+' width=200
height=200></img></td><td>'+str(row[2])+'</td><td>'+str(row[3])+'</td><td>'+
str(row[4])+'</td></tr>'
    context= {'data':strdata}
    return render(request, 'MonitorPost.html', context)


def AddBullyingWords(request):
    if request.method == 'POST':
      message = request.POST.get('t1', False)
      label = request.POST.get('t2', False)
      message = message.strip('\n')
      message = message.strip()
      message = message.lower()
      message = re.sub(r'[^a-zA-Z\s]+', '', message)
      file = open('dataset.txt','a+')
      file.write(message+","+label+"\n")
      file.close()
```

```
        context= {'data':'Cyber Words added to dataset as '+label}
        return render(request, 'AddCyberMessages.html', context)


def Signup(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        contact = request.POST.get('t3', False)
        email = request.POST.get('t4', False)
        address = request.POST.get('t5', False)
        db_connection = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root',
password = 'root', database = 'cyber',charset='utf8')
        db_cursor = db_connection.cursor()
        student_sql_query = "INSERT INTO
users(username,password,contact_no,email,address,status)
VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"','
Accepted')"
        db_cursor.execute(student_sql_query)
        db_connection.commit()
        print(db_cursor.rowcount, "Record Inserted")
        if db_cursor.rowcount == 1:
         context= {'data':'Signup Process Completed'}
         return render(request, 'Register.html', context)
        else:
         context= {'data':'Error in signup process'}
```

```
        return render(request, 'Register.html', context)



def UserLogin(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        index = 0
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password =
'root', database = 'cyber',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM users")
            rows = cur.fetchall()
            for row in rows:
                if row[0] == username and password == row[1] and row[5] == 'Accepted':
                    index = 1
                    break
        if index == 1:
         file = open('session.txt','w')
         file.write(username)
         file.close()
         context= {'data':'welcome '+username}
         return render(request, 'UserScreen.html', context)
        else:
```

```
        context= {'data':'login failed'}
        return render(request, 'Login.html', context)


def AdminLogin(request):
    if request.method == 'POST':
      username = request.POST.get('t1', False)
      password = request.POST.get('t2', False)
      if username == 'admin' and password == 'admin':
       context= {'data':'welcome '+username}
       return render(request, 'AdminScreen.html', context)
      else:
       context= {'data':'login failed'}
       return render(request, 'Admin.html', context)


def ViewUsers(request):
    if request.method == 'GET':
       color='<font size="" color=black>'
       strdata = '<table border=1 align=center
width=100%><tr><th>Username</th><th>Password</th><th>Contact
No</th><th>Email ID</th><th>Address</th><th>Status</th></tr><tr>'
       con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password =
'root', database = 'cyber',charset='utf8')
       with con:
          cur = con.cursor()
          cur.execute("select * FROM users")
```

```
        rows = cur.fetchall()
        for row in rows:

strdata+='<td>'+color+row[0]+'</td><td>'+color+row[1]+'</td><td>'+color+row[
2]+'</td><td>'+color+str(row[3])+'</td><td>'+color+str(row[4])+'</td><td>'+col
or+row[5]+'</td></tr>'
    context= {'data':strdata+'</table><br/><br/><br/>'}
    return render(request, 'ViewUsers.html', context)


def ViewUserPost(request):
    if request.method == 'GET':
        strdata = '<table border=1 align=center width=100%><tr><th>Sender
Name</th><th>File Name</th><th>Message</th><th>Post Time</th>
<th>Status</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password =
'root', database = 'cyber',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM posts")
            rows = cur.fetchall()
            for row in rows:
                strdata+='<td>'+str(row[0])+'</td><td><img
src=/static/photo/'+str(row[1])+' width=200
```

```
height=200></img></td><td>'+str(row[2])+'</td><td>'+str(row[3])+'</td><td>'+
str(row[4])+'</td></tr>'
    context= {'data':strdata}
    return render(request, 'ViewUserPost.html', context)

def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts
def prediction(X_test, cls): #prediction done here
    y_pred = cls.predict(X_test)
    for i in range(len(X_test)):
      print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
    return y_pred

def cal_accuracy(y_test, y_pred, details):
    msg = ''
    cm = confusion_matrix(y_test, y_pred)
```

