e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

PHISHING WEBSITES DETECTION USING MACHINE LEARNING

Prof. Rohit Mane[*1], Mr. Prithviraj Satpute[*2], Mr. Shivam Saraswat[*3], Mr. Siddharth Kulkarni[*4], Mrs. Gayatri Suryawanshi[*5], Mrs. Shrushti Chalke[*6]

[*1,2,3,4,5,6]Shivaji University, Department Of Computer Science , Dr. J.J. Magdum College Of Engineering, Jaysingpur, Maharashtra, India.

ABSTRACT

Phishing is a cyber-attack technique where malicious actors send deceptive messages that appear to originate from trusted sources, aiming to steal sensitive information or infect systems. This study presents a machine learning-based approach to detect phishing URLs in real time. The proposed system utilizes the Gradient Boosting Classifier to differentiate between legitimate and phishing websites by analyzing various URL features. The model is trained on a dataset containing both legitimate and phishing links, extracting distinguishing characteristics such as URL length, presence of special characters, domain age, and security indicators. Among several algorithms tested, including SVM, Neural Networks, Random Forest, and Decision Tree, the Gradient Boosting Classifier demonstrated superior accuracy. The results indicate that the model can effectively identify and block phishing attempts, offering a practical solution for enhancing cybersecurity. This approach contributes to reducing the risk of phishing attacks by providing users with timely and accurate detection of malicious URLs.

Keywords: Phishing, Machine Learning, Gradient Boosting, URL Detection, Cybersecurity, Real-Time Classification.

I. INTRODUCTION

Phishing is a fraudulent cyber technique that combines social engineering and technological deception to steal sensitive information such as user credentials and financial details. As digital platforms become increasingly integral to everyday life—including sectors like banking, education, communication, health, and commerce—the volume and impact of cyber-attacks have grown significantly. The convenience of mobile and wireless technologies has expanded internet access globally, but also exposed users to greater security risks. Among various cyber threats, phishing attacks are the most widespread and harmful, leading to financial losses and compromising sensitive data.

Phishing typically involves attackers sending deceptive emails or messages designed to appear as if they originate from legitimate institutions. These messages often contain malicious links or attachments that redirect users to fake websites, replicating the appearance of real ones to trick users into submitting their confidential information. Due to the easy replication of logos and website designs via HTML, phishing has become increasingly sophisticated and difficult to detect.

Conventional phishing detection methods, such as blacklists and heuristic-based approaches, often fail to identify new or evolving phishing attacks. Blacklists cannot detect zero-hour attacks, and heuristic methods may generate high false positives. In response, researchers have turned toward machine learning-based solutions. These systems can analyze large datasets and learn patterns associated with phishing websites, offering real-time and adaptive detection capabilities.

This study aims to address the challenges of phishing detection by implementing a model based on machine learning algorithms such as Logistic Regression, K-Nearest Neighbors, Support Vector Classification, Random Forest, Decision Tree, XGBoost Classifier, and Naïve Bayes. The proposed system uses features extracted from user-submitted URLs to accurately classify websites as legitimate or phishing. With phishing attacks continuing to evolve and expand, developing an intelligent, automated detection system is both necessary and timely to safeguard users in the digital age.

1.1 PROBLEM STATEMENT

1. The Cyber-attacks are growing faster than usual rate, it became evident that necessary steps should be taken in-order to get them under control. Among various cyber-attacks, Phishing websites is one of the popular and commonly used attack to steal users personal information and financial information by manipulating the website URL and IP addresses.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

2. The main focus in this project is to implement the better model for detecting these phishing websites using ML algorithms.

1.2 EXISTING SYSTEM

"Phishing Detection Using Machine Learning": This paper proposes an approach of phishing detection system to detect blacklisted URL also known as phishing websites, so that individual can be alerted while browsing or accessing a particular website. Therefore, it can be utilized for identification and authentication and become a legitimate tool to prevent an individual from getting tricked. The system fosters many features in comparison of other software. Its unique features such as capturing blacklisted URL's from the browser directly to verify the validity of the website, notifying user on blacklisted websites while they are trying to access through popup, and also notifying through email. This system will assist user to be alert when they are trying to access a blacklisted website.

1.2.1 LIMITATIONS

i. As a list of blacklisted URLs is being used to detect the accuracy of predicting the correct results may be very low. There isn't any standard list of URLs published by any standard organization.

ii. As a part of daily life, we may encounter many new URL's which seems to be same as original ones and not present in the list. It is hard to differentiate URL'S.

1.3 PROPOSED SYSTEM

As part of our project, we are implementing a new framework to detect these phished websites using Machine Learning Algorithms. As we aren't using any list of URLs, this can be used to detect any new URL that the user encounters. We use URL features as parameters to the model. There are 30 features that are being considered as major decision parameters. Using these 30 features an ML model is made to meet the challenges in existing system.

1.3.1 ADVANTAGES

The major advantages of this project are:

- We can identify fake websites and helps users safe from exploitation of their information.
- We can stop unsafe transactions identifying these fake websites.
- We can add this approach to any browser to make it secure browser.

1.4 SIGNIFICANCE AND RELEVANCE OF WORK

a. In today's society, as the phishing attacks have become evident, the need of counteractions are necessary.

b. The main problem of Phishing has raised abundantly in the last 3 years due to Covid –

c. 19. Number of Cases encountering on a daily basis has increased by 125% in 2021 from 2020. If this is the scenario happening, the phishing cases may raise up to 500 per day by 2025.

d. Phishing detection techniques do suffer low detection accuracy and high false alarm especially when novel phishing approaches are introduced.

e. Besides, the most common technique used, blacklist-based method is inefficient in responding to emanating phishing attacks since registering new domain has become easier, no comprehensive blacklist can ensure a perfect up-to-date database.

f. Therefore, we are developing a website to predict the phishing website URL's which helps the society to cut down these attacks and save themselves from the frauds.

1.5 OBJECTIVES

a. To develop a novel approach to detect malicious URL and alert users.

b. To apply ML techniques in the proposed approach in order to analyze the real time URLs and produce effective results.

c. Using zero - hour mechanism to cut-down Phishing attacks.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

II. METHODOLOGY

The proposed system for phishing website detection is developed through the following sequential phases:

1) Data Collection

The dataset is collected from reliable sources such as Kaggle and includes various features that help distinguish between legitimate and phishing websites. Features include domain identity, security and encryption criteria, page content-based attributes, and URL characteristics.

