

A
Project Report
on
**WEB HAZARD IDENTIFICATION AND DETECTION USING MACHINE
LEARNING APPROACH**

Submitted to
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTAPURAMU

in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
Submitted by

T. SUJITESH	(199E1A05C5)
R. BALAJI KUMAR REDDY	(199E10A5B8)
G. VENKATESH	(209E5A0510)
J. LAKSHMI RAJ	(199E1A0586)

Under the Guidance of
Mr. N. Sathish., MTech.,
Assistant Professor,
Department of CSE.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SRI VENKATESWARA ENGINEERING COLLEGE

(Approved by AICTE, New Delhi NBA & NAAC Accredited Institution with UGC section 2(f) & 12(b)

& Affiliated to JNTUA, Ananthapuramu)

Karakambadi Road, TIRUPATI – 517507

2019-2023

SRI VENKATESWARA ENGINEERING COLLEGE

(Approved by AICTE, New Delhi Accredited by NBA & NAAC UGC section 2(f) & 12(B) & Affiliated to JNTUA, Ananthapuramu) Karakambadi Road, TIRUPATI – 517507

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project report entitled
“WEB HAZARD IDENTIFICATION AND DETECTION USING MACHINE LEARNING APPROACH”

a bonafide record of the project work done and submitted by

T. SUJITESH	(199E1A05C5)
R. BALAJI KUMAR REDDY	(199E1A05B8)
G. VENKATESH	(209E5A0510)
J. LAKSHMI RAJ	(199E1A0586)

*for the partial fulfillment of the requirements for the award of B.Tech Degree in **COMPUTER SCIENCE AND ENGINEERING**, JNT University Anantapur, Ananthapuramu.*

GUIDE

Head of the Department

External Viva-Voce Exam Held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project report entitled **“WEB HAZARD IDENTIFICATION AND DETECTION USING MACHINE LEARNING APPROACH.”** done by us under the guidance of **S Jasmin Sabeena**, and is submitted in partial fulfillment of the requirements for the award of the Bachelor’s degree in **Computer Science and Engineering**. This project is the result of our own effort and it has not been submitted to any other University or Institution for the award of any degree or diploma other than specified above.

T. SUJITESH	(199E1A05C5)
R. BALAJI KUMAR REDDY	(199E1A05B8)
G. VENKATESH	(209E5A0510)
J. LAKSHMI RAJ	(199E1A0586)

ACKNOWLEDGEMENT

We are thankful to our guide **Mr. N. Sathish** for her valuable guidance and encouragement. Her helping attitude and suggestions have helped us in the successful completion of the project.

We would like to express our gratefulness and sincere thanks to **Dr. K. Santi**, Head of the Department of COMPUTER SCIENCE AND ENGINEERING, for her kind help and encouragement during the course of our study and in the successful completion of the project work.

We have great pleasure in expressing our hearty thanks to our beloved Principal **Dr. C. Chandrasekhar**, for spending his valuable time with us to complete this project.

Successful completion of any project cannot be done without proper support and encouragement. We sincerely thank to the **Management** for providing all the necessary facilities during the course of study.

We would like to thank our parents and friends, who have the greatest contributions in all our achievements, for the great care and blessings in making us successful in all our endeavors.

T. SUJITESH	(199E1A05C5)
R. BALAJI KUMAR REDDY	(199E1A05B8)
G. VENKATESH	(209E5A0510)
J. LAKSHMI RAJ	(199E1A0586)

TABLE OF CONTENTS

Chapter No.	Description	Page No.
	Abstract	i
	List of Figures	ii
1	Introduction	1
2	Literature Survey	3
3	Problem Definition	6
4	Computational Environment	7
	4.1 Hardware Specification	7
	4.2 Software Specification	7
	4.3 Functional and non-functional requirements	7
	4.4 Software Features	8
5	Feasibility Study	28
	5.1 Economical Feasibility	28
	5.2 Technical Feasibility	28
	5.3 Social Feasibility	28
6	System Analysis	30
	6.1 Existing Method	30
	6.2 Proposed Method	30
7	System Design	31
	7.1 UML Diagrams	31
	7.1.1 Class Diagram	32
	7.1.2 Use Case Diagram	32
	7.1.3 Sequence Diagram	33
	7.1.4 Collaboration Diagram	33
	7.1.5 Deployment Diagram	34
	7.1.6 Activity Diagram	34
	7.1.7 Component Diagram	35
	7.1.8 ER Diagram	36
	7.1.9 DFD Diagram	36
	7.1.10 Architecture	38

8	System Implementation	39
	8.1 Implementation Process	39
	8.2 Modules	39
9	Algorithms	41
	9.1 Random Forest Classifier	41
	9.2 ADABOOST Classifier	42
	9.3 XGBoost	43
	9.4 Gradient Boosting Classifier	44
	9.5 Support Vector Machine	45
10	System Testing	49
	10.1 Types of Tests	49
	10.1.1 Unit Testing	49
	10.1.2 Integration Testing	49
	10.1.3 Functional Testing	50
	10.1.4 White Box Testing	50
	10.1.5 Black Box Testing	50
	10.1.6 Acceptance Testing	51
11	Source Code	52
12	Screen Layout	68
13	Conclusion and Future enhancements	73
14	Bibliography	74
	References	74

ABSTRACT

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 analyse 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.

LIST OF FIGURES

S.No.	Figure No.	Description	Page No.
1	Fig.1	Software Installation	9
2	Fig.2	UML Diagrams	32
3	Fig.3	Algorithms	41
4	Fig.4	Web Pages	68

CHAPTER -1

INTRODUCTION

Hazard Identification and Detections the process of identifying and flagging websites that attempt to impersonate legitimate websites with the goal of stealing sensitive information such as login credentials, credit card numbers, and personal identification information. Hazard attacks have become increasingly sophisticated over the years, and attackers often use tactics such as social engineering and fake login screens to trick users into giving up their sensitive information. Hazard Identification and Detection may also use URL spoofing to make it appear that the user is on a legitimate. Hazard attacks are a common type of cybercrime that involves the use of fraudulent emails, messages, or websites to trick users into revealing sensitive information. One of the most effective ways to combat Hazard attacks is through the detection and blocking of Hazard websites. Hazard Identification and Detection involves the use of various techniques and technologies to identify and flag websites that are designed to deceive users. One common technique used by attackers is to create websites that closely mimic the appearance of legitimate sites, such as banks or e-commerce sites. These Hazard sites are often hosted on compromised servers or using domain names that are similar to the real sites.

One approach to detecting Hazard Identification And Detection is to use machine learning algorithms that can analyze website content, metadata, and other features to identify potential Hazard sites. These algorithms can be trained on large datasets of known Hazard Identification And Detection to identify common patterns and characteristics. Some machine learning models may also incorporate real-time data feeds to identify and flag new Hazard sites as they are created. Another approach to Hazard Identification And Detection is to use reputation-based systems that maintain lists of known malicious websites. These systems can use various sources of information, such as blacklists, user reports, and threat intelligence feeds to identify and block Hazard sites. Some web browsers also use reputation-based systems to warn users when they attempt to access a known Hazard site. Hazard Identification And Detection may also involve the use of behavioural analysis techniques that can detect unusual or suspicious activity on a website. For example, these techniques may look for patterns of user behaviour that differ from normal usage, such as a sudden increase in requests for login credentials or an unusual number of redirects to other sites. Overall, the detection and blocking of Hazard Identification And Detection is an essential component of

any effective cyber-security strategy. By using a combination of machine learning algorithms, reputation-based systems, and behavioural analysis techniques, organizations can protect their users and prevent sensitive information from falling into the hands of attackers. Therefore, it is crucial to remain vigilant and stay up-to-date with the latest Hazard threat trends and detection methods.

Objective

The objective of Hazard Identification and Detection using machine learning is to develop an automated system that can accurately and efficiently identify websites that are designed to steal sensitive information from users. This application is to investigate a specific problem of whether it is valuable or not to use machine learning techniques to predict the type of website.

Scope

In this application, we design a new classification system to analyse and detect the malicious web pages using machine learning classifiers. We use URL (Uniform Resource Locator) based on extricated features the classifiers are trained to predict the malicious web pages.

CHAPTER – 2

LITERATURE REVIEW

[1] J. Shad and S. Sharma, “A Novel Machine Learning Approach to Detect Hazard Identification And Detection Jaypee Institute of Information Technology”.

In the last few years, many fake websites have developed on the World Wide Web to harm users by stealing their confidential information such as account ID, user name, password, etc. Hazard is the social engineering attacks and currently attacks on mobile devices. That might result in the form of financial losses. In this paper, we described many detection techniques using URL, Hyperlinks features that can be used to differentiate between the defective and non-defective website. There are six main approaches such as heuristic, blacklist, Fuzzy Rule, machine learning, image processing, and CANTINA based approach. It delivers a good consideration of the Hazard issue, a present machine learning solution, and future study about Hazard threats by using machine learning Approach.

[2] Y. Sönmez, T. Tuncer, H. Gökal, and E. Avci, “Hazard web sites features classification based on extreme learning machine,” 6th Int. Symp. Digit. Forensic Secur. ISDFS - Proceeding.

Hazard is a common attack on credulous people by making them to disclose their unique information using counterfeit websites. The objective of Hazard website URLs is to purloin the personal information like user name, passwords and online banking transactions. Phishers use the websites which are visually and semantically similar to those real websites. As technology continues to grow, Hazard techniques started to progress rapidly and this needs to be prevented by using anti-Hazard mechanisms to detect Hazard. Machine learning is a powerful tool used to strive against Hazard attacks. This paper surveys the features used for detection and detection techniques using machine learning.

[3] T. Peng, I. Harris, and Y. Sawa, “Detecting Hazard Attacks Using Natural Language Processing and Machine Learning,” Proc. - 12th IEEE Int. Conf. Semant. Comput.

Hazard attacks are one of the most common and least defended security threats today. We present an approach which uses natural language processing techniques to analyse text and detect inappropriate statements which are indicative of Hazard attacks. Our approach is

novel compared to previous work because it focuses on the natural language text contained in the attack, performing semantic analysis of the text to detect malicious intent. To demonstrate the effectiveness of our approach, we have evaluated it using a large benchmark set of Hazard emails.

[4] M. Karabatak and T. Mustafa, “Performance comparison of classifiers on reduced Hazard website dataset,” 6th Int. Symp. Digit. Forensic Secur. ISDFS - Proceeding.

These days, numerous enemy of Hazard frameworks are being created to recognize Hazard substance in online correspondence frameworks. In spite of the accessibility of hordes hostile to Hazard frameworks, Hazard proceeds with unabated because of lacking recognition of a zero-day assault, pointless computational overhead and high bogus rates. In spite of the fact that Machine Learning approaches have accomplished promising exactness rate, the decision and the exhibition of the component vector limit their successful location. Hazard is a typical assault on guileless individuals by making them to unveil their one of a kind data utilizing fake sites. In this work, an upgraded AI based prescient model is proposed to improve the effectiveness of against Hazard plans. The prescient model comprises of Feature Selection Module which is utilized for the development of a successful element vector. These highlights are removed from the URL, website page properties and site page conduct utilizing the gradual segment-based framework to introduce the resultant component vector to the prescient model. The proposed framework utilizes CNN, KNN AND SVM which have been prepared on a 30-dimensional list of capabilities. AI is an incredible asset used to endeavor against Hazard assaults

[5] K. Shima et al., “Classification of URL bitstreams using bag of bytes,” in 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN).

In present days, websites are main responsible for the rapid growth of criminal activities in the internet and corresponding activities which results in the many illegal things. So, there are many preventive steps to be taken to stop these kinds of activities. Here we propose a model which will classify the given URL into any of the three possible classes, i.e. Benign, spam and malware. Our model will the detect the classification of the URL without using any websites content.

CHAPTER – 3

PROBLEM DEFINITION

Web hazard detection and identification is a critical area of research in the field of cybersecurity. The internet has become an integral part of our lives, and with the increasing reliance on the internet, the number and sophistication of web-based hazards have also increased. Malware, phishing, spam, and other forms of cyber attacks can cause significant financial losses, identity theft, and other forms of harm to users. Therefore, detecting and identifying web hazards is essential to safeguard users' interests.

