

# **CHAPTER 1**

## **PREAMBLE**

### **1.1 Introduction**

With the rapid development of the web, more and more services like internet banking, e-commerce, social networking, shopping, making a bill payment, e-learning, etc. are available to users and they are surfing the internet via browsers or web application. As the browsers are come up with different advanced features and functionalities which leads to risk by losing their personal and sensitive information.

As the naive users are not aware of the different malware so they are easily trapped by the intruder by just a single click on the malicious web sites which allows the invaders to detect the vulnerabilities on the web page and inject the payloads to get remote access to victim's web page. Therefore, the precise identification of web pages in an ever-growing web environment is very important. Blacklisting services were embedded in the browsers to face the challenges but it has several disadvantages like incorrect listing. In this article, we explore a self-learning approach to classify the web page based on a small feature set. We use four machine learning classifiers to classify the web site into two classes benign and malicious web pages.

To identify the web pages that are malicious, three different techniques i.e., blacklisting, static analysis, and dynamic analysis are suggested by research practitioners. Each approach has some objective to satisfy and we have discussed some of these techniques sequentially. Tao et al. presented a novel framework for detecting the web page as malicious or benign automatically using supervised machine learning approaches. The web pages were distinguished as malicious or not based upon features. Benign web pages were collected from dataset.

Internet surfing has become a vital part of our daily life. So, to catch the attention of the users' different browser vendors compete to set up the new functionality and advanced features that become the source of attacks for the intruder and the websites are put at hazard. However, the existing approaches are not adequate to protect the surfers which require an expeditious and precise model that can be able to distinguish between the benign or malicious webpages. In this, we design a new classification system to analyze and detect the malicious web pages using machine learning

classifiers such as, random forest, support vector machine, naive Bayes, logistic regression and Some special URL (Uniform Resource Locator) based on extricated features the classifiers are trained to predict the malicious web pages.

### 1.2 Existing System

In the existing methods typically detect malicious URLs of a single attack type and acquiring perfectly balanced and highly related dataset is almost impossible. It can work with small number of datasets when the dataset is huge it fails to extract the data and whenever the number of parameters is more this system fails to predict the correct output. Although large quantities of data are available but still extracting relevant data is a complex job. In this, we propose method using machine learning to detect malicious URLs of all the popular attack types and identify the nature of attack a malicious URL attempt. To overcome all this, we use machine learning algorithms to detect harmful URLs.

#### **Disadvantages of Existing System:**

- Accuracy low
- Operating cost is high
- Difficult to handle

### 1.3 Proposed System

Our method uses a variety of discriminative features including textual properties, link structures, webpage contents, DNS information, and network traffic. Many of these features are novel and highly effective. Our experimental studies with 40,000 benign URLs and 32,000 malicious URLs obtained from real-life Internet sources show that our method delivers a superior performance: the accuracy was over 98% in detecting malicious URLs and over 93% in identifying attack types. We also report our studies on the effectiveness of each group of discriminative features, and discuss their evitability.

#### **Advantages:**

- accuracy is maximum
- prediction is accurate

## **1.4 Plan of Implementation**

The implementation of the project is based on detection of hazard webpages using machine learning algorithms. We have divided into two be two classes like benign or malicious webpages. We have used some classifiers are random forest, support vector machine, naive Bayes and logistic regression. The implementation of this project is based on Machine Learning algorithms like Support Vector Machine (SVM) because machine learning models can be used to predict the result in less amount of time. Support Vector Machine (SVM) is one of the prominent algorithms that can be used for classification purpose. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts  $P(Y=1)$  as a function of  $X$ . Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. Bayes Theorem provides a way that we can calculate the probability of a piece of data belonging to a given class, given our prior knowledge. The language we have used for the development of the project is Python due to it is more supportable for developing the Machine Learning model due to its built-in libraries. The development of the project has done using PyCharm tool and uses high-level Python web framework that enables rapid development of secure and maintainable websites. This project consists of many files regarding to the detection of hazard web pages. The completion time for the project is approximately 4 to 6 weeks.

## **1.5 Problem Statement**

Detecting the harmful webpages by using a variety of discriminative features including textual properties, link structures, webpage contents, DNS information, and network traffic. Many of these features are novel and highly effective. Predicting if a given website is malicious or benign using machine learning classifiers are logistic regression, random forest, naive Bayes and SVM.

## **1.6 Objective**

The main objectives of the project:

- To detect harmful webpages.
- Gain knowledge about Machine learning models.
- To know about the harmful and harmless webpages.
- Dividing the webpages into two classes like malicious and benign.
- Building the own machine learning model by using logistic regression, random forest, naive Bayes and SVM.
- Training the dataset using the classifiers which can be capable of dividing the classes into two classes like malicious and benign.

## CHAPTER 2

### LITERATURE SURVEY

#### **1. Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection.**

**Authors:** Altay, Betel, Tansel Dokeroglu, and Ahmet Cosar.

Conventional malicious webpage detection methods use blacklists in order to decide whether a webpage is malicious or not. The blacklists are generally maintained by third-party organizations. However, keeping a list of all malicious Web sites and updating this list regularly is not an easy task for the frequently changing and rapidly growing number of webpages on the web. In this study, we propose a novel context-sensitive and keyword density-based method for the classification of webpages by using three supervised machine learning techniques, support vector machine, maximum entropy, and extreme learning machine. Features (words) of webpages are obtained from HTML contents and information is extracted by using feature extraction methods: existence of words, keyword frequencies, and keyword density techniques. The performance of proposed machine learning models is evaluated by using a benchmark data set which consists of one hundred thousand webpages. Experimental results show that the proposed method can detect malicious webpages with an accuracy of 98.24%, which is a significant improvement compared to state-of-the-art approaches. We propose Maxent because of its success on document classification and it has not been implemented for any malicious webpage detection study until now. ELM provides faster learning speed and less human intervention than SVM. We study with ELM because of its learning speed with 100 thousand webpages and 800 thousand features. By increasing 2 the efforts on data processing phase, we are able to increase the accuracy level of detecting the malicious webpages up to 98.28% true positive ratio. Even the most successful recent studies provide 97.8% true positive ratio with 2.2% false positive.

## **2. WebMon: ML-and YARA-based malicious webpage detection**

**Authors:** Kim, Sungjin, Jinkook Kim, Seokwoo Nam, and Dohoon Kim.

Attackers use the openness of the Internet to facilitate the dissemination of malware. Their attempts to infect target systems via the Web have increased with time and are unlikely to abate. In response to this threat, we present an automated, low-interaction malicious webpage detector, WebMon, that identifies invasive roots in Web resources loaded from WebKit2-based browsers using machine learning and YARA signatures. WebMon effectively detects hidden exploit codes by tracing linked URLs to confirm whether the relevant websites are malicious. WebMon detects a variety of attacks by running 250 containers simultaneously. In this configuration, the proposed model yields a detection rate of 98%, and is 7.6 times faster (with a container) than previously proposed models. Most importantly, Web Mon's focus on extracting malicious paths in a domain is a novel approach that has not been explored in previous studies. Since the initial development of Web browsers, there have been a growing number of attempts to infect online systems by transmitting malware through browsers. In this Web architecture, one malicious webpage can contaminate several thousand user PCs in minutes. Therefore, the concealment of malware on webpages is one of the most dangerous types of cyberattack, and poses a significant threat to the integrity of critical systems. More recent types of malwares, such as ransomware, in conjunction with exploit toolkits, have evolved to become more complex, automated, and impossible to decrypt. Thus, detecting websites that propagate malware and developing techniques to neutralize them is crucial. Malicious URLs that activate drive-by downloads are a popular form of exploitation and malware delivery. Hence, fast detection of malicious URLs is useful, because the URLs can then be distributed to blacklists maintained by various security systems. (At present, these databases might contain a great deal of outdated information.) Thus, rapidly finding malicious URLs from countless webpages, which are live only for very limited amounts of time, is a duty of security research.

### **3. Detection of malicious web pages based on hybrid analysis. Journal of Information Security and Applications**

**Authors:** Wang, Rong, Yan Zhu, Jiefan Tan, and Binbin Zhou.

Malicious web pages have become an increasingly serious threat to web security in recent years. In this paper, we propose a new detection method that consists of static and dynamic analyses for detecting malicious web pages. Static analysis utilizes classification algorithms in machine learning to identify certain benign and malicious web pages. As a complement to static analysis, dynamic analysis mainly checks the unknown web pages to determine whether they have malicious shellcodes during their execution. Because of the combination of static and dynamic analyses, the proposed detection method achieves high performance, and it has a light weight and is simple to use. With the rapid development of the web, more and more services like internet banking, e-commerce, social networking, shopping, making a bill payment, e-learning, etc. are available to users and they are surfing the internet via browsers or web application. As the browsers are come up with different advanced features and functionalities which leads to risk by losing their personal and sensitive information. As the naïve users are not aware of the different malware so they are easily trapped by the intruder by just a single click on the malicious web sites which allows the invaders to detect the vulnerabilities on the web page and inject the payloads to get remote access to victim's web page. Therefore, the precise identification of web pages in an ever-growing web environment is very important. In this article, we explore a self-learning approach to classify the web page based on a small feature set. We use four machine learning classifiers to classify the web site into two classes benign and malicious web pages. Internet surfing has become a vital part of our daily life. So, to catch the attention of the users' different browser vendors compete to set up the new functionality and advanced features that become the source of attacks for the intruder and the websites are put at hazard. However, the existing approaches are not adequate to protect the surfers which require an expeditious and precise model that can be able to distinguish between the benign or malicious webpages. In this research article, we design a new classification system to analyze and detect the malicious web pages using machine learning classifiers such as, random forest, support vector machine. naïve Bayes, logistic regression and Some special URL (Uniform Resource Locator) based on extricated features the classifiers are trained to predict the malicious web pages. The experimental results have shown that the performance of the random forest classifier achieves better accuracy of 95% in comparison to other machine learning classifiers.

#### **4. Machine Learning Classifiers to Detect Malicious Websites.**

**Authors:** Urcuqui, Christian, Andres Navarro, Jose Osorio, and Melisa Garcia.

A risk that exists in Internet is the access of websites with malicious content, because they might be open doors for cybercrimes or be the mechanism to download files in order to affect organizations, persons and the environment. What is more, the attack registers through websites have been part of cyberattacks reports during the last years; this information includes attacks made by the currently risks found in new technologies, such as the IoT. Due the computer security complexity, studies have been working in to use machine learning algorithms to identify web malicious content. This article explores the application of a data analysis process through a framework that includes dynamic, static analysis, updated websites and a low interaction client honeypot in order to classify a website. Furthermore, it evaluates the capacity of the classification of four machine learning through the information analyzed. Several studies have been performed previously that use various aspects of websites to detect whether it contains malicious content. In this the authors used machine learning to detect malware in websites using IP address features with an argument that IP address is a much more stable feature of websites as compared to URL and DNS. The goal of their study was to develop a powerful technique to compensate the previous techniques. They used octet, extended octet and bit string-based features to classify websites. The authors argue that malicious websites have a distinct signature for all three feature systems and can be used to differentiate them from benign/safe websites. While the results from this study showed promising accuracy and was able to detect even unknown malicious websites, a limitation was that it misclassified many benign websites as malicious simply because they were hosted on the same hosting service. This is done by analyzing the software that is downloaded in result of opening the website and whether the software tries to access sensitive information or not. “Using website URL features such as textual properties, link structures, webpage contents, DNS information, and network traffic detects malicious websites exploiting state-of-the-art machine learning algorithms”. The authors state that most traditional methods are aimed at targeting a single type of malicious attack, while their approach not only detect a malicious attack but also identifies its type. Their major focus is to increase the quality of features extraction techniques. In summary, there have been a variety of different approaches that have been tried in the literature. Each study has its own set of limitations and one thing that has been common across all of them is that they



use fairly simple machine learning models and their feature set doesn't contain demographic information such as country of registration and dates of registration

### **5. Two-phase malicious web page detection scheme using misuse and anomaly detection.**

**Authors:** Yoo, Suyeon, Sehun Kim, Anil Choudhary, O. P. Roy, and T. Tuithung.

Misuse detection method and anomaly detection method are widely used for the detection of malicious web pages. Both are based on machine learning. Misuse detection can detect known malicious web pages, but it cannot detect new ones. In contrast, anomaly detection can detect unknown malicious web pages, but it has a high false positive rate. In order to achieve a high detection rate through precisely detecting known and unknown malicious web pages, we propose a two-phase detection scheme. In the first phase, the misuse detection model is built based on the C4.5 decision tree algorithm, which allows known malicious web pages to be detected. In the second phase, the anomaly detection model with a one-class support vector machine is used to detect new types of malicious web pages. The experimental results show that our proposed method has significantly higher malicious web page detection rate than conventional ones with the expense of slightly higher false positive rate.

## **CHAPTER 3**

### **THEORETICAL BACKGROUND**

#### **3.1 Overview of domain description**

##### **Machine Learning:**

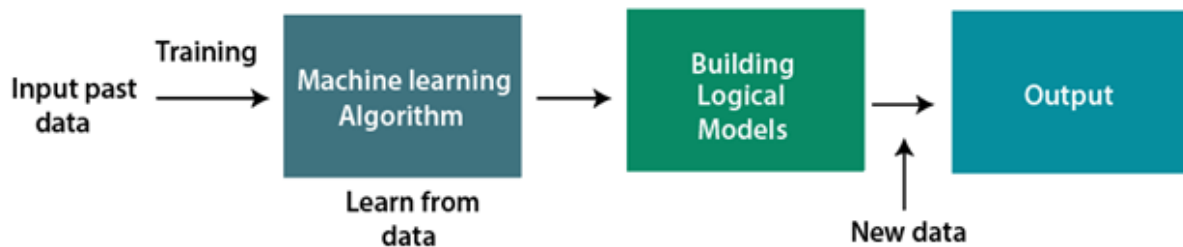
The use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyze and draw inferences from patterns in data. The term machine learning was first introduced by Arthur Samuel in 1959. Machine learning is a branch of artificial intelligence and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is more dependent on human intervention to learn. Human experts determine the set of features to understand the differences between data inputs, usually requiring more structured data to learn. It is focused on teaching computers to learn from data and to improve with experience instead of being explicitly programmed to do so. In machine learning, algorithms are trained to find patterns and correlations in large data sets and to make the best decisions and predictions based on that analysis. Machine learning applications improve with use and become more accurate the more data they have access to. Applications of machine learning are all around us in our homes, our shopping carts, our entertainment media, and our healthcare. A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem.