2) Data Preprocessing

The dataset is preprocessed to handle missing values, normalize numerical features, and encode categorical data. This ensures uniformity and improves model performance. The dataset is then split into training and testing subsets.

3) Feature Selection

Relevant features contributing significantly to phishing detection are selected using statistical methods and domain expertise. This step helps in reducing the dimensionality and enhancing the model's accuracy and efficiency.

4) Model Development

The Gradient Boosting Classifier (GBC), a robust ensemble learning technique, is employed for classification. It combines multiple weak learners to form a strong predictive model. The model is trained using the training data and tuned for hyperparameters using cross-validation techniques.

5) Model Evaluation

The trained model is evaluated using the testing data. Performance metrics such as Accuracy, Precision, Recall, and F1-Score are used to assess the effectiveness of the classifier in identifying phishing websites.

6)Web Application Development

A user-friendly web application is developed using Flask as the backend framework. The front end is designed using HTML, CSS, and JavaScript. Users can input a website URL, and the system will predict whether the site is phishing or legitimate.

7) Integration and Deployment

The machine learning model is integrated into the web application. The complete system is tested for usability and performance, and then deployed for real-time use.

FEASIBILITY STUDY

A feasibility study is a crucial preliminary step in the planning and initiation of a project, providing an in-depth assessment of the practicality, viability, and potential success of the proposed project. The primary objective is to determine whether the project is worth pursuing by evaluating various aspects such as technical feasibility, operational feasibility, economic feasibility, legal feasibility, and scheduling feasibility. The findings of a feasibility study provide stakeholders, including project managers, investors, and decision-makers, with valuable insights to make informed decisions about whether to proceed with the project. If the study reveals that the project is not feasible, stakeholders can save resources by avoiding investments in an endeavor that may not yield the desired outcomes. Conversely, if the study indicates feasibility, it serves as a foundation for detailed project planning and execution.

TECHNICAL FEASIBILITY

Technical feasibility, a key component of the broader feasibility study, is a comprehensive evaluation that focuses on the technological aspects of a proposed project. This assessment is critical in determining whether the envisaged solution aligns with the existing technology infrastructure and can be effectively implemented within the technical capabilities of the organization. In the initial stages of technical feasibility analysis, there is a thorough examination of the required technology. This involves assessing the availability and suitability of essential components, such as hardware, software, and any specialized tools or systems necessary for project execution. This step ensures that the proposed project's technological needs can be met either through existing resources or by acquiring or developing new technology.

An essential aspect of technical feasibility is evaluating the technical skills and expertise available within the

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

organization. This entails assessing the competency of the development team, understanding whether additional training is required, or if there is a need to augment the team with individuals possessing specific technical skills. A well-equipped and skilled team is fundamental to overcoming technical challenges and ensuring the successful implementation of the project.

Risk analysis is inherent in technical feasibility, aiming to identify and mitigate potential obstacles that could hinder project success. This includes assessing risks related to technology obsolescence, security vulnerabilities, or dependencies on external systems. Proactive identification and mitigation of these risks.

OPERATIONAL FEASIBILITY

Operational feasibility is a crucial aspect of a feasibility study that assesses the practicality and viability of implementing a proposed project within an organization's existing operational environment. This evaluation focuses on whether the proposed solution can seamlessly integrate into the daily business operations and whether it is acceptable and adaptable for end-users. The primary goal is to ensure that the project aligns with the organizational processes and can be effectively utilized by stakeholders without causing disruptions.Key considerations in operational feasibility include user acceptance, training requirements, and potential changes in roles and responsibilities. Evaluating whether the proposed project can be seamlessly integrated into the day-to-day activities of the organization ensures that the implementation process is smooth and does not adversely affect productivity. A positive operational feasibility assessment indicates that the proposed solution is not only technically and economically viable but is also practical and operationally sound within the context of the organization's current operational landscape.

ECONOMIC FEASIBILITY

Economic feasibility is a critical component of a feasibility study that evaluates the financial viability and potential economic benefits of a proposed project. This assessment involves a thorough analysis of the costs associated with project development and implementation against the expected economic returns and benefits. The primary objective is to determine whether the project is financially justifiable and aligns with the organization's budgetary constraints.In the economic feasibility analysis, various costs are considered, including development costs, operational costs, maintenance costs, and any other expenditures associated with the project's lifecycle. These costs are compared to the anticipated benefits, which may include revenue generation, cost savings, increased efficiency, or economic advantages. This comparison helps stakeholders understand the financial implications of undertaking the project and whether the expected benefits outweigh the incurred costs.

The dataset used in this machine learning project was obtained from Kaggle, a well- known platform for data science competitions and datasets. 11430 URLs with 30 retrieved characteristics are part of the supplied dataset. The dataset is intended to serve as the benchmark for phishing detection systems that employ machine learning. The collection comprises precisely 45% genuine URLs and 55% phishing URLs. Each instance contains 30 features. Each feature is associated with a rule. If the rule satisfies, it is termed as phishing. If the rule doesn‘t satisfy then it is termed as legitimate. The features take three discrete values. 1 if the rule is satisfied, 0 if the rule is partially satisfied and -1 if the rule is not satisfied. For this research implementation, this dataset is used since it is the most recent dataset accessible in the public domain.