Machine learning is a powerful tool for detecting and identifying web hazards. Machine learning models can analyze vast amounts of data and identify patterns that may not be apparent to humans. The models can learn to recognize the characteristics of hazardous content and provide real-time protection against them. However, developing a robust and accurate machine learning model for web hazard detection and identification is a complex task that involves several challenges.

One of the primary challenges is data collection. Machine learning models require large amounts of high-quality data to learn and make accurate predictions. In the case of web hazard detection, the data must be diverse and representative of the different types of hazards present on the internet. Data collection involves identifying relevant data sources, collecting and pre-processing the data, and labelling the data to train the machine learning model.

Another challenge is feature selection. Machine learning models require features that can effectively differentiate between hazardous and non-hazardous content. Feature selection involves identifying the most relevant features that can accurately represent the characteristics of hazardous content. In the case of web hazard detection, features such as URL structure, HTML tags, and page content are commonly used.

Model selection is another challenge in web hazard detection and identification. There are several machine learning algorithms available, each with its strengths and weaknesses. Selecting the most appropriate algorithm for a given problem is essential for developing an accurate and efficient model. The model's accuracy and efficiency depend on various factors such as data quality, feature selection, and algorithm selection.

Finally, maintaining and updating the model is another challenge. The internet is constantly evolving, and new types of hazards emerge regularly. The machine learning model must be continually updated to keep up with these changes. The model must also adapt to new data sources and changes in the web landscape to remain effective.

In conclusion, web hazard detection and identification using machine learning is a complex problem that requires addressing several challenges. Data collection, feature selection, model selection, and model maintenance are all critical aspects of developing an accurate and efficient machine learning model. Nevertheless, machine learning can be a powerful tool for protecting users against web-based hazards and ensuring a safer and more secure internet.

CHAPTER - 4

COMPUTATIONAL ENVIRONMENT

4.1 HARDWARE SPECIFICATION

- Processor : I3/Intel Processor
- Hard Disk : More than 500 GB
- Key Board : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse
- Monitor : SVGA
- RAM : 8 GB

4.2 SOFTWARE SPECIFICATION

- Operating System : Windows 7/8/10
- Software's : Python 3.6 or high version
- IDE : PyCharm
- Framework : Flask

4.3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

- 1) Authentication of user whenever he/she logs into the system

- 2) System shutdown in case of a cyber-attack
- 3) A verification email is sent to user whenever he/she register for the first time on some software system.

Non-functional requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioural requirements.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

- 1) Emails should be sent with a latency of no greater than 12 hours from such an activity.
- 2) The processing of each request should be done within 10 seconds
- 3) The site should load in 3 seconds whenever of simultaneous users are > 10000

4.4 SOFTWARE FEATURES

SOFTWARE INSTALLATION FOR MACHINE LEARNING PROJECTS

Installing Python:

1.To download and install Python visit the official website of Python <https://www.python.org/downloads/> and choose your version.

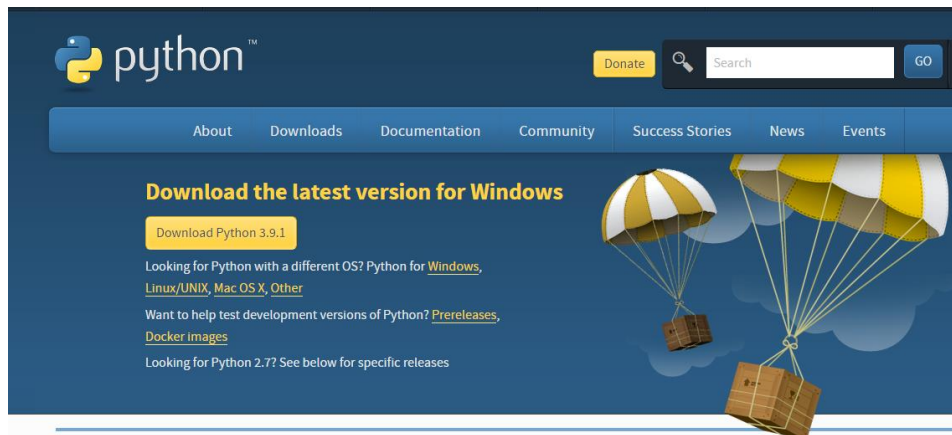


Fig.1.1. Python Installation

2. Once the download is complete, run the exe for install Python. Now click on Install Now.
3. You can see Python installing at this point.
4. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

Installing PyCharm:

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

Download PyCharm

[Windows](#) [Mac](#) [Linux](#)

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Download

Free trial

Community

For pure Python development

Download

Free, open-source

Fig.1.2. PyCharm Installation

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.