```
    accuracy = accuracy_score(y_test,y_pred)*100
    msg+=details+"<br/>"
    msg+="Accuracy : "+str(accuracy)+"<br/>"
    #msg+="Report : "+str(classification_report(y_test, y_pred))+"<br/>"
    #msg+="Confusion Matrix : "+str(cm)+"<br/>"
    return msg

def classifyPost(vec1, vec2):
    vector1 = np.asarray(vec1)
    vector2 = np.asarray(vec2)
    return dot(vector1, vector2)/(norm(vector1)*norm(vector2))

def PostSent(request):
    if request.method == 'POST' and request.FILES['t2']:
        output = ''
        myfile = request.FILES['t2']
        msg = request.POST.get('t1', False)
        fs = FileSystemStorage()
        filename =
fs.save('E:/venkat/2021/May21/Cyber/CyberBullying/static/photo/'+str(myfile),
myfile)
        now = datetime.datetime.now()
        current_time = now.strftime("%Y-%m-%d %H:%M:%S")
        text = 'msg,label\n'+msg+",?"
        f = open("test.txt", "w")
```

```
f.write(text)
f.close()
df = pd.read_csv('test.txt')
X1 = df.iloc[:, :-1].values

dataset = ''
for k in range(len(corpus)):
    dataset+=corpus[k]+","

dataset = dataset[0:len(dataset)-1]
dataset+="\n"

for i in range(len(X1)):
    line = str(X1[i]).strip('\n')
    line = line.strip()
    line = line.lower()
    line = re.sub(r'[^a-zA-Z\s]+', '', line)
    print(line)
    wordCount = word_count(line.strip())
    value = ''
    for j in range(len(corpus)):
        if corpus[j] in wordCount.keys():
            value+=str(wordCount[corpus[j]])+","
        else:
            value+="0,"
```

```
    value = value[0:len(value)-1]
    dataset+=value+"\n"

print(dataset)
f = open("test.txt", "w")
f.write(dataset)
f.close()
test = pd.read_csv("test.txt")
X1 = test.values[:, 0:label_count]
result = classifier.predict(X1)
#classify = 0
#score = 0
#for i in range(len(X)):
#    cosine = classifyPost(X[i], X1)
#    if cosine > score:
#    score = cosine
#    classify = Y[i]
status = 'Non-Cyber Harassers'
print(result)
if result[0] == 0:
    status = 'Non-Cyber Harassers'
else:
    status = 'Cyber Harassers'

user = ''
```

```
        with open("session.txt", "r") as file:
          for line in file:
             user = line.strip('\n')

        db_connection = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root',
password = 'root', database = 'cyber',charset='utf8')
        db_cursor = db_connection.cursor()
        student_sql_query = "INSERT INTO
posts(sender,filename,msg,posttime,status)
VALUES('"+user+"','"+str(myfile)+"','"+msg+"','"+current_time+"','"+status+"')"
        db_cursor.execute(student_sql_query)
        db_connection.commit()
        print(db_cursor.rowcount, "Record Inserted")
        if db_cursor.rowcount == 1:
            context= {'data':'Posts details added'}
            return render(request, 'SendPost.html', context)
        else:
            context= {'data':'Error in adding post details'}
            return render(request, 'SendPost.html', context)

