PHISHING WEBSITE DETECTION

A Mini Project Report submitted to JNTU
Hyderabad in partial fulfillment
of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

by

KUMARI NIKITA	**21S11A1228**
CHENNA SRIKANTH	**21S11A1253**
M. BHARATHI NAIK	**21S11A1204**
PUTUMBAKA REVANTH SAI	**21S11A1239**

Under the Guidance of

Dr. VAKA MURALI MOHAN
B. Tech, M. Tech, Ph.D.

Professor & Principal

DEPARTMENT OF INFORMATION TECHNOLOGY
MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE
(Approved by AICTE New Delhi and Affiliated to JNTUH)
(Accredited by NBA & NAAC with "A" Grade)
An ISO 9001: 2015 Certified Institution
Maisammaguda, Medchal (M), Hyderabad-500100, T. S.

JULY 2024

DEPARTMENT OF INFORMATION TECHNOLOGY

MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE

(Approved by AICTE New Delhi and Affiliated to JNTUH)

(Accredited by NBA & NAAC with "A" Grade)
An ISO 9001: 2015 Certified Institution
Maisammaguda, Medchal (M), Hyderabad-500100, T. S.
JULY 2024

CERTIFICATE

This is to certify that the mini project entitled **"PHISHING WEBSITE DETECTION"** has been submitted by **KUMARI NIKITA (21S11A1228), CHENNA SRIKANTH (21S11A1253) M. BHARATHI NAIK (21S11A1204), PUTUMBAKA REVANTH SAI (21S11A1239)** in partial fulfilment of the requirements for the award of **BACHELOR OF TECHNOLOGY** in **INFORMATION TECHNOLOGY**. This record of bonafide work was carried out by them under my guidance and supervision. **The result embodied in this mini- project report has not been submitted to any other University or Institute for the award of any degree.**

Dr. VAKA MURALI MOHAN	**Dr. A. Nagaraju**
Professor & Principal	*Head of The Department*
Project Guide	

External Examiner

ACKNOWLEDGEMENT

The Mini Project work carried out by our team in the Department of Information Technology, Malla Reddy Institute of Technology and Science, Hyderabad. ***This work is original and has not been submitted in part or full for any degree or diploma of any other university.***

We wish to acknowledge our sincere thanks to our project guide **Dr. Vaka Murali Mohan**, Professor for formulation of the problem, analysis, guidance and his continuous supervision during the course of work.

We acknowledge our sincere thanks to **Dr. Vaka Murali Mohan,** Principal and **Dr. A. Nagaraju**, Head of the Department and Coordinator, faculty members of IT Department for their kind cooperation in making this Mini Project work a success.

We extend our gratitude to **Sri. Ch. Malla Reddy**, Founder Chairman MRGI and **Sri. Ch. Mahender Reddy,** Secretary MRGI, **Dr.Ch. Bhadra Reddy**, President MRGI, **Sri. Ch. Shalini Reddy**, Director MRGI, **Sri. P. Praveen Reddy**, Director MRGI, for their kind cooperation in providing the infrastructure for completion of our Mini Project.

We acknowledge our special thanks to the entire teaching faculty and non-teaching staff members of the Information Technology Department for their support in making this project work a success

KUMARI NIKITA	**21S11A1228**	____________________
CHENNA SRIKANTH	**21S11A1253**	____________________
M. BHARATHI NAIK	**21S11A1204**	____________________
PUTUMBAKA REVANTH SAI	**21S11A1239**	____________________

.

INDEX

ABSTRACT

Phishing sites which expects to take the victims confidential data by diverting them to surf a fake website page that resembles a honest to goodness one is another type of criminal acts through the internet and its one of the especially concerns toward numerous areas including e-managing an account and retailing. Phishing site detection is truly an unpredictable and element issue including numerous components and criteria that are not stable. On account of the last and in addition ambiguities in arranging sites because of the intelligent procedures programmers are utilizing, some keen proactive strategies can be helpful and powerful tools can be utilized, for example, fuzzy, neural system and data mining methods can be a successful mechanism in distinguishing phishing sites. We applied Random Forest (RF), one of the different types of machine learning based algorithms used for detection of Phishing websites. Finally we measured and compared the performance of the classifier in terms of accuracy.

LIST OF FIGURES

CHAPTER-1: SYSTEM ANALYSIS

1.1Existing system

Anti-phishing strategies involve educating netizens and technical defense. In this paper, we mainly review the technical defense methodologies proposed in recent years. Identifying the phishing website is an efficient method in the whole process of deceiving user information Along with the development of machine learning techniques, various machine learning based methodologies have emerged for recognizing phishing websites to increase the performance of predictions. The primary purpose of this paper is to survey effective methods to prevent phishing attacks in a real-time environment. Existing research works have suggested various strategies for detecting phishing websites.

1.1.1Disadvantages of Existing System

1.Timeliness

- Traditional methods, such as blacklists and signature-based techniques, may not keep up with rapidly evolving phishing attacks.
- New phishing sites can emerge quickly, rendering existing blacklists outdated.

2.Zero-Day Attacks:

- Existing systems struggle to defend against **zero-day** phishing attacks—those exploiting previously unknown vulnerabilities.
- These attacks bypass known patterns and heuristics.

3.Targeted Attacks:

- Phishers increasingly use **sophisticated obfuscation techniques** to evade detection.
- Existing methods relying on familiar login interfaces can be vulnerable to targeted attacks

4.Low Accuracy:

- Conventional approaches provide low accuracy and can recognize only about 20% of phishing attacks.
- False positives and false negatives are common, impacting overall effectiveness.

1.2 Proposed System

The most frequent type of phishing assault, in which a cybercriminal impersonates a well-known institution, domain, or organization to acquire sensitive personal information from the victim, such as login credentials, passwords, bank account information, credit card information, and so on. Emails containing malicious URLs in this sort of phishing email contain a lot of personalization information about the potential victim. To spear phish a "whale," here a top-level executive such as CEO, this sort of phishing targets corporate leaders such as CEOs and top-level management employees To infect the target, the fraudster or cyber-criminal employs a URL link.

1.2.1 Advantages of Proposed System

1. **Adaptability**:
    - Machine learning models can learn from data and adapt to new types of phishing attacks without manual updates.
    - They stay effective even as attack techniques evolve.
2. **Real-Time Analysis**:
    - Machine learning provides real-time detection, crucial for timely protection.
    - It can swiftly identify suspicious patterns and URLs.
3. **Versatility**:
    - Machine learning can examine large datasets, capturing both well-known and unique phishing attacks.
    - It considers various features beyond simple rules or heuristics.
4. **Accuracy:**
    - Machine learning models can be more accurate and effective than traditional methods (e.g., blacklists or heuristics).
    - They learn from patterns and generalize well to detect subtle variations in phishing attempts.

1.3 INTRODUCTION

1.3.1 OUTLINE OF THE PROJECT

Phishing is a type of extensive fraud that happens when a malicious website act like a real one keeping in mind that the end goal to obtain touchy data, for example, passwords, account points of interest, or MasterCard numbers.

In spite of the fact that there are a few contrary to phishing programming and methods for distinguishing potential phishing endeavours in messages & identifying phishing substance on sites, phishes think of new and half breed strategies to go around the accessible programming and systems.

Phishing is a trickery system that uses a blend of social designing what's more, innovation to assemble delicate and individual data, for example, passwords and charge card subtle elements by taking on the appearance of a dependable individual or business in an electronic correspondence. Phishing makes utilization of spoof messages that are made to look valid and implied to be originating from honest to goodness sources like money related foundations, ecommerce destinations and so forth, to draw clients to visit fake sites through joins gave in the phishing email. The misleading sites are intended to emulate the look of a genuine organization site page.

The employing so as to phishing invader's trap clients diverse social building strategies, for example, debilitating to suspend client accounts on the off chance thatthey don't finish the account upgrade process, give other data to approve their records or a few different motivations to get the clients to visit their satirize page.

Supervised learning (Classification Technique) accommodates a vastlyimproved precision while unsupervised learning accommodates a quick and dependable way to deal with infer information from a dataset. That's why we used supervised learning in our work.

1.3.2 MACHINE LEARNING

Machine learning could be a subfield of computer science (AI). The goal of machine learning typically is to know the structure information of knowledge of information and match that data into models which will be understood and used by folks. Although machine learning could be a field inside technology, it differs from ancient process approaches.

In ancient computing, algorithms are sets of expressly programmed directions employed by computers to calculate or downside solve. Machine learning algorithms instead give computers to coach on knowledge inputs and use applied math analysisso as to output values that fall inside a particular vary. thanks to this, machine learning facilitates computers in building models from sample knowledge so as to modify decision-making processes supported knowledge inputs.

1.3.3 SCOPE AND OBJECTIVE

The system should be useful in many e-commercial websites for maintaining thesecurity and reliability of customers and people online
The system should be useful in preventing online frauds leading to leakage ofimportant and private user data
The scope of using Machine Language over other Traditional Detecting MethodsObjectives:

- Understanding phishing domain (or Fraudulent Domain) characteristics, its distinguishing features from legitimate domains
- Why it is so important to detect this domain and how they can be detected using machine learning and natural language processing techniques
- Reviewing the state-of-the-art machine learning techniques for malicious URL detection in literature
- Understanding the newly emerging concept of Malicious URL Detection as a service and the principles to be used while designing such a system.

CHAPTER-2 LITERATURE REVIEW

Many scholars have done some sort of analysis on the statistics of phishing URLs. Our technique incorporates key concepts from past research. We review past work in the detection of phishing sites using URL features, which inspired our current approach. Happy describe phishing as "one of the most dangerous ways for hackers to obtain users' accounts such as usernames, account numbers and passwords, without their awareness." Users are ignorant of this type of trap and will ultimately, they fall into Phishing scam. This could be due to a lack of a combination of financial aid and personal experience, as well as a lack of market awareness or brand trust. In this article, Mehmet et al. suggested a method for phishing detection based on URLs. To compare the results, the researchers utilized eight different algorithms to evaluate the URLs of three separate datasets using various sorts of machine learning methods and hierarchical architectures. The first method evaluates various features of the URL; the second method investigates the website's authenticity by determining where it is hostedand who operates it; and the third method investigates the website's graphic presence. We employ Machine Learning techniques and algorithms to analyze these many properties of URLs and websites. Garera et al. classify phishing URLs using logistic regression over hand-selected variables. The inclusion of red flag keywords in the URL, as well as features based on Google's Web page and Google's Page Rank quality recommendations, are among the features. Without access to the same URLs and features as our approach, it's difficult to conduct a direct comparison.