The machine learning process includes the following steps.

1. Identify relevant datasets and prepares them for analysis.
2. Choose the kind of machine learning algorithms to be used.

3. Builds an analytical model based on chosen algorithm.
4. Trains the model on test datasets, revising it as needed.
5. Runs the model to generate the results.



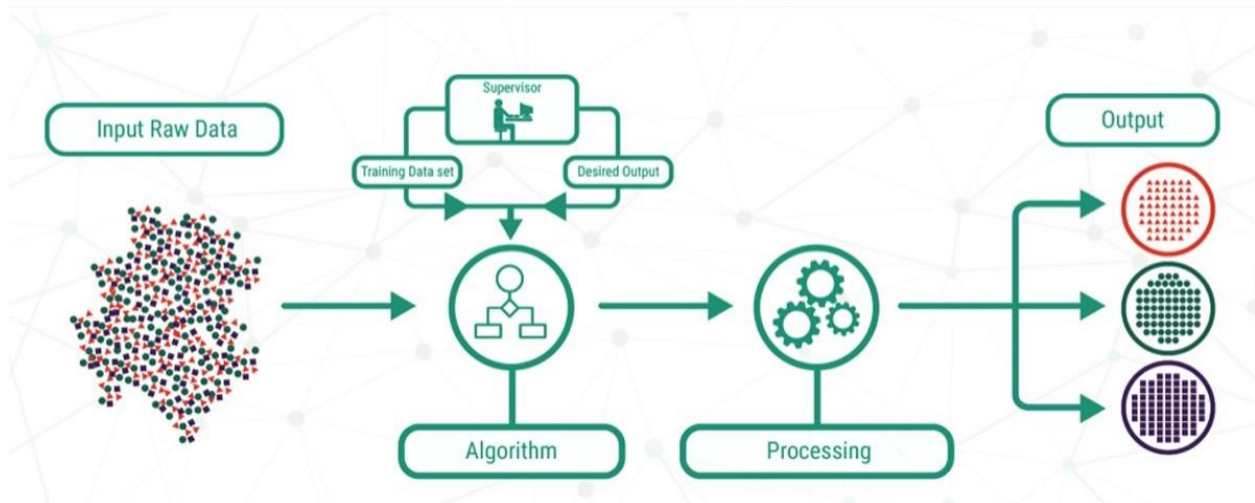
**Fig 3.1:** Process of Machine Learning

At a broad level, machine learning can be classified into three types:

- 1) Supervised learning
- 2) Unsupervised learning
- 3) Reinforcement learning

### **1) Supervised Learning:**

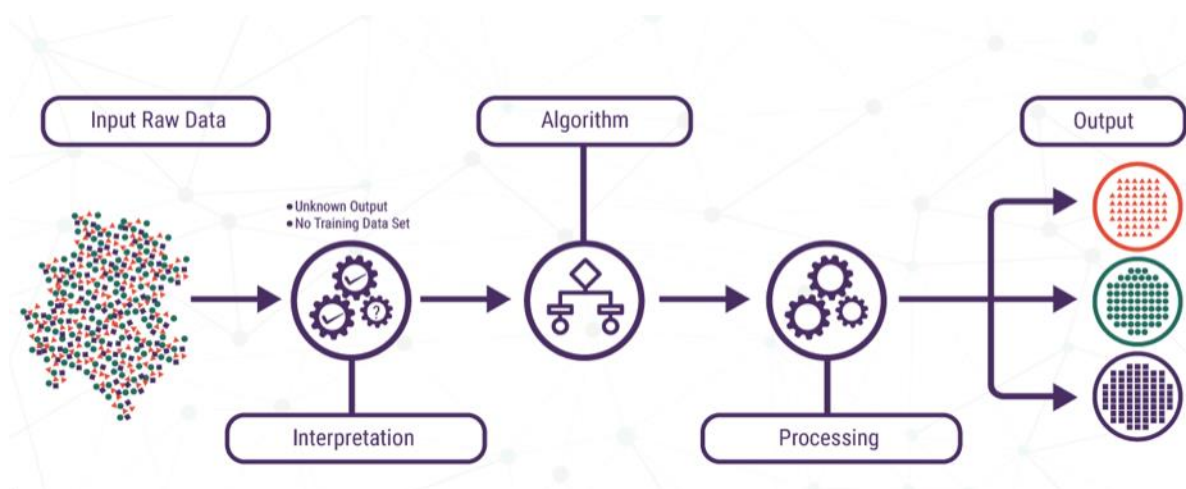
Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not. The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher.



**Fig 3.2:** Supervised Learning

## 2) Unsupervised Learning:

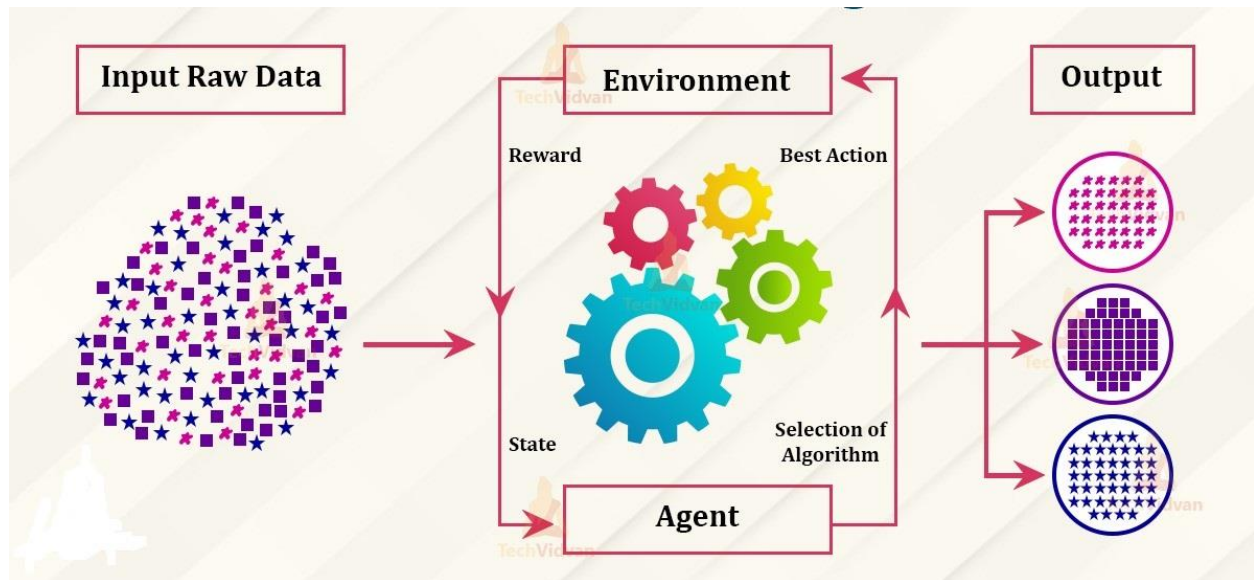
Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labelled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classified into two categories of



**Fig 3.3:** Unsupervised Learning

### 3) Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance. The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.



**Fig 3.4:** Reinforcement Learning

#### Features of Machine Learning:

1. Machine learning uses data to detect various patterns in a given dataset.
2. It can learn from past data and improve automatically.
3. It is a data-driven technology.
4. Machine learning is much similar to data mining as it also deals with the huge amount of the data.

## **3.2 Technology**

### **3.2.1 Python:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

### **History:**

- Python was conceived in the late 1980s by **Guido van Rossum** at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.
- Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker.
- There is a fact behind choosing the name Python. Guido van Rossum was reading the script of a popular BBC comedy series "Monty Python's Flying Circus". It was late on-air 1970s. Van Rossum wanted to select a name which unique, sort, and little-bit mysterious. So, he decided to select naming Python after the "Monty Python's Flying Circus" for their newly created programming language. The comedy series was creative and well random. It talks about everything. Thus, it is slow and unpredictable, which made it very interesting.

### **3.2.2 Features of Python Programming Language:**

#### **1. Easy**

When we say the word 'easy', we mean it in different contexts.

##### **a. Easy to code**

As we have seen in earlier lessons, Python is very easy to code. Compared to other popular languages like Java and C++, it is easier to code in Python. Anyone can learn Python syntax

in just a few hours. Through sure, mastering Python requires learning about all its advanced concepts and packages and modules. That takes time. Thus, it is programmer-friendly.

- b. Easy to read Being a high-level language, Python code is quite like English. Looking at it, you can tell what the code is supposed to do. Also, since it is dynamically-typed, it mandates indentation. This aids readability.

### **2. Expressive**

First, let's learn about expressiveness. Suppose we have two languages A and B, and all programs that can be made in A can be made in B using local transformations. However, there are some programs that can be made in B, but not in A, using local transformations. Then, B is said to be expressive than A. Python provides us with a myriad of constructs that help us focus on the solution rather than on the syntax. This is one of the outstanding python features that tells why you should learn Python.

### **3. Free and Open-source**

Firstly, Python is freely available, Secondly, it is open-source. This means that its source code is available to the public. You can download it, change it, use it, and distribute it. This is called FLOSS (Free/Libre and Open-Source Software). As the Python community, we're all headed toward one goal-an ever-bettering Python.

### **4. High-level**

It is a high-level language. This means that as programmers, we don't need to remember the system architecture. Nor we need to manage the memory. This makes it more programmer-friendly and is one of the key python features.

### **5. Portable**

Let's assume you've written a Python code for your windows machine. Now, if you want to run it on any machine, there is no need to write different code for different machines. This makes Python a portable language. However, you must avoid any system-dependent features in this case.

### **6. Interpreted**

If you're any familiar with languages like C++ or Java, you must first compile it, and then run it. But in Python, there is no need to compile it. Internally, its source code is converted.



## **7. Object-Oriented**

Python is known as Object Oriented programming language because it can model the real world. Python basically focuses on an object and combines functions and data. Oppositely a language which is more procedure-oriented revolves around the function that is coded and reused. In Python programming, both Procedure-oriented and Object-oriented language is supported. This is one more key feature of Python. Unlike Java, it also underpins various inheritance. A class is an abstract data type and holds no values. Class is a blueprint, for such objects.

## **8. Extensible**

Python can be extended to other languages which make it extensible language. This feature allows you to write some of the Python codes ion other languages C++ or Java.

## **9. Embeddable**

In the above feature extensible we got to know that other languages code can be used in Python source code. Now, Embeddable means we can also put our Python code in different languages source code, like C++. This enables the users to coordinate scripting capabilities into the program of other languages.

## **10. Large Standard Library**

Python has a large library inbuilt that can be used while you are coding a programme so that you don't have to write your own code for every single thing. In Python, the library implies a gathering of modules where you will locate all sort of stuff. Modules are records, much the same as books which contain functions that should be imported in your program. For example, on the off chance that you need to take a stab at something else for the graphics you can go to and check the Python Imaging Library. The Pygame of the Pyglet libraries can be utilized if you want to make a game. If you are among the enthusiastic science fans, you have a SciPy library.

## **11. GUI Programming**

In Python, you can create basic GUI's by using Tk.

## **12. Dynamically Typed**

Python is dynamically-typed. This implies the Type for esteem is chosen at runtime, not progress of time. This is the reason we don't have to determine the sort of information while pronouncing it.



## 3.3 Tools

### 3.3.1 PyCharm:

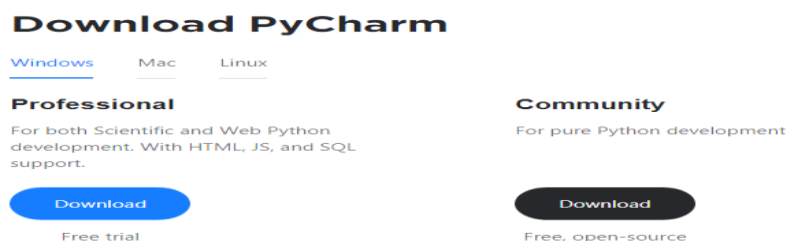
PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features released under a subscription-funded proprietary license and also an educational version.

#### Features of PyCharm:

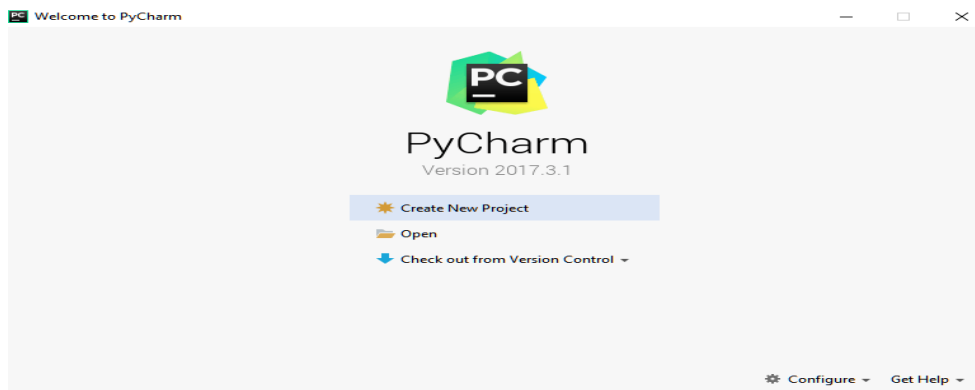
- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes.
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages.
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others.
- Support for web frameworks: Django, web2py and Flask.
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development

#### Installing PyCharm:

1. To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/> and click the "DOWNLOAD" link under the Community Section.



2. Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.
3. On the next screen, Change the installation path if required. Click “Next”.
4. On the next screen, you can create a desktop shortcut if you want and click on “Next”.
5. Choose the start menu folder. Keep selected JetBrains and click on “Install”.
6. Wait for the installation to finish.
7. Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.
8. After you click on "Finish," the Following screen will appear.