def RunAlgorithm(request):
    global classifier
    global label_count
    global X
    global Y
```

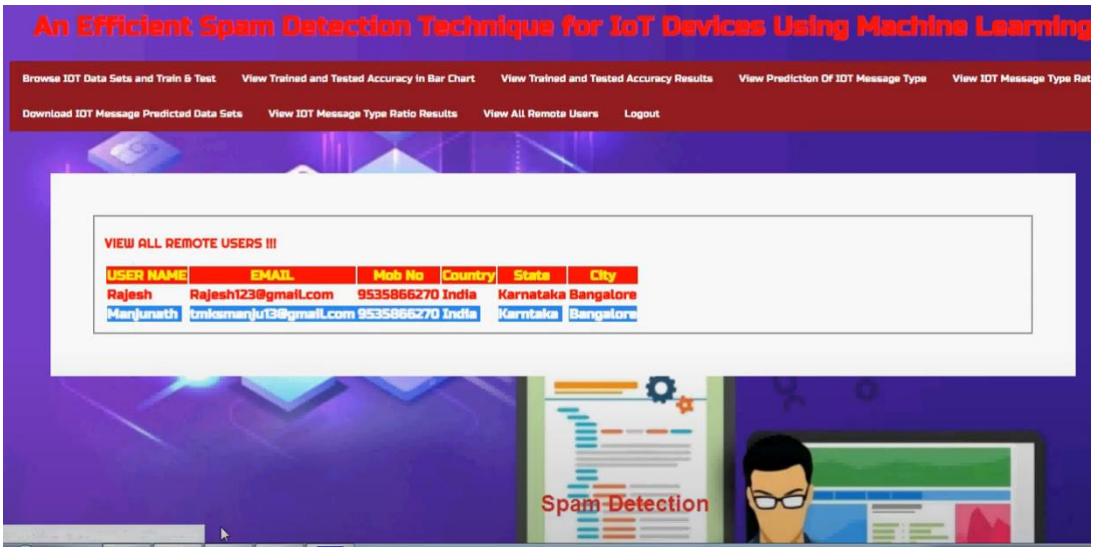
```
msg = ''
if request.method == 'POST':
  name = request.POST.get('t1', False)
  stop_words = set(stopwords.words('english'))
  corpus.clear()
  posts = []
  df = pd.read_csv('dataset.txt')
  X = df.iloc[:, :-1].values
  Y = df.iloc[:, -1].values
  for i in range(len(Y)):
     if Y[i] == 'Non-Bullying':
        Y[i] = 0
     else:
        Y[i] = 1

  for i in range(len(X)):
     line = str(X[i]).strip('\n')
     line = line.strip()
     line = line.lower()
     line = re.sub(r'[^a-zA-Z\s]+', '', line)
     arr = line.split(" ")
     for k in range(len(arr)):
        word = arr[k].strip("\n").strip()
        if len(word) > 2 and word not in corpus and word not in stop_words:
           corpus.append(word)
```

```
    posts.append(line)
    dataset = ''
for k in range(len(corpus)):
    dataset+=corpus[k]+","
dataset+='Label\n'

for k in range(len(posts)):
    text = posts[k];
    wordCount = word_count(text.strip())
    for j in range(len(corpus)):
        if corpus[j] in wordCount.keys():
            dataset+=str(wordCount[corpus[j]])+","
        else:
            dataset+="0,"
    dataset+=str(Y[k])+"\n"

f = open("features.txt", "w")
f.write(dataset)
f.close()

train = pd.read_csv("features.txt")
cols = train.shape[1]
features = cols - 2
label = cols - 1
label_count = label
```

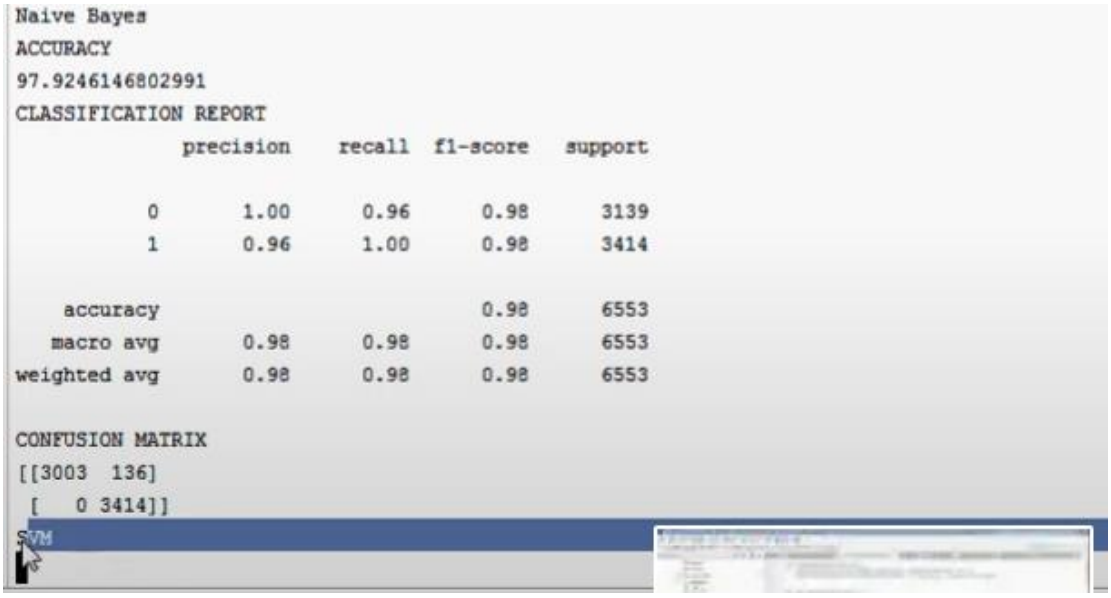
```
    X = train.values[:, 0:label]
    Y = train.values[:, label]
    print(Y)
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2,
random_state = 0)