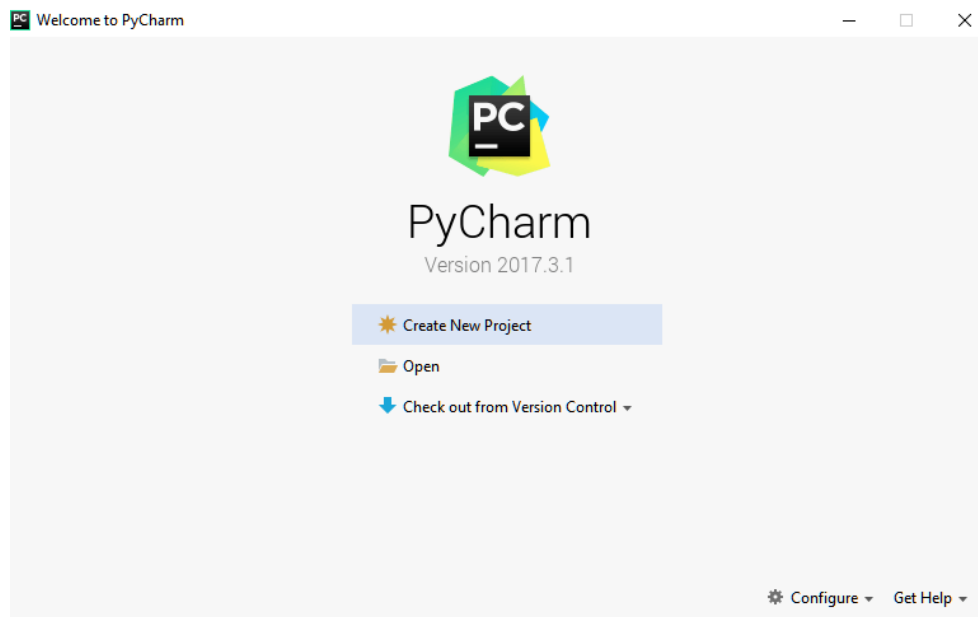


Fig.1.3. PyCharm Home Screen

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, b
Installing collected packages: numpy
Successfully installed numpy-1.18.5
```

Fig.1.4. Pip Installation

INTRODUCTION TO PYTHON

What Is A Script?

Up to this point, I have concentrated on the interactive programming capability of Python. This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode

Scripts are reusable

Basically, a script is a text file containing the statements that comprise a Python program. Once you have created the script, you can execute it over and over without having to retype it each time.

Scripts are editable

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor. Then you can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

Text editor

Just about any text editor will suffice for creating Python script files. You can use Microsoft Notepad, Microsoft WordPad, Microsoft Word, or just about any word processor if you want to.

Difference between a script and a program

Script:

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, whereas the applications they control are traditionally compiled to native machine code.

Program:

The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled)

Python

What is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don't know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to "big name" languages. Hopefully I can explain it for you.

Python concepts

If you're not interested in the hows and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it's a great one to start programming with.

- Open source general-purpose language.
- Object Oriented, Procedural, Functional
- Easy to interface with C/ObjC/Java/Fortran
- Easy-is to interface with C++ (via SWIG)
- Great interactive environment

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before ,executing it. This is similar to PERL and PHP.
- Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include

- Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read – Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain – Python's source code is fairly easy-to-maintain.
- A broad standard library – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable – you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases – Python provides interfaces to all major commercial databases.

- GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Dynamic vs. Static

Types Python is a dynamic-typed language. Many other languages are static typed, such as C/C++ and Java. A static typed language requires the programmer to explicitly tell the computer what type of “thing” each data value is.

For example, in C if you had a variable that was to contain the price of something, you would have to declare the variable as a “float” type. This tells the compiler that the only data that can be used for that variable must be a floating point number, i.e. a number with a decimal point. If any other data value was assigned to that variable, the compiler would give an error when trying to compile the program. Python, however, doesn’t require this. You simply give your variables names and assign values to them. The interpreter takes care of keeping track of what kinds of objects your program is using. This also means that you can change the size of the values as you develop the program. Say you have another decimal number (a.k.a. a floating point number) you need in your program.

With a static typed language, you have to decide the memory size the variable can take when you first initialize that variable. A double is a floating point value that can handle a much larger number than a normal float (the actual memory sizes depend on the operating environment). If you declare a variable to be a float but later on assign a value that is too big to it, your program will fail; you will have to go back and change that variable to be a double. With Python, it doesn't matter. You simply give it whatever number you want and Python will take care of manipulating it as needed. It even works for derived values.

For example, say you are dividing two numbers. One is a floating point number and one is an integer. Python realizes that it's more accurate to keep track of decimals so it automatically calculates the result as a floating point number

Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory. Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types –

- Numbers
- String
- List
- Tuple
- Dictionary

Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them.

Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([]) and their elements and size can be changed, while tuples are enclosed in parentheses (()) and cannot be updated. Tuples can be thought of as read-only lists.

Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any

Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Different modes in python

Python has two basic modes: normal and interactive.

The normal mode is the mode where the scripted and finished .py files are run in the Python interpreter.

Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole 20 Python libraries

1. Requests. The most famous http library written by Kenneth remits. It's a must have for every python developer.
2. Scrappy. If you are involved in web scraping then this is a must have library for you. After using this library you won't use any other.
3. Python. A guy toolkit for python. I have primarily used it in place of tinder. You will really love it.
4. Pillow. A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.
5. SQLAlchemy. A database library. Many love it and many hate it. The choice is yours.
6. Beautiful Soup. I know it's slow but this xml and html parsing library is very useful for beginners.
7. Twisted. The most important tool for any network application developer. It has a very beautiful ape and is used by a lot of famous python developers.
8. Numbly. How can we leave this very important library? It provides some advance math functionalities to python.

9. Skippy. When we talk about numbly then we have to talk about spicy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.
10. Matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyser.
11. Pygmy. Which developer does not like to play games and develop them? This library will help you achieve your goal of 2d game development.
12. Piglet. A 3d animation and game creation engine. This is the engine in which the famous python port of mine craft was made
13. Pit. A GUI toolkit for python. It is my second choice after python for developing GUI's for my python scripts.
14. Pit. Another python GUI library. It is the same library in which the famous Bit torrent client is created.
15. Scaly. A packet sniffer and analyser for python made in python.
16. Pywin32. A python library which provides some useful methods and classes for interacting with windows.
17. Notch. Natural Language Toolkit – I realize most people won't be using this one, but it's generic enough. It is a very useful library if you want to manipulate strings. But its capacity is beyond that. Do check it out.
18. Nose. A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.
19. Simply. Simply can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.
20. I Python. I just can't stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more. Make sure that you take a look at it.

Numpy

Numpy's main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In numpy dimensions are called axes. The number of axes is rank.

- Offers Matlab-ish capabilities within Python
- Fast array operations
- 2D arrays, multi-D arrays, linear algebra etc.

Matplotlib

- High quality plotting library.

Python class and objects

These are the building blocks of OOP. Class creates a new object. This object can be anything, whether an abstract data concept or a model of a physical object, e.g. a chair. Each class has individual characteristics unique to that class, including variables and methods. Classes are very powerful and currently “the big thing” in most programming languages. Hence, there are several chapters dedicated to OOP later in the book.

The class is the most basic component of object-oriented programming. Previously, you learned how to use functions to make your program do something. Now will move into the big, scary world of Object-Oriented Programming (OOP). To be honest, it took me several months to get a handle on objects. When I first learned C and C++, I did great; functions just made sense for me. Having messed around with BASIC in the early '90s, I realized functions were just like subroutines so there wasn't much new to learn. However, when my C++ course started talking about objects, classes, and all the new features of OOP, my grades definitely suffered. Once you learn OOP, you'll realize that it's actually a pretty powerful tool. Plus many Python libraries and APIs use classes, so you should at least be able to understand what the code is doing.

One thing to note about Python and OOP: it's not mandatory to use objects in your code in a way that works best; maybe you don't need to have a full-blown class with initialization code and methods to just return a calculation. With Python, you can get as technical as you want. As you've already seen, Python can do just fine with functions. Unlike