In this research, Yong et al. created a novel approach for detecting phishing websites that focuses on detecting a URL which has been demonstrated to be an accurate and efficient way of detection. To offer you a better idea, our new capsule-based neural network is divided into several parallel components. One method involves removing shallow characteristics from URLs. The other two, on the other hand, construct accurate feature representations of URLs and use shallow features to evaluate URL legitimacy. The final output of our system is calculated by adding the outputs of all divisions. Extensive testing on a dataset collected from the Internet indicate that our system can compete with other cutting-edge detection methods while consuming a fair amount of time. For phishing detection, Vahid Shahrivari et al. used machine learning approaches. They used the logistic regression classification method, KNN, Adaboost algorithm, SVM, ANN and random forest. They found random forest algorithm provided good accuracy. Dr.G. Ravi Kumar used a variety of machine learning methods to detect phishing assaults.

For improved results, they used NLP tools. They were able to achieve high accuracy using a Support Vector Machine and data that had been pre-processed using NLP approaches. Amani Alswailem et al. tried different machine learning model for phishing detection but was able to achieve more accuracy in random forest. Hossein et al. created the "Fresh-Phish" open-source framework. This system can be used to build machine-learning data for phishing websites. They used a smaller feature set and built the query in Python. They create a big, labelled dataset and test several machine-learning classifiers on it. Using machine-learning classifiers, this analysis yields very high accuracy. These studies look at how long it takes to train a model. X. Zhang suggested a phishing detection model based on mining the semantic characteristics of word embedding, semantic feature, and multi-scale statistical features in Chinese web pages to detect phishing performance successfully. To obtain statistical aspects of web pages, eleven features were retrieved and divided into five classes. To obtain statistical aspects of web pages, eleven featureswere retrieved and divided into five classes. To learn and evaluate the model, AdaBoost, Bagging, Random Forest, and SMO are utilized. The legitimate URLs dataset came from DirectIndustry online guides, and the phishing data came from China's Anti-Phishing Alliance. With novel methodologies, M. Aydin approaches a framework for extracting characteristics that is versatile and straightforward. Phish Tank provides data, and Google provides authentic URLs. C# programming and R programming were utilized to obtain the text attributes. The dataset and third-party service providers yielded a total of 133 features. The feature selection approaches of CFS subset based and Consistency subset-based feature selection were employed and examined with the WEKA tool. The performance of the Nave Bayes and Sequential Minimal Optimization (SMO) algorithms was evaluated, and the author prefers SMO to NB for phishing detection.

1. **“Detecting Phishing Websites Using Machine Learning Technique”**:
    - **Abstract**: This study proposes a URL detection technique based on machine learning approaches. A recurrent neural network (RNN) method is employed to detect phishing URLs. The proposed method outperforms recent approaches in detecting malicious URLs.
    - **Author**: Ashit Kumar Dutta
    - **Published in**: PLoS ONE, October 11, 2021

2. **“Machine Learning-Based Phishing Detection Using URL Features: A Comprehensive Survey”**:
    - **Abstract**: This survey provides a comprehensive review of research on URL-based phishing detectors using machine learning. It covers feature extraction, datasets, algorithms, experimental design, and results for each work.

CHAPTER-3 : SYSTEM DESIGN

3.1 System Architecture

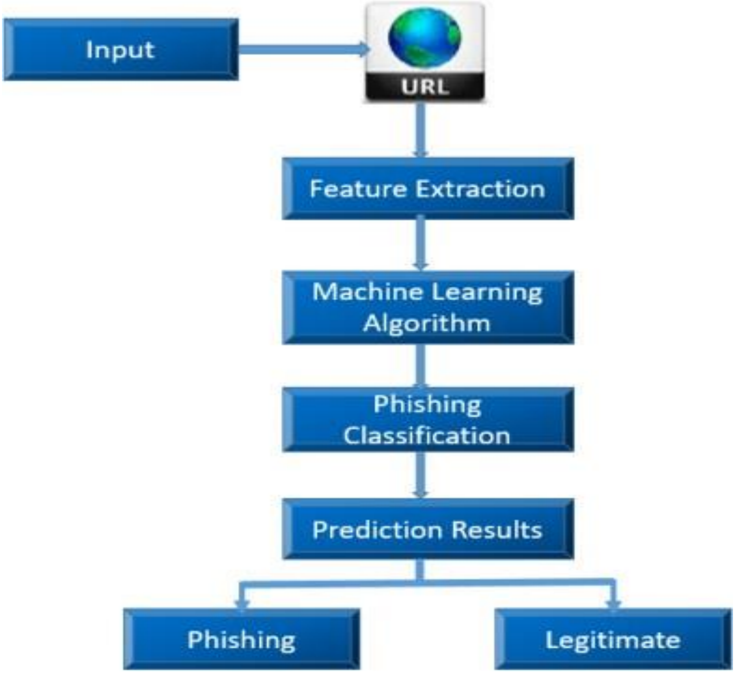


Fig 3.1 System Architecture

3.2 Flow Diagram

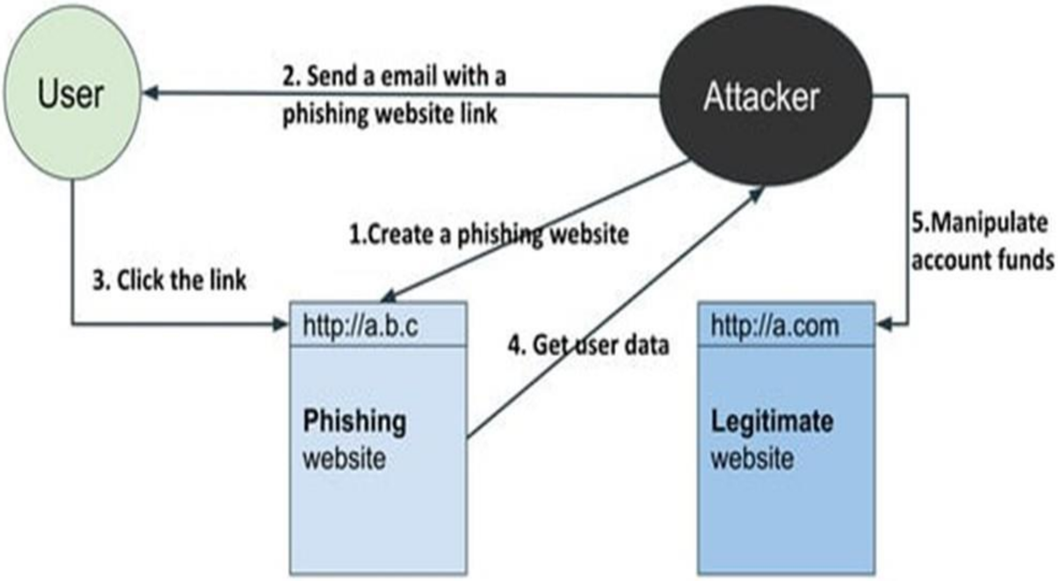


Fig 3.2 Flow Diagram

3.3 UML DIAGRAMS

The Unified Modeling Language (UML) is a standardized specification language for object modeling. UML is a general-purpose modeling language that includes a graphical notation used to create an abstract model of a system, referred to as a UML model.

Definition: UML is a general purpose visual modeling language that is used to

- specify
- visualize
- construct

UML is a language: it will provide vocabulary and rules for communications and function on conceptual and physical representation. So it is a modeling language.

UML specifying: specifying means building models which are precise, unambiguous and complete. In particular, the UML address the specification of all the important analysis, design and implementation decisions that must be made in developing and displaying a software intensive system.

UML visualization: the UML includes both graphical and textual representation. It makes easy to visualize the system and for better understanding.

UML constructing: UML models can be directly connected to a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.

UML documenting: UML provides variety of documents in addition to raw executable codes. The system is designed using on UML. The UML modeling and design is a new way of thinking about problems using models organized around real world concepts. UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software system, as well as for business modeling and other non-software 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. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Reasons to model:

- To communicate the desired structure and behavior of the system
- To visualize and control the system’s architecture.

- To better understand the system and expose opportunities for specification and reuse.
- To manage risk

Uses of UML:

The UML is intended primarily for software intensive systems. It has been used effectively for such domain as Enterprise information system, banking and financial services, telecommunications, transportation, defense/aerospace, retails, medical, electronics, scientific fields, distributed wed .

The UML standard specifies exactly how the diagrams are to be drawn and what each component in the diagram means. UML is not dependent on any particular programming language, instead it focuses on the fundamental concepts and ideas that model a system. Using UML enables anyone familiar with its specifications to instantly read and understand diagrams drawn by other people. There are UML diagram for modeling static class relationships, dynamic temporal interactions between objects, the usages of objects, the particulars of an implementation, and the state transitions of systems.

UML diagram consists of the following features:

- **Entities:** These may be classes, objects, users or systems behaviors.
- **Relationship:** Lines that model the relationships between entities in the system.
- **Generalization:** a solid line with an arrow that points to a higher abstraction of the present item.
- **Association:** a solid line that represents that one entity uses another entity as part of its behavior.
- **Dependency:** a dotted line with an arrowhead that shows one entity depends on the behavior of another entity.

Types of UML Diagrams:

UML defines nine types of Diagrams in which they are divided into two categories. They are Static diagrams and Dynamic diagrams.