    output = ""
    if name == "SVM Algorithm":
        cls = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random_state = 2)
        cls.fit(X_train, y_train)
        prediction_data = prediction(X_test, cls)
        msg = cal_accuracy(y_test, prediction_data,'SVM Accuracy')
        output = 'SVM Algorithm Output details<br/><br/>'+msg
        classifier = cls

    if name == "Decision Tree":
        cls = DecisionTreeClassifier(max_depth=None,
min_samples_split=2,random_state=0)
        cls.fit(X_train, y_train)
        prediction_data = prediction(X_test, cls)
        msg = cal_accuracy(y_test, prediction_data,'Decision Tree Accuracy')
        output = 'Decision Tree Algorithm Output details<br/><br/>'+msg

    if name == "KNearest Neighbors":
```

```
        cls = BaggingClassifier(KNeighborsClassifier(),max_samples=0.5,
max_features=0.5)
        cls.fit(X_train, y_train)
        prediction_data = prediction(X_test, cls)
        msg = cal_accuracy(y_test, prediction_data,'KNearest Neighbor Accuracy')
        output = 'KNearest Neighbor Algorithm Output details<br/><br/>'+msg

    if name == "Random Forest":
        cls =
RandomForestClassifier(n_estimators=1,max_depth=0.9,random_state=None)
        cls.fit(X_train, y_train)
        prediction_data = prediction(X_test, cls)
        msg = cal_accuracy(y_test, prediction_data,'Random Forest Accuracy')
        output = 'Random Forest Algorithm Output details<br/><br/>'+msg

    if name == "Naive Bayes":
        cls = MultinomialNB()
        cls.fit(X_train, y_train)
        prediction_data = prediction(X_test, cls)
        msg = cal_accuracy(y_test, prediction_data,'Naive Bayes Accuracy')
        output = 'Naive Bayes Algorithm Output details<br/><br/>'+msg