languages such as Java, you aren't tied down to a single way of doing things; you can mix functions and classes as necessary in the same program. This lets you build the code. Objects are an encapsulation of variables and functions into a single entity. Objects get their variables and functions from classes. Classes are essentially a template to create your objects.

Here's a brief list of Python OOP ideas:

- The class statement creates a class object and gives it a name. This creates a new namespace.
- Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: `ClassName.Attribute`.
- Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class.
- Calling a class (just like a function) creates a new instance of the class.

This is where the multiple copies part comes in.

- Each instance gets ("inherits") the default class attributes and gets its own namespace. This prevents instance objects from overlapping and confusing the program.
- Using the term `self` identifies a particular instance, allowing for per-instance attributes. This allows items such as variables to be associated with a particular instance.

Inheritance

First off, classes allow you to modify a program without really making changes to it. To elaborate, by subclassing a class, you can change the behavior of the program by simply adding new components to it rather than rewriting the existing components. As we've seen, an instance of a class inherits the attributes of that class. However, classes can also inherit attributes from other classes. Hence, a subclass inherits from a superclass allowing you to make a generic superclass that is specialized via subclasses. The subclasses can override the logic in a superclass, allowing you to change the behavior of your classes without changing the superclass at all.

Operator Overloads

Operator overloading simply means that objects that you create from classes can respond to actions (operations) that are already defined within Python, such as addition, slicing, printing, etc. Even though these actions can be implemented via class methods, using overloading ties the behavior closer to Python's object model and the object interfaces are more consistent to Python's built-in objects, hence overloading is easier to learn and use. User-made classes can override nearly all of Python's built-in operation methods

Exceptions

I've talked about exceptions before but now I will talk about them in depth. Essentially, exceptions are events that modify program's flow, either intentionally or due to errors. They are special events that can occur due to an error, e.g. trying to open a file that doesn't exist, or when the program reaches a marker, such as the completion of a loop. Exceptions, by definition, don't occur very often; hence, they are the "exception to the rule" and a special class has been created for them. Exceptions are everywhere in Python. Virtually every module in the standard Python library uses them, and Python itself will raise them in a lot of different circumstances.

Here are just a few examples:

- Accessing a non-existent dictionary key will raise a Key Error exception.
- Searching a list for a non-existent value will raise a Value Error exception
- Calling a non-existent method will raise an Attribute Error exception.
- Referencing a non-existent variable will raise a Name Error exception.
- Mixing data types without coercion will raise a Type Error exception.

One use of exceptions is to catch a fault and allow the program to continue working; we have seen this before when we talked about files. This is the most common way to use exceptions. When programming with the Python command line interpreter, you don't need to worry about catching exceptions. Your program is usually short enough to not be hurt too much if an exception occurs. Plus, having the exception occur at the command line is a quick and easy way to tell if your code logic has a problem. However, if the same error occurred in your real program, it will fail and stop working. Exceptions can be created manually in the code by raising an exception. It operates exactly as a system-caused exceptions, except that

the programmer is doing it on purpose. This can be for a number of reasons. One of the benefits of using exceptions is that, by their nature, they don't put any overhead on the code processing. Because exceptions aren't supposed to happen very often, they aren't processed until they occur. Exceptions can be thought of as a special form of the if/else statements. You can realistically do the same thing with if blocks as you can with exceptions. However, as already mentioned, exceptions aren't processed until they occur; if blocks are processed all the time. Proper use of exceptions can help the performance of your program. The more infrequent the error might occur, the better off you are to use exceptions; using if blocks requires Python to always test extra conditions before continuing. Exceptions also make code management easier: if your programming logic is mixed in with error-handling if statements, it can be difficult to read, modify, and debug your program.

User-Defined Exceptions

I won't spend too much time talking about this, but Python does allow for a programmer to create his own exceptions. You probably won't have to do this very often but it's nice to have the option when necessary. However, before making your own exceptions, make sure there isn't one of the built-in exceptions that will work for you. They have been "tested by fire" over the years and not only work effectively, they have been optimized for performance and are bug-free. Making your own exceptions involves object-oriented programming, which will be covered in the next chapter. To make a custom exception, the programmer determines which base exception to use as the class to inherit from, e.g. making an exception for negative numbers or one for imaginary numbers would probably fall under the Arithmetic Error exception class. To make a custom exception, simply inherit the base exception and define what it will do.

Python modules

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library.

To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a module; definitions from a module can be imported into other modules or into the main module.

Testing code

As indicated above, code is usually developed in a file using an editor. To test the code, import it into a Python session and try to run it. Usually there is an error, so you go back to the file, make a correction, and test again. This process is repeated until you are satisfied that the code works. This entire process is known as the development cycle. There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid. This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

Functions in Python

It is possible, and very useful, to define our own functions in Python. Generally speaking, if you need to do a calculation only once, then use the interpreter. But when you or others have need to perform a certain type of calculation many times, then define a function.

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task. To carry out that specific task, the function might or might not need multiple inputs. When the task is carved out, the function can or cannot return one or more values.

There are three types of functions in python:

Help (), min (), print ().

Python Namespace

Generally speaking, a namespace (sometimes also called a context) is a naming system for making names unique to avoid ambiguity. Everybody knows a name spacing system from daily life, i.e. the naming of people in first name and family name (surname). An example is a network: each network device (workstation, server, printer,) needs a unique

name and address. Yet another example is the directory structure of file systems. The same file name can be used in different directories, the files can be uniquely accessed via the pathnames. Many programming languages use namespaces or contexts for identifiers. An identifier defined in a namespace is associated with that namespace. This way, the same identifier can be independently defined in multiple namespaces. (Like the same file names in different directories) Programming languages, which support namespaces, may have different rules that determine to which namespace an identifier belongs. Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces.

Some namespaces in Python:

- Global names of a module
- Local names in a function or method invocation
- Built-in names: this namespace contains built-in functions (e.g. `abs()`, `camp()`, ...) and built-in exception names.

Garbage Collection

Garbage Collector exposes the underlying memory management mechanism of Python, the automatic garbage collector. The module includes functions for controlling how the collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

Python XML Parser

XML is a portable, open source language that allows programmers to develop applications that can be read by other applications, regardless of operating system and/or developmental language.

What is XML? The Extensible Markup Language XML is a markup language much like HTML or SGML. This is recommended by the World Wide Web Consortium and available as an open standard. XML is extremely useful for keeping track of small to medium amounts of data without requiring a SQL-based backbone. XML Parser Architectures and APIs the Python standard library provides a minimal but useful set of interfaces to work with