9. You need to install some packages to execute your project in a proper way.
10. Open the command prompt/ anaconda prompt or terminal as administrator.
11. The prompt will get open, with specified path, type “pip install package name” which you want to install (like NumPy, pandas, sea born, scikit-learn, Matplotlib, Pyplot)

Ex: Pip install NumPy

```
C:\WINDOWS\system32>pip install numpy==1.18.5
Collecting numpy==1.18.5
  Downloading numpy-1.18.5-cp36-cp36m-win_amd64.whl (12.7 MB)
    | 12.7 MB 939 kB/s
ERROR: tensorboard 2.0.2 has requirement setuptools>=41.0.0, but
Installing collected packages: numpy
Successfully installed numpy-1.18.5
```

### **3.4 Package Description**

#### **1. NUMPY:**

NumPy is a Python library used for working with arrays of any dimensions. Numpy also has functions for working in the domain of linear algebra and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely and many numerical functions can be done easily. NumPy stands for Numerical Python. In Python we have lists that serve the purpose of arrays, but they are slow in process and less reliable so we use numpy arrays to solve these problems. NumPy aims to provide an array (one or two dimensions) object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, which provides a lot of supporting functions that make working with ndarray very easy and quick. We have to install NUMPY before importing it by using the following command: `Pip install numpy`. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

#### **2. PANDAS:**

Pandas is one of the Python libraries used for working with data sets. It has functions for analyzing, cleaning, exploring, manipulating and many other functions that deals with data. The name "Pandas" which has reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in early 2008. Pandas allows the users to analyze big data and make conclusions based on statistical theories which make them easy to study and perform required operations. Pandas can clean messy data sets, and make them readable and relevant that majorly makes the use of pandas as relevant data is very important to a data scientist.

#### **3. PICKLE:**

Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is

called pickling or serialization or flattening or marshallng. We can convert the byte stream (generated through pickling) back into python objects by a process called as unpickling.

#### **4. SCIKIT-LEARN:**

Scikit-learn (Sklearn) is the most useful and robust libraries for machine learning and deep learning in Python. It provides a wide range and selection of efficient tools for machine learning, deep learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a continuous interface in Python. Tensorflow can also be an alternative for some metrics for scikit-learn in deep learning.

#### **5. LOGISTIC REGRESSION:**

Logistic regression is a fundamental classification technique. It belongs to the group of linear classifiers and is somewhat similar to polynomial and linear regression. Logistic regression is fast and relatively uncomplicated, and it's convenient for you to interpret the results. Although it's essentially a method for binary classification, it can also be applied to multiclass problems.

#### **6. SCIPY:**

SciPy is a free and open-source Python library used for scientific computing and technical computing. It is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data. As mentioned earlier, SciPy builds on NumPy and therefore if you import SciPy, there is no need to import NumPy.

#### **7. PILLOW:**

It is a fork of PIL (Python Image Library) that developed into an easy-to-use and efficient tool for image manipulation in Python.

With the use of pillow, we can

1. Open and save images of different file types (JPEG, PNG, GIF, PDF, etc.).
2. Create thumbnails for images.

3. Use a collection of image filters (e.g. SMOOTH, BLUR, SHARPEN).

## **8. MATPLOTLIB:**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt or GTK. Matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. In matplotlib.pyplot various states are preserved across function calls, so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes (please note that "axes" here and in most places in the documentation refers to the axes part of a figure and not the strict mathematical term for more than one axis).

## **9. RANDOM FOREST CLASSIFIER:**

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. The sub-sample size is controlled with the max sample parameter if bootstrap == True (default), otherwise the whole dataset is used to build each tree.

## **10. HTTPRESPONSE:**

The http or Hyper Text Transfer Protocol works on client server model. Usually, the web browser is the client and the computer hosting the website is the server. Upon receiving a request from client, the server generates a response and sends it back to the client in certain format. It will return data in the form of HTML.

## **12. NAVIE BAYES ALGORITHM:**

Bayes' Theorem provides a way that we can calculate the probability of a piece of data belonging to a given class, given our prior knowledge. Bayes' Theorem is stated as:

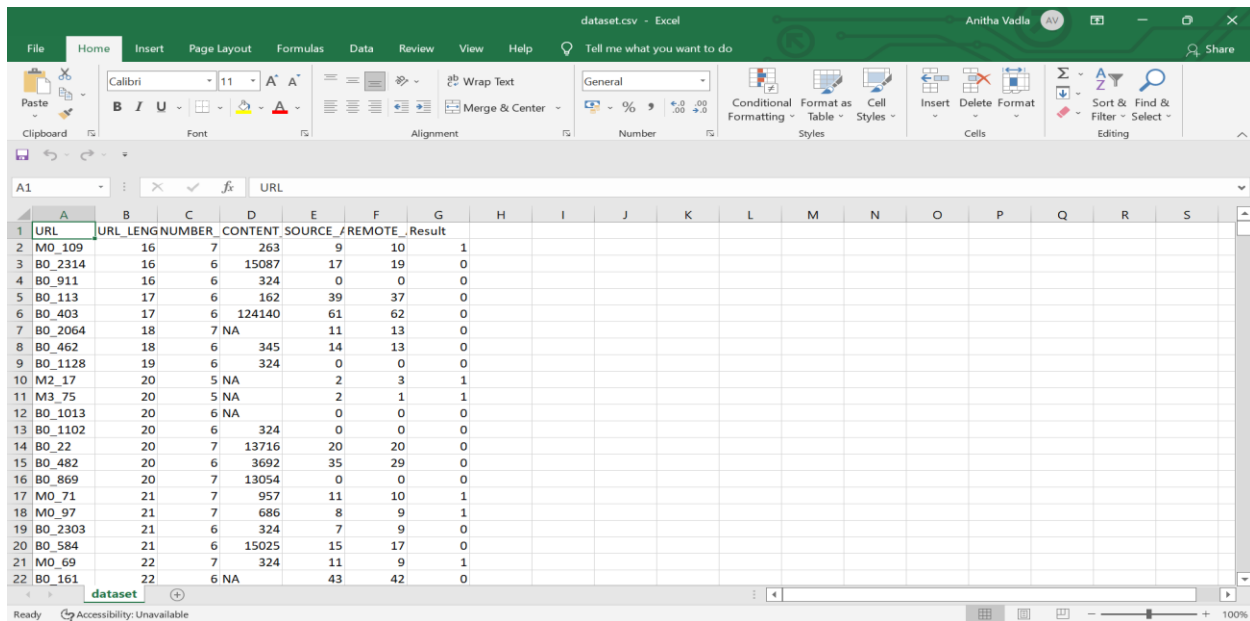
$$P(\text{class data}) = (P(\text{data class}) * P(\text{class})) / P(\text{data})$$

### 13. OS:

It is possible to automatically perform many operating system tasks. The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

### 3.5 Datasets

The Project consists of statistical dataset i.e., .CSV file which contains some parameters as features. This is a .CSV file which consists of the values such as URL, URL\_length, number\_special\_characters, content\_length, source\_app\_packets, remote\_app\_packets, Result. This proposed dataset used to detect the harmful webpages. Here by using machine learning classifiers we divide the datasets into two classes like malicious and benign. We will detect whether the webpage is harmful or not by using the below features.



URL	URL_LENGTH	NUMBER	CONTENT	SOURCE	REMOTE	Result
MO_109	16	7	263	9	10	1
BO_2314	16	6	15087	17	19	0
BO_911	16	6	324	0	0	0
BO_113	17	6	162	39	37	0
BO_403	17	6	124140	61	62	0
BO_2064	18	7	NA	11	13	0
BO_462	18	6	345	14	13	0
BO_1128	19	6	324	0	0	0
M2_17	20	5	NA	2	3	1
M3_75	20	5	NA	2	1	1
BO_1013	20	6	NA	0	0	0
BO_1102	20	6	324	0	0	0
BO_22	20	7	13716	20	20	0
BO_482	20	6	3692	35	29	0
BO_869	20	7	13054	0	0	0
MO_71	21	7	957	11	10	1
MO_97	21	7	686	8	9	1
BO_2303	21	6	324	7	9	0
BO_584	21	6	15025	15	17	0
MO_69	22	7	324	11	9	1
BO_161	22	6	NA	43	42	0

**Fig 3.5:** Proposed Dataset

Based on the classification algorithms and by considering the below features the it divided into two classes one is 0-malicious and 1- benign the result is obtained.

### **3.6 Data Pre-processing**

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data preprocessing methods. These data preprocessing methods may be applied to the data to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

There are few methods for the data preprocessing namely:

**1. Data Cleaning:** Data cleaning can be applied to remove noise and correct inconsistencies in the data. Data Cleaning means the process of identifying the incorrect, incomplete, inaccurate, irrelevant or missing part of the data and then modifying, replacing or deleting them according to the necessity. Data cleaning refers to identifying and correcting errors in the dataset that may negatively impact a predictive model. Data cleaning is used to refer to all kinds of tasks and activities to detect and repair errors in the data. It is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

**2. Data Integration:** The process of combining multiple sources into a single dataset. The Data integration process is one of the main components in data management.

**3. Data Reduction:** This process helps in the reduction of the volume of the data which makes the analysis easier yet produces the same or almost the same result. This reduction also helps to reduce storage space. There are some of the techniques in data reduction are Dimensionality reduction, Numerosity reduction, Data compression.

- **Dimensionality reduction:** This process is necessary for real-world applications as the data size is big. In this process, the reduction of random variables or attributes is done so that the dimensionality of the data set can be reduced. Combining and merging the attributes of the data without losing its original characteristics. This also helps in the

reduction of storage space and computation time is reduced. When the data is highly dimensional the problem called “Curse of Dimensionality” occurs.

- **Numerosity Reduction:** In this method, the representation of the data is made smaller by reducing the volume. There will not be any loss of data in this reduction.
- **Data compression:** The compressed form of data is called data compression. This compression can be lossless or lossy. When there is no loss of information during compression it is called lossless compression. Whereas lossy compression reduces information but it removes only the unnecessary information.

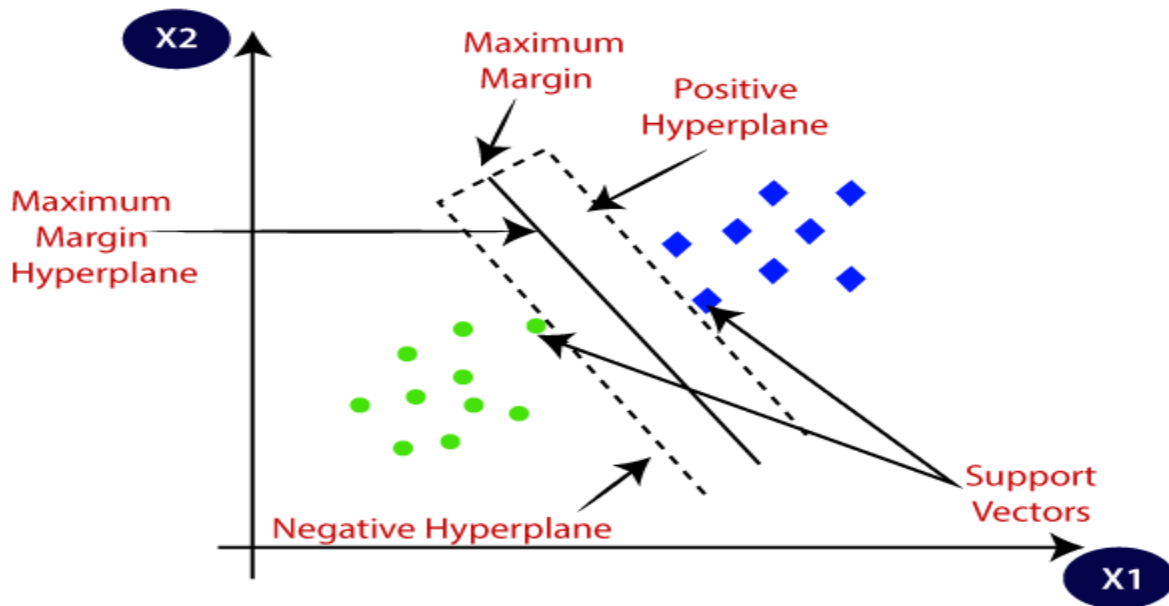
### **3.7 List of algorithms**

#### **3.7.1 Support Vector Machine (SVM):**

Support-vector machines are supervised learning models using learning algorithms that evaluate data for classification and regression analysis in machine learning. SVMs, which are based on statistical learning frameworks, are one of the most reliable prediction approaches. An SVM training algorithm creates a model that assigns new examples to one of two categories, making it a non-probabilistic binary linear classifier, given a series of training examples, each marked as belonging to one of two categories. SVM maps training examples to points in space in order to widen the distance between the two categories as much as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. A support-vector machine, in more technical terms, creates a hyper plane or set of hyper planes in a high- or infinite-dimensional space that can be used for classification, regression, or other tasks such as outlier detection. Intuitively, the hyper plane with the greatest distance to the nearest training-data point of any class (so-called functional margin) achieves a decent separation, because the larger the margin, the lower the classifier's generalization error. Even if the initial problem is expressed in a finite-dimensional space, the sets to discriminate are frequently not linearly separable in that space. As a result, it was suggested that the original finite-dimensional space be transferred into a much higher-dimensional region, purportedly making separation easier there. The mappings used by SVM schemes are designed to ensure that dot products of pairs of input data vectors can be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function. The goal of the SVM algorithm is to create the best line or



decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a **hyperplane**. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.



**Fig 3.6:** Classification using SVM

There are 2 types of svm:

**Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

**Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

The Support Vector Machine is mainly used for classification purposes, particularly those such as classifying objects from unseen data samples. In this project SVM is used to test various algorithms and determine whether each one has a security level of strong, acceptable, weak. This purpose

requires several inputs that can be treated as features or feature vectors. Suppose a series of samples consists of  $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3) \dots (X_n, Y_n)$ , in which  $X_i$  signifies the inputs and  $Y_i$  signifies the output. The dimensions of the data depend upon the number of features, as demonstrated below:

For 2-D dataset:  $Y = (X_1, X_2)$

For 3-D dataset:  $Y = (X_1, X_2, X_3)$

For n-D dataset:  $Y = (X_1, X_2, X_3 \dots X_n)$

where  $X_1$  and  $X_2$  are two independent features on the basis of which SVM classifies the output labels ( $Y_i$ ). For a dataset, it is not necessary that the number of features and the number of classes are equal. Instead, the number of classes may vary according to the required output. In the case of a two-dimensional dataset, a line (support vector) is required to separate the data with maximum margins. That margin between the data points represents the maximum distance between the closest data points. In the case of a higher-dimensional dataset, though, a plane may be used to separate the data instead of a line. As the data used in this work is seven dimensional (7-D), which means seven different features are used to predict the final output label, we are required to find the best plane through which to classify the data with a minimum rate of error. We can define the classification function as follows:

$$\mathbf{F(x)} = \mathbf{S.X + B}$$

Where  $S$  is the weight vector and  $B$  is the intercept. Weight ( $S$ ) can be defined as:

$$\mathbf{S = \frac{x_f - x_p}{y_f - y_p}}$$

For the linearly separable structure, all the input points should be classified according to equation to maximize the margin, a hyperplane is used, here the margin is signifying the distance from the hyperplane to the nearest data points. To achieve the maximum margin, the factor ‘ $w$ ’ should be minimum. This equation can be written as:

$$\mathbf{Max_{mar} = 1/|w|}$$

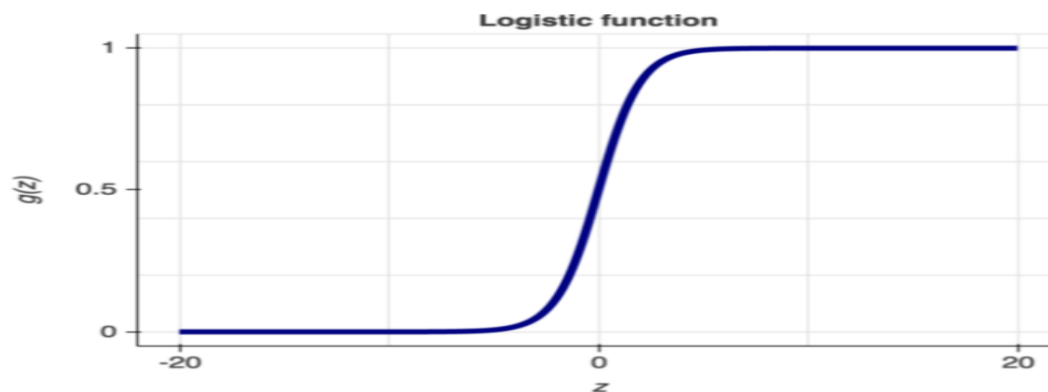
### 3.7.2 Logistic regression:

Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts  $P(Y=1)$  as a function of  $X$ .