1. Class diagram
2. Object diagram
3. Use case diagram
4. Sequence diagram
5. Collaboration diagram

6. State chart diagram

7. Activity diagram

8. Component diagram

9. Deployment diagram

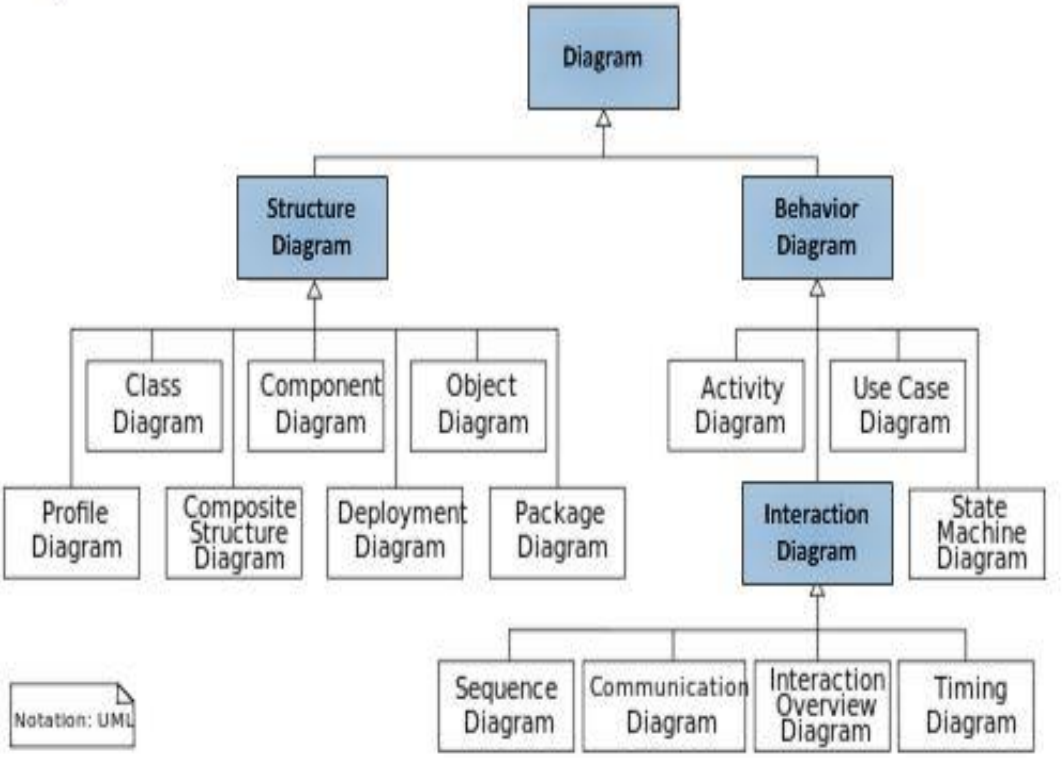


Fig 3.3 Types of UML Diagrams

1. Class Diagram:

A Class Diagram shows a set of Classes, interfaces, Collaborations and their relationships. Class Diagram models class structure and contents using design elements such as classes, packages and objects. It also displays relationships such as containment, inheritance, associations and others.

2. Object Diagram:

An Object Diagram shows a set of Objects and their relationships.

3. Use case diagram:

A use case diagram shows the relationship among actors and use cases within a system. Use case diagrams show elements from the use case model.

The use case model represents functionality of a system or a class as manifested to external actors with the system. A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and the use cases, and generalizations among the use cases.

4. Sequence diagram:

A sequence diagram shows an interaction arranged in time sequence. In particular, it shows the objects participating in the interaction by their "lifelines" and the messages that they exchange arranged in time sequence. It does not show the associations among the objects. A sequence diagram represents an Interaction, which is a set of messages exchanged among objects within collaboration to effect a desired operation or result.

5. Collaboration Diagram:

A Collaboration Diagram is an interaction Diagram that emphasizes the structural organization of the objects that send and receive messages. It shows a set of objects, links among those objects and messages sent & received by those objects.

6. State Chart Diagram:

A State Chart Diagram shows a state machine, consisting of states, transitions, events and activities. State Diagram emphasizes the event-ordered behavior of objects which is especially useful in modeling reactive systems.

7. Activity diagram:

An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state. Activity diagrams show the flow of activities through the system. Diagrams are read from top to bottom and have branches and forks to describe conditions and parallel activities. A fork is used when multiple activities are occurring at the same time. The branch describes what activities will take place based on a set of conditions. All branches at some point are followed by a merge to indicate the end of the conditional behavior started by that branch. After the merge all of the parallel activities must be combined by a join before transitioning into the final activity state.

3.3.1 Use Case Diagram

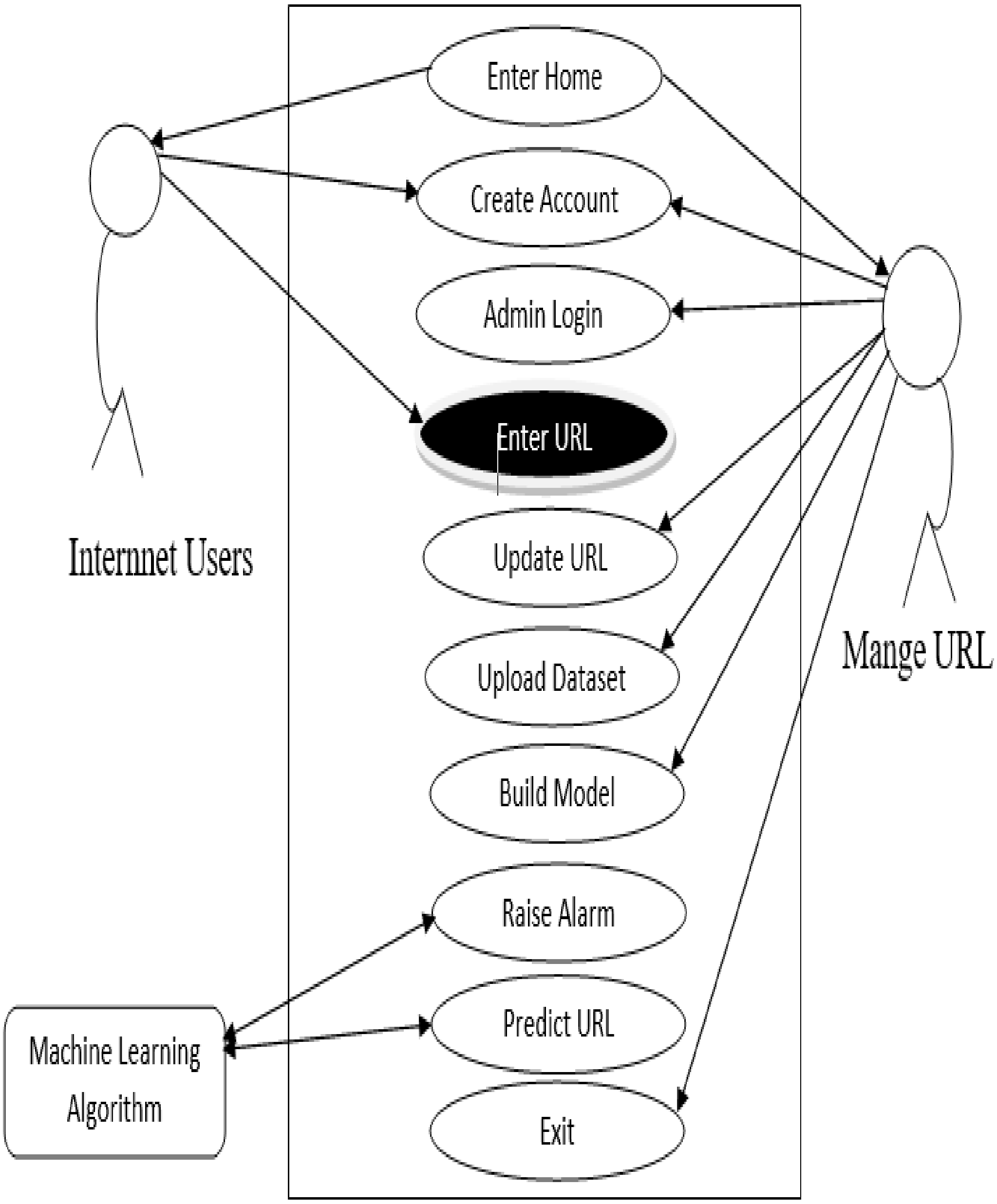
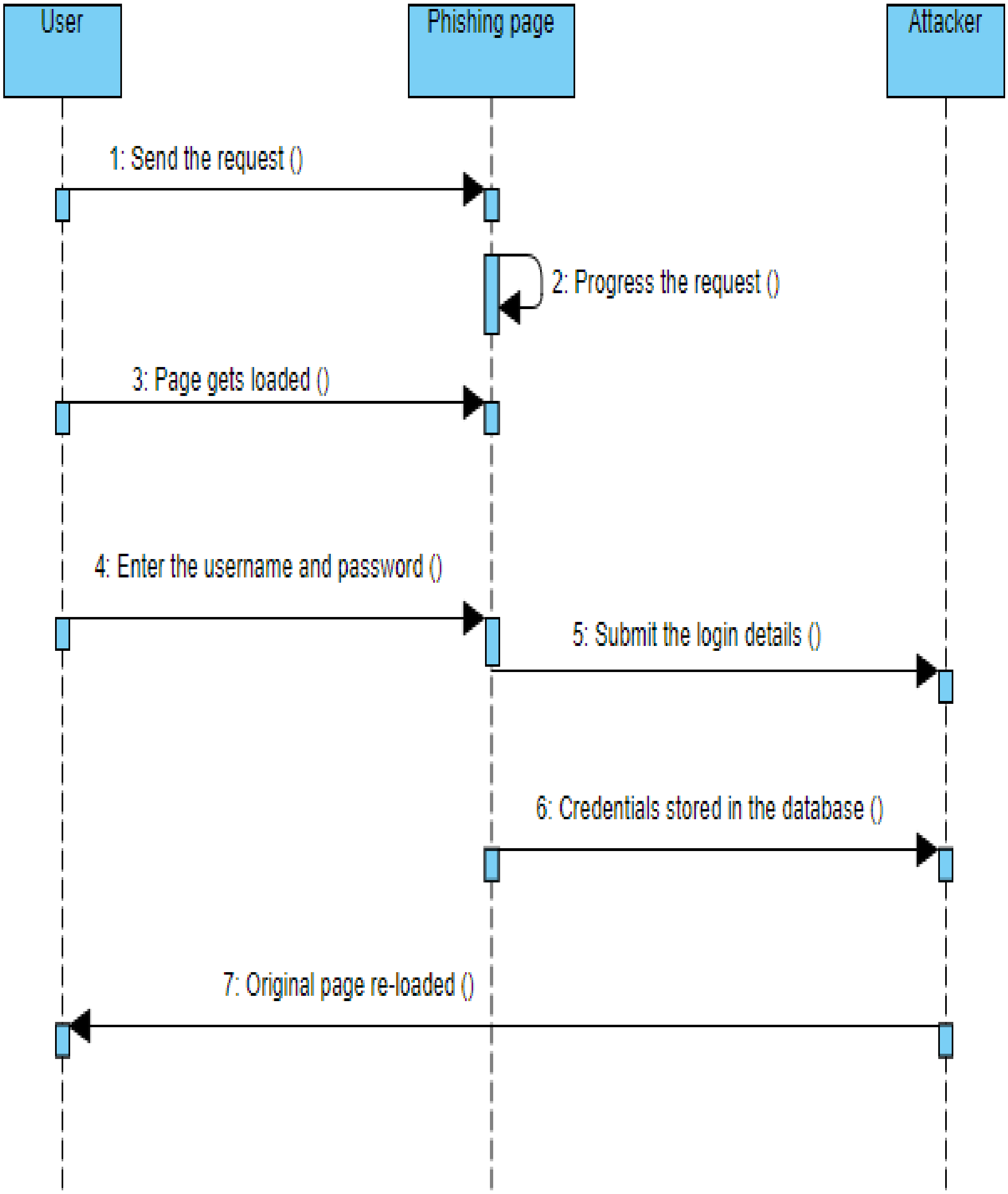


