Project Report on

"URLGuard: A Holistic Hybrid Machine Learning Approach for Phishing Detection"

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Engineering in

Artificial Intelligence and Data Science

by

Mr.Siddheshwar Patil (B190352050)

Mr.Abhishek Kubde (B190352037)

Mr.Rohit Mehetre (B190352044)

Mr.Shubham Gaikwad(B190352017)

Under the guidance of

Dr.P.M.Paithane

Department of Artificial Intelligence and Data Science

Vidya Pratishthan's

Kamalnayan Bajaj Institute of Engineering and Technology

Baramati-413133, Dist-Pune (M.S.) India

April 2023-24

VPKBIET, Baramati

Certificate

This is to certify that the Project Stage II Report on

URLGuard: A Holistic Hybrid Machine Learning Approach for Phishing Detection

SUBMITTED BY

Siddheshwar Patil (B190352050) **Abhishek Kubde (B190352037)**

Rohit Mehetre (B190352044) **Shubham Gaikwad (B190352017)**

in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in Artificial Intelligence and Data Science at Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati, under the Savitribai Phule Pune University, Pune. This work is done during year 2023-24 Semester-II, under our guidance.

Dr.P.M.Paithane	Dr. P. M. Paithane	Dr. R. S. Bichkar
Project Guide	Head of Dept.	Principal

Examiner 1: - - - - - - -

Examiner 2: - - - - - - -

Acknowledgements

It gives immense pleasure in submitting this project report on **”URLGuard: A Holistic Hybrid Machine Learning Approach for Phishing Detection”** The Successful project reflects our project work and effort of our guide in giving us good information.

We would like to express our heartiest thanks to our project guide **Dr.P.M.Paithane** for his undying support which makes us possible to make this project knowledgeable. She not only provided us the literature and guidance to study but also the platform which required for us to prepare best for this project. We also equally indebted to our project co-ordinator **Mrs.Rohini Naik** who has been a constant source of inspiration and guiding star in achieving my goal. We thank **Dr.P.M.Paithane**, Head of Department of AI&DS Engineering and respected Principal **Dr.R.S.Bichkar** for supporting and providing all facilities to complete the work. We are thankful to and fortunate enough to get constant encouragement, support and guidance from entire teaching staff of Department of AI&DS engineering which helped us in successfully completing our project work.

Siddheshwar Patil (B190352050)
Abhishek Kubde (B190352037)
Rohit Mehetre (B190352044)
Shubham Gaikwad (B190352017)

Abstract

The fast growth of Internet technology has significantly changed online users experiences, while security concerns are becoming increasingly overpowering. Among these concerns, phishing stands out as a prominent criminal activity that uses social engineering and technology to steal a victim's identification data and account information. According to the Anti-Phishing Working Group (APWG), the number of phishing detections increased by 46 in the first quarter of 2018 compared to the fourth quarter of 2017. So to overcome these situations below paper introduces a phishing detection system using a hybrid machine learning approach based on URL attributes. It addresses the growing threat of phishing attacks that exploit email manipulation and fake websites to deceive users and steal sensitive data. The study employs a phishing URL dataset with over 11,000 websites, extracted from a reputable repository. After preprocessing, a hybrid machine learning model, which includes Decision Tree, Random Forest, and XGB is employed to safeguard against phishing URLs. The proposed approach undergoes evaluation with key metrics such as precision(0.96), accuracy(0.96), recall(0.97), and F1-score(0.96). Results demonstrate that the proposed method surpasses the state of art.

Key Words: Anti-Phishing Working Group (APWG), Decision Tree, and Random Forest, and XGB, hybrid machine learning.

List of Figures

List of Tables

Notation and Abbreviations

- DT: Decision Tree
- RF: Random Forest
- XGB: Extreme Gradient Boosting

Contents

Chapter 1

Introduction

1.1 Introduction

In the ever-evolving landscape of cybersecurity, the prevalence of phishing attacks poses a significant threat to individuals and organizations alike. To combat the escalating threat of phishing[6] attacks on the internet effectively, the comprehensive URL-based phishing detection system utilizing a variety of machine learning algorithms.

Phishing detection using hybrid machine learning models outlines the growing threat of phishing attacks in the digital era and highlights the need for advanced, accurate detection methods. It introduces the concept of hybrid machine learning, emphasizing its potential to enhance detection accuracy by combining the strengths of multiple algorithms.

Phishing, often executed through deceptive URLs, exploits human vulnerability, aiming to extract sensitive information or deploy malicious payloads. This report delves[2] into the development and functionality of a phishing URL detection system, shedding light on its importance in mitigating cyber risks and fortifying the defenses against malicious actors.[6] As we explore the intricacies of such a system, we uncover the pivotal role it plays in ensuring a secure digital environment for users, ultimately contributing to the ongoing battle against cyber threats.