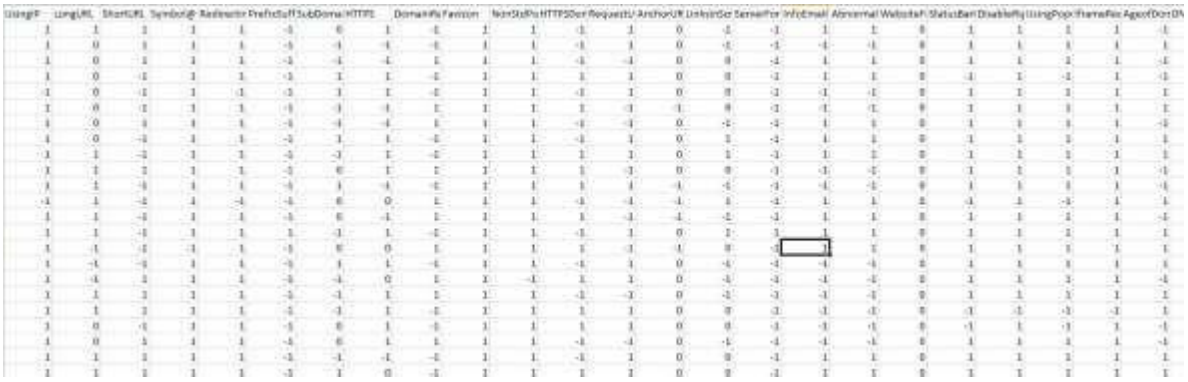

Fig: Sample of dataset

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

FEATURES OF DATASET

The dataset contains several factors that should be considered when deciding whether a website URL is licit or phishing. The factors for discovery and bracket of phishing websites are as follows

- Address Bar based features
- Abnormal based features
- HTML and JavaScript based features
- Domain based features.

Address Bar Based Features

1. Having an IP Address

If an IP address is used in the URL instead of the domain name, such as http://217.102.24.235/sample.html.

2. Length of URL

Phishers may conceal the suspicious element of the URL in the address bar by using a lengthy URL.

3. URL Shortening Service

Provides access to a website with a lengthy URL. The URL http://sharif.hud.ac.uk/, for example, may be abbreviated to bit.ly/1sSEGTB.

4. Using the @ sambol

@ in the URL causes the browser to disregard anything before the @ symbol, and the true address often follows the @ symbol.

5. Double Slash Redirection

The presence of / in a URL indicates that the user will be redirected to another website.

6. SSL Status

Indicates whether or not a website employs SSL.

7. Domain Registration Length

Because a phishing website only exists for a brief time,

8. Favicon

A favicon is a visual image (icon) that is connected with a particular website. If the favicon is loaded from a domain different than the one displayed in the address bar, the site is most certainly a Phishing attempt.

9. Using Non-Standard Ports

It is much preferable to just open the ports that you need to regulate invasions. Several firewalls, proxy servers, and Network Address Translation (NAT) servers will, by default, block all or most of the ports and only allow access to those that are explicitly allowed.

10.HTTPS token

Using a deceptive https token in the URL. For instance, http://www.mellat-phish.ir

Abnormal Based Features

11 Anchor URL

An anchor is an element specified by the a > tag. This feature is processed in the same way as Request URL.

12 Links in Tags

Websites often utilize Meta tags to provide information about the HTML content, Script tags to generate a client-side script, and Link tags to fetch external online resources. If the domain name in SFHs differs from the domain name of the site.

13 Server from Handler (SFH)

Because process should be performed upon that reported file, SFHs that contains an empty string or roughly clean are suspicious.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

HTML And JavaScript Based Features

14 Website Redirect Count

If the redirection occurs more than four times, it is considered excessive.

15 Status Bar Customization

Utilize JavaScript to display a bogus URL to users in the status bar.

16 Disabling Right Click

This is the same as using on Mouse Over to conceal the Link.

17 Using Pop-up Window

Demonstrating the presence of pop-up windows on the website.

18 IFrame

An IFrame is an HTML element that displays an extra website inside the one that is now shown.

Domain Based Features

19 Domain Age

If the domain is less than a month old.

20 DNS Record

Possessing a DNS record

21 Web Traffic

The number of visits to a website is used to determine its popularity.

22 Page Rank

Page rank is a number that ranges from 0 to 1. PageRank attempts to determine the importance of a site on the Internet.

23 Google Index

This function determines whether or not a website is included in Google's index.

24 Number of Links Pointing To Page

The number of links that point to the web page.

25 Statistical Report

Determine whether or not the IP address

FUNCTIONAL REQUIREMENTS

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality.

In this system following are the functional requirements:

1. Training dataset must be loaded
2. Hoefflin Anytime tree model must be trained from the dataset
3. User enters the URL in URL locator.
4. URL must be detected and display whether it is legitimate or non-fake website.

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:

1. Product Requirements
2. Organizational Requirements
3. User Requirements

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

4. Basic Operational Requirements Some of them are as follows:

- Reusability

The same code with limited changes can be used for detecting phishing attacks variants like smishing, vishing, etc.

- Maintainability

The implementation is very basic and includes print statements that makes it easy to debug.

- Usability

The software used is very user friendly and open source. It also runs on any operating system.

- Scalability

The implementation can include detection of vishing, smishing, etc.

REQUIREMENT

a. For desired performance, transferred data size, speed of connection, response time, processing speed must be considered.

b. System should work real – time which means there should be an acceptable time delay between request and response.

c. The system should be reliable to use by the user.

ML PACKAGES

1. 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 is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation.

2. Pandas

Pandas is a python package designed for fast and flexible data processing, manipulation and analysis. Pandas has a number of fundamental data structures (a data management and storage format). Pandas Data Frame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

3. 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. The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as easy of use, code quality, collaboration, documentation and performance.

4. Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. Matplotlib comes with a wide variety of plots. Plots helps to understand trends, patterns, and to make correlations. They're typically instruments for reasoning about quantitative information. Pyplot is a Matplotlib module which provides a MATLAB-like interface. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python and the advantage of being free and open-source. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. The various plots we can utilize using Pyplot are Line Plot, Histogram, Scatter, 3D Plot, Image, contour, and Polar.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

5. Seaborn

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily. As Matplotlib is also used for the same purpose of Data Visualization, Seaborn uses fascinating themes whereas Matplotlib is used only for Basic Graphs.

6. Flask

Flask is a web framework, it's a Python module that lets you develop web applications easily. It's has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like URL routing, template engine. It is a WSGI web app framework.

7. PyMySQL

PyMySQL is a pure-Python MySQL client library, based on PEP 249. Most public APIs are compatible with mysqlclient and MySQLdb. PyMySQL works with MySQL

5.5+ and MariaDB 5.5+.

ML LIBRARIES

1. Whois

pywhois is a Python module for retrieving WHOIS information of domains. pywhois works with Python 2.4+ and no external dependencies.

XGBoost (Extreme Gradient Boosting) belongs to a family of boosting algorithms and uses the gradient boosting (GBM) framework at its core. It is an optimized distributed gradient boosting library.