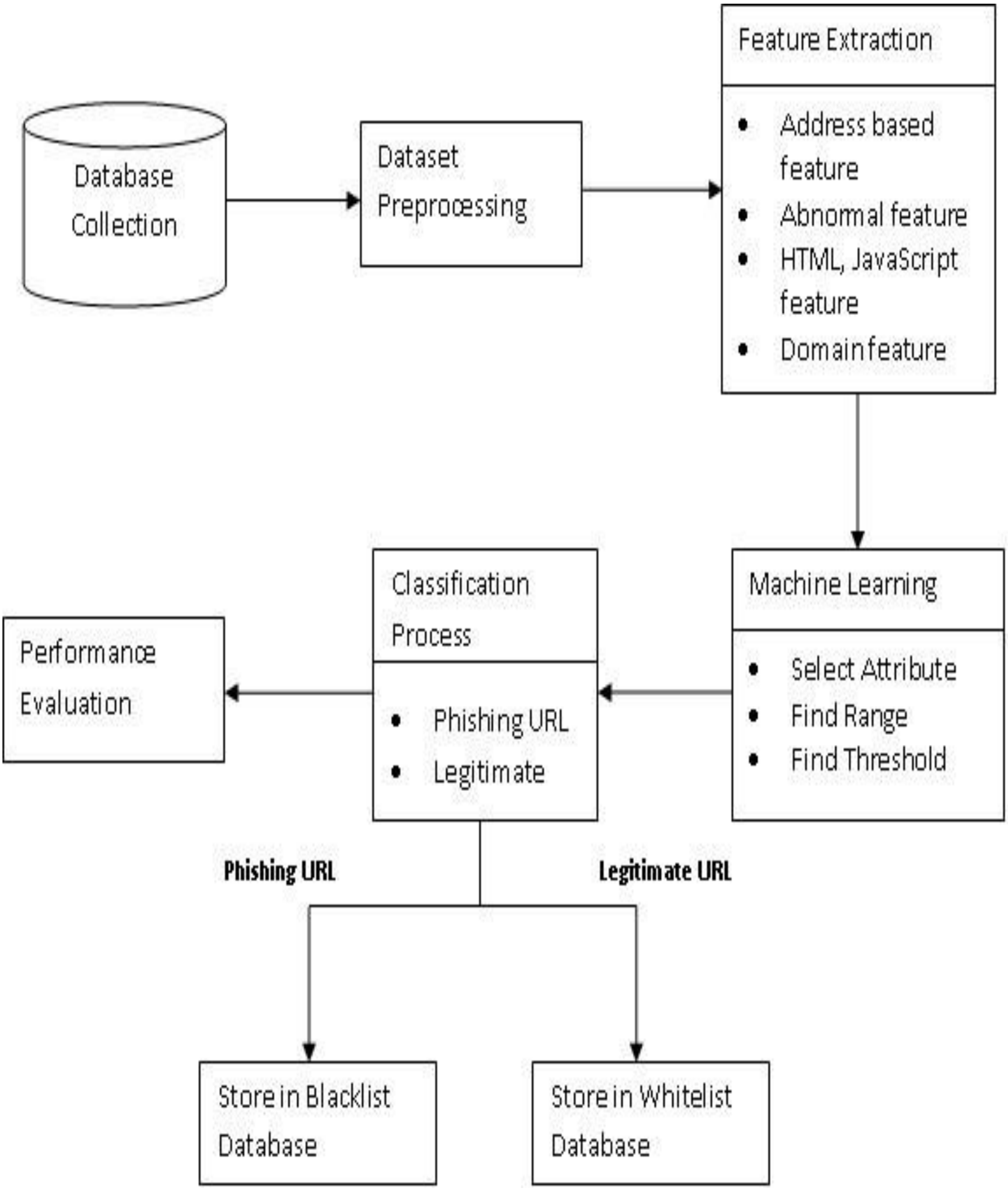
Fig 3.3.1 Use Case Diagram

3.3.2 Sequence Diagram


User
Phishing page
Attacker
1: Send the request ()
2: Progress the request ()
3: Page gets loaded ()
4: Enter the username and password ()
5: Submit the login details ()
6: Credentials stored in the database ()
7: Original page re-loaded ()

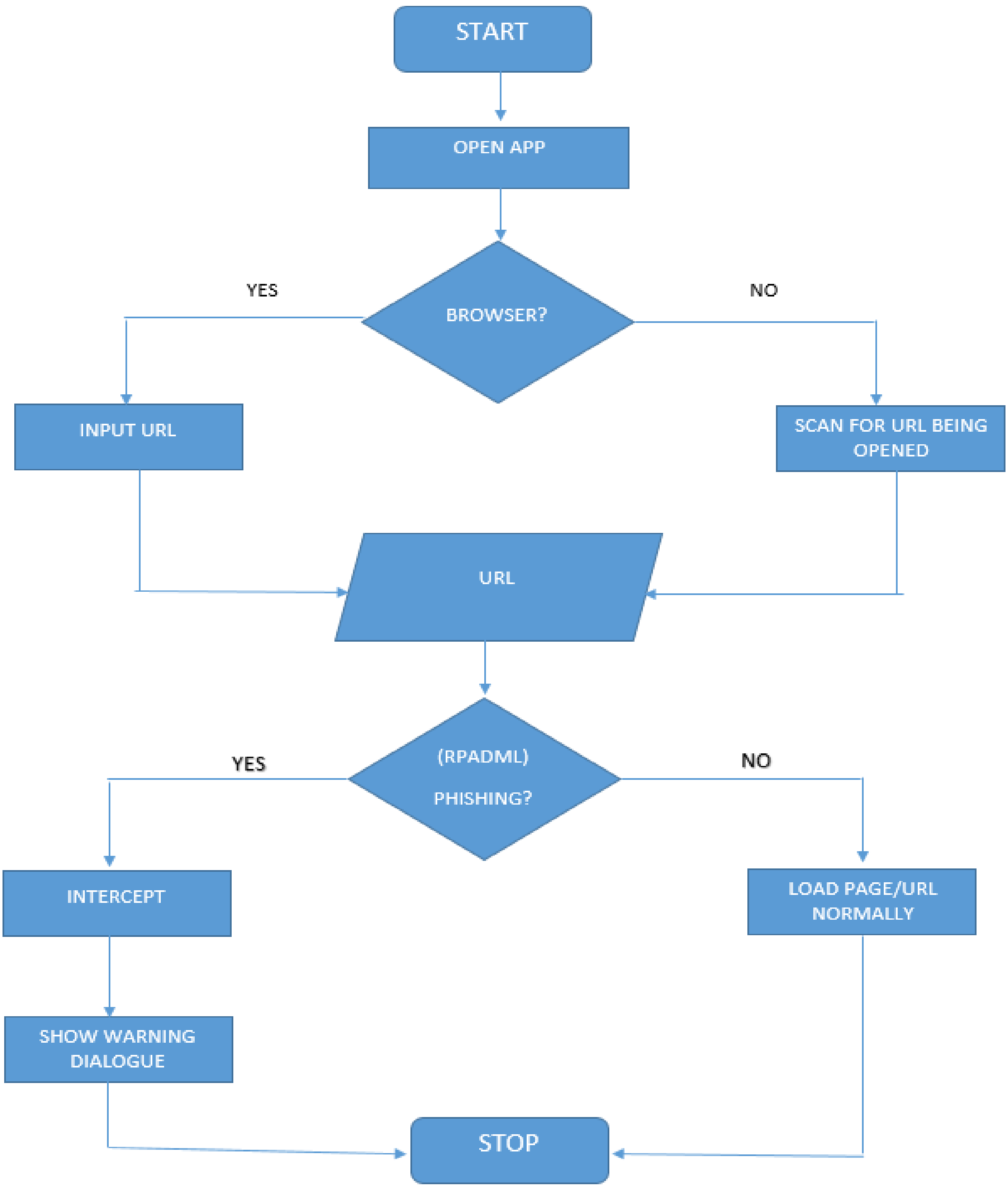
3.3.2 Sequence Diagram

3.3.3 Class Diagram



3.3.3 Class Diagram

3.3.4 Activity Diagram



3.3.4 Activity Diagram

3.4 Modules

3.4.1 PHISHING WEBSITES FEATURES

- One of the challenges faced by our research was the unavailability of reliable training datasets. In fact, this challenge faces any researcher in the field. However, although plenty of articles about predicting phishing websites using data mining techniques have been disseminated these days, no reliable training dataset has been published publically, maybe because there is no agreement in literature on the definitive features that characterize phishing websites, hence it is difficult to shape a dataset that covers all possible features.
- In this article, we shed light on the important features that have proved to be sound and effective in predicting phishing websites. In addition, we proposed some new features, experimentally assign new rules to some well-known features and update some other features.