#### Step1: Logistic regression hypothesis

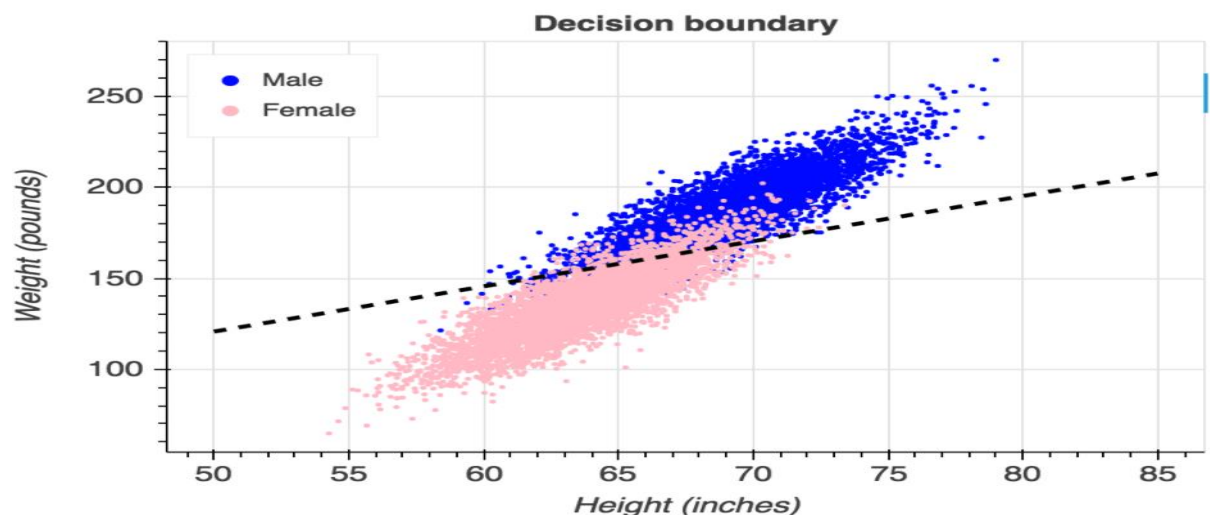
The logistic regression classifier can be derived by analogy to the logistic *regression* the function  $g(z)$  is the logistic function also known as the *sigmoid function*.

The logistic function has asymptotes at 0 and 1, and it crosses the y-axis at 0.5.



#### Step (1b): Logistic regression decision boundary

Since our data set has two features: height and weight, the logistic regression hypothesis is the following:



### 3.7.3 Random Forest classifier:

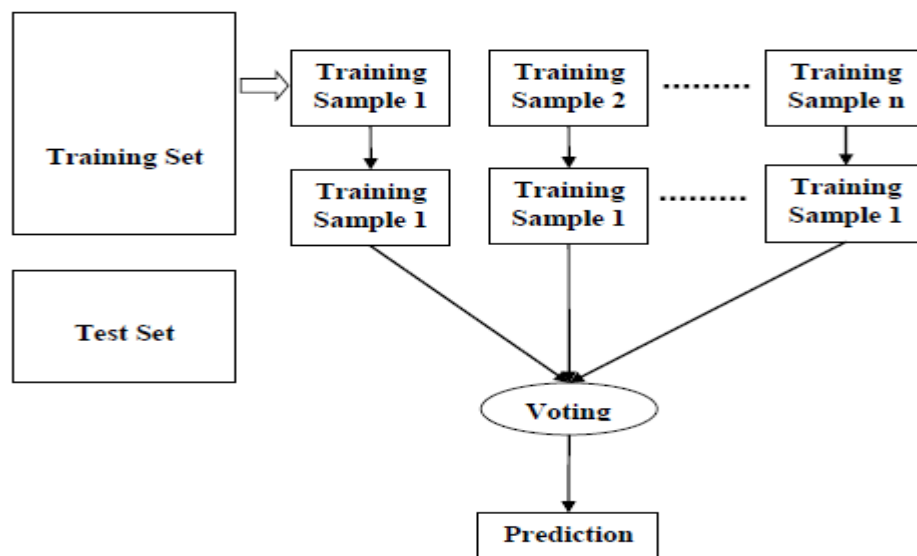
Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

#### Working of Random Forest Algorithm

We can understand the working of Random Forest algorithm with the help of following steps –

- **Step 1** – First, start with the selection of random samples from a given dataset.
- **Step 2** – Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- **Step 3** – In this step, voting will be performed for every predicted result.
- **Step 4** – At last, select the most voted prediction result as the final prediction result.

The following diagram will illustrate its working



**Fig 3.7:** Working of Random Forest

### 3.7.4 Naive Bayes algorithm:

Bayes' Theorem provides a way that we can calculate the probability of a piece of data belonging to a given class, given our prior knowledge. Bayes' Theorem is stated as:

- $P(\text{class}|\text{data}) = (P(\text{data}|\text{class}) * P(\text{class})) / P(\text{data})$

Where  $P(\text{class}|\text{data})$  is the probability of class given the provided data.

**Step 1: Separate by Class** - This means that we will first need to separate our training data by class. A relatively straightforward operation. We can create a dictionary object where each key is the class value and then add a list of all the records as the value in the dictionary.

**Step 2: Summarize Dataset** - We need two statistics from a given set of data. We'll see how these statistics are used in the calculation of probabilities in a few steps. The two statistics we require from a given dataset are the mean and the standard deviation (average deviation from the mean).

The mean is the average value and can be calculated as:

- $\text{mean} = \text{sum}(x)/n * \text{count}(x)$

Where  $x$  is the list of values or a column we are looking.

**Step 3: Summarize Data By Class** - We require statistics from our training dataset organized by class. Above, we have developed the `separate_by_class()` function to separate a dataset into rows by class. And we have developed `summarize_dataset()` function to calculate summary statistics for each column. We can put all of this together and summarize the columns in the dataset organized by class values.

**Step 4: Gaussian Probability Density Function** - Calculating the probability or likelihood of observing a given real-value like  $X_1$  is difficult. A Gaussian distribution can be summarized using only two numbers: the mean and the standard deviation. We can estimate the probability of a given value. This piece of math is called a Gaussian Probability Distribution Function (or Gaussian PDF) and can be calculated as:  $f(x) = (1 / \sqrt{2 * \pi} * \sigma) * \exp(-(x-\text{mean})^2 / (2 * \sigma^2))$   
Where  $\sigma$  is the standard deviation for  $x$ ,  $\text{mean}$  is the mean for  $x$  and  $\pi$  is the value of pi.

## **CHAPTER 4**

### **SYSTEM REQUIREMENTS SPECIFICATION**

#### **4.1 Software requirements**

- Operating system: Windows 10
- Programming Language: Python 3
- IDE: Python IDLE
- Tool: PyCharm
- Dependencies: pandas, sklearn, matplotlib, sklearn, pandas, logistic regression, random forest classifiers, naive bayes and svm

#### **4.2 Hardware requirements**

- Processor: Intel Pentium processor (64-bit system configuration recommended)
- Ram: Minimum 256Mb of RAM capacity, Minimum 32Mb graphic card RAM capacity
- Recommended Hard disk space of 10Gb or more

#### **4.3 Network requirements**

- Access to internet
- Bandwidth of minimum 2 mb/s

## **CHAPTER 5**

### **FEASIBILITY STUDY**

The project that is being proposed is a machine learning model that can be accessed by anyone who has internet access. This machine learning model can be easily used by any one by simply entering your credentials without revealing any personal information. One of the most appealing features of this model is to check the secure webpage using certain features. Without the permission of the admin, no other user or individual would have access to this model. Because the Machine learning algorithmic technique used in the development of this model is completely free, there will be no service tax charged to the end user. Another unique feature of this application is that, unlike most others, it does not require many installations, only a very few installations can be done and even take up a small amount of space on your device, saving the user's memory storage.

Feasibility study is an important phase in the software development process. Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. Feasibility study should be performed on the basis of various criteria and parameters. The various feasibility studies are:

- Technical Feasibility
- Operation Feasibility
- Economic Feasibility

#### **5.1 Technical feasibility**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?

### 5.2 Operational feasibility

- **User-friendly:** Not only programmers have to analyze data and Python can be useful for almost everyone in an office job. The problem is non-technical people are scared to death of making even the tiniest change to the code. So, it is very user-friendly.
- **Security:** As we are developing our model on PyCharm, no one can access your own private PyCharm and will be protected from hacking, virus etc.
- **Portability:** The application will be developed using google chrome as there are no restrictions. So, we can use both on Windows and Linux o/s. Hence portability problems will not rise.
- **Availability:** This software will be available always.
- **Performance:** Use the computing power of the Google servers instead of your own machine. Running python scripts requires often a lot of computing power and can take time. By running scripts in the cloud, you don't need to worry. Your local machine performance won't drop while executing your Python scripts.

### 5.3 Economic feasibility

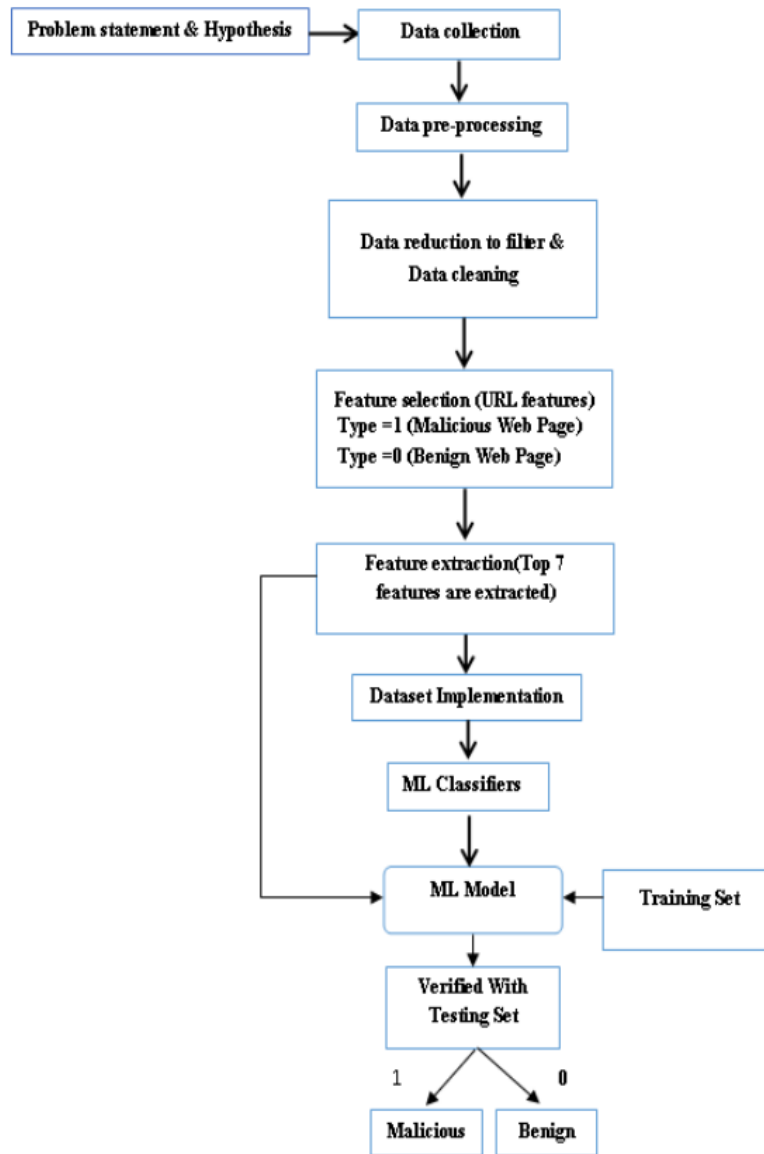
A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any additional hardware or software. The implementation of the model need not requires any separate software installation, it reduces the budget of the project.



## CHAPTER 6

### SYSTEM DESIGN

#### 6.1 System development methodology



**Fig 6.1:** Architecture for developed model

For the proposed dataset we have to collect the data from various resources. And these collected data consist of various noisy data and consists of redundancy. This can be

eliminated by using some data preprocessing methods. After that the training for the model can be done by selecting various machine learning algorithms. He testing can be done by selecting the trained model by giving the user input like URL, URL\_length, number\_special\_characters, content\_length, source\_app\_packets, remote\_app\_packets.

### **6.2 Model Phases**

1. Data Collection
2. Building Dataset
3. Building the models
4. Select the model for training
5. Evaluation the model

#### **1. Data Collection:**

This is the first real step towards the real development of a machine learning model, collecting data. This is a first and critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform. We have to select the data which is suitable for the fast development of the model and dataset that consists of a smaller number of parameters. For the developing of the project, we have collected a lot of raw data and we have collected different types of URLs from various resources. These raw data consist of many details regarding the URLs. We have taken the features from the gathered data. The collected data can be used for the training the model.