XML. The two most basic and broadly used APIs to XML data are the SAX and DOM interfaces. Simple API for XML SAX: Here, you register call backs for events of interest and then let the parser proceed through the document. This is useful when your documents are large or you have memory limitations, it parses the file as it reads it from disk and the entire file is never stored in memory.

Document Object Model DOM API : This is a World Wide Web Consortium recommendation wherein the entire file is read into memory and stored in a hierarchical tree – based form to represent all the features of an XML document.

SAX obviously cannot process information as fast as DOM can when working with large files. On the other hand, using DOM exclusively can really kill your resources, especially if used on a lot of small files. SAX is read-only, while DOM allows changes to the XML file. Since these two different APIs literally complement each other, there is no reason why you cannot use them both for large projects.

Python Web Frameworks

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

Why are web frameworks useful?

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Common web framework functionality

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URL routing
2. HTML, XML, JSON, and other output format templating
3. Database manipulation
4. Security against Cross-site request forgery (CSRF) and other attacks
5. Session storage and retrieval

Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the "batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

Comparing web frameworks

There is also a repository called `compare-python-web-frameworks` where the same web application is being coded with varying Python web frameworks, templating engines and object.

Web framework resources

- When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
- Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
- What is a web framework? Is an in-depth explanation of what web frameworks are and their relation to web servers?
- Jingo vs. Flash vs. Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
- This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.
- Python's web frameworks benchmarks is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.

- What web frameworks do you use and why are they awesome? Is a language agnostic Reedit discussion on web frameworks? It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
- This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?" The votes aren't as important as the list of the many frameworks that are available to Python developers.

Web frameworks learning checklist

1. Choose a major Python web framework (Jingo or Flask are recommended) and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.
2. Work through a detailed tutorial found within the resources links on the framework's page.
3. Study open source examples built with your framework of choice so you can take parts of those projects and reuse the code in your application. Build the first simple iteration of your web application then go to the deployment section to make it accessible on the web.

CHAPTER – 5

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

5.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the

users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER – 6

SYSTEM ANALYSIS

6.1 EXISTING METHOD

Existing systems for Hazard Identification and Detection using machine learning often rely on supervised learning techniques that require labeled data for training. This means that a large dataset of both legitimate and Hazard Identification and Detection must be manually labeled, which can be time-consuming and costly.

Drawbacks :

1. Disadvantage is that these systems may be vulnerable to adversarial attacks, where attackers can manipulate the website features to evade detection. Additionally, machine learning-based systems may require significant computational resources, which could limit their scalability.
2. Finally, these systems may produce false positives or false negatives, which can impact the user experience and reduce the effectiveness of the system. These limitations highlight the need for ongoing research and development in the field of machine learning-based Hazard website detection.

6.2 PROPOSED METHOD

The proposed system for Hazard Identification and Detection using machine learning algorithms aims to overcome the limitations of existing systems. One approach is to use supervised learning techniques that do not require labelled data for training. This can reduce the time and cost of data labelling and improve scalability.

Advantages :

1. Another approach is to incorporate multiple machine learning algorithms to enhance the accuracy and robustness of the system. The system can also be augmented with additional features such as website behaviour analysis and user behaviour monitoring to improve its ability to detect Hazard websites.
2. Overall, the proposed system aims to improve the accuracy, efficiency, and effectiveness of Hazard Identification and Detection using machine learning algorithms.

CHAPTER – 7

SYSTEM DESIGN

7.1 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML 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 Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful insists the modeling 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.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

7.1.1 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

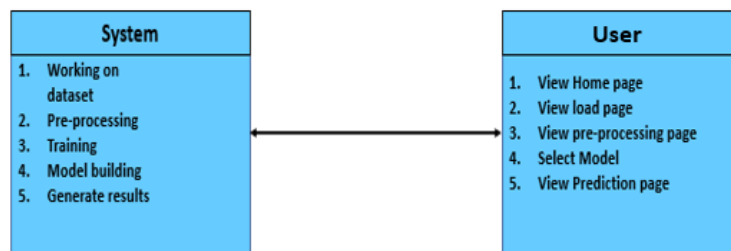


Fig.2.1. Class Diagram

7.1.2 USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

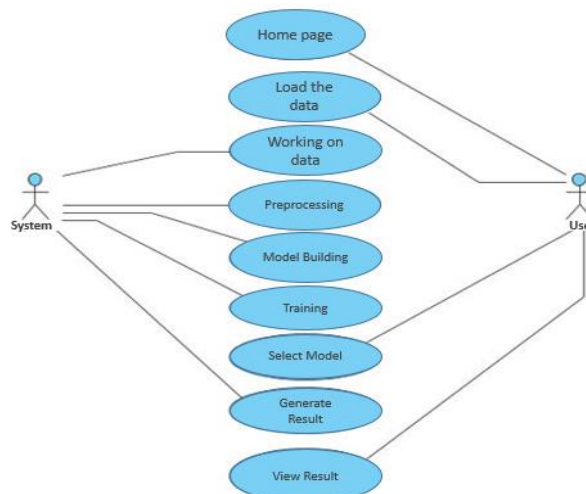


Fig.2.2. Use Case Diagram

7.1.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

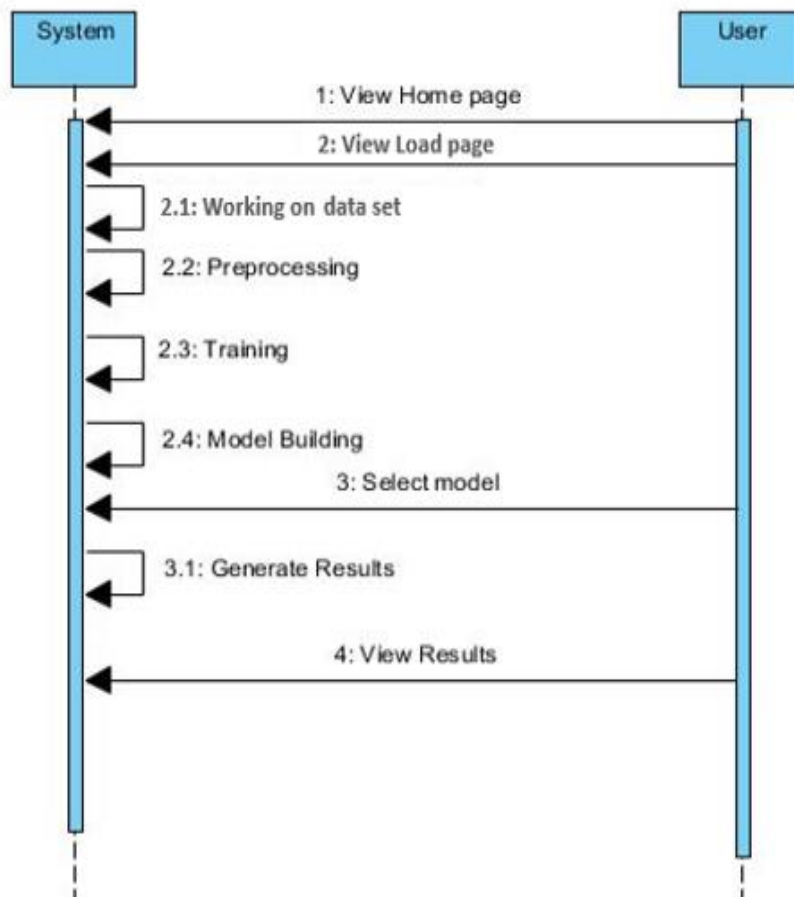


Fig.2.3. Sequence Diagram

7.1.4 COLLABORATION DIAGRAM

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that

the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

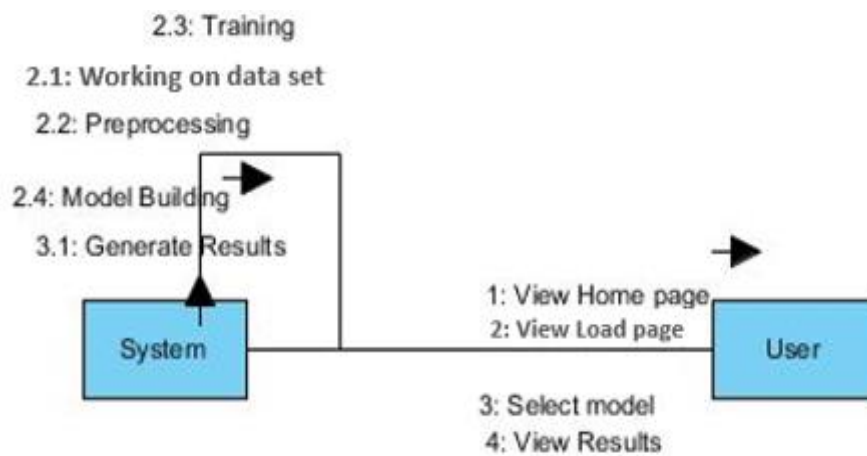


Fig.2.4. Collaboration Diagram

6.1.5 DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

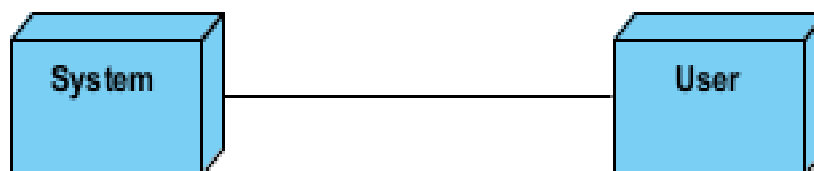


Fig.2.5. Deployment Diagram

7.1.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

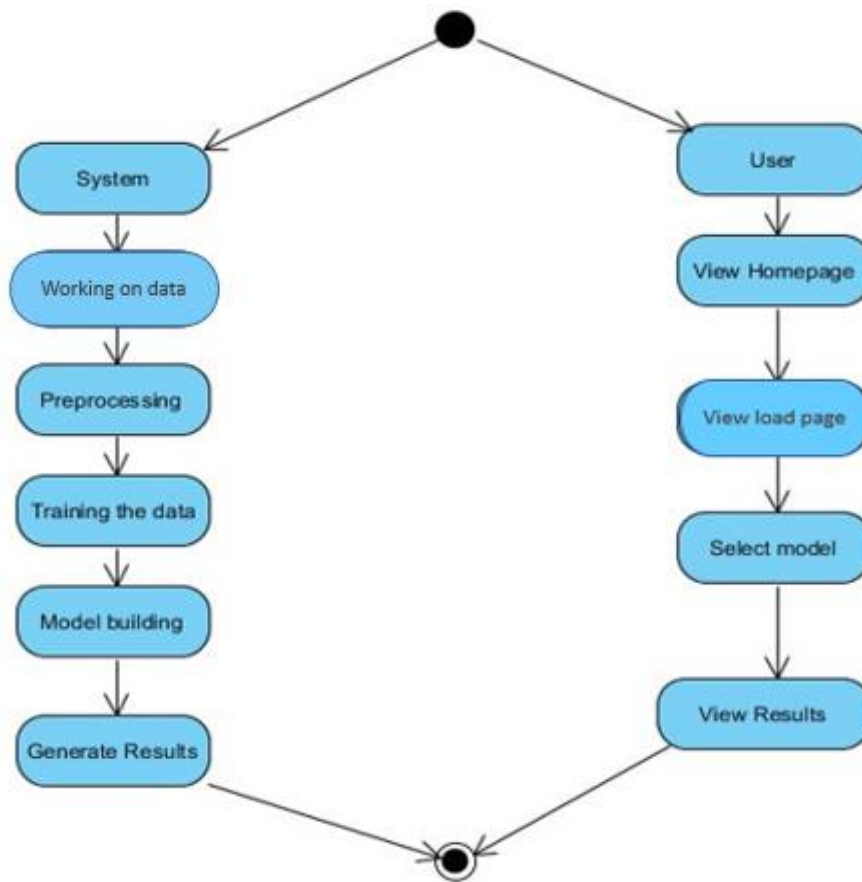


Fig.2.6. Activity Diagram

7.1.7 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



Fig.2.7. Component Diagram

7.1.8 ER DIAGRAM

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

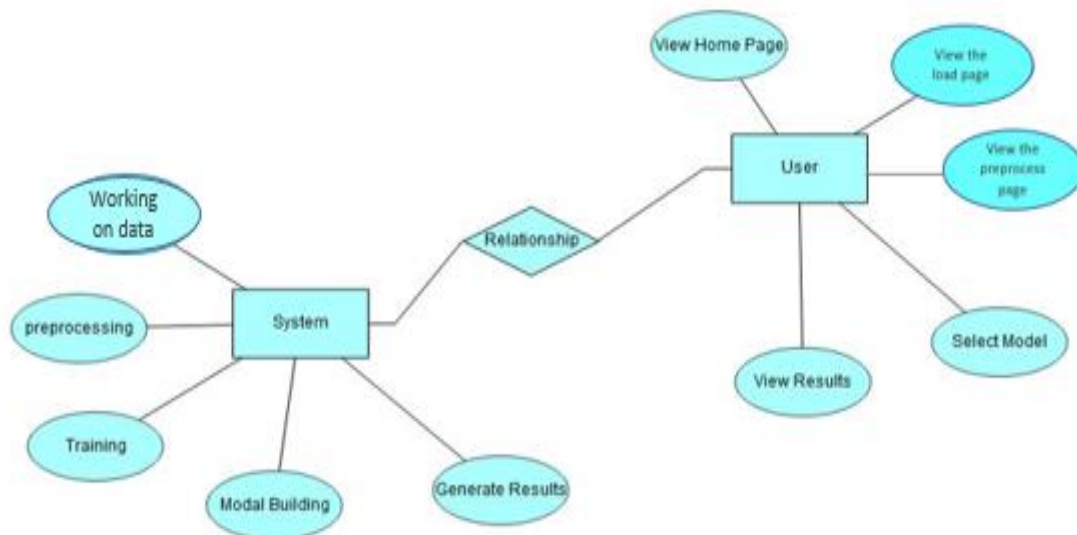


Fig.2.8. ER Diagram

7.1.9 DFD DIAGRAM

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

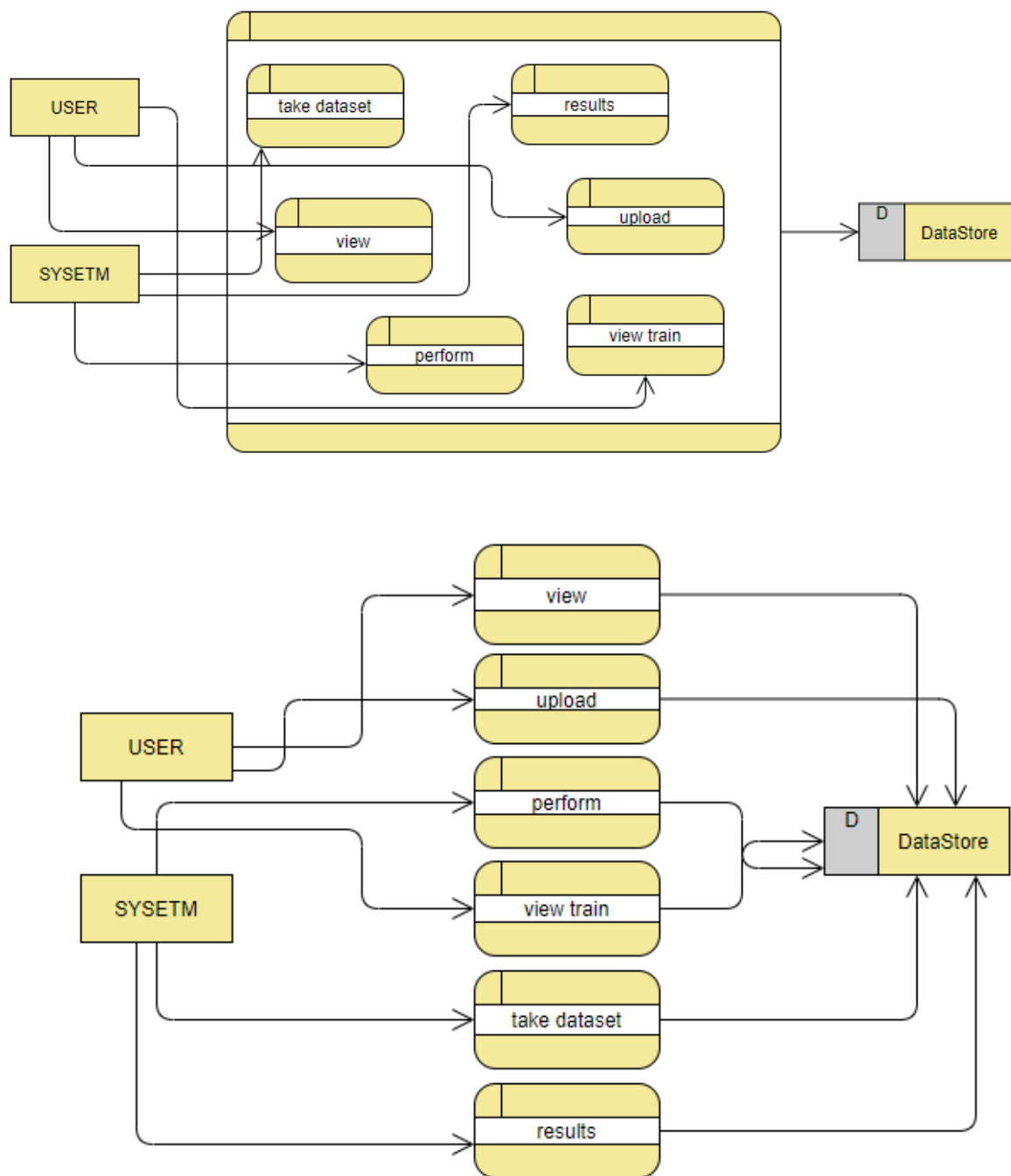


Fig.2.9. DFD Diagram

7.1.10 ARCHITECTURE

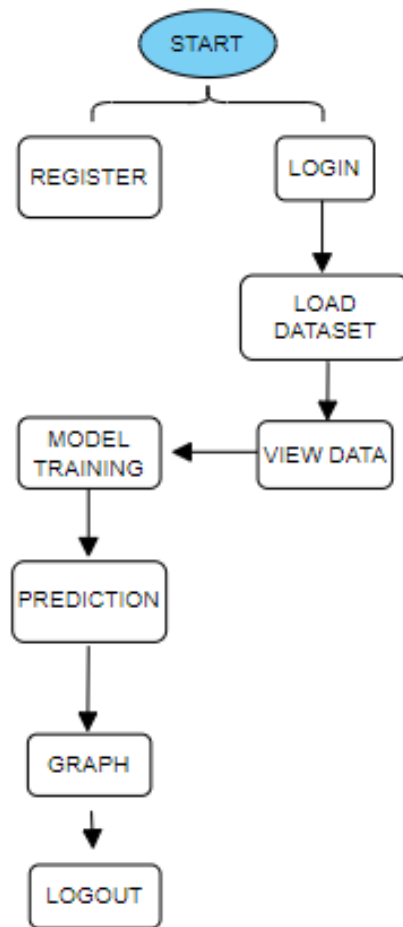


Fig.2.10. Architecture

CHAPTER – 8

SYSTEM IMPLEMENTATION

8.1 IMPLEMENTATION PROCESS

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

8.2 MODULES

1. User:

1.1 View Home page:

Here user view the home page of the Hazard website prediction web application.

1.2 View Upload page:

In the about page, users can learn more about the Hazard prediction.

View Page:

In view page, user can see the dataset.

1.3 Input Model:

The user must provide input values for the certain fields in order to get results.

1.4 View Results:

User view's the generated results from the model.

1.5 View score:

Here user have ability to view the score in %

2. System:

2.1 Working on dataset:

System checks for data whether it is available or not and load the data in csv files.

2.2 Pre-processing:

Data need to be pre-processed according the models it helps to increase the accuracy of the model and better information about the data.

2.3 Training the data:

After pre-processing the data will split into two parts as train and test data before training with the given algorithms.

2.4 Model Building

To create a model that predicts the personality with better accuracy, this module will help user.

2.5 Generated Score:

Here user view the score in %

2.6 Generate Results:

We train the machine learning algorithm and calculate the personality prediction.

CHAPTER – 9

ALGORITHMS

9.1 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

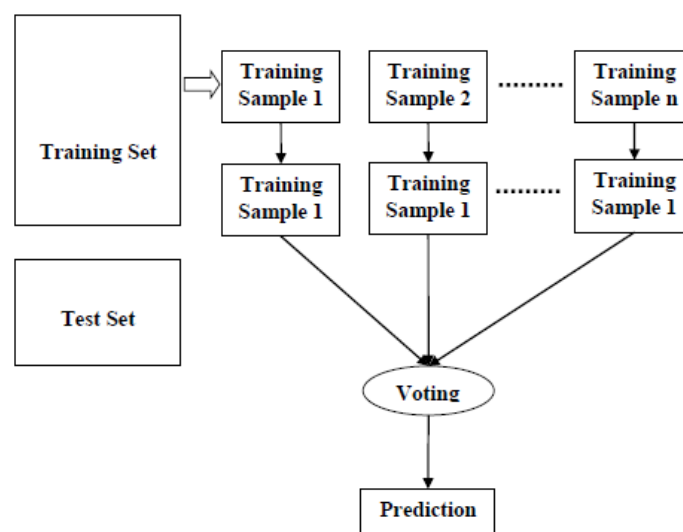


Fig.3.1. Random Forest Algorithm

9.2 ADABOOST CLASSIFIER

AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances. Boosting is used to reduce bias as well as variance for supervised learning. It works on the principle of learners growing sequentially. Except for the first, each subsequent learner is grown from previously grown learners. In simple words, weak learners are converted into strong ones. The AdaBoost algorithm works on the same principle as boosting with a slight difference. Let's discuss this difference in detail.

First, let us discuss how boosting works. It makes 'n' number of decision trees during the data training period. As the first decision tree/model is made, the incorrectly classified record in the first model is given priority. Only these records are sent as input for the second model. The process goes on until we specify a number of base learners we want to create. Remember, repetition of records is allowed with all boosting techniques.

This figure shows how the first model is made and errors from the first model are noted by the algorithm. The record which is incorrectly classified is used as input for the next model. This process is repeated until the specified condition is met. As you can see in the figure, there are 'n' number of models made by taking the errors from the previous model. This is how boosting works. The models 1,2, 3,..., N are individual models that can be known as decision trees. All types of boosting models work on the same principle.