3.4.2 DATA SET

One of the challenges faced by our research was the unavailability of reliable trainingdatasets. In fact, this challenge faces any researcher in the field. However, although plenty of articles about predicting phishing websites using data mining techniques have been disseminated these days, no reliable training dataset has been published publically, maybe because there is no agreement in literature on the definitive features that characterize phishing websites, hence it is difficult to shape a dataset that covers all possible features.

In this article, we shed light on the important features that have proved to be sound and effective in predicting phishing websites. In addition, we proposed some new features, experimentally assign new rules to some well-known features and update some other features.

3.4.3 Detection Techniques

Machine Learning-Based Approaches: These involve training models on labelled data to distinguish between legitimate and phishing websites. Common algorithms include Naïve Bayes, neural networks, decision trees, and support vector machines.

Address Bar-Based Features: Analysing the URL itself, checking for suspicious patterns (e.g., IP addresses instead of domain names).

Abnormal-Based Features: Identifying abnormal behaviour or patterns in website structure or content.

HTML and JavaScript-Based Features: Examining the code for suspicious elements.

Domain-Based Features: Comparing domain names to known legitimate domains.

3.5 System Requirements

3.5.1 Hardware Requirements

Processor	-	Ryzen 7
RAM	-	4 GB (min)
Hard Disk	-	20 GB

3.5.2 Software Requirements:

Operating System	-	Windows 11
Coding Language	-	Python
Front End	-	HTML, CSS, JavaScript.
Back End	-	MySQL.

CHAPTER-4: INPUT AND OUTPUT DESIGN

4.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

4.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing arecommunicated to the users and to other system through outputs. In output design itis determined how the information is to be displaced for immediate need and also thehard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help userdecision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of theFuture.
- Signal important events, opportunities,
  problems, or warnings.
- Trigger an action.
- Confirm an action.

CHAPTER-5: SYSTEM ENVIRONMENT

5.1 Python

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 toPERL 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 ofapplications from simple text processing to WWW browsers to games.

5.1.1 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, Small Talk, 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.

5.1.2 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 andcross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** − Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.

- **Portable** − Python can run on a wide variety of hardware platforms and hasthe 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 programsthan shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, feware 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 forbuilding large applications.

- It provides very high-level dynamic data types and supports dynamic typechecking.

- It supports automatic garbage collection.

- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

- Python is available on a wide variety of platforms including Linux and Mac OS

5.1.3 What is Python

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

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

5.2 What is Machine Learning –

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

5.2.1 Categories Of Machine Learning

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

5.2.2Applications of Machines Learning

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML -

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention

5.2.3 How to Start Learning Machine Learning

Arthur Samuel coined the term **"Machine Learning"** in 1959 and defined it as a **"Field of study that gives computers the capability to learn without being explicitly programmed".** And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it. So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

How to start learning ML

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!

Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python**.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

- **Training** – The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction** – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning** – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning** – This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning** – This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning** – This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

5.2.4 Advantages of Machine learning

- **Easily identifies trends and patterns**

  Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

- **No human intervention needed (automation)**

  With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares, they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

- **Continuous Improvement**

  As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

- **Handling multi-dimensional and multi-variety data**

  Machine Learning algorithms are good at handling data that are multi-dimensional and multi- variety, and they can do this in dynamic or uncertain environments.

5.2.5 Disadvantages of Machine Learning

- **Data Acquisition**

  Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

- **Time and Resources**

  ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

- **Interpretation of Results**

  Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

- **High error-susceptibility**

  Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

5.3 LIBRARIES USED

- **TensorFlow**

  TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

- **Numpy**

  Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

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

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

- **Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

- **Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

- **Scikit – learn**

  Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

5.4 MACHINE LEARNING MODELS

Decision Tree Classifier: For classification and regression applications, decision trees are commonly used models. They basically learn a hierarchy of if/else questions that leads to a choice. Learning a decision tree is memorizing the sequence of if/else questions that leads to the correct answer in the shortest amount of time. The method runs through all potential tests to discover the one that is most informative about the target variable to build a tree.

Random Forest Classifier: Random forests are one of the most extensively used machine learning approaches for regression and classification. A random forest is just acollection of decision trees, each somewhat different from the others. The notion behind random forests is that while each tree may do a decent job of predicting, it will almost certainly overfit on some data. They are incredibly powerful, frequently operate effectively without a lot of parameters adjusting, and don't require data scalability.

5.5 MACHINE LEARNING ALGORITHMS

5.5.1 DECISION TREE ALGORITHM

There's not much mathematics involved here. Since it is very easy to use and interpret it is one of the most widely used and practical methods used in Machine Learning. It is atool that has applications spanning several different areas. Decision trees can be used for classification as well as regression problems.

The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.

Root Nodes – It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.

Decision Nodes – the nodes we get after splitting the root nodes are called Decision Node.

Leaf Nodes – the nodes where further splitting is not possible are called leaf nodes or terminal nodes.

Sub-tree – just like a small portion of a graph is called sub-graph similarly a subsection of this decision tree is called sub-tree.

Pruning – is nothing but cutting down some nodes to stop overfitting.

Entropy: Entropy is nothing but the uncertainty in our dataset or measure of disorder. Let me try to explain this with the help of an example. Suppose you have a group of friends who decides which movie they can watch together on Sunday. There are 2 choices for movies, one is "Lucy" and the second is "Titanic" and now everyone has to tell their choice. After everyone gives their answer we see that "Lucy" gets 4 votes and "Titanic" gets 5 votes. Which movie do we watch now? Isn't it hard to choose 1 movie now because the votes for both the movies are somewhat equal. This is exactly what we call disorderness, there is an equal number of votes for both the movies, and we can't really decide which movie we should watch. It would have been much easier if the votes for "Lucy" were 8 and for "Titanic" it was 2. Here we could easily say that the majority of votes are for "Lucy" hence everyone will be watching this movie.

Information Gain: Information gain measures the reduction of uncertainty given some feature and it is also a deciding factor for which attribute should be selected as a decision node or root node. It is just entropy of the full dataset – entropy of the dataset given some feature. To understand this better let's consider an example: Suppose our entire population has a total of 30 instances.

The dataset is to predict whether the person will go to the gym or not. Let's say people go to the gym and 14 people don't Now we have two features to predict whether he/she will go to the gym or not. Feature 1 is "Energy" which takes two values "high" and "low" Feature 2 is "Motivation" which takes 3 values "No motivation", "Neutral" and "Highlymotivated".

Let's see how our decision tree will be made using these 2 features. We'll use information gain to decide which feature should be the root node and which feature should be placed after the split.

Let's calculate the entropy: To see the weighted average of entropy of each node we will do as follows: Now we have the value of E(Parent) and E(Parent|Energy), information gain will be: Our parent entropy was near 0.99 and after looking at this value of information gain, we can say that the entropy of the dataset willdecrease by 0.37 if we make "Energy" as our root node. Similarly, we will do this with the other feature "Motivation" and calculate its information gain. In this example "Energy" will be our root node and we'll do the same for sub-nodes. Here we can see that when the energy is "high" the entropy is low and hence we can say a person will definitely go to the gym if he has high energy, but what if the energy is low? We will again split the node based on the new feature which is "Motivation".

You must be asking this question to yourself that when do we stop growing our tree? Usually, real-world datasets have a large number of features, which will result in a large number of splits, which in turn gives a huge tree. Such trees take time to build and can lead to overfitting. That means the tree will give very good accuracy on the training dataset but will give bad accuracy in test data. There are many ways to tackle this problem through hyperparameter tuning. We can set the maximum depth of our decision tree using the max_depth parameter. The more the value of max_depth, the more complex your tree will be. The training error will off-course decrease if we increase the max_depth value but when our test data comes into the picture, we will geta very bad accuracy. Hence you need a value that will not overfit as well as underfit our data and for this, you can use Grid Search CV.

Another way is to set the minimum number of samples for each spilt. It is denoted by min_samples split. Here we specify the minimum number of samples required to do a split. For example, we can use a minimum of 10 samples to reach a decision. That means if a node has less than 10 samples then using this parameter, we can stop the further splitting of this node and make it a leaf node.

Pruning: It is another method that can help us avoid overfitting. It helps in improving the performance of the tree by cutting the nodes or sub-nodes which are not significant. It removes the branches which have very low importance.

There are mainly 2 ways for pruning:

Pre-pruning – we can stop growing the tree earlier, which means we can prune/remove/cut a node if it has low importance while growing the tree.

Post-pruning – once our tree is built to its depth, we can start pruning the nodes based on their significance.

5.5.2 RANDOM FOREST ALGORITHM

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One ofthe most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems. Let’s dive into a real-life analogy to understand this concept further. A student named X wants to choose a course after his 10+2, and he is confused about the choice of course based on his skill set. So he decides to consult various people like his cousins, teachers, parents, degree students, and working people. He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc. Finally, after consulting various people about the course he decides to take the course suggested by most of the people.

Ensemble uses two types of methods:

1.**Bagging–** It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.

2.**Boosting–** It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST,XG BOOST.