2. Favicon

Favicons are used in browser tabs, browser history, toolbar apps, bookmarks dropdown, search bar, and search bar recommendations. In all of these, especially in the bookmarks and history tabs, that consist of lists of URLs all looking the same, the favicon makes it faster to find that web-site you're looking for.

3. Requests

Requests library is one of the integral part of Python for making HTTP requests to a specified URL. Whether it be REST APIs or Web Scraping, requests is must to be learned for proceeding further with these technologies. When one makes a request to a URI, it returns a response. Python requests provides inbuilt functionalities for managing both the request and response.

4. Beautiful soup

Python library that is used for web scraping purposes to pull the data out of HTML and XML files. It creates a parse tree from page source code that can be used to extract data in a hierarchical and more readable manner.

5. Google Search

If you want to develop a search service utilizing the power of Google search, you can do so using the google module in Python. You can use this to develop a backend service for a desktop application or implement a website search or app search with the python code running on your server.

ABOUT THE TECHNOLOGY / SOFTWARE

In this section, the tools and methodology to implement text summarization and classification is detailed.

MACHINE LEARNING

Machine learning is a field of artificial intelligence that focuses on the development of algorithms and models capable of learning patterns from data, enabling systems to make predictions or decisions without explicit programming. It encompasses a variety of techniques, from supervised learning, where models learn from labeled data, to unsupervised learning, where patterns are identified without labeled examples. Machine learning algorithms adapt their performance over time as they encounter more data, making them versatile tools for tasks such as classification, regression, clustering, and pattern recognition.

In the context of phishing website detection, machine learning is a pivotal tool employed through a systematic process. It begins with the compilation of a labeled dataset, encompassing both phishing and legitimate websites. Features, such as URL structure and content analysis, are then extracted from this dataset.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

Following data preprocessing, an appropriate machine learning algorithm is selected, and the model is trained to recognize patterns distinguishing between malicious and genuine websites. Validation and hyperparameter tuning ensure the model's efficacy, with evaluation metrics like accuracy and precision guiding the optimization process. Once validated, the model is deployed for real-time detection, often integrated into web browsers or email clients. Continuous monitoring and updates are crucial, given the evolving nature of phishing techniques, and measures are taken to enhance the model's robustness against adversarial attacks. The integration of the machine learning model into broader cybersecurity systems provides a multi-layered defense against phishing threats. This comprehensive approach, combining machine learning with other security measures, strengthens the overall security posture and reduces the risk of falling victim to phishing attacks.

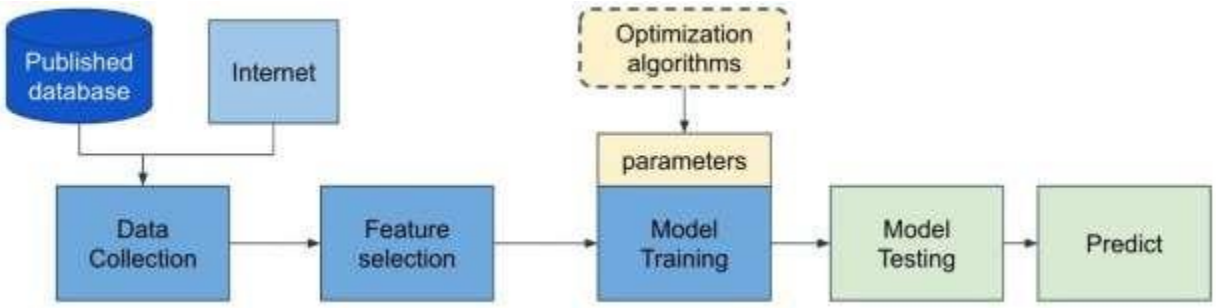

Fig: Phishing Website Detection using Machine Learning

III. INTRODUCTION TO PYTHON

Python is a powerful high-level, object-oriented programming language created by Guido van Rossum. Python is a general-purpose language. It has a wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D).

In the late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language extensible. This led to the design of a new language which was later named Python.

Python is a great and friendly language to use and learn. and can be adapted to both small and large projects. Python will cut a project's development time greatly and overall, it's much faster to write Python than other languages. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks. Its standard library is made up of many functions that come with Python when it is installed. Represent your data, a few of the most important machine learning algorithms, and how to evaluate the performance of your machine learning algorithm.

PYTHON PROGRAM USING ANACONDA

Use Anaconda Navigator to launch an application. Then, create and run a simple Python program with Jupyter Notebook.

Anaconda is a free and open-source distribution of the Python and R programming language for scientific computing (data science , machine learning applications, large-scale data processing, predictive analytics etc.,)that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS

OPEN ANACONDA NAVIGATOR

Choose the instructions for your operating system. Click the Start icon and search for the Navigator