Since we now know the boosting principle, it will be easy to understand the AdaBoost algorithm. Let's dive into AdaBoost's working. When the random forest is used, the algorithm makes an 'n' number of trees. It makes proper trees that consist of a start node with several leaf nodes. Some trees might be bigger than others, but there is no fixed depth in a random forest. With AdaBoost, however, the algorithm only makes a node with two leaves, known as Stump.

The figure here represents the stump. It can be seen clearly that it has only one node with two leaves. These stumps are weak learners and boosting techniques prefer this. The order of stumps is very important in AdaBoost. The error of the first stump influences how other stumps are made. Let's understand this with an example.

Here's a sample dataset consisting of only three features where the output is in categorical form. The image shows the actual representation of the dataset. As the output is in binary/categorical form, it becomes a classification problem. In real life, the dataset can have any number of records and features in it. Let us consider 5 datasets for explanation purposes. The output is in categorical form, here in the form of Yes or No. All these records will be assigned a sample weight. The formula used for this is ' $W=1/N$ ' where N is the number of records. In this dataset, there are only 5 records, so the sample weight becomes $1/5$ initially. Every record gets the same weight. In this case, it's $1/5$.

Learn AdaBoost Model from Data

Ada Boosting is best used to boost the performance of decision trees and this is based on binary classification problems. AdaBoost was originally called AdaBoost.M1 by the author. More recently it may be referred to as discrete Ada Boost. As because it is used for classification rather than regression. AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners.

9.3 XGBOOST

XGBoost stands for "Extreme Gradient Boosting". XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.

Boosting

Boosting is an ensemble learning technique to build a strong classifier from several weak classifiers in series. Boosting algorithms play a crucial role in dealing with bias-variance trade-off. Unlike bagging algorithms, which only controls for high variance in a model, boosting controls both the aspects (bias & variance) and is considered to be more effective.

Below are the few types of boosting algorithms:

- AdaBoost (Adaptive Boosting)
- Gradient Boosting

- XGBoost
- CatBoost
- Light GBM

XGBoost

XGBoost stands for eXtreme Gradient Boosting. It became popular in the recent days and is dominating applied machine learning and Kaggle competitions for structured data because of its scalability. XGBoost is an extension to gradient boosted decision trees (GBM) and specially designed to improve speed and performance.

9.4 GRADIENT BOOSTING CLASSIFIER

Gradient boosting algorithm is one of the most powerful algorithms in the field of machine learning. As we know that the errors in machine learning algorithms are broadly classified into two categories i.e. Bias Error and Variance Error. As gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

Unlike, Ada boosting algorithm, the base estimator in the gradient boosting algorithm cannot be mentioned by us. The base estimator for the Gradient Boost algorithm is fixed and i.e. Decision Stump. Like, AdaBoost, we can tune the `n_estimator` of the gradient boosting algorithm. However, if we do not mention the value of `n_estimator`, the default value of `n_estimator` for this algorithm is 100.

Gradient boosting algorithm can be used for predicting not only continuous target variable (as a Regressor) but also categorical target variable (as a Classifier). When it is used as a regressor, the cost function is Mean Square Error (MSE) and when it is used as a classifier then the cost function is Log loss.

Let us now understand the working of the Gradient Boosting Algorithm with the help of one example. In the following example, Age is the Target variable whereas LikesExercising, GotoGym, DrivesCar are independent variables. As in this example, the target variable is continuous, Gradient Boosting Regressor is used here.

Let us now find out the estimator-2. Unlike AdaBoost, in the Gradient boosting algorithm, residues ($\text{age}_i - \mu$) of the first estimator are taken as root nodes as shown below. Let us suppose for this estimator another dependent variable is used for prediction. So, the records with False GotoGym.

9.5 SUPPORT VECTOR MACHINE (SVM)

The objective of the support vector machine algorithm is to find a hyper plane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

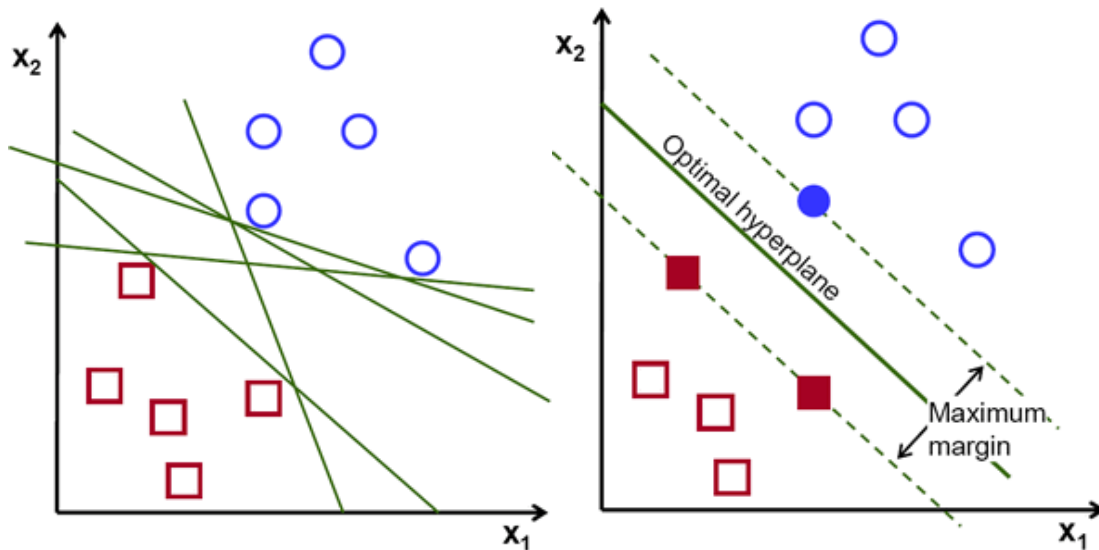


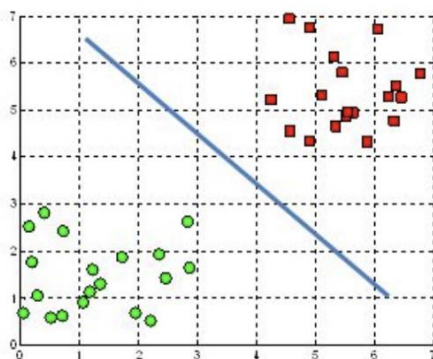
Fig.3.2. SVM

Possible hyper planes :

To separate the two classes of data points, there are many possible Hyper planes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyper planes and Support Vectors

A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane

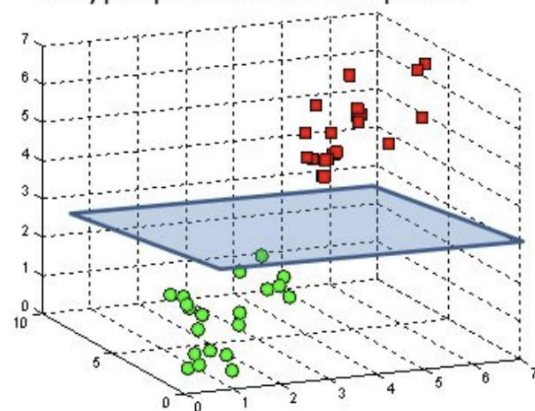


Fig.3.3. Hyper Planes

Hyper planes in 2D and 3D feature space

Hyper planes are decision boundaries that help classify the data points. Data points falling on either side of the hyper plane can be attributed to different classes. Also, the dimension of the hyper plane depends upon the number of features. If the number of input features is 2, then the hyper plane is just a line. If the number of input features is 3, then the hyper plane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

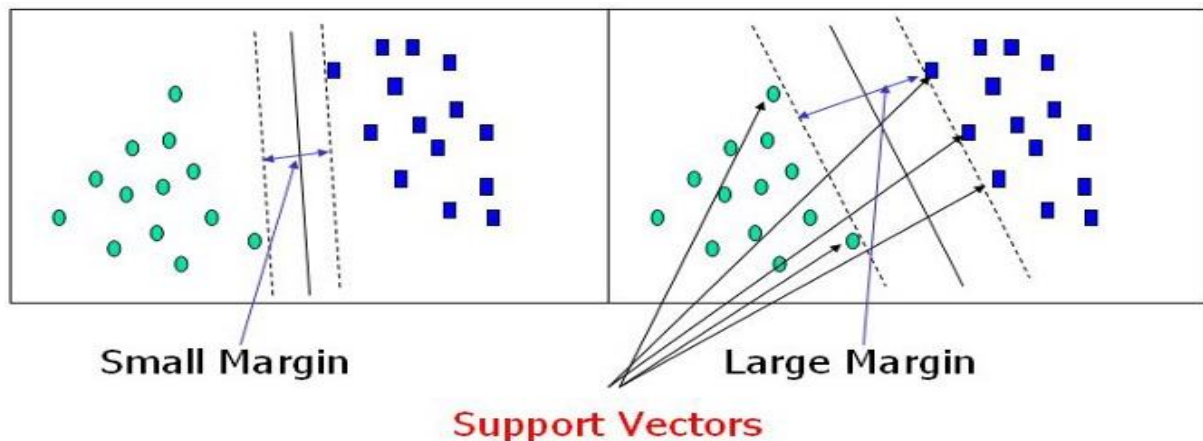


Fig.3.4. SVM Margins

Support Vectors

Support vectors are data points that are closer to the hyper plane and influence the position and orientation of the hyper plane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyper plane. These are the points that help us build our SVM.

Large Margin Intuition

In logistic regression, we take the output of the linear function and squash the value within the range of $[0,1]$ using the sigmoid function. If the squashed value is greater than a threshold value (0.5) we assign it a label 1, else we assign it a label 0. In SVM, we take the output of the linear function and if that output is greater than 1, we identify it with one class and if the output is -1, we identify it with another class. Since the threshold values are changed to 1 and -1 in SVM, we obtain this reinforcement range of values $([-1, 1])$ which acts as margin.

Cost Function and Gradient Updates

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyper plane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

$$c(x, y, f(x)) = (1 - y * f(x))_+$$

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^n (1 - y_i \langle x_i, w \rangle)_+$$

Hinge loss function (function on left can be represented as a function on the right)

The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter to the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost function looks as below

Loss function for SVM

Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.

$$\frac{\delta}{\delta w_k} \lambda \|w\|^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} (1 - y_i \langle x_i, w \rangle)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

Gradients

When there is no misclassification, i.e. our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.

$$w = w - \alpha \cdot (2\lambda w)$$

Gradient Update — No misclassification

When there is a misclassification, i.e. our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update.

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

CHAPTER – 10

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring. Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

10.1 TYPES OF TESTS

10.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

10.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

10.1.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

10.1.4 WHITE BOX TESTING

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

10.1.5 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

10.1.6 ACCEPTANCE TESTING

Acceptance testing involves planning and execution of functional tests, performance tests and stress tests in order to check whether the system implemented satisfies the requirements specifications. Quality assurance people as well as customers may simultaneously develop acceptance tests and run them. In addition to functional and performance tests, stress test are performed to determine the limits/limitations of the system developed. For example, a compiler may be tested for its symbol table overflows or a real-time system may be tested for multiple interrupts of different/same priorities.

TESTING OBJECTIVES

Testing is a process of execution a program with the intent of finding on errors. A good test is one that has a high probability of finding an undiscovered errors. Testing is vital to the success of the system. System testing is the state of implementation, which ensures that the system works accurately before live operations commence. System testing makes a logical assumption that the system is correct and that the system is correct and that the goals are successfully achieved.

CHAPTER – 11

SOURCE CODE