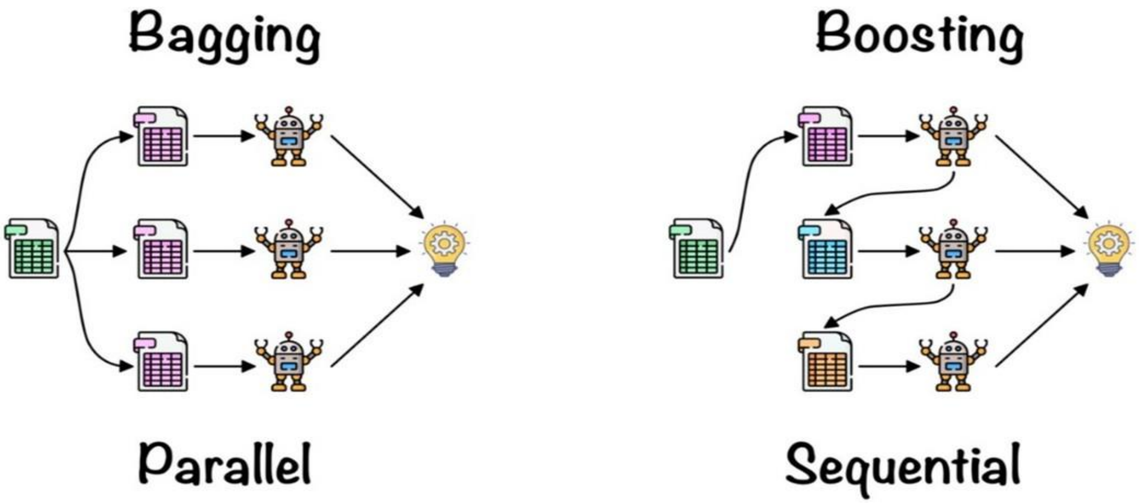


Fig 5.5.2 Random Forest models

Bagging: Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

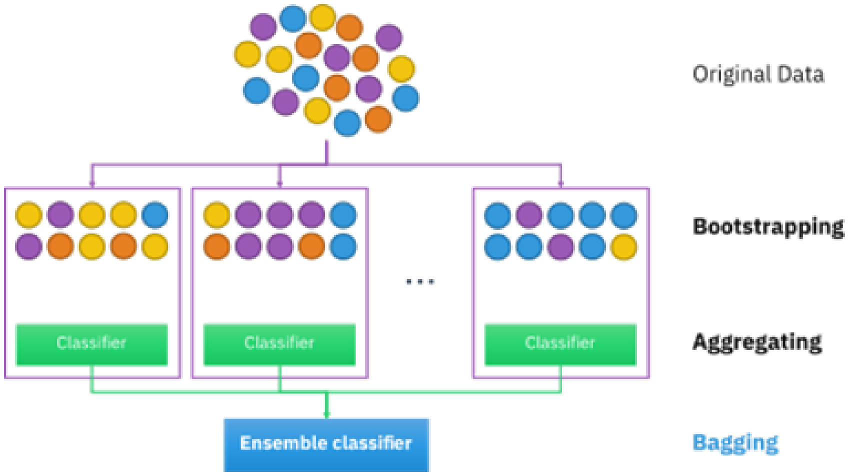


Fig 5.5.2 Bagging

Steps involved in random forest algorithm:

Step 1: In Random forest n number of random records are taken from the data sethaving k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

For example: consider the fruit basket as the data as shown in the figure below. Now a number of samples are taken from the fruit basket and an individual decision tree is constructed for each sample. Each decision tree will generate an output as shown in the figure. The final output is considered based on majority voting. In the below figure you can see that the majority decision tree gives output as an apple when compared to a banana, so the final output is taken as an apple.

Important Features of Random Forest

- **Diversity-** Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- **Dimensionality**- Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization**-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
- **Train-Test split-** In a random forest we don’t have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability-** Stability arises because the result is based on majority voting/ averaging.

CHAPTER-6: SYSTEM STUDY

6.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposalis 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. Thisis 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.

The feasibility study investigates the problem and the information needs of thestakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of sucha solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

- Three key considerations involved in the feasibility analysis are
  - ECONOMICAL FEASIBILITY
  - TECHNICAL FEASIBILITY
  - SOCIAL FEASIBILITY

6.1.1 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 justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available.

6.1.2 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.

6.1.3 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.

CHAPTER-7: SYSTEM TESTING

7. 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 that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test.

7.1 TYPES OF TESTS

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logicis 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. 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.

7.1.2 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 satisfaction, as shown by successfully 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.

7.1.3 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 accepted.

Invalid Input: identified classes of invalid input must 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.

7.1.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which 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.

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.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach: Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested:

- Verify that the entries are of the correct format No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

7.1.5 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is averification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

CHAPTER-8: RESULTS

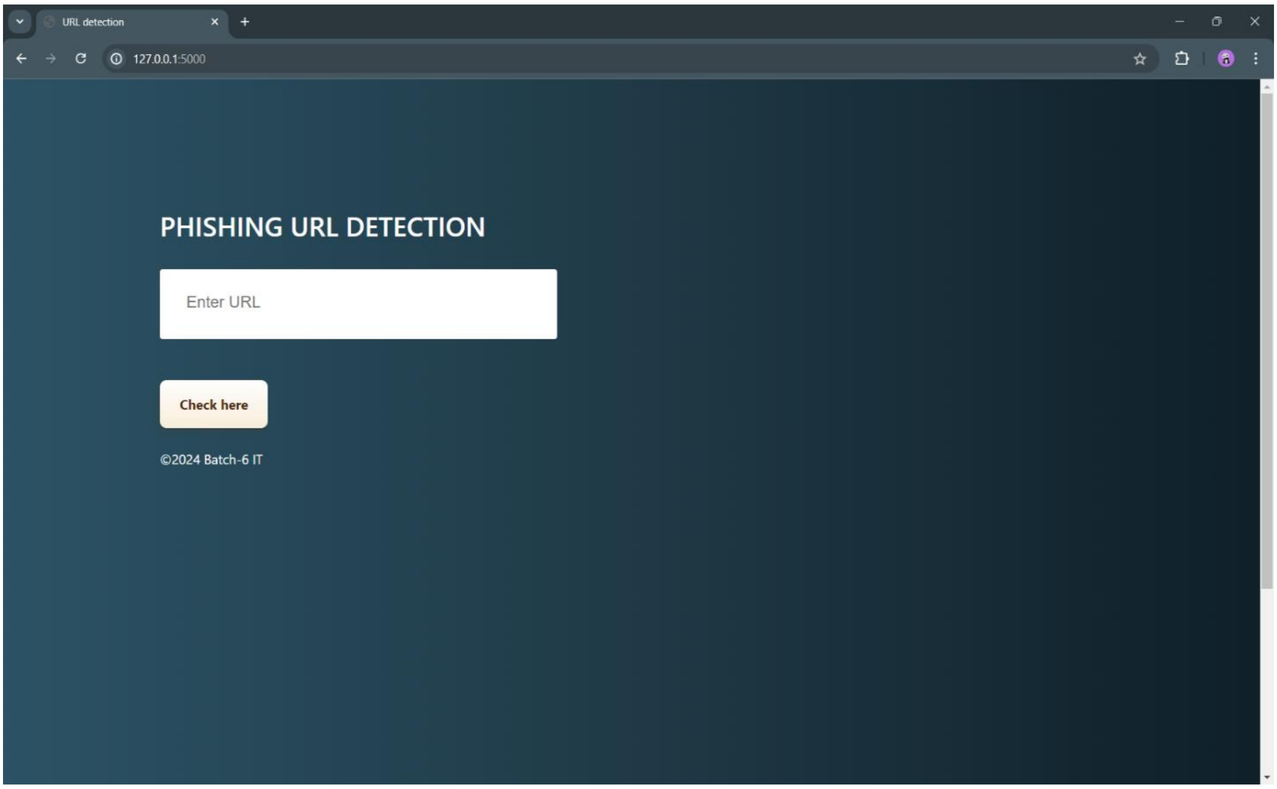
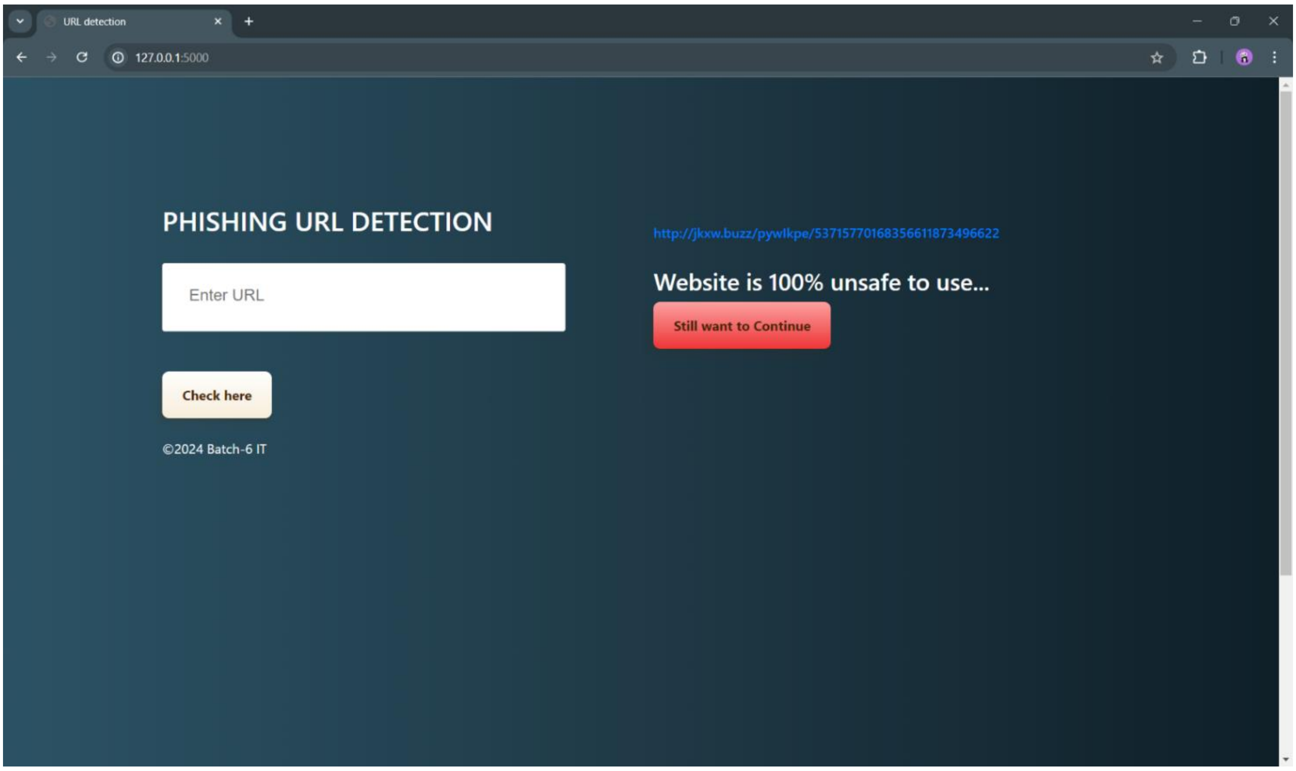