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

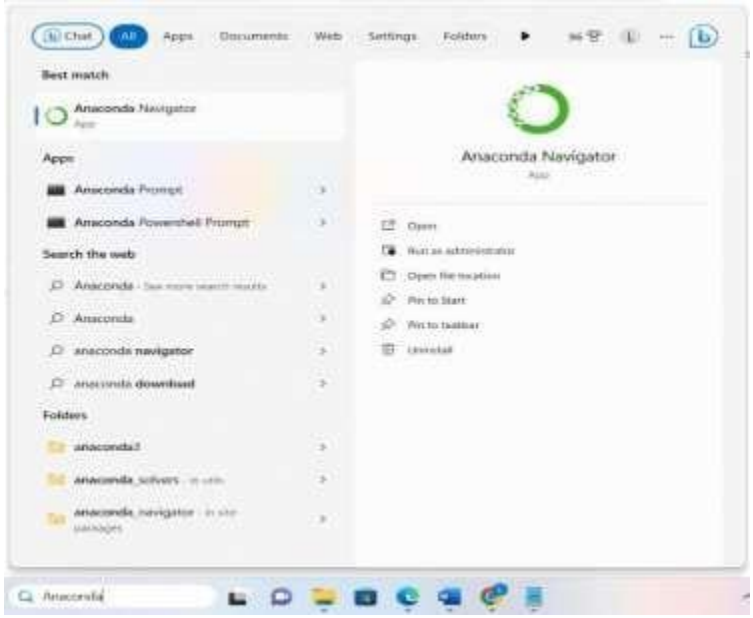

Fig: (a) Anaconda Navigator Search

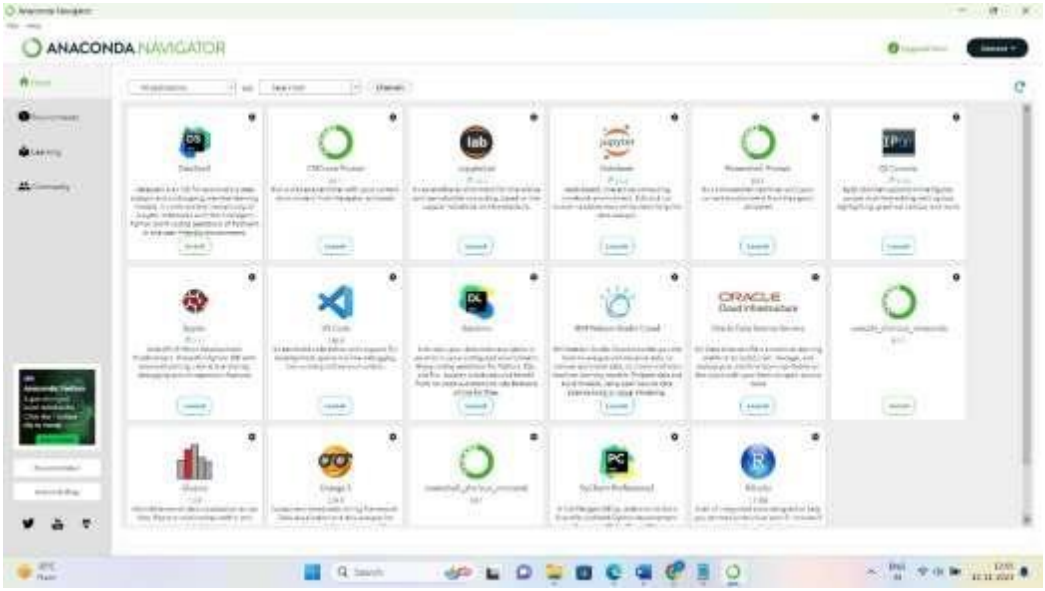

Fig: (b) Aanaconda Navigator Homescreen

RUN PYTHON IN A JUPYTER NOTEBOOK

On Navigator's Home tab, in the Applications pane on the right, scroll to the Jupyter Notebook tile and click the Install button to install Jupyter Notebook.

NOTE: If Jupyter Notebook is already installed, jump right to the Launch step.

- Launch Jupyter Notebook by clicking Jupyter Notebook's Launch button.
- This will launch a new browser window (or a new tab) showing the Notebook Dashboard.
- On the top of the right hand side, there is a drop down labeled "New".
- Create a new Notebook with the Python version installed.
- In the first line of the Notebook, type or copy/paste print("Hello Anaconda").
- Save Notebook by either clicking the save and checkpoint icon or select File - Save and Checkpoint in the top menu.
- Run new program by clicking the Run button or selecting Cell - Run All from the top menu.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

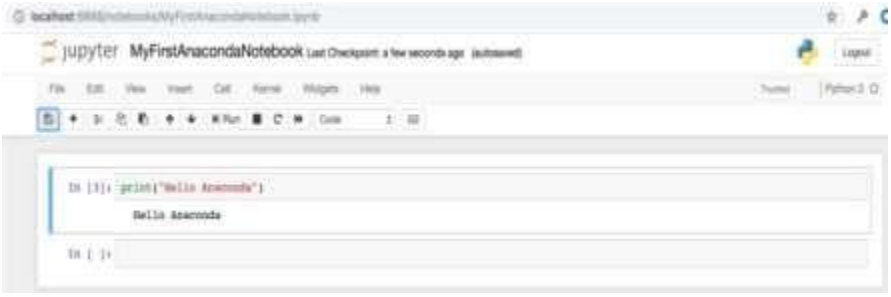

Fig: Jupyter Notebook display

CLOSE JUPYTER NOTEBOOK

From Jupyter Notebooks' top menu bar, select File - Close and Halt. Then click the Quit button at the upper right of the Notebook Dashboard. Close the window or tab.

PYTHON PROGRAMMING

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management.

IV. MODELING AND ANALYSIS

We present the machine learning model used for phishing detection along with the materials and techniques adopted in the research. The model was implemented in Python using libraries such as Scikit-learn, Pandas, NumPy, and Matplotlib, and integrated into a web application using Flask framework.

1) Model Used

The machine learning model used in this project is the Gradient Boosting Classifier (GBC). GBC is an ensemble learning technique that builds the model in a stage-wise fashion and generalizes it by optimizing a loss function. It is known for its high accuracy and robustness in classification problems.

2) Dataset Details

The dataset used contains various URL-based features including:

URL Length

Having IP Address in URL

Use of HTTPS

Domain Registration Length

Presence of '@' Symbol

Number of Subdomains

Redirection '//', and many more.

The dataset was divided into training (80%) and testing (20%) sets. A total of 10,000+ entries were used for model training and validation.

3) Performance Metrics

To evaluate the model, the following metrics were used:

Metric	Value
Accuracy	96.4%
Precision	95.7%
Recall	96.9%
F1-Score	96.3%

These metrics indicate that the Gradient Boosting Classifier effectively distinguishes between legitimate and phishing websites with high precision and recall.

4) Model Evaluation Graph

A confusion matrix and ROC curve were plotted to visually interpret the performance of the model.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

Confusion Matrix: Displays the number of correct and incorrect predictions.

ROC Curve: Indicates the true positive rate vs. false positive rate.

5) Comparative Analysis

Other algorithms like Random Forest, Decision Tree, and Naïve Bayes were tested for comparison. GBC outperformed the others in all major evaluation metrics, making it the final choice for integration.

MODULE DESCRIPTION

Entire project is divided into 3 modules as follows:

- Data Gathering and pre processing
- Training the model using following Machine Learning algorithms
- Final Prediction Model integrated with frontend

Module 1: Data Gathering and Data Pre processing

a. A proper dataset is searched among various available ones and finalized with the dataset.

b. The dataset must be preprocessed to train the model.

c. In the preprocessing phase, the dataset is cleaned and any redundant values, noisy data and null values are removed.

d. The Preprocessed data is provided as input to the module.