#### **2. Building Dataset:**

The dataset we have used for the development of this project is .CSV file which consists of various features which are extracted from the various resources. In the proposed dataset, we have different types of parameters which is taken as features for URLs. We have different type of features like URL, URL\_length, number\_special\_characters, content\_length, source\_app\_packets, remote\_app\_packets, Result. These features are used to extract the data it removes the outliers, null values and gives the data.

### **3. Building the models:**

After the collection of data and creation of dataset the next step is developing a model for the detection of either URLs are malicious or benign. For building the models we have used a machine learning algorithm like SVM, Logistic regression, Random Forest, Naive Bayes algorithm.

To get the final result we are going to build a model using the available dataset. While building a model for the implementation of project we have to perform the following actions:

1. Import all the modules which are necessary for developing the project
2. Building the architecture required for the training purpose
3. Selection of data type like Images/Text/CSV

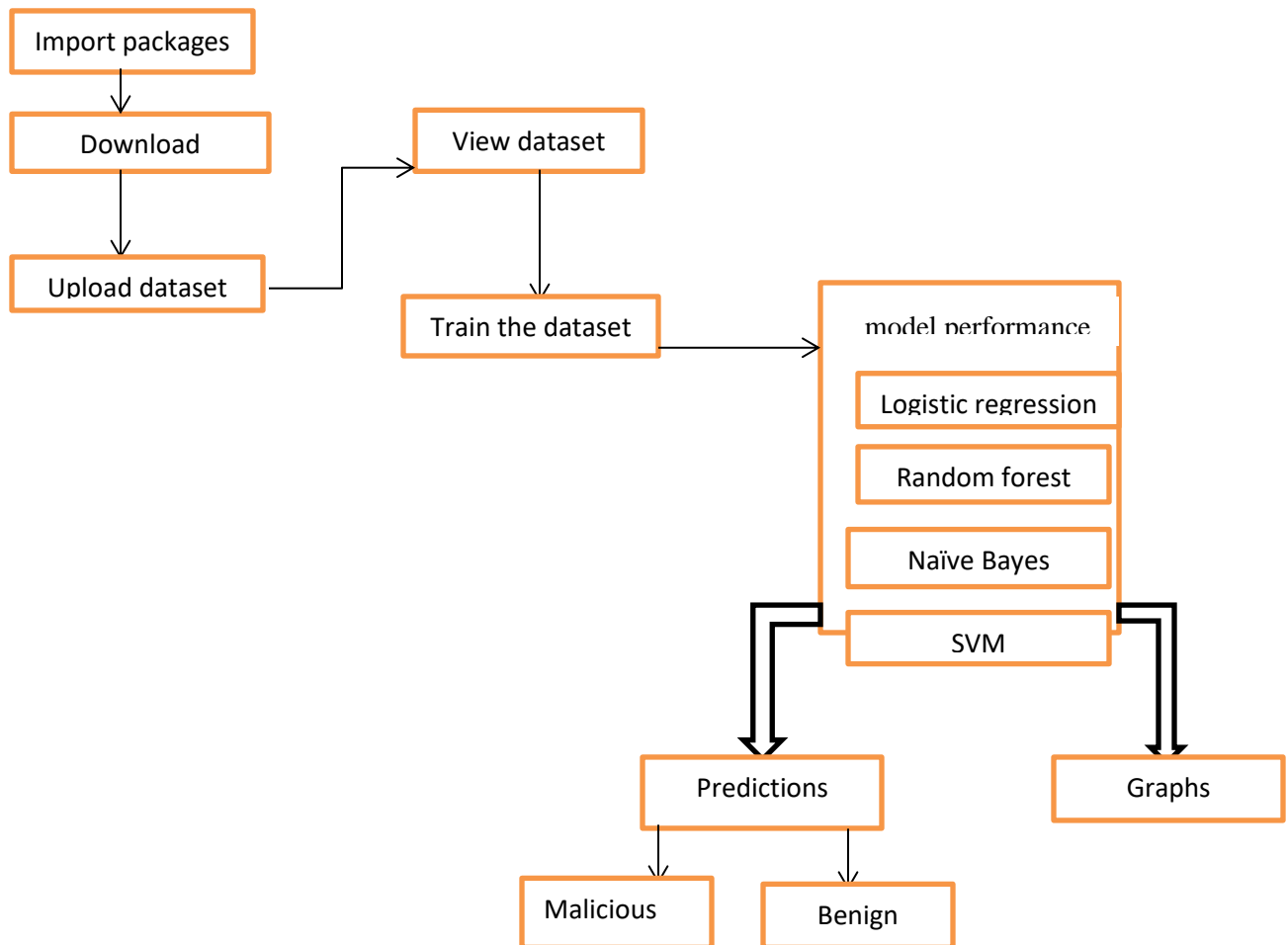
### **4. Select the model for training:**

After all, doing all the necessary activities for building the model, we have to do the further implementations. While building a model for dataset we have to make sure that data should be relevant, uniform, representative, diverse. We are going to build a model using Support Vector Machine (SVM). As Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers' detection. We have to train the given data by selecting one of the developed algorithm. We have to train the model based up on their features. The training for each model can be given by selecting the respective machine learning algorithm.

### **5. Evaluation of model:**

The last step is to evaluate the model whether our trained model is working properly for the user input. For the evaluation purpose of the proposed model, we have done some experimental analysis on some evaluation metrics like confusion matrix, precision and recall, F1 score, classification accuracy.

### 6.3 System Architecture



**Fig 6.2:** System Architecture

For this project we have used a dataset which consists of various features. At the starting phase we will train the model by using the proposed dataset. After training the model we will test the model by the user input as image, along with that user is capable of selecting the encryption algorithm. The encryption process can be done based on the training. And the model can be able to predict whether the given encryption algorithm is strong, acceptable, weak.

### 6.4 UML Diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed,

and was created by, the Object Management Group. The goal of UML is to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

### **Class Diagram:**

A class diagram shows a set of classes, interfaces, and collaboration and their relationship. It expresses the static structures of a system that are divided into different parts called classes, as well as the relationships between the classes. Class diagrams can be useful in both the early stages of a project and during the detailed design phase. A class diagram is made up of parts such as classes, associations, and generalizations, and it can have multiple levels.

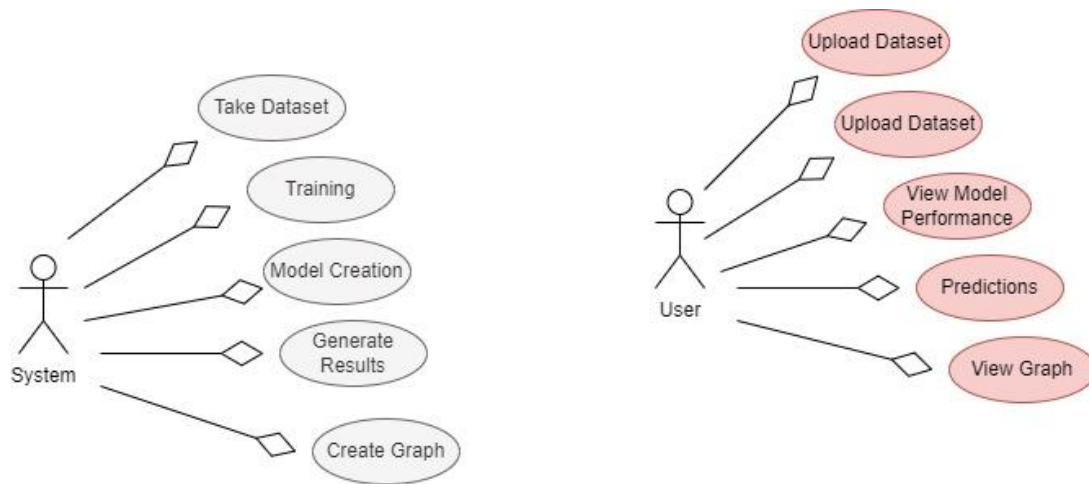


**Fig 6.3:** Class Diagram

### **Use Case diagram:**

A Use case diagram shows a set of use cases and actors and their relationships. Use case diagrams are behavior diagrams that are used to describe a set of actions (use cases) that a system or systems (subject) should or can perform in collaboration with one or more external users (actors). Each use case should result in some observable and valuable outcome for the system's actors or other stakeholders. In the early stages of a software development project, use case

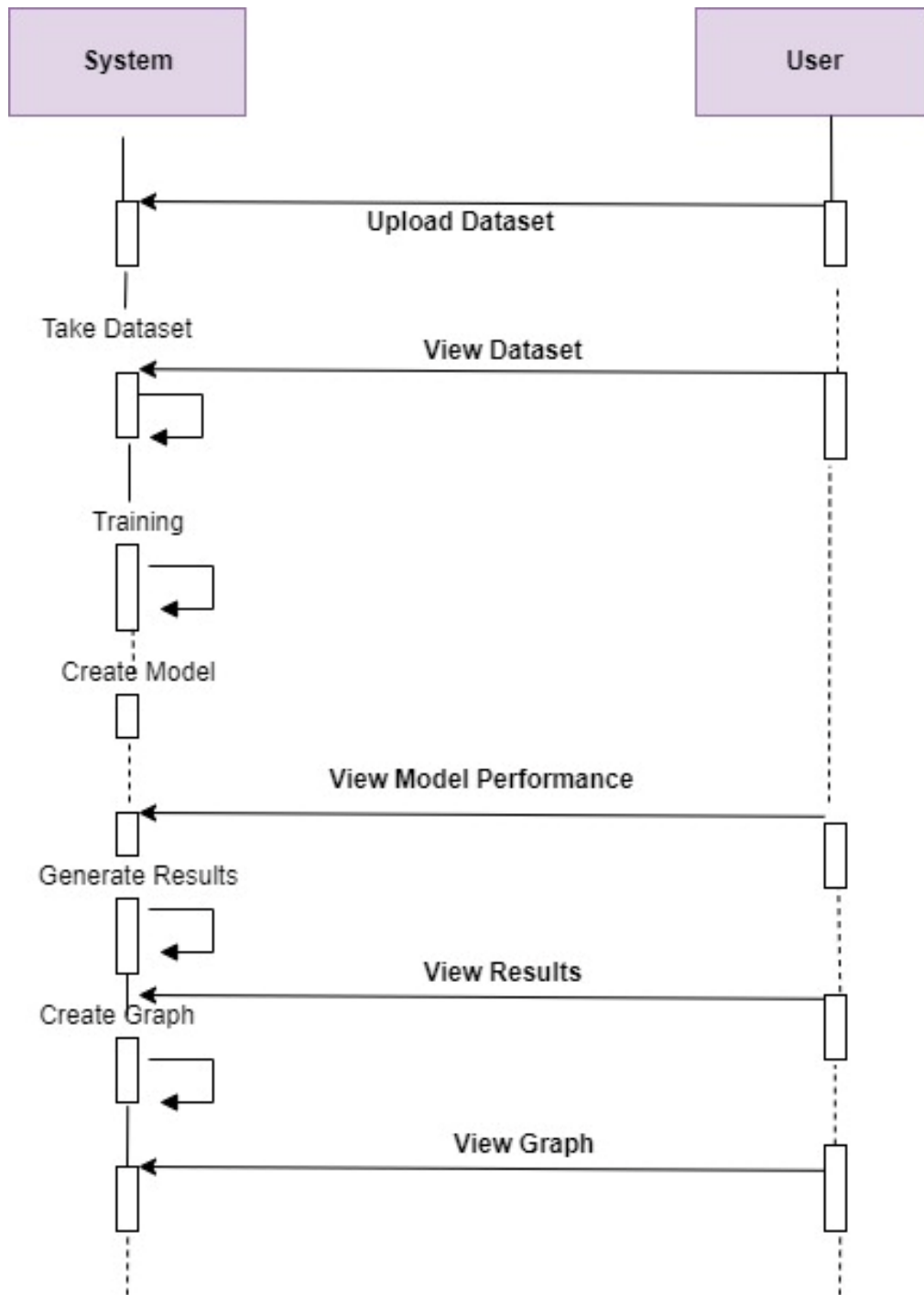
diagrams are created. They describe how the final system should be able to be used. Use cases are an intuitive and easy-to-understand way to express the functional requirements of a software system, and they can be used in negotiation with non-programmers. Use cases, one or more actors, and relationships, as well as associations and generalizations between them, are all participants in a UML use case diagram.



**Fig 6.4:** Use case diagram

### **Sequence Diagram:**

IT depicts the interaction between a group of objects by displaying the messages that can be sent between them. It's common to focus the model on scenarios specified by use-cases because the diagrams are made up of interacting objects and actors with messages in between. It's also a good idea to use it as input to the detailed class diagram when trying to model something specific.

**Fig 6.5:** Sequence Diagram

## CHAPTER 7

### IMPLEMENTATION

#### 7.1 Data Analysis

Data analysis is the process of collecting, modeling, and analyzing data to extract insights that support decision-making. There are several methods and techniques to perform analysis depending on the industry and the aim of the investigation. For the development of this project, we have collection of raw data which consists of various image features for the training purpose. These data is collected from various resources so the data is incomplete, noisy, and inconsistent. To avoid all the above problems, we have a few steps regarding to the data analysis.

1. **Data gathering:** Collect the data, remember that the collected data must be processed or organized for Analysis. As you collected data from various sources, you must have to keep a log with a collection date and source of the data.
2. **Data cleaning:** The data should be cleaned and error free. This phase must be done before Analysis because based on data cleaning, your output of Analysis will be closer to your expected outcome.
3. **Data analysis:** Once the data is collected, cleaned, and processed, it is ready for Analysis. As you manipulate data, you may find you have the exact information you need, or you might need to collect more data. During this phase, you can use data analysis tools and software which will help you to understand, interpret, and derive conclusions based on the requirements.
4. **Data interpretation:** After analyzing your data, it's finally time to interpret your results. You can choose the way to express or communicate your data analysis either you can use simply in words or maybe a table or chart. Then use the results of your data analysis process to decide your best course of action.
5. **Data visualization:** Data visualization is very common in your day-to-day life; they often appear in the form of charts and graphs. In other words, data shown graphically so that it will be easier for the human brain to understand and process it. Data visualization often used to discover unknown facts and trends.



## 7.2 Data Preprocessing

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms. Preprocessing of data is mainly to check the data quality. For the preprocessing of the data, we have different types of techniques.