    context= {'data':"+output}
    return render(request, 'RunAlgorithms.html', context)
```

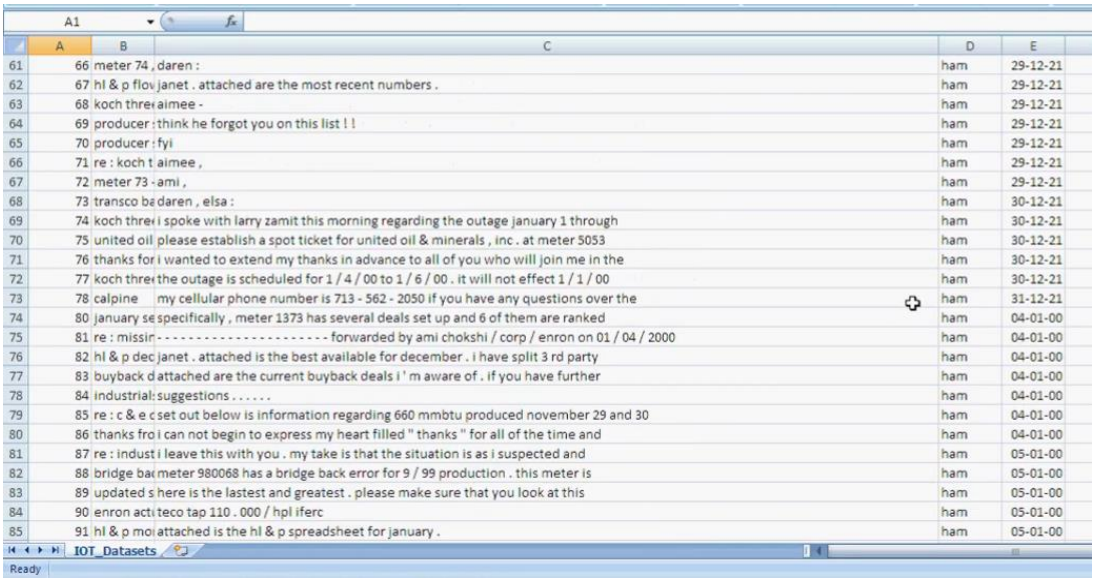
SCREENS/ FORMS



Screen 1: In above screen user adding signup details and then click on 'Register' button to get below screen



Screen 2: In above screen user signup completed and now click on 'User' link to get below login screen

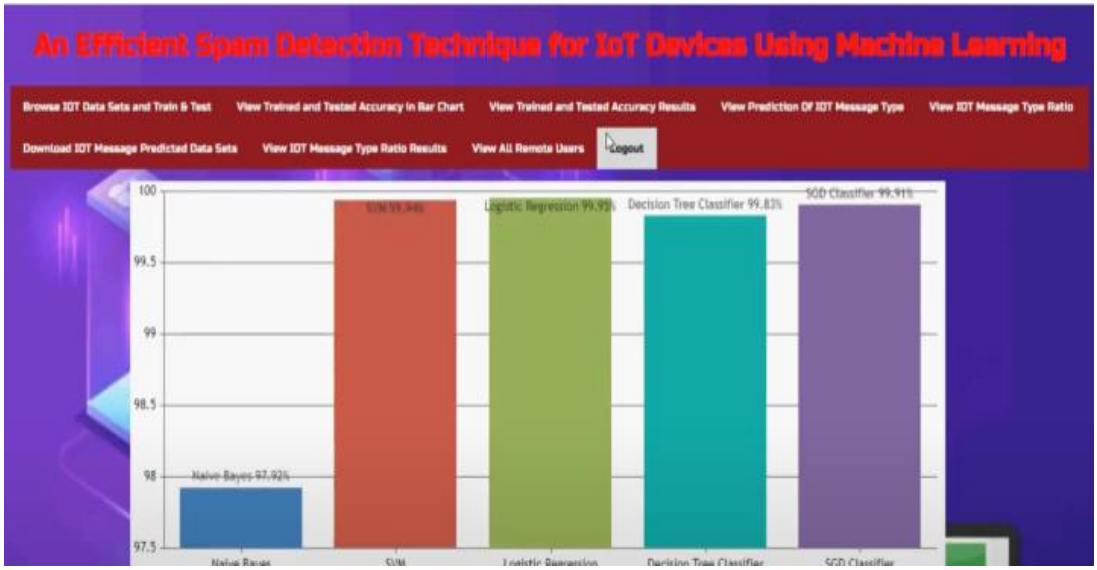


	A	B	C	D	E
61	66	meter 74 ,	daren :	ham	29-12-21
62	67	hl & p flo	janet . attached are the most recent numbers .	ham	29-12-21
63	68	koch thre	aimee -	ham	29-12-21
64	69	producer :	think he forgot you on this list ! !	ham	29-12-21
65	70	producer :	fyi	ham	29-12-21
66	71	re : koch t	aimee ,	ham	29-12-21
67	72	meter 73 -	ami ,	ham	29-12-21
68	73	transco ba	daren , elsa :	ham	30-12-21
69	74	koch thre	i spoke with larry zamit this morning regarding the outage january 1 through	ham	30-12-21
70	75	united oil	please establish a spot ticket for united oil & minerals , inc . at meter 5053	ham	30-12-21
71	76	thanks for	i wanted to extend my thanks in advance to all of you who will join me in the	ham	30-12-21
72	77	koch thre	the outage is scheduled for 1 / 4 / 00 to 1 / 6 / 00 . it will not effect 1 / 1 / 00	ham	30-12-21
73	78	calpine	my cellular phone number is 713 - 562 - 2050 if you have any questions over the	ham	31-12-21
74	80	january se	specifically , meter 1373 has several deals set up and 6 of them are ranked	ham	04-01-00
75	81	re : missir	- forwarded by ami chokshi / corp / enron on 01 / 04 / 2000	ham	04-01-00
76	82	hl & p dec	janet . attached is the best available for december . i have split 3 rd party	ham	04-01-00
77	83	buyback d	attached are the current buyback deals i ' m aware of . if you have further	ham	04-01-00
78	84	industrial:	suggestions	ham	04-01-00
79	85	re : c & e c	set out below is information regarding 660 mmbtu produced november 29 and 30	ham	04-01-00