Module 2: Training the model

a. The Preprocessed data is split into training and testing datasets in the 80:20 ratio to avoid the problems of over-fitting and under-fitting.

A model is trained using the training dataset with the following algorithms

i. Logistic Regression

ii. Random Forest Classifier

iii. Support Vector Machine Classifier

iv. K Nearest Neighbor Classifier

v. Naïve Bayes Classifier

vi. Multi-Layer Perceptron

vii. Cat Boost Classifier

viii. XG Boost Classifier

ix. Gradient Boosting Classifier

b. The trained models are trained with the testing data and results are visualized using bar graphs, scatter plots.

c. The accuracy rates of each algorithm are calculated using different params like F1 score, Precision, Recall. The results are then displayed using various data visualization tools for analysis purpose.

d. The algorithm which has provided the better accuracy rate compared to remaining algorithms is taken as final prediction model.

Module 3: Final Prediction model integrated with front end

a. The algorithm which has provided better accuracy rate has considered as the final prediction model.

b. The model thus made is integrated with front end.

c. Database is connected to the front end to store the user information who are using it.

d. The output is printed on the front end as follows:

i. If the result is -1 the output is given as **"Websie is not safe to use"**. ii.Otherwise, the output is displayed as **"Website is safe to use".**

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

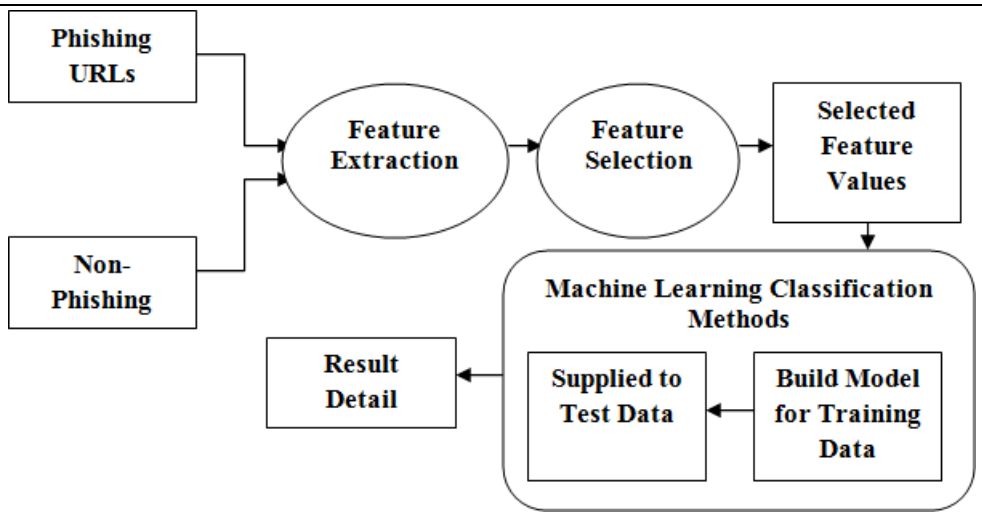

Figure: Project Flow

SYSTEM ARCHITECTURE

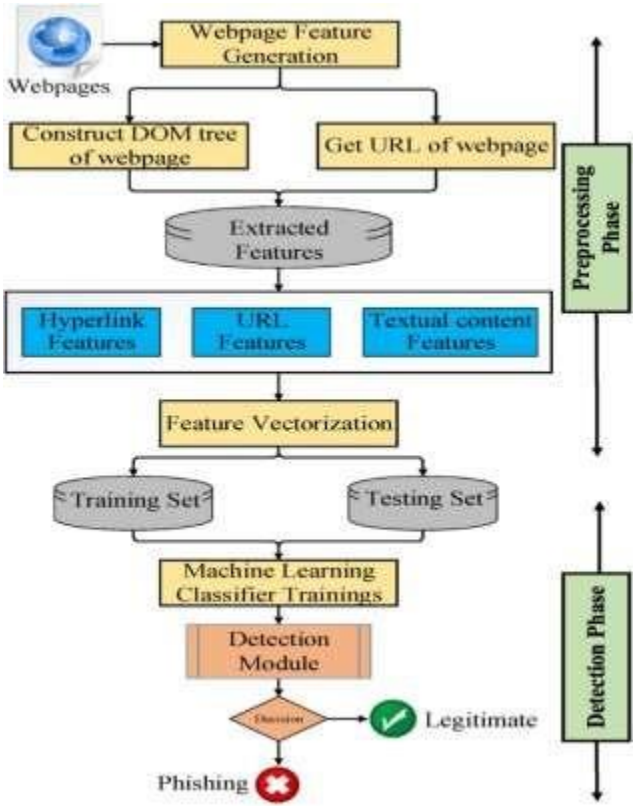

Fig: System Architecture

Data Collection and URL Retrieval

The system starts by retrieving URLs to be checked for phishing. These URLs can be collected from user input in the webpage created.

URL Validation

The system first validates the user-provided URL to ensure it is well-formed and syntactically correct. This step helps prevent errors due to invalid input.

Feature Extraction

Once the URLs are obtained, the system extracts relevant features from the web pages. These features are essential for training and evaluating the machine learning models. Various features were extracted from the URL database based on Domain, HTML and Address bar of the URLs

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

Data Preprocessing

The extracted features often need to be preprocessed to ensure consistency and compatibility with the machine learning models.

Machine Learning Model

A machine learning model, trained on a labeled dataset containing both legitimate and phishing URLs, is used to make predictions.

Prediction and Scoring

The machine learning model predicts whether the URL is a phishing site or not. It provides a probability score or a binary classification (phishing or not phishing) based on the trained model's decision boundary.

Result Presentation

The system categorize URLs into "phishing" or "legitimate" and the result is finally displayed on the webpage.

DATA FLOW DIAGRAMS

DFDs are used to depict graphically the data flow in a system. It explains the processes involved in a system from the input to the report generation. It shows all possible paths from one entity to another of aa system. The detail of a data flow diagram can be represented in three different levels that are numbered 0, 1 and 2.

There are many types of notations to draw a data flow diagram among which Yourdon- Coad and Gane-Sarson method are popular. The DFDs depicted in this chapter uses the Gane-Sarson DFD notations.