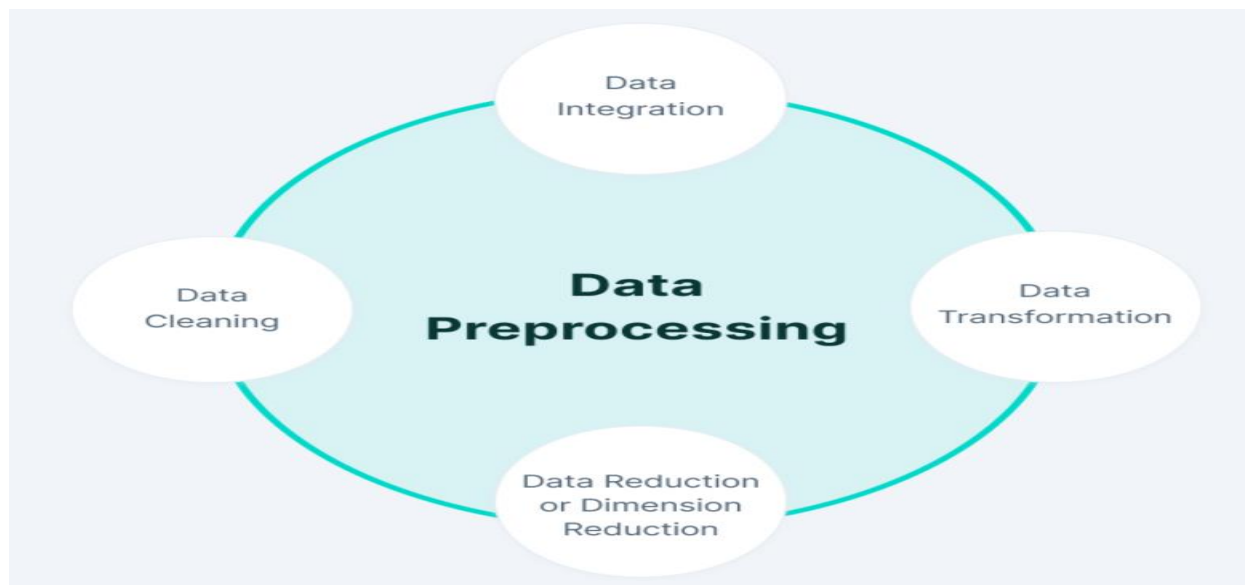
### Why is Data Preprocessing important:

The majority of the real-world datasets are highly susceptible to missing, inconsistent, and noisy data due to their heterogeneous origin.

Applying data mining algorithms on this noisy data would not give quality results as they would fail to identify patterns effectively. Data Processing is, therefore, important to improve the overall data quality.

- Duplicate or missing values may give an incorrect view of the overall statistics of data.
- Outliers and inconsistent data points often tend to disturb the model's overall learning, leading to false predictions.

There are different types of data preprocessing techniques:



**Fig 7.1:** Data Preprocessing

### **Data Cleaning:**

Data Cleaning is particularly done as part of data preprocessing to clean the data by filling missing values, smoothing the noisy data, resolving the inconsistency, and removing outliers. There are few techniques for data cleaning:

1. Remove duplicates.
2. Remove irrelevant data.
3. Standardize capitalization.
4. Convert data type.
5. Clear formatting.
6. Fix errors.
7. Language translation.
8. Handle missing values.

### **Data Integration:**

Data Integration is one of the data preprocessing steps that are used to merge the data present in multiple sources into a single larger data store like a data warehouse. Data Integration is needed especially when we are aiming to solve a real-world scenario like detecting the presence of nodules from CT Scan images. The only option is to integrate the images from multiple medical nodes to form a larger database. There are some problems to be considered during data integration.

- Schema integration: Integrates metadata (a set of data that describes other data) from different sources.
- Entity identification problem: Identifying entities from multiple databases. For example, the system or the use should know student\_id of one database and student name of another database belongs to the same entity.
- Detecting and resolving data value concepts: The data taken from different databases while merging may differ. Like the attribute values from one database may differ from another database. For example, the date format may differ like “MM/DD/YYYY” or “DD/MM/YYYY”.

### **Data Transformation:**

Data transformation is the process of converting raw data into a format or structure that would be more suitable for the model or algorithm and also data discovery in general. It is an

essential step in the feature engineering that facilitates discovering insights. This article mainly covers techniques of numeric data transformation.

### **Why need data transformation?**

- The algorithm is more likely to be biased when the data distribution is skewed.
- Transforming data into the same scale allows the algorithm to compare the relative relationship between data points better

### **When to apply data transformation?**

When implementing supervised algorithms, training data and testing data need to be transformed in the same way. This is usually achieved by feeding the training dataset to building the data transformation algorithm and then apply that algorithm to the test set.

### **Data Reduction:**

The size of the dataset in a data warehouse can be too large to be handled by data analysis and data mining algorithms. One possible solution is to obtain a reduced representation of the dataset that is much smaller in volume but produces the same quality of analytical results. Here is a walkthrough of various Data Reduction strategies.

**Data cube aggregation:** It is a way of data reduction, in which the gathered data is expressed in a summary form.

**Dimensionality reduction:** Dimensionality reduction techniques are used to perform feature extraction. The dimensionality of a dataset refers to the attributes or individual features of the data. This technique aims to reduce the number of redundant features we consider in machine learning algorithms. Dimensionality reduction can be done using techniques like Principal Component Analysis etc.

**Data compression:** By using encoding technologies, the size of the data can significantly reduce. But compressing data can be either lossy or non-lossy. If original data can be obtained after reconstruction from compressed data, this is referred to as lossless reduction; otherwise, it is referred to as lossy reduction.

**Discretization:** Data discretization is used to divide the attributes of the continuous nature into data with intervals. This is done because continuous features tend to have a smaller chance of correlation with the target variable.

**Numerosity reduction:** The data can be represented as a model or equation like a regression model. This would save the burden of storing huge datasets instead of a model.

**Attribute subset selection:** It is very important to be specific in the selection of attributes. Otherwise, it might lead to high dimensional data, which are difficult to train due to underfitting/overfitting problems. Only attributes that add more value towards model training should be considered, and the rest all can be discarded.

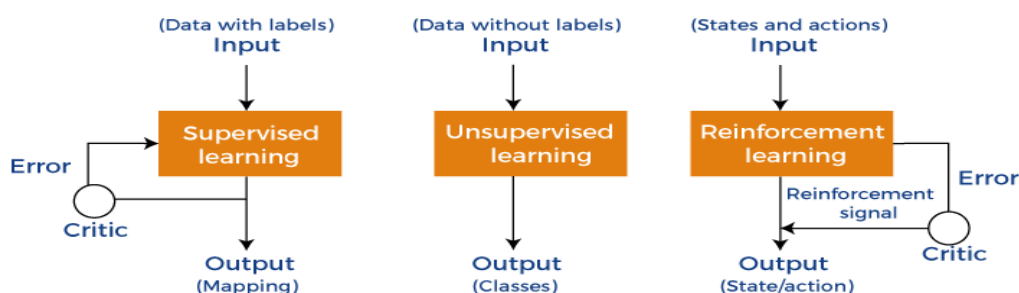
### 7.3 Machine learning models

Machine Learning models can be understood as a program that has been trained to find patterns within new data and make predictions. These models are represented as a mathematical function that takes requests in the form of input data, makes predictions on input data, and then provides an output in response.

#### Classification of Machine Learning Models:

Based on different business goals and data sets, there are three learning models for algorithms. Each machine learning algorithm settles into one of the three models:

- i. Supervised Learning
- ii. Unsupervised Learning
- iii. Reinforcement Learning



**Fig 7.2:** Machine Learning models

**Based on the type of tasks, we can classify machine learning models into the following types:**

- Classification Models
- Regression Models
- Clustering
- Dimensionality Reduction
- Deep Learning etc.

### **1) Classification**

With respect to machine learning, classification is the task of predicting the type or class of an object within a finite number of options. The output variable for classification is always a categorical variable. For example, predicting an email is spam or not is a standard binary classification task. Some important models for classification problems.

1. K-Nearest neighbors' algorithm – simple but computationally exhaustive.
2. Naive Bayes – Based on Bayes theorem.
3. Logistic Regression – Linear model for binary classification.
4. SVM – can be used for binary/multiclass classifications.

### **2) Regression**

In the machine, learning regression is a set of problems where the output variable can take continuous values. For example, predicting the airline price can be considered as a standard regression task. Some important regression models used in practice.

1. Linear Regression – Simplest baseline model for regression task, works well only when data is linearly separable and very less or no multicollinearity is present.
2. Lasso Regression – Linear regression with L2 regularization.
3. Ridge Regression – Linear regression with L1 regularization.
4. SVM regression
5. Decision Tree Regression etc.

### **3) Clustering**

In simple words, clustering is the task of grouping similar objects together. It helps to identify similar objects automatically without manual intervention. We cannot build effective supervised machine learning models (models that need to be trained with manually curated or labeled data) without homogeneous data. Following are some of the widely used clustering models:

1. K-means - Simple but suffers from high variance.
2. K medoids.
3. DBSCAN – Density-based clustering algorithm etc.

### **4) Dimensionality Reduction**

Dimensionality is the number of predictor variables used to predict the independent variable or target. Often, in real-world datasets, the number of variables is too high. Too many variables also bring the curse of overfitting to the models. In practice, among these large numbers of variables, not all variables contribute equally towards the goal, and in a large number of cases, we can actually preserve variances with a lesser number of variables. Some commonly used models for dimensionality reduction.

1. PCA – It creates lesser numbers of new variables out of a large number of predictors. The new variables are independent of each other but less interpretable.
2. TSNE – Provides lower dimensional embedding of higher-dimensional data points.
3. SVD – Singular value decomposition is used to decompose the matrix into smaller parts to efficiently calculate.

### **5) Deep Learning**

Deep learning is a subset of machine learning which deals with neural networks. Based on the architecture of neural networks, Some important deep learning models:

1. Artificial Neural Networks
2. Convolution Neural Networks
3. Recurrent Neural Networks

## **CHAPTER 8**

### **TESTING**

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as “A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item”.

#### **8.1 Testing Methodologies**

Any Software product can be tested in one of two ways

##### **8.1.1 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.

##### **8.1.2 Black Box Testing:**

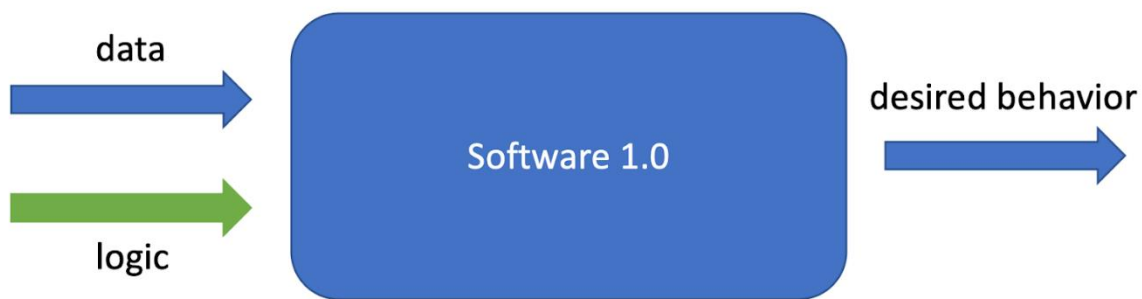
In Black Box testing by understanding and knowing the various program internal operation of a application or product or program, Black Box Testing can be conducted to guarantee that all gears mesh of the internal activities of the product or program or application can be tested. The process provides 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.

The steps involved in black box test case design are:

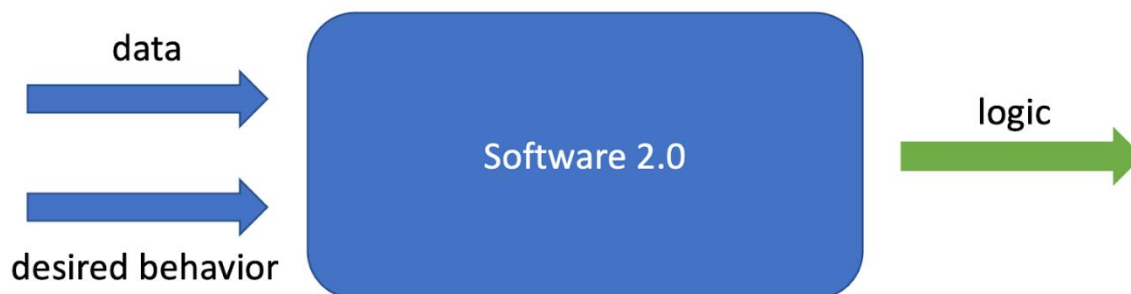
- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

## 8.2 Testing machine learning models

In traditional software systems, humans write the logic which interacts with data to produce a desired behavior. Our software tests help ensure that this written logic aligns with the actual expected behavior.



However, in machine learning systems, humans provide desired behavior as examples during training and the model optimization process produces the logic of the system. How do we ensure this learned logic is going to consistently produce our desired behavior?





Let's start by looking at the best practices for testing traditional software systems and developing high-quality software.

A typical software testing suite will include:

**Unit Testing:** unit tests which operate on atomic pieces of the codebase and can be run quickly during development.

**Regression Testing:** regression tests replicate bugs that we've previously encountered and fixed.

**Integration Testing:** integration tests which are typically longer-running tests that observe higher-level behaviours that leverage multiple components in the codebase.

For machine learning systems, we should be running model evaluation and model tests in parallel.

**Model evaluation:** Model evaluation covers metrics and plots which summarize performance on a validation or test dataset.

**Model testing:** Model testing involves explicit checks for behaviours that we expect our model to follow.

### 8.3 System Testing

Testing has become an integral part of any system especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed.

### 8.4 Quality Assurance

Quality assurance (QA) is a way of preventing mistakes and defects in manufactured products and avoiding problems when delivering products or services to customers; which ISO 9000 defines as "part of quality management focused on providing confidence that quality requirements will be fulfilled". This defect prevention in quality assurance differs subtly from defect detection and rejection in quality control and has been referred to as a shift left since it focuses on quality earlier in the process.

Quality assurance comprises administrative and procedural activities implemented in a quality system so that requirements and goals for a product, service or activity will be fulfilled. It is the systematic measurement, comparison with a standard, monitoring of processes and an associated feedback loop that confers error prevention. This can be contrasted with quality control, which is focused on process output.

### **Types of Quality Assurance:**

#### **User Acceptance Testing:**

Though a programmer may build an application with a specific intended use, consumers behave in a variety of ways.