Fig 8.1 Input page

Navigation steps:

- **Step 1 :** Copy & paste the URL you want to check
- **Step 2 :** Click on check here
- **Step 3 :** It will display the output



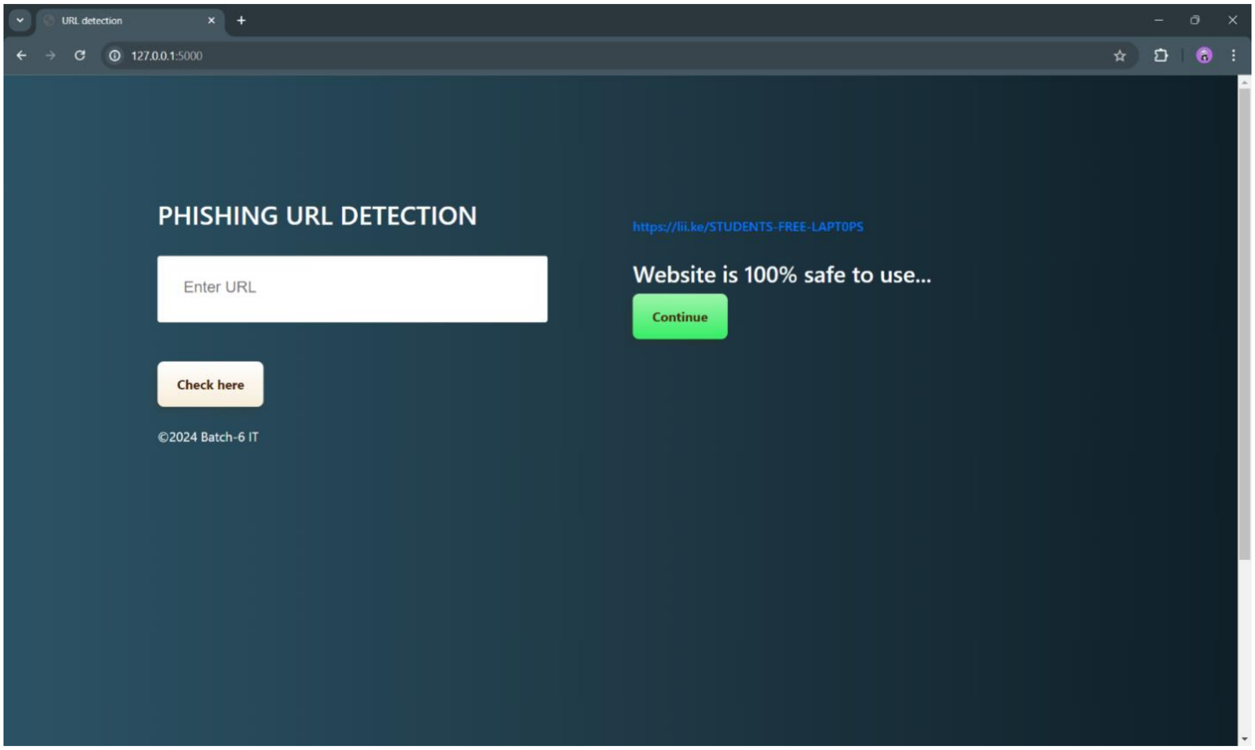
http://jkxw.buzz/pywlkpe/5371577016835661187349662

Website is 100% unsafe to use...

Fig 8.2 Output Page

Navigation steps:

- **Step 1 :** It will display the output whether it is safe/unsafe
- **Step 2 :** Click on still want to continue to open page



PHISHING URL DETECTION

https://lii.ke/STUDENTS-FREE-LAPTOPS

Website is 100% safe to use...

Fig 8.3 Output Page

Navigation steps:

- **Step 1 :** It will display the output whether it is safe/unsafe
- **Step 2 :** Click on continue to open page

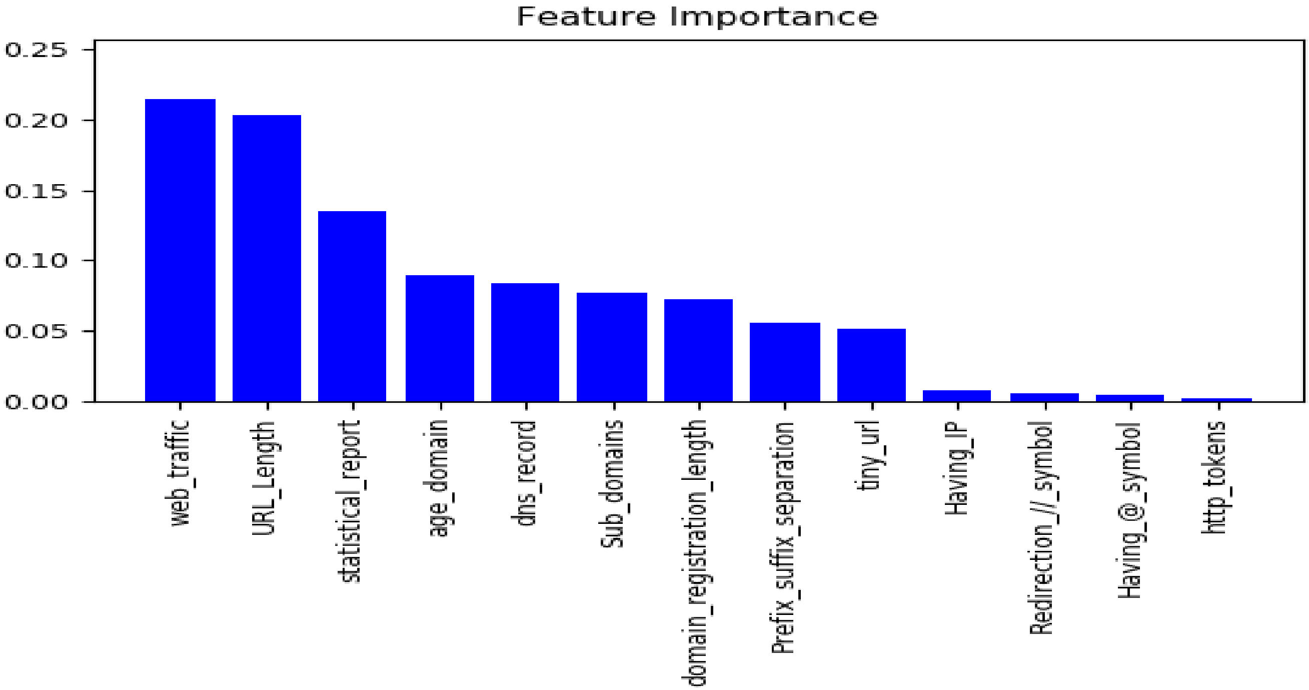


Fig 8.4 Accuracy graph

CHAPTER-9: CONCLUSION & FUTURE ENHANCEMENT

9.1 Conclusion

This survey presented various algorithms and approaches to detect phishing websites by several researchers in Machine Learning. On reviewing the papers, we came to a conclusion that most of the work done by using familiar machine learning algorithms like Naïve Bayesian, SVM, Decision Tree and Random Forest. Some authors proposed a new system like Phish Score and Phish Checker for detection. The combinations of features with regards to accuracy, precision, recall etc. were used. Experimentally successful techniques in detecting phishing website URLs were summarized As phishing websites increases day by day, some features may be included or replaced with new ones to detect them.

Phishing is a cyber crime procedure utilizing both social building and specialized deception to take individual sensitive data. Besides, Phishing is considered as another extensive type of fraud. Experimentations against recent dependable phishing data sets utilizing different classification algorithm have been performed which received different learning methods. The base of the experiments is accuracy measure.

The aim of this research work is to predict whether a given URL is phishing website or not. It turns out in the given experiment that Random forest based classifiers are the best classifier with great classification accuracy of 82.644% for the given dataset of phishing site. As a future work we might use this model to other Phishing dataset with larger size then now and then testing the performance of those classification algorithm's in terms of classification accuracy.

9.2 Future Enhancement

A future work will focus on collecting phishing and non-phishing web-sites that are currently accessible in the www and extract a list of features that are different from the one commonly used in phishing detection, such as the "time-to-response".

A potential area of research is comparing several rules learner techniques to be used with the Fuzzy Inference Process. Another potential area is reducing the number of rules to enhance the system performance and studying which rule has more impact on the system output.