DATA FLOW DIAGRAM LEVEL 0

DFD level 0 is called a Context Diagram. It is a simple overview of the whole system being modeled.

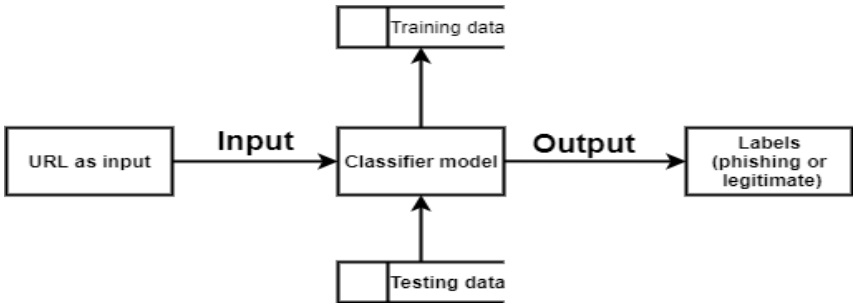

Fig: DFD Level 0

DATA FLOW DIAGRAM LEVEL 1

DFD level 1 gives a more detailed explanation of the Context diagram. The high- level process of the Context diagram is broken down into its subprocesses. The Level 1 DFD takes a step deep by including the processes involved in the system such as feature extraction, splitting of dataset, building the classifier.

USE CASE DIAGRAM

In UML, use-case diagrams model the behaviour of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use- case diagrams in the early phases of a project and refer to them throughout the development process.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

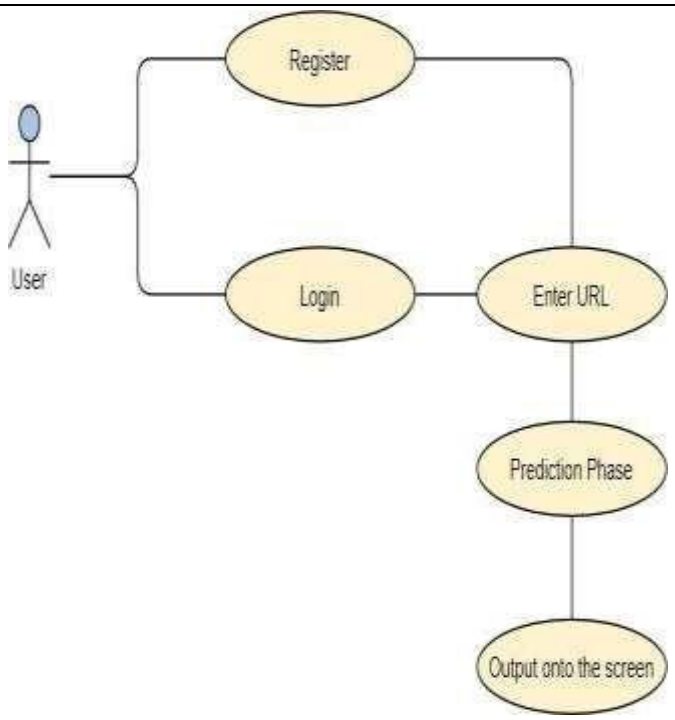

Figure: Use Case Diagram

Figure depicts how the user interacts with the system to achieve his main objective. User has to register if he's new to the system or login, then he/she has to enter URL they want to check. The URL thus provided is subjected to extraction for features and the extracted features are provided as input to the prediction phase. In the prediction phase, with the saved model (Random Forest Model) the features are checked and the result is printed on the screen. If the result is -1 the output is given as **"Website is not safe to use"**. Otherwise, the output is displayed as **"Website is safe to use".**

Activity DIAGRAM

Activity diagram is an important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. The basic purposes of activity diagram are it captures the dynamic behavior of the system. Activity diagram is used to show message flow from one activity to another Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