80	86	thanks fro	i can not begin to express my heart filled " thanks " for all of the time and	ham	04-01-00
81	87	re : indust	i leave this with you . my take is that the situation is as i suspected and	ham	05-01-00
82	88	bridge ba	meter 980068 has a bridge back error for 9 / 99 production . this meter is	ham	05-01-00
83	89	updated s	here is the lastest and greatest . please make sure that you look at this	ham	05-01-00
84	90	enron acti	teco tap 110 . 000 / hpl iferc	ham	05-01-00
85	91	hl & p moi	attached is the hl & p spreadsheet for january .	ham	05-01-00

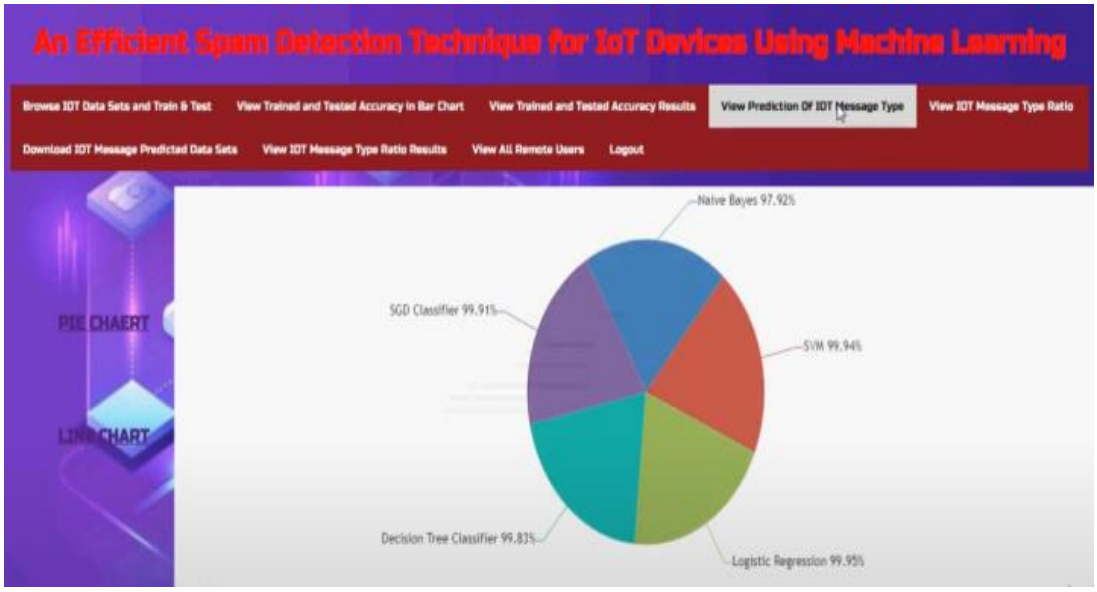
IOT_Datasets

Ready

Screen 3: In above screen click on 'Register Here' link to get below signup screen



Screen 4: Chart Showing Accuracy values using Various ML Methods



`

Screen 5: Chart Showing Accuracy values using Various ML Methods

REFERENCES

[1] Fatima Hussain,Rasheed Hussain,Syed Ali HassanHossain. Machine Learning in IoT Security: Current Solutions and Future Challenges

[2] Choi, J.; Jeoung, H.; Kim, J.; Ko, Y.; Jung, W.; Kim, H.; Kim, J. Detecting and identifying faulty IoT devices in smart homes with context extraction. In Proceedings of
The 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018, Luxembourg, 25– 28 June 2018; pp. 610–621.

[3] Tang, S.; Gu, Z.; Yang, Q.; Fu, S. Smart Home IoT Anomaly Detection based on Ensemble Model Learning from Heterogeneous Data. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 4185–4190.

[4] Makkar A.; Garg S.; Kumar, N.; Hossain, M.S.; Ghoneim, A.; Alrashoud, M. An Efficient Spam Detection Technique for IoT Devices using Machine Learning. IEEE Trans. Ind. Inform. 2020.

[5] Ameema Zainab, Shady S. Refaat and Othmane Bouhali;Ensemble-Based Spam Detection in Smart Home IoT Devices Time Series Data Using Machine Learning Techniques

[6] L. University, “Refit smart home dataset,” https://repository.lboro.ac.uk/ articles/REFIT Smart Home dataset/2070091, 2019 (accessed April 26, 2019