CHAPTER-10: BIBILOGRAPHY

[1] ‘APWG | Unifying The Global Response To Cybercrime’ (n.d.) available: https://apwg.org/

[2] 14 Types of Phishing Attacks That IT Administrators Should Watch For [online] (2021)https://www.blog.syscloud.com,available:https://www.blog.syscloud.comtypes-o f-phishing/

[3] Lakshmanarao, A., Rao, P.S.P., Krishna, M.M.B. (2021) ‘Phishing website detection using novel machine learning fusion approach’, in 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Presented at the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 1164–1169

[4] H. Chapla, R. Kotak and M. Joiser, "A Machine Learning Approach for URL Based Web Phishing Using Fuzzy Logic as Classifier", 2019 International Conference on Communication and Electronics Systems (ICCES), pp. 383-388, 2019, July

[5] Vaishnavi, D., Suwetha, S., Jinila, Y.B., Subhashini, R., Shyry, S.P. (2021) ‘A Comparative Analysis of Machine Learning Algorithms on Malicious URL Prediction’, in 2021 5th International Conference on Intelligent Computing and ControlSystems (ICICCS), Presented at the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 1398–1402

[6] Microsoft, Microsoft Consumer safety report. https://news.microsoft.com/en-sg/2014/02/11/microsoft-consumersafety-index-reveals- impact-of-poor-online-safety-behaviours-in-singapore/sm.001xdu50tlxsej410r11kqvks u4nz.

[7] Internal Revenue Service, IRS E-mail Schemes. Available at https://www.irs.gov/uac/newsroom/consumers-warnedof-new-surge-in-irs-email-schem es-during-2016-tax-season-tax-industry-also-targeted.

[8] Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S. (2007), A comparison of machinelearning techniques for phishing detection.

Proceedings of the Anti-phishing Working Groups 2nd Annual ECrime ResearchersSummit on - ECrime ’07.

doi:10.1145/1299015.1299021.

[9] E., B., K., T. (2015)., Phishing URL Detection: A Machine Learning and WebMining-based Approach. International Journal of

Computer Applications,123(13), 46-50. doi:10.5120/ijca2015905665.

[10] Wang Wei-Hong, L V Yin-Jun, CHEN Hui-Bing, FANG Zhao-Lin., A StaticMalicious Javascript Detection Using SVM, In Proceedings of the 2nd International Conference on Computer Science and Electrical Engineering (ICCSEE 2013).

[11] Ningxia Zhang, Yongqing Yuan, Phishing Detection Using Neural Net-work, In Proceedings of International Conference on Neural Information Processing, pp. 714–719. Springer, Heidelberg (2004).

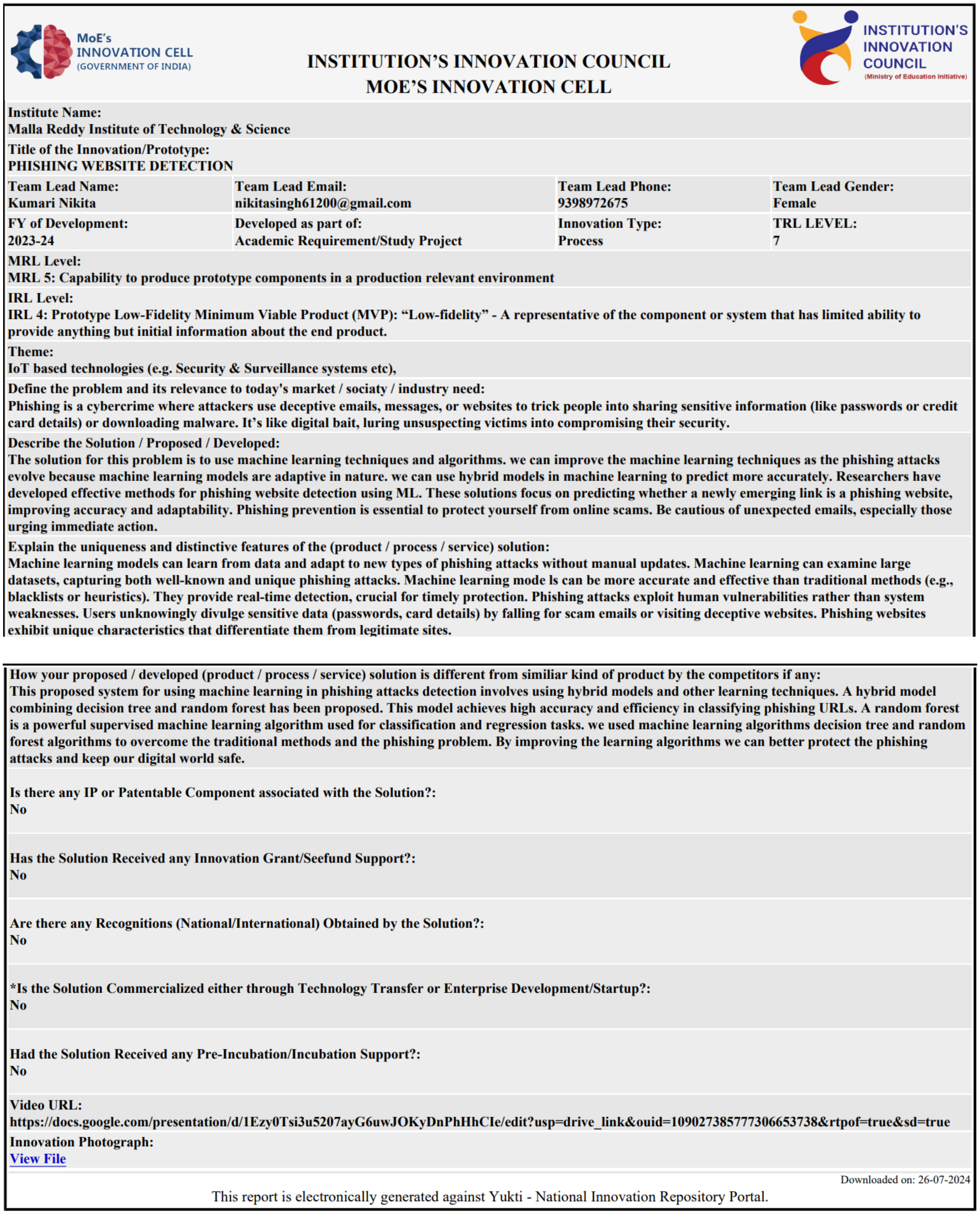
[12] Ram Basnet, Srinivas Mukkamala et al, Detection of Phishing Attacks: A MachineLearning Approach, In Proceedings of the International World Wide Web Conference (WWW), 2003.

[13] Sci-kit learn, SVM library. http://scikit-learn.org/stable/modules/svm.html.

[14] Sklearn, ANN library. http://scikit-learn.org/stable/modules/ann.html.

[15]Sclera,Randomforestlibrary.http://scikitlearn.org/stable/modules/Randomforese ts.html.

CHAPTER-11 : YUKTHI INNOVATION CERTIFICATE



MoE's
INNOVATION CELL
(GOVERNMENT OF INDIA)

INSTITUTION'S INNOVATION COUNCIL
MOE'S INNOVATION CELL

INSTITUTION'S
INNOVATION
COUNCIL
(Ministry of Education Initiative)

Institute Name:
Malla Reddy Institute of Technology & Science

Title of the Innovation/Prototype:
PHISHING WEBSITE DETECTION

Team Lead Name: **Kumari Nikita**	**Team Lead Email:** **nikitasingh61200@gmail.com**	**Team Lead Phone:** **9398972675**	**Team Lead Gender:** **Female**
FY of Development: **2023-24**	**Developed as part of:** **Academic Requirement/Study Project**	**Innovation Type:** **Process**	**TRL LEVEL:** **7**

MRL Level:
MRL 5: Capability to produce prototype components in a production relevant environment

IRL Level:
IRL 4: Prototype Low-Fidelity Minimum Viable Product (MVP): "Low-fidelity" - A representative of the component or system that has limited ability to provide anything but initial information about the end product.

Theme:
IoT based technologies (e.g. Security & Surveillance systems etc),

Define the problem and its relevance to today's market / sociaty / industry need:
Phishing is a cybercrime where attackers use deceptive emails, messages, or websites to trick people into sharing sensitive information (like passwords or credit card details) or downloading malware. It's like digital bait, luring unsuspecting victims into compromising their security.

Describe the Solution / Proposed / Developed:
The solution for this problem is to use machine learning techniques and algorithms. we can improve the machine learning techniques as the phishing attacks evolve because machine learning models are adaptive in nature. we can use hybrid models in machine learning to predict more accurately. Researchers have developed effective methods for phishing website detection using ML. These solutions focus on predicting whether a newly emerging link is a phishing website, improving accuracy and adaptability. Phishing prevention is essential to protect yourself from online scams. Be cautious of unexpected emails, especially those urging immediate action.

Explain the uniqueness and distinctive features of the (product / process / service) solution:
Machine learning models can learn from data and adapt to new types of phishing attacks without manual updates. Machine learning can examine large datasets, capturing both well-known and unique phishing attacks. Machine learning mode ls can be more accurate and effective than traditional methods (e.g., blacklists or heuristics). They provide real-time detection, crucial for timely protection. Phishing attacks exploit human vulnerabilities rather than system weaknesses. Users unknowingly divulge sensitive data (passwords, card details) by falling for scam emails or visiting deceptive websites. Phishing websites exhibit unique characteristics that differentiate them from legitimate sites.

How your proposed / developed (product / process / service) solution is different from similiar kind of product by the competitors if any:
This proposed system for using machine learning in phishing attacks detection involves using hybrid models and other learning techniques. A hybrid model combining decision tree and random forest has been proposed. This model achieves high accuracy and efficiency in classifying phishing URLs. A random forest is a powerful supervised machine learning algorithm used for classification and regression tasks. we used machine learning algorithms decision tree and random forest algorithms to overcome the traditional methods and the phishing problem. By improving the learning algorithms we can better protect the phishing attacks and keep our digital world safe.

Is there any IP or Patentable Component associated with the Solution?:
No

Has the Solution Received any Innovation Grant/Seefund Support?:
No

Are there any Recognitions (National/International) Obtained by the Solution?:
No

***Is the Solution Commercialized either through Technology Transfer or Enterprise Development/Startup?:**
No

Had the Solution Received any Pre-Incubation/Incubation Support?:
No

Video URL:
https://docs.google.com/presentation/d/1Ezy0Tsi3u5207ayG6uwJOKyDnPhHhCIe/edit?usp=drive_link&ouid=109027385777306653738&rtpof=true&sd=true

Innovation Photograph:
View File

Downloaded on: 26-07-2024

This report is electronically generated against Yukti - National Innovation Repository Portal.