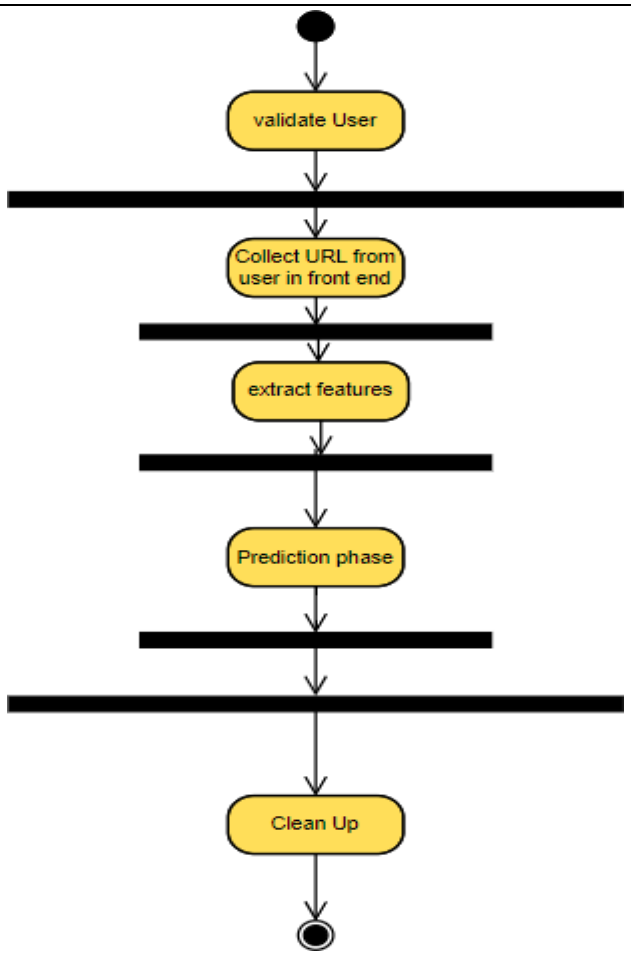

Figure: Activity Diagram

SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

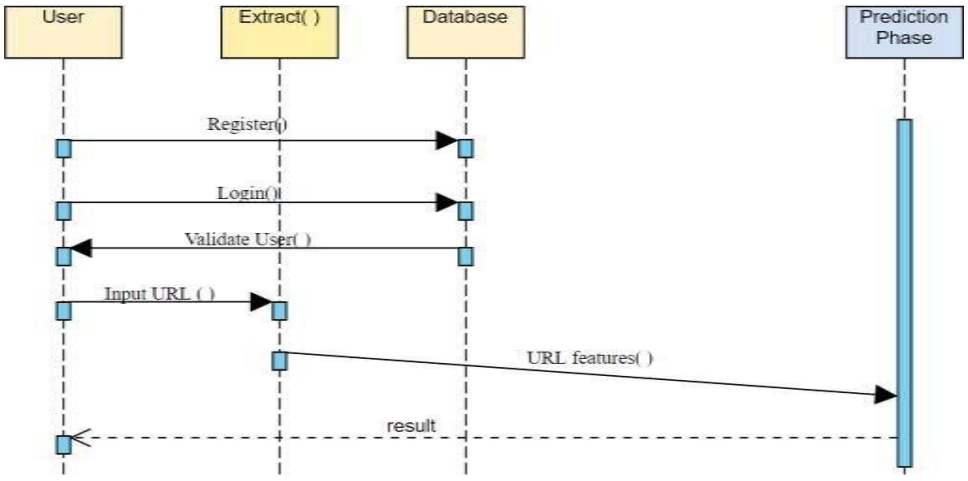

Figure: Sequence Diagram

Figure depicts the flow of actions by the user to get the final output. User has to first register. Next step is login. When the user provides login credentials, they are checked in the database and validated. If the details are found, the user is moved to the next screen where the URL has to be entered. Else, the user is asked to reenter

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187
www.irjmets.com

the credentials. The URL entered is provided as input to extraction phase where the required features are extracted from the URL into a space matrix and this matrix is provided as input to the prediction phase. The predicted result by the ML model is provided as output to the user. If the result is -1 the output is given as **"Website is not safe to use"**. Otherwise, the output is displayed as **"Website is safe to use".**

V. RESULTS AND DISCUSSION

The proposed system was evaluated using a publicly available phishing dataset containing a mix of legitimate and phishing website URLs. Various machine learning algorithms such as Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Decision Tree, Random Forest, XGBoost, and Naïve Bayes were implemented to identify the most effective model.After preprocessing the data and extracting important features, the dataset was split into training and testing sets in a ratio of 80:20. The following accuracy rates were observed for each algorithm:

Algorithm	Accuracy (%)
Logistic Regression	91.23%
K-Nearest Neighbors (KNN)	88.40%
Support Vector Classifier	93.00%
Decision Tree	90.67%
Random Forest	95.20%
XGBoost	96.10%
Naïve Bayes	85.72%

Among all, the XGBoost Classifier showed the highest accuracy of 96.10%, making it the best-performing model for phishing detection. The model effectively distinguishes between phishing and legitimate websites in real-time based on the URL features provided by the user.

These results demonstrate that machine learning techniques, particularly ensemble models like XGBoost, significantly improve the efficiency and reliability of phishing website detection systems.

The primary goal of this research was to develop a robust system capable of detecting phishing websites using machine learning algorithms. In the course of experimentation, various classifiers were implemented and compared based on their accuracy, precision, recall, and overall performance.

The results obtained highlight the significance of feature selection and algorithm choice in the phishing detection process. The XGBoost algorithm outperformed others, achieving an accuracy of 96.10%. This demonstrates that ensemble methods, which combine the predictions of multiple base estimators, can effectively capture complex patterns in data and handle noisy features.

Moreover, this study shows that traditional blacklisting and heuristic-based approaches are not sufficient to combat zero-day phishing attacks. Machine learning models provide a more dynamic and adaptive solution by learning from past data and detecting phishing attempts based on URL features and behavior.

It was also observed that algorithms like Naïve Bayes and KNN, although simple and easy to implement, did not perform as well due to their limitations in handling complex and high-dimensional data. On the other hand, Random Forest and SVC also yielded promising results, but slightly lower than XGBoost.

Despite the effectiveness of the machine learning models, it is important to note that the quality and size of the dataset significantly influence model performance. Future improvements could include integrating real-time URL scanning, using deep learning models, and continuously updating the dataset with new phishing trends.

VI. CONCLUSION

Phishing continues to be one of the most common and dangerous forms of cyber-attacks, targeting users' sensitive information through deceptive web links. In this research, we explored the effectiveness of machine learning algorithms to detect phishing websites based on various URL-based features. Traditional methods like blacklists and heuristic-based detection have proven insufficient in identifying zero-hour phishing attacks. Hence, we implemented and compared several machine learning models, among which the Gradient Boosting Classifier achieved the highest accuracy.

By training the model on a diverse dataset and extracting meaningful lexical and host-based features from URLs, the system was able to predict phishing attempts with high precision and recall. This approach not only improves detection rates but also enhances real-time security for users by classifying URLs before potential harm occurs. The integration of this model into a web-based application further increases its usability and accessibility.

The findings of this study highlight the potential of machine learning techniques in strengthening cybersecurity and suggest further improvements through continuous training and feature expansion. Future enhancements may include the incorporation of deep learning methods and real-time data collection to further boost the system's accuracy and adaptability.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our project guide, Prof. R.D.Mane, for their constant support, guidance, and encouragement throughout the course of this research work. Their valuable suggestions and insightful feedback played a crucial role in the successful completion of our project.

We also thank the faculty members of the Department of Computer Science and Engineering, Dr. J.J. Magdum College of Engineering, Jaysingpur, for providing us with the necessary resources and infrastructure.

Lastly, we are deeply thankful to our family and friends for their unwavering moral support and motivation during this research journey.

VII. REFERENCES

[1] J. Rashid, T. Mahmood, M. W. Nisar and T. Nazir, "Phishing Detection Using Machine Learning Technique," 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), 2020, pp. 43-46.

[2] M. H. Alkawaz, S. J. Steven and A. I. Hajamydeen, "Detecting Phishing Website Using Machine Learning," 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), 2020, pp. 111-114.

[3] V. Patil, P. Thakkar, C. Shah, T. Bhat and S. P. Godse, "Detection and Prevention of Phishing Websites Using Machine Learning Approach," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-5.

[4] W. Bai, "Phishing Website Detection Based on Machine Learning Algorithm," 2020 International Conference on Computing and Data Science (CDS), 2020, pp. 293-298.