```
import os
from flask import *
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from urllib.parse import urlparse
import ipaddress
import re
from bs4 import BeautifulSoup
import whois
import urllib
import urllib.request
from datetime import datetime
import requests

from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import mysql.connector
db=mysql.connector.connect(user="root",password="",port='3306',database='phishing')
cur=db.cursor()

app = Flask(__name__)
app.secret_key = "fghhdfgdfgrthrttgdfsadfsaffgd"

app.config['upload folder'] =r'uploads'
top_doms = pd.read_csv('top-1m.csv', header=None)
```

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

@app.route('/login',methods=['POST','GET'])
def login():
    if request.method=='POST':
        useremail=request.form['useremail']
        session['useremail']=useremail
        userpassword=request.form['userpassword']
        sql="select      count(*)      from      user      where      Email='%s'      and
Password='%s'"%(useremail,userpassword)

        # cur.execute(sql)
        # data=cur.fetchall()
        # db.commit()
        x=pd.read_sql_query(sql,db)
        print(x)
        print('#####')
        count=x.values[0][0]

        if count==0:
            msg="user Credentials Are not valid"
            return render_template("login.html",name=msg)
        else:
            s="select      *      from      user      where      Email='%s'      and
Password='%s'"%(useremail,userpassword)
            z=pd.read_sql_query(s,db)
            session['email']=useremail
            pno=str(z.values[0][4])
            print(pno)
            name=str(z.values[0][1])
            print(name)
```

```
session['pno']=pno
session['name']=name
return render_template("userhome.html",myname=name)
return render_template('login.html')
@app.route('/registration',methods=["POST","GET"])
def registration():
    if request.method=='POST':
        username=request.form['username']
        useremail = request.form['useremail']
        userpassword = request.form['userpassword']
        conpassword = request.form['conpassword']
        Age = request.form['Age']

        contact = request.form['contact']
        if userpassword == conpassword:
            sql="select      *      from      user      where      Email='%s'      and
Password='%s'"%(useremail,userpassword)
            cur.execute(sql)
            data=cur.fetchall()
            db.commit()
            print(data)
            if data==[]:

                sql = "insert into user(Name,Email>Password,Age,Mob)values(%s,%s,%s,%s,%s)"
                val=(username,useremail,userpassword,Age,contact)
                cur.execute(sql,val)
                db.commit()
                flash("Registered successfully","success")
                return render_template("login.html")
            else:
                flash("Details are invalid","warning")
                return render_template("registration.html")
```

```
    else:
        flash("Password doesn't match", "warning")
        return render_template("registration.html")
    return render_template('registration.html')

@app.route('/load data',methods = ["POST","GET"])
def load_data():
    if request.method == "POST":
        file = request.files['file']
        filetype = os.path.splitext(file.filename)[1]
        print(filetype)
        if filetype == '.csv':
            mypath = os.path.join(app.config['upload folder'],file.filename)
            file.save(mypath)
            return render_template('load data.html',msg = 'success')
        else:
            return render_template('load data.html',msg = 'invalid')
    return render_template('load data.html')

@app.route('/view data',methods = ["POST","GET"])
def view_data():
    path = os.listdir(app.config['upload folder'])
    file = os.path.join(app.config['upload folder'],path[0])
    df = pd.read_csv(file)
    df = pd.read_csv('uploads/url_data_modified.csv')
    return render_template('view data.html',col_name = df.columns,row_val =
list(df.values.tolist()))

@app.route('/model',methods = ['GET',"POST"])
def model():
    global score1,score2,score3, score4,score5
```

```
if request.method == "POST":
    model = int(request.form['selected'])
    print(model)
    path = os.listdir(app.config['upload folder'])
    file = os.path.join(app.config['upload folder'], path[0])
    df = pd.read_csv(file)
    df = pd.read_csv('uploads/url_data_modified.csv')
    print(df.columns)
    print('#####')
    X = df.drop(['Label', 'Domain', 'Web_Traffic'], axis = 1)
    y = df.Label
    x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 20)
    print(df)
    if model == 1:
        from sklearn.ensemble import RandomForestClassifier
        rfr = RandomForestClassifier()
        x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 20)
        rfr.fit(x_train, y_train)
        pred = rfr.predict(x_test)
        score1 = accuracy_score(y_test, pred) * 100
        print(score1)
        msg = 'The accuracy obtained by Random Forest Classifier is ' + str(score1) + str('%')
        return render_template('model.html', msg=msg)
    elif model == 2:
        classifier = AdaBoostClassifier()
        classifier.fit(x_train, y_train)
        pred = classifier.predict(x_test)
        score2 = accuracy_score(y_test, pred) * 100
        print(score2)
        msg = 'The accuracy obtained by AdaBoost Classifier is ' + str(score2) + str('%')
        return render_template('model.html', msg=msg)
    elif model == 3:
```



```
from xgboost import XGBClassifier
xgb = XGBClassifier()
xgb.fit(x_train,y_train)
pred = xgb.predict(x_test)
score3 = accuracy_score(y_test, pred)*100
print(score3)
msg = 'The accuracy obtained by XGBoost Classifier is ' + str(score3) + str('%')
return render_template('model.html',msg=msg)
elif model ==4:
    cf = SVC(kernel='linear')
    cf.fit(x_train,y_train)
    pred = cf.predict(x_test)
    score4 = accuracy_score(y_test, pred)*100
    print(score4)
    msg = 'The accuracy obtained by Support Vector Machine is ' + str(score4) + str('%')
    return render_template('model.html',msg=msg)
elif model ==5:
    gb = GradientBoostingClassifier()
    gb.fit(x_train,y_train)
    pred = gb.predict(x_test)
    score5 = accuracy_score(y_test, pred)*100
    print(score5)
    msg = 'The accuracy obtained by Gradient Boosting Classifier is ' + str(score5) +
str('%')
    return render_template('model.html',msg=msg)
return render_template('model.html')

@app.route('/prediction', methods=["POST","GET"])
def prediction():
    if request.method == "POST":
        url1 = request.form['a']
        def getDomain(url):
```

```
domain = urlparse(url).netloc
if re.match(r"^www.", domain):
    domain = domain.replace("www.", "")
return domain

def havingIP(url):
    try:
        ipaddress.ip_address(url)
        ip = 1
    except:
        ip = 0
    return ip

def haveAtSign(url):
    if "@" in url:
        at = 1
    else:
        at = 0
    return at

def getLength(url):
    if len(url) < 54:
        length = 0
    else:
        length = 1
    return length

def getDepth(url):
    s = urlparse(url).path.split('/')
    depth = 0
    for j in range(len(s)):
        if len(s[j]) != 0:
```

```
        depth = depth + 1
    return depth

def redirection(url):
    pos = url.rfind('/')
    if pos > 6:
        if pos > 7:
            return 1
        else:
            return 0
    else:
        return 0

def httpDomain(url):
    domain = urlparse(url).netloc
    if 'https' in domain:
        return 1
    else:
        return 0

shortening_services =
r"bit\.ly|goo\.gl|shorte\.st|go2l\.ink|x\.co|ow\.ly|t\.co|tinyurl|tr\.im|is\.gd|cli\.gs|" \

r"yfrog\.com|migre\.me|ff\.im|tiny\.cc|url4\.eu|twit\.ac|su\.pr|twurl\.nl|snipurl\.com|" \

r"short\.to|BudURL\.com|ping\.fm|post\.ly|Just\.as|bkite\.com|snipr\.com|fic\.kr|loopt\.us|" \

r"doiop\.com|short\.ie|kl\.am|wp\.me|rubyurl\.com|om\.ly|to\.ly|bit\.do|t\.co|lnkd\.in|db\.tt|" \

r"qr\.ae|adf\.ly|goo\.gl|bitly\.com|cur\.lv|tinyurl\.com|ow\.ly|bit\.ly|ity\.im|q\.gs|is\.gd|" \

r"po\.st|bc\.vc|twitthis\.com|u\.to|j\.mp|buzurl\.com|cutt\.us|u\.bb|yourls\.org|x\.co|" \
```

```
r"prettylinkpro\.com|scrnch\.me|filoops\.info|vzturl\.com|qr\.net|lurl\.com|tweez\.me|v\.gd|" \
r"tr\.im|link\.zip\.net"
```

```
def tinyURL(url):
    match = re.search(shortening_services, url)
    if match:
        return 1
    else:
        return 0
```

```
def prefixSuffix(url):
    if '-' in urlparse(url).netloc:
        return 1 # phishing
    else:
        return 0 # legitimate
```

```
# def web_traffic(url):
#     try:
#         # Filling the whitespaces in the URL if any
#         url = urllib.parse.quote(url)
#         rank = \
#
```

```
BeautifulSoup(urllib.request.urlopen("http://data.alexas.com/data?cli=10&dat=s&url=" +
url).read(),
#         "xml").find(
#         "REACH")['RANK']
#     rank = int(rank)
# except TypeError:
#     return 1
#     if rank < 100000:
#         return 1
```

```
# else:
#     return 0

def domainAge(domain_name):
    creation_date = domain_name.creation_date
    expiration_date = domain_name.expiration_date
    if (isinstance(creation_date, str) or isinstance(expiration_date, str)):
        try:
            creation_date = datetime.strptime(creation_date, '%Y-%m-%d')
            expiration_date = datetime.strptime(expiration_date, "%Y-%m-%d")
        except:
            return 1
    if ((expiration_date is None) or (creation_date is None)):
        return 1
    elif ((type(expiration_date) is list) or (type(creation_date) is list)):
        return 1
    else:
        ageofdomain = abs((expiration_date - creation_date).days)
        if ((ageofdomain / 30) < 6):
            age = 1
        else:
            age = 0
    return age

def domainEnd(domain_name):
    expiration_date = domain_name.expiration_date
    if isinstance(expiration_date, str):
        try:
            expiration_date = datetime.strptime(expiration_date, "%Y-%m-%d")
        except:
            return 1
    if (expiration_date is None):
```

```
        return 1
    elif (type(expiration_date) is list):
        return 1
    else:
        today = datetime.now()
        end = abs((expiration_date - today).days)
        if ((end / 30) < 6):
            end = 0
        else:
            end = 1
    return end

def iframe(response):
    if response == "":
        return 1
    else:
        if re.findall(r"<iframe>|<frameBorder>", response.text):
            return 0
        else:
            return 1

def mouseOver(response):
    if response == "":
        return 1
    else:
        if re.findall("<script>.+onmouseover.+</script>", response.text):
            return 1
        else:
            return 0

def rightClick(response):
    if response == "":
```

```
        return 1
    else:
        if re.findall(r"event.button ?== ?2", response.text):
            return 0
        else:
            return 1

def forwarding(response):
    if response == "":
        return 1
    else:
        if len(response.history) <= 2:
            return 0
        else:
            return 1

def featureExtraction(url):
    features = []
    # Address bar based features (10)
    features.append(getDomain(url))
    features.append(havingIP(url))
    features.append(haveAtSign(url))
    features.append(getLength(url))
    features.append(getDepth(url))
    features.append(redirection(url))
    features.append(httpDomain(url))
    features.append(tinyURL(url))
    features.append(prefixSuffix(url))
    # Domain based features (4)
    dns = 0
    try:
        domain_name = whois.whois(urlparse(url).netloc)
```

```
except:
    dns = 1
features.append(dns)
# features.append(web_traffic(url))
features.append(1 if dns == 1 else domainAge(domain_name))
features.append(1 if dns == 1 else domainEnd(domain_name))

# HTML & Javascript based features (4)
try:
    response = requests.get(url)
except:
    response = ""
features.append(iframe(response))
features.append(mouseOver(response))
features.append(rightClick(response))
features.append(forwarding(response))
# features.append(label)
return features

data0 = pd.read_csv('uploads/url_data_modified.csv')
data = data0.drop(['Domain','Web_Traffic'], axis=1).copy()
data = data.sample(frac=1).reset_index(drop=True)
y = data['Label']
X = data.drop('Label', axis=1)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=12)
from sklearn.ensemble import RandomForestClassifier
# instantiate the model
forest = RandomForestClassifier(max_depth=5)
# fit the model
forest.fit(X_train, y_train)
print('aa')
```



```
print(url1)
print(type(url1))
my_features = featureExtraction(url1)
prob_of_doms = top_doms[1].values
if my_features[0] in prob_of_doms:
    return render_template('prediction.html',msg = 'success')
else:
    pred1 = forest.predict([my_features[1:]])
    print(pred1)
    if pred1==0:
        msg=""
    else:
        # email=session.get('email')
        # name=session.get('pno')
        # pno=session.get('pno')
        # ts = time.time()
        # date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
        # timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        # msg = 'The website you are trying to visit not legitimate'
        # t = 'Regards,'
        # t1 = 'Phishing Website.'
        # mail_content = 'Dear ' + name + ', '\n'+msg + '\n' + '\n' + t + '\n' + t1
        # sender_address = "
        # sender_pass = "
        # receiver_address = email
        # message = MIMEMultipart()
        # message['From'] = sender_address
        # message['To'] = receiver_address
        # message['Subject'] = 'Phishing Website'
        # message.attach(MIMEText(mail_content, 'plain'))
        # ses = smtplib.SMTP('smtp.gmail.com', 587)
        # ses.starttls()
```

```
# ses.login(sender_address, sender_pass)
# text = message.as_string()
# ses.sendmail(sender_address, receiver_address, text)
# ses.quit()
# url = "https://www.fast2sms.com/dev/bulkV2"
# message = 'Dear ' + name + ',' + '\n'+msg
# no = pno
# data1 = {
#     "route": "q",
#     "message": message,
#     "language": "english",
#     "flash": 0,
#     "numbers": no,
# }
# headers = {
#
#                                     "authorization":
"UwmaiQR5OoA6lSTz93nP0tDxsFEhI7VJrfKkvYjbM2C14Wde8g9lvA2Ghq5VNCjrZ4TH
WkF1KOwp3Bxd",
#     "Content-Type": "application/json"
# }
# response = requests.post(url, headers=headers, json=data1)
# print(response)
msg="Phishing Mail Sent"
return render_template('prediction.html',result=pred1,msg = msg)
return render_template('prediction.html')

@app.route('/graph')
def graph ():
#pic=pd.DataFrame({'Models':['RandomForestClassifier','XGBoostClassifier','AdaBoostClassifier','GradientBoostingClassifier','SupportVectorMachine'],'Accuracy':[score1,score3,score2,score5,score4]})
# pic
```

```
# plt.figure(figsize = (10,6))
# sns.barplot(y = pic.Accuracy,x = pic.Models)
# plt.xticks(rotation = 'vertical')
# plt.show()
return render_template('graph.html')

if __name__ == '__main__':
    app.run(debug=True)
```

CHAPTER – 12

RESULTS

Home page

Here user view the home page of Hazard Identification and Detection prediction web application.



Fig.4.1 Home Page

Login

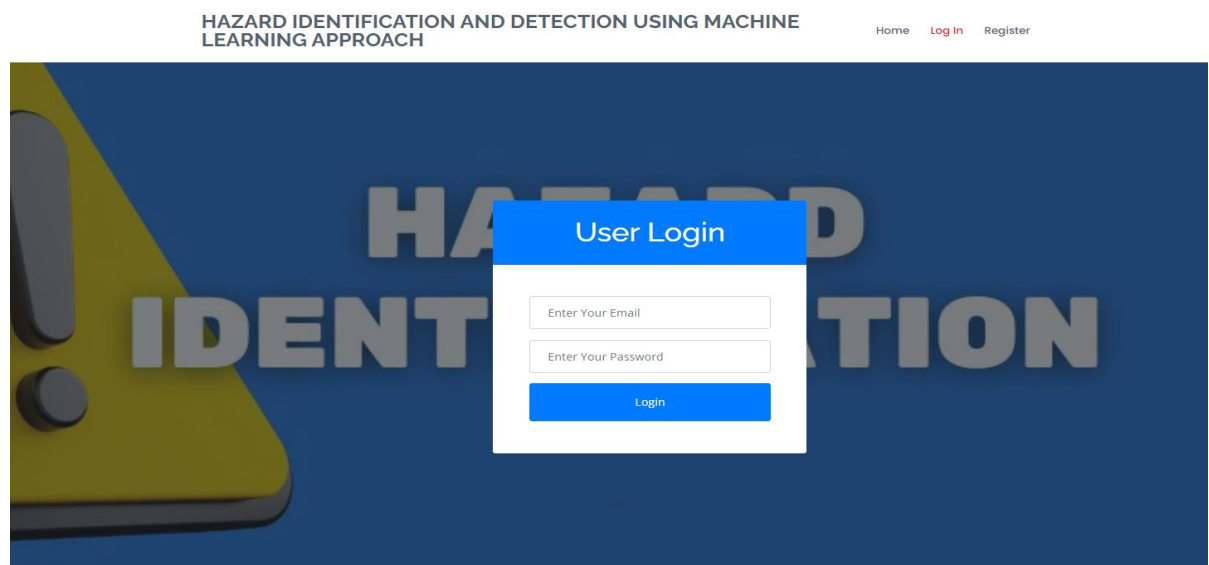
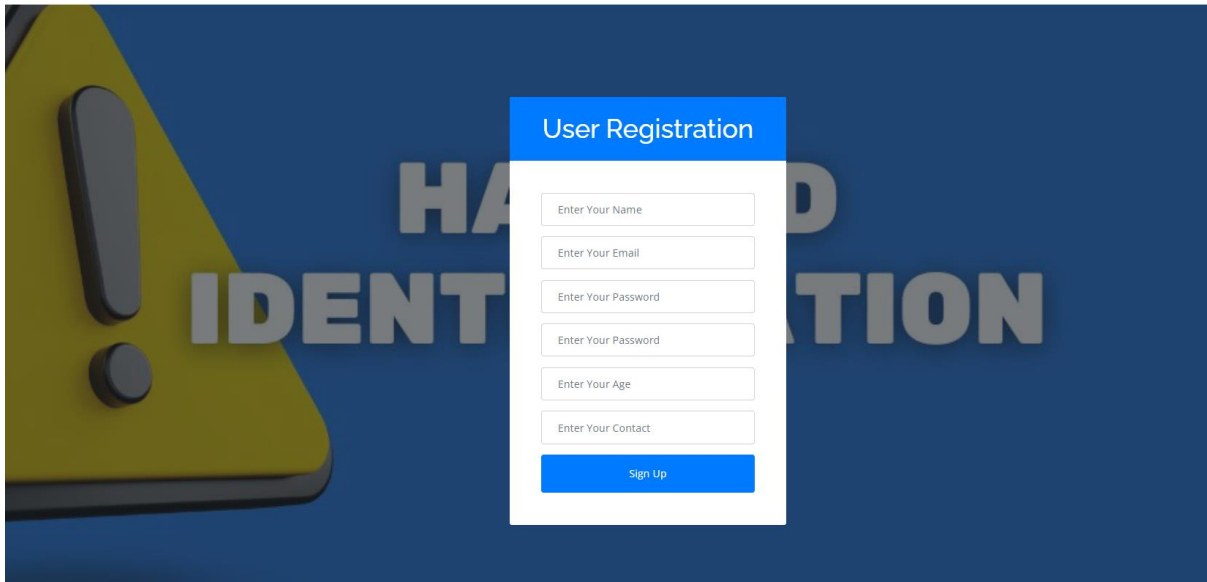