#### **Software Performance Testing:**

The designers of an app always want to confirm the specs and abilities of the program. Performance testing helps to define those limits and ensure accuracy by checking for speed, scalability, and stability through a series of experiments.

#### **Data Conversion Testing:**

When a company migrates data to a new software, it becomes vulnerable. Once the transfer of information has begun, digital assets are quite literally hanging in the balance. Any errors could cause massive file corruption or even data loss. That's why extensive conversion testing to confirm the compatibility between old and new systems is important.

### **8.5 Functional Testing:**

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on simulation of actual system usage but does not develop any system structure assumptions.

It is basically defined as a type of testing which verifies that each function of the software application works in conformance with the requirement and specification. This testing is not concerned about the source code of the application. Each functionality of the software application is tested by providing appropriate test input, expecting the output and comparing the

actual output with the expected output. This testing focuses on checking of user interface, APIs, database, security, client or server application and functionality of the Application Under Test.

Functional testing can be manual or automated.

### **Functional Testing Process:**

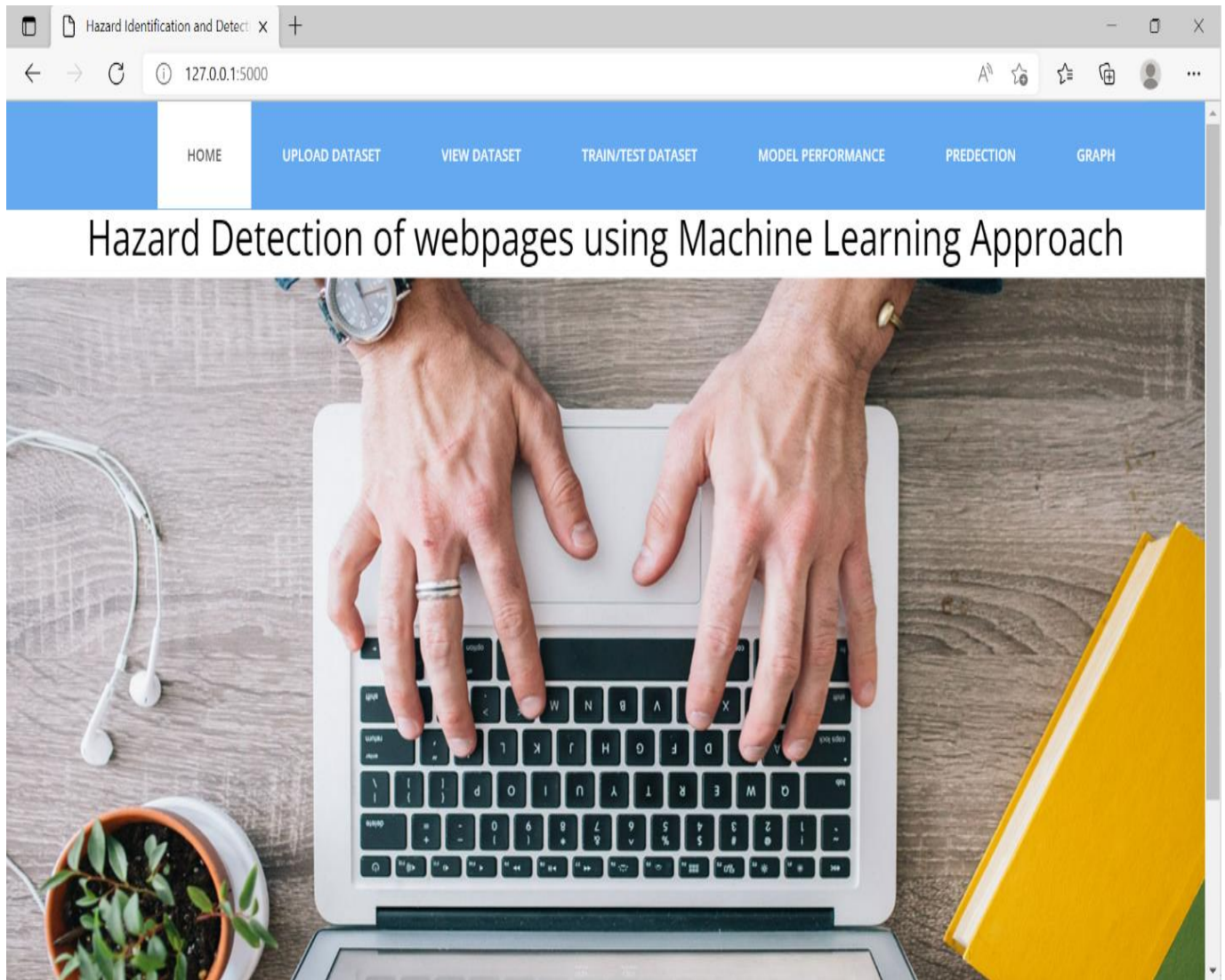
Functional testing involves the following steps:

1. Identify function that is to be performed.
2. Create input data based on the specifications of function.
3. Determine the output based on the specifications of function.
4. Execute the test case.
5. Compare the actual and expected output.

## CHAPTER 9

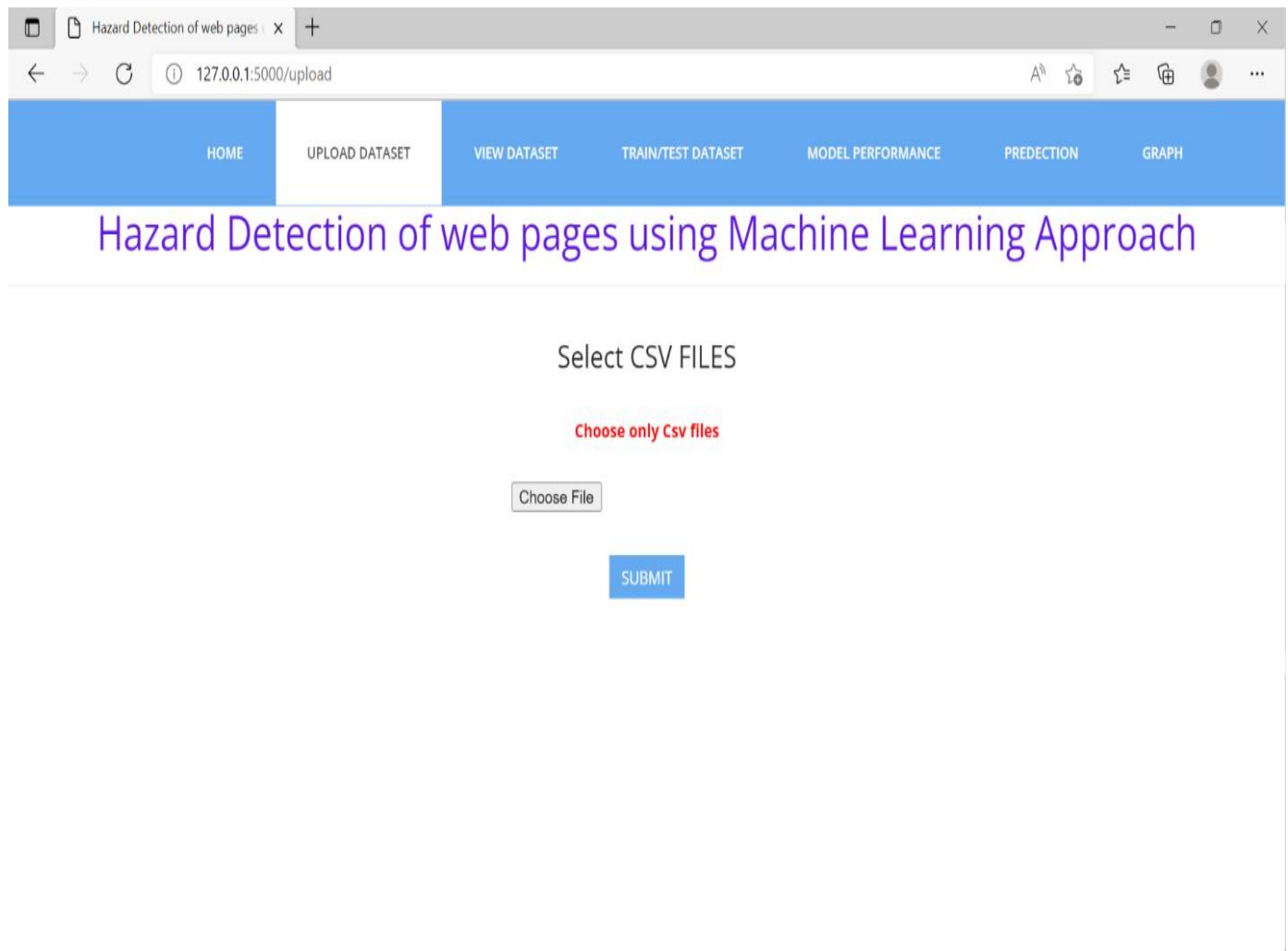
### RESULTS AND PERFORMANCE ANALYSIS

#### 9.1 Screenshots



**Fig 9.1:** Home page

This is the home page for the developed project which consists of different types of buttons regarding the different type of pages to navigate to the respective page.



**Fig 9.2:** Upload Dataset page

This upload page can be used to upload the proposed dataset i.e. **.CSV** file which consists of various parameters for the development of the project. And we can choose the file from the system itself. The uploaded dataset can be submitted to developed model by clicking the submit button.

URL	URL LENGTH	NUMBER SPECIAL CHARACTERS	CONTENT LENGTH	SOURCE APP PACKETS	REMOTE APP PACKETS	Result
80_109	16	7	263.0	9	10	1
80_2314	16	6	15087.0	17	19	0
80_911	16	6	324.0	0	0	0
80_113	17	6	162.0	39	37	0
80_403	17	6	124140.0	61	62	0
80_462	18	6	345.0	14	13	0
80_1128	19	6	324.0	0	0	0
80_1102	20	6	324.0	0	0	0
80_22	20	7	13716.0	20	20	0
80_482	20	6	3692.0	35	29	0
80_869	20	7	13054.0	0	0	0
80_71	21	7	957.0	11	10	1
80_97	21	7	686.0	8	9	1
80_2303	21	6	324.0	7	9	0
80_584	21	6	15025.0	15	17	0
80_69	22	7	324.0	11	9	1
80_2122	22	6	318.0	8	10	0
80_2176	22	6	224.0	4	6	0

**Fig 9.3:** View Dataset page

The above page (View Dataset page) can be used to see details of the dataset. It can be used to see the detailed information about the dataset like URL name, URL length, Special characters available in the dataset etc. and information of each parameter is visible through the view dataset page.

HOME UPLOAD DATASET VIEW DATASET TRAIN/TEST DATASET MODEL PERFORMANCE PREDECTION GRAPH

## Hazard Detection of web pages using Machine Learning Approach

Select Test DataSet Size

Choose upto 0.1 to 0.9 ratio

SUBMIT

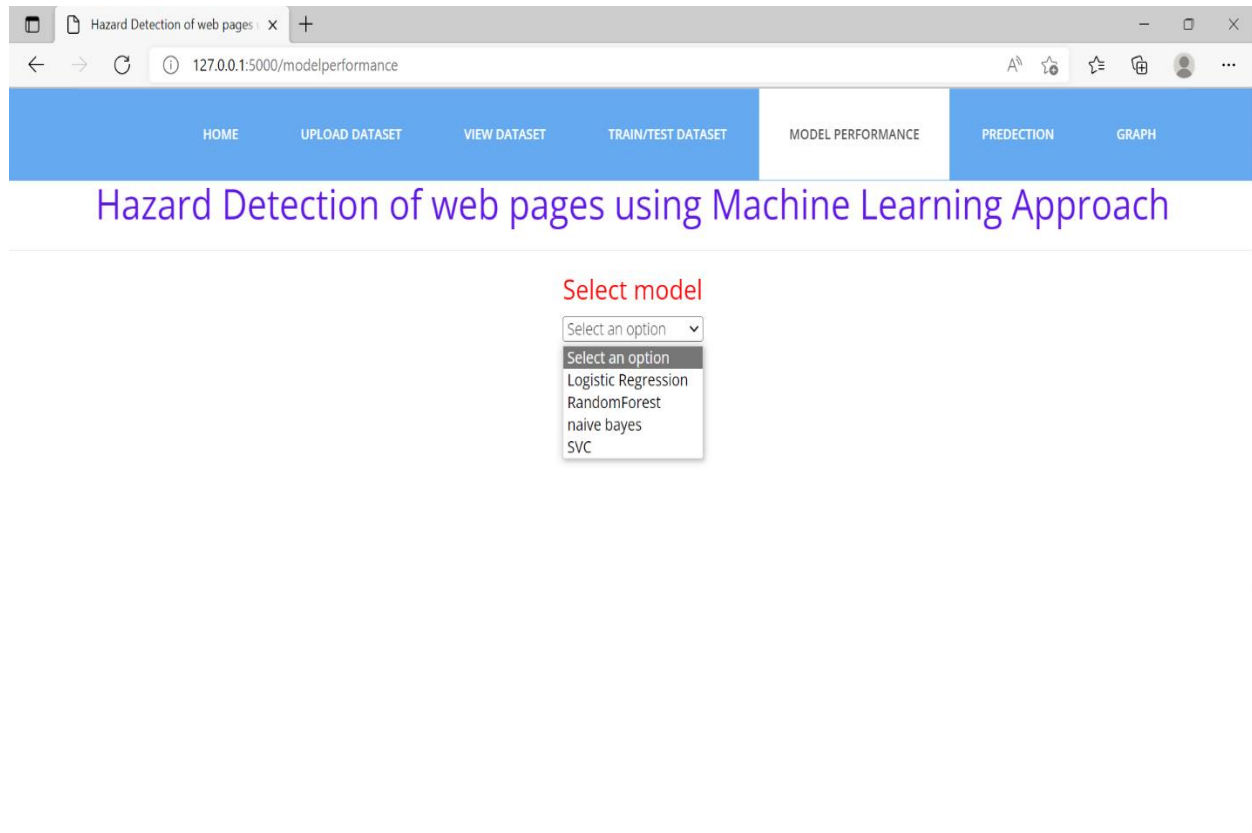
Train Data Length is : 678

Test Data Length is : 291

	URL	URL_LENGTH	NUMBER_SPECIAL_CHARACTERS	CONTENT_LENGTH	SOURCE_APP_PACKETS	REMOTE_APP_PACKETS
207	B0_2271	32	6	199.0	6	7
15	M0_71	21	7	957.0	11	10
994	B0_2119	53	11	8466.0	26	13
1020	B0_230	54	10	18952.0	1198	1284
867	B0_16	49	10	1148.0	11	9
1310	B0_933	66	11	241.0	0	0
1713	B0_529	120	20	100163.0	19	18

**Fig 9.4:** Train Dataset page

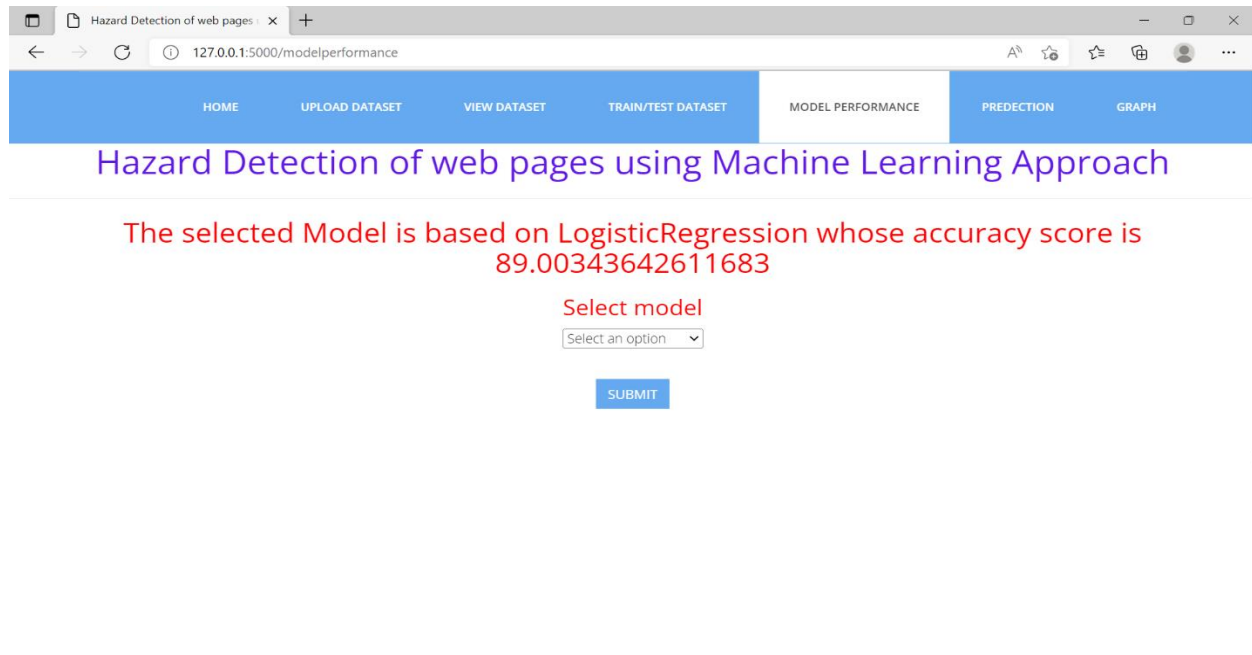
The above page can be used to train the developed model by choosing the ratio that up to extent the training should be done. After selecting the ratio of the dataset size we can see normalized details regarding the dataset and there will not be any duplicated data in the dataset after performing this activity.



**Fig 9.5:** Selection of ML algorithm page

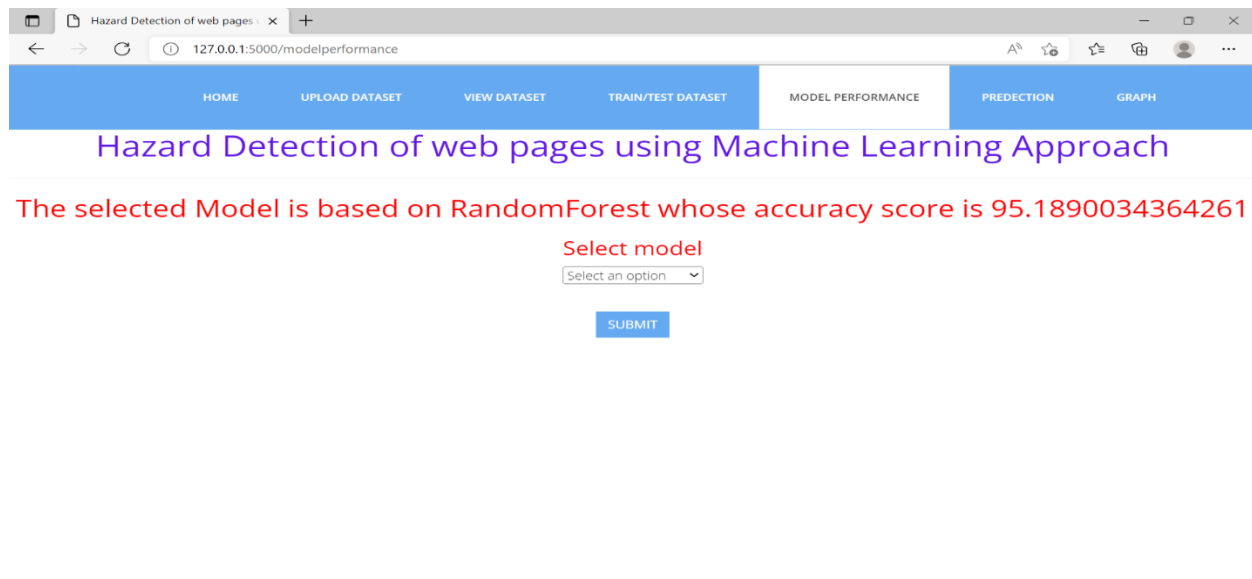
The above page can be used to select the machine learning algorithm on which you want to perform the training. By selecting the one of the proposed models as shown above the figure. So, that model can be able to predict the output and accuracy of the selected algorithm.





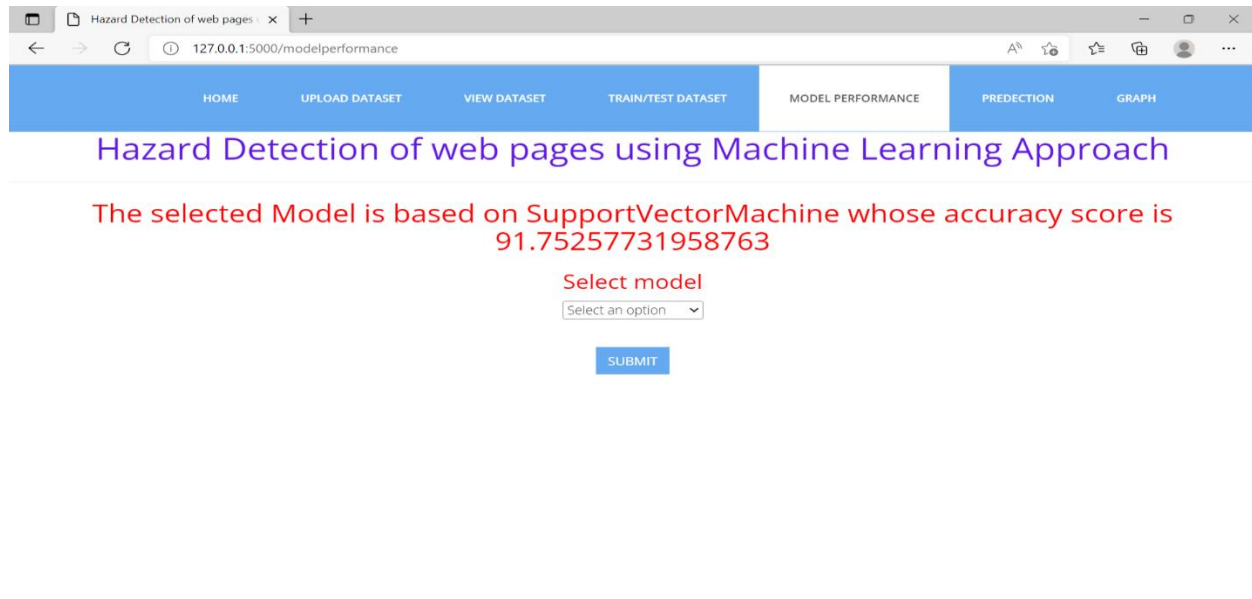
**Fig 9.6:** Accuracy of Logistic Regression

The above figure shows the accuracy of Logistic Regression which is 89.00343642611683. This accuracy will depend on based up on the dataset size and the number of features that are used for the development of the model.



**Fig 9.7:** Accuracy of Random Forest

The above figure shows the accuracy of Random Forest which is 95.1890034364261. This algorithm shows the best accuracy among three algorithms that we proposed.



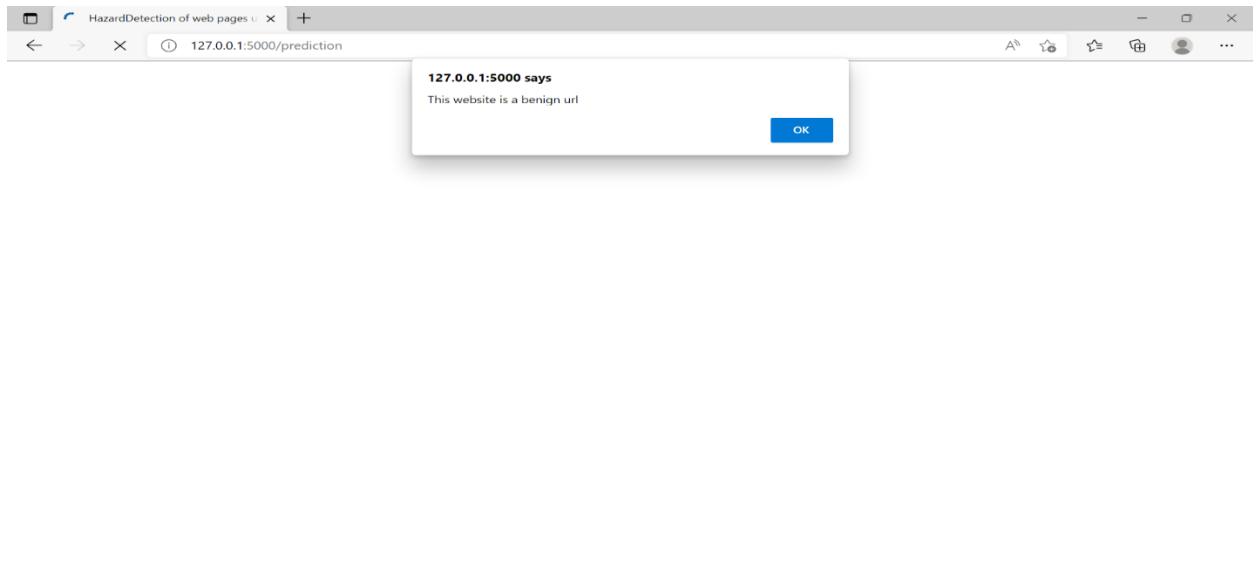
**Fig 9.8:** Accuracy of Support Vector Machine

The above figure shows the accuracy of Support Vector Machine which is 91.75257731958763

The screenshot shows a web browser window with the URL `127.0.0.1:5000/prediction`. The page has a blue navigation bar with links: HOME, UPLOAD DATASET, VIEW DATASET, TRAIN/TEST DATASET, MODEL PERFORMANCE, PREDECTION (active), and GRAPH. Below the navigation bar, the title "Hazard Detection of web pages using Machine Learning Approach" is displayed in purple. The main content area contains five input fields with labels: "ENTER URL LENGTH:" (value: 12), "ENTER NUMBER OF SPECIAL CHARACTERS:" (value: 7), "ENTER CONTENT\_LENGTH:" (value: 12), "ENTER SOURCE\_APP\_BYTES:" (value: 17), and "ENTER APP\_PACKETS:" (value: 10). A blue "Submit" button is located below the input fields.

**Fig 9.9:** User input for prediction page

The above page can be used to give the user own input based up on the features available in the proposed dataset. It consists of multiple text fields to give the input. And submit button can be used to submit the details to the system.



**Fig 9.10:** Output for Benign URL

The above figure shows the output for the benign URL. The message shows after the execution of the project will be like “This website is a benign url” as shown in the above figure.



**Fig 9.11:** Bar chart for analysis

The above figure shows the analysis about the project regarding different machine learning. This analysis can be done using the evaluation parameters like precision, recall, accuracy. By using this analysis, we can see the performance of the model.

## **CHAPTER 10**

### **CONCLUSION**

Malicious web page identification is an emerging topic in cybersecurity. Though several research studies have been performed relating to the issues of malicious web page detection these are very costly as they consume more time and resources. In this research article, we employed a new web site classification system based on URL features to predict the web pages as malicious or benign using machine learning algorithms. The machine learning classifiers Random Forest (RF) achieves a higher accuracy of 95%. The experimental results have shown that our method can perform effectively for detecting the malicious web page.

## **REFERENCES**

- [1] Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020) IEEE Xplore Part Number: CFP20K74-ART.
- [2] Altay, Betul, Tansel Dokeroglu, and Ahmet Cosar. "Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection." *Soft Computing* 23, no. 12 (2019): 4177-4191.
- [3] Kim, Sungjin, Jinkook Kim, Seokwoo Nam, and Dohoon Kim. "WebMon: ML-and YARA-based malicious webpage detection." *Computer Networks* 137 (2018): 119-131.
- [4] Wang, Rong, Yan Zhu, Jiefan Tan, and Binbin Zhou. "Detection of malicious web pages based on hybrid analysis." *Journal of Information Security and Applications* 35 (2017): 68-74.74.
- [5] Sun, Bo, Mitsuaki Akiyama, Takeshi Yagi, Mitsuhiro Hatada, and Tatsuya Mori. "Automating URL blacklist generation with similarity search approach." *IEICE TRANSACTIONS on Information and Systems* 99, no. 4 (2016): 873-882.
- [6] Krishnaveni, S., and K. Sathiyakumari. "SpiderNet: An interaction tool for predicting malicious web pages." In *International Conference on Information Communication and Embedded Systems (ICICES2014)*, pp. 1-6. IEEE, 2014.
- [7] Yue, Tao, Jianhua Sun, and Hao Chen. "Fine-grained mining and classification of malicious Web pages." In *2013 Fourth International Conference on Digital Manufacturing & Automation*, pp. 616-619. IEEE, 2013.
- [8] Aldwairi, Monther, and Rami Alsalman. "Malurls: A lightweight malicious website classification based on URL features." *Journal of Emerging Technologies in Web Intelligence* 4, no. 2 (2012): 128-133.