Fig.4.2 Login

Registration

HAZARD IDENTIFICATION AND DETECTION USING MACHINE LEARNING APPROACH

[Home](#) [Log In](#) [Register](#)



User Registration

Enter Your Name

Enter Your Email

Enter Your Password

Enter Your Password

Enter Your Age

Enter Your Contact

Sign Up

Fig.4.3 Registration

User Home Page



Fig.4.4 User Home Page

Load data

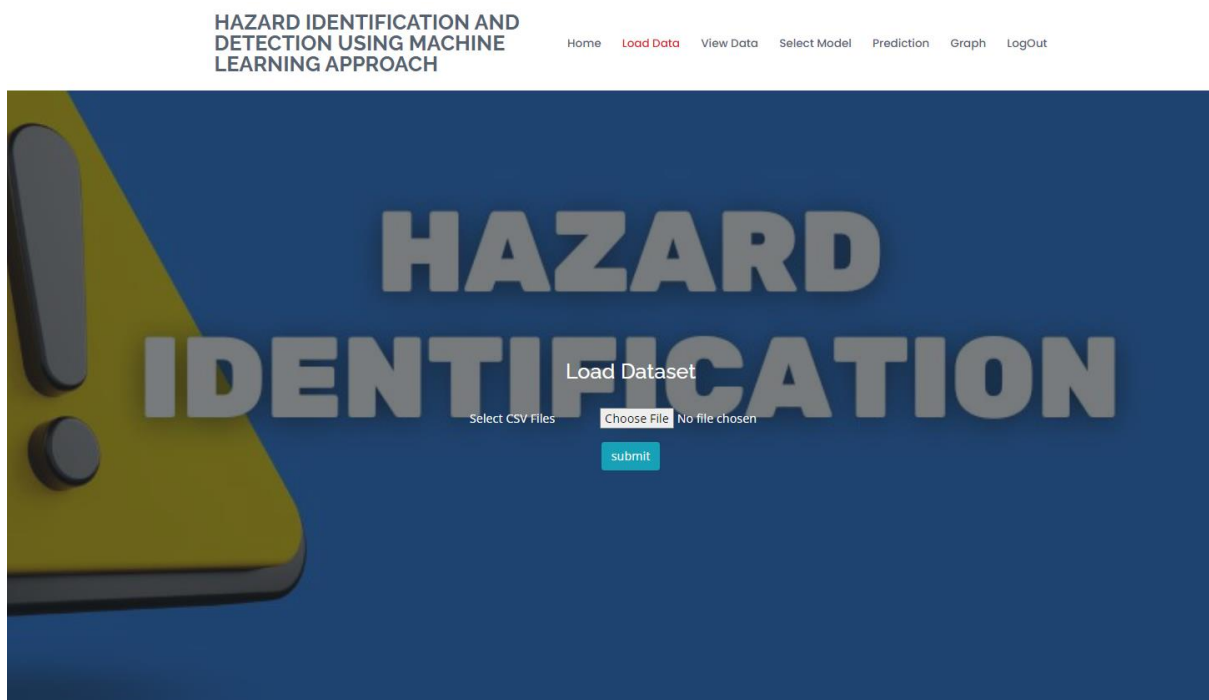


Fig.4.5 Load Data

View

HAZARD IDENTIFICATION AND DETECTION USING MACHINE LEARNING APPROACH

Home Load Data **View Data** Select Model Prediction Graph LogOut

S/N	Domain	Have_IP	Have_At	URL_Length	URL_Depth	Redirection	https_Domain	TinyURL	Pr
1	graphicriver.net	0	0	1	1	0	0	0	
2	ecnavi.jp	0	0	1	1	1	0	0	
3	huppages.com	0	0	1	1	0	0	0	
4	extratorrent.cc	0	0	1	3	0	0	0	
5	icicibank.com	0	0	1	3	0	0	0	
6	nypost.com	0	0	1	4	0	0	1	
7	kienthuc.net.vn	0	0	1	2	0	0	0	
8	thenextweb.com	0	0	1	6	0	0	0	
9	tobogo.net	0	0	1	2	0	0	0	
10	akhbarelyom.com	0	0	1	5	0	0	0	
11	tunein.com	0	0	1	5	0	0	0	
12	tune.pk	0	0	1	3	0	0	0	
13	sfglobe.com	0	0	1	4	0	0	0	
14	mic.com	0	0	1	3	0	0	0	
15	thenextweb.com	0	0	1	6	0	0	0	
16	couchtuner.eu.com	0	0	1	3	0	0	0	
17	olx.in	0	0	1	3	0	0	0	
18	venturebeat.com	0	0	1	4	0	0	1	

Fig.4.6 View

Model

Here we can train our data using different algorithm.

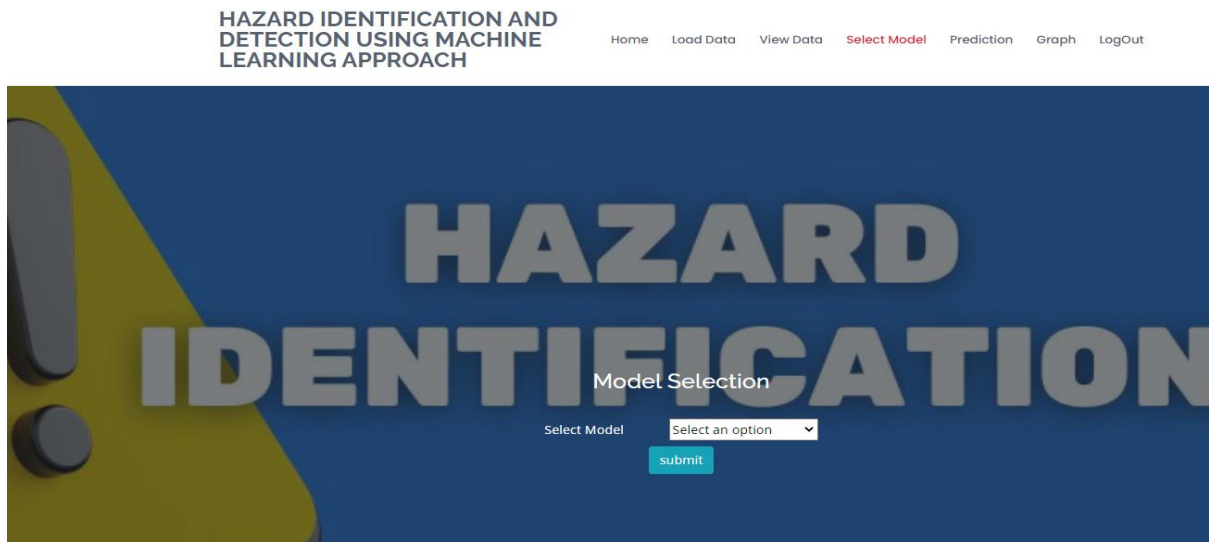


Fig.4.7 Model

Prediction

This page show the detection result that whether the website is a Hazard or legitimate.



Fig.4.8 Prediction

Graph

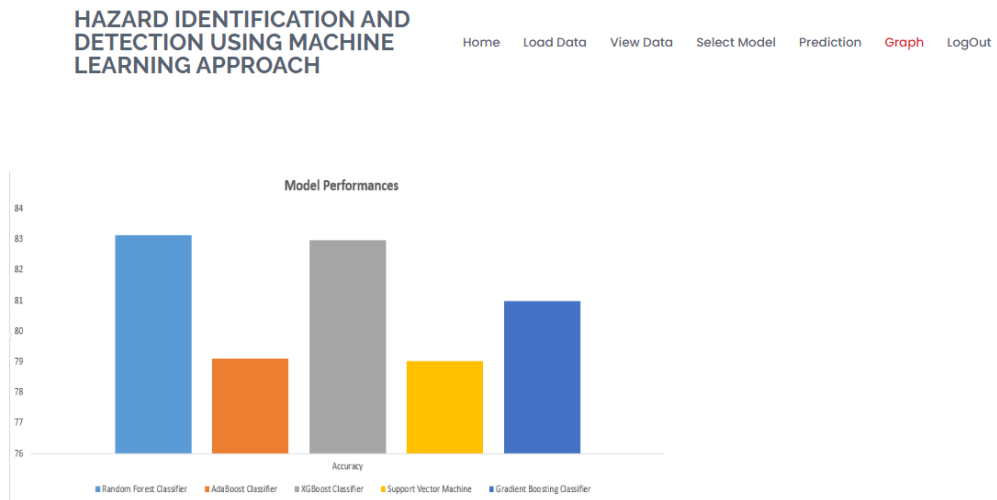


Fig.4.9 Graph

CHAPTER – 13

CONCLUSION AND FUTURE ENHANCEMENTS

Hazard Identification and Detection using machine learning is a promising approach to combat the growing threat of online fraud. Machine learning algorithms can be trained to detect patterns in the behavior and characteristics of Hazard allowing them to identify and block suspicious sites before they can do harm. Recent studies have shown that machine learning algorithms can achieve high levels of accuracy in detecting Hazard websites. These algorithms can analyse various features of a website, such as its URL structure, content, and user interface, to determine whether it is likely to be a Hazard site. However, it is important to note that machine learning algorithms are not perfect and can sometimes produce false positives or false negatives. Additionally, Hazard attackers are constantly evolving their tactics, so machine learning models must be continuously updated and refined to stay effective. Overall, Hazard Identification and Detection using machine learning is a valuable tool in the fight against online fraud, but it should be used in conjunction with other security measures to provide the most comprehensive protection for users.

CHAPTER – 14

BIBLIOGRAPHY

REFERENCES

1. J. Shad and S. Sharma, “A Novel Machine Learning Approach to Detect Hazard Identification And DetectionJaypee Institute of Information Technology,” pp. 425–430, 2018.
2. Y. Sönmez, T. Tuncer, H. Gökal, and E. Avci, “Hazard web sites features classification based on extreme learning machine,” 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding, vol. 2018–Janua, pp. 1–5, 2018.
3. T. Peng, I. Harris, and Y. Sawa, “Detecting Hazard Attacks Using Natural Language Processing and Machine Learning,” Proc. - 12th IEEE Int. Conf. Semant. Comput. ICSC 2018, vol. 2018–Janua, pp. 300–301, 2018.
4. M. Karabatak and T. Mustafa, “Performance comparison of classifiers on reduced Hazard website dataset,” 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding, vol. 2018–Janua, pp. 1–5, 2018.
5. S. Parekh, D. Parikh, S. Kotak, and P. S. Sankhe, “A New Method for Detection of Hazard Websites: URL Detection,” in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, vol. 0, no. Icicct, pp. 949–952.
6. K. Shima et al., “Classification of URL bitstreams using bag of bytes,” in 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), 2018, vol. 91, pp. 1–5.
7. A. Vazhayil, R. Vinayakumar, and K. Soman, “Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks,” in 2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018, 2018, pp. 1–6.
8. W. Fadheel, M. Abusharkh, and I. Abdel-Qader, “On Feature Selection for the Prediction of Hazard Websites,” 2017 IEEE 15th Intl Conf Dependable, Auton.

- Secur. Comput. 15th Intl Conf Pervasive Intell. Comput. 3rd Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr., pp. 871–876, 2017.
9. X. Zhang, Y. Zeng, X. Jin, Z. Yan, and G. Geng, “Boosting the Hazard Detection Performance by Semantic Analysis,” 2017.
 10. L. MacHado and J. Gadge, “Hazard Sites Detection Based on C4.5 Decision Tree Algorithm,” in 2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017, 2018, pp. 1–5.