1.2 Motivation

Building a robust phishing URL detection system is crucial in safeguarding individuals and organizations from cyber threats.[14] By developing and implementing such a system, you contribute to the overall cybersecurity landscape, protecting users from potential financial losses, identity theft, and other malicious activities. Your efforts can make a meaningful impact in creating a safer digital environment for everyone. Keep pushing forward for a more secure online world!

1.3 Problem Statement

To combat the escalating threat of phishing attacks on the internet effectively, the comprehensive URL-based phishing detection system utilizing a variety of machine learning algorithms.

Chapter 2

Literature Survey

- **Phishing detection using hybrid machine learning model(RF+DT+XGB)**

  Prediction performance is improved with a hybrid model that combines Random Forest (RF), Decision Trees (DT), and XGBoost (XGB). XGBoost handles complicated interactions in big datasets and is well-known for its gradient boosting and regularization. An ensemble of decision trees called Random Forest provides resistance against noise and outliers, improving variance reduction and generalization. Interpretability and feature importance insights are offered by decision trees. Decision Trees help with feature analysis, Random Forest adds variation to predictions, and XGBoost finds complex patterns in this hybrid model.[16] The model balances the advantages of either approach voting or averaging to produce a dependable and accurate predictive tool that can be used for a variety of tasks, such as regression, classification, and anomaly detection.

- **Phishing detection using hybrid machine learning model(LR+SVC+DT)(2023)**:

  S. Raman Kumar Jog, ABDUL KARIM, MOBEEN SHAHROZ, KHABIB MUSTOFA, AND SAMIR BRAHIM BELHAOUARI [1]: Machine Learning Models: Logistic Regression (LR), Support Vector Classifier (SVC), and Decision Trees (DT) combined into a hybrid model. In summary,[7] the hybrid model (LSD) that has been suggested seeks to improve phishing detection efficiency and accuracy. It makes use of grid search, hyperparameter optimization, and canopy feature selection. On the other hand, false positive or negative rates are not discussed, and the study does not provide any information on scalability for huge datasets.

- Working
    - ◇ Feature Extraction and Preprocessing: Extract features from URLs (e.g., length, special characters, domain age) and preprocess data (normalize, handle missing values).
    - ◇ Training Individual Models:
        - · Logistic Regression: Trains on the dataset to estimate the probability of a URL being phishing.
        - · Support Vector Classifier: Uses a suitable kernel to classify data, effective in high-dimensional spaces.
        - · Decision Tree: Learns decision rules to classify URLs based on features.
    - ◇ Model Predictions: Each model predicts whether a URL is phishing. LR provides probabilities, SVC offers class margins, and DT gives class labels.
    - ◇ Ensemble Methods:
        - · Voting: Combines predictions using majority voting (hard voting) or weighted probabilities (soft voting) to decide the final class.
        - · Stacking: Uses predictions from LR, SVC, and DT as inputs for a meta-classifier (e.g., another LR) to make the final decision.
    - ◇ Evaluation: The hybrid model's performance is assessed using metrics like accuracy, precision, recall, and F1-score on a validation dataset.

- **Phishing Detection Using Machine Learning Techniques(2020)**

  Machine Learning Models [2]: Logistic Regression, Ada Booster, Random Forest[16] [17] [20], K-Nearest Neighbors (KNN) [13] [18], Neural Networks, Support Vector Machines (SVM) [14], Gradient Boosting, Naive Bayes [15], and XGBoost.

  The study focuses on phishing prevention techniques, including email filtering, user education, and[13] real-time machine learning detection. It also gives us a detailed explanation of the workings of different machine learning models. However, it does not explore potential vulnerabilities or weaknesses in the machine learning models, and dataset details are omitted, impacting generalizability.

- **Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism(2019)**

  YONG FANG, CHENG ZHANG, CHENG HUANG, LIANG LIU, AND YUE YANG[4] Machine Learning Models: Deep learning model named THEMIS (advanced version of RCNN). [8]THEMIS enhances embedding for improved performance detection. This approach allows for modeling emails[4] at multiple levels, including the email header, email body, character level, and word level simultaneously, enhancing the model's ability to capture relevant features. However, the study notes that some phishing emails without an email header may reduce efficiency and accuracy. It is more accurate than previous methods, and it is also more efficient and robust.

  - Data Collection and Preprocessing: Collect a dataset of phishing and legitimate emails.Preprocess the emails by cleaning the text (removing HTML tags, special characters, stop words)[1] and tokenizing the text into words or subwords
  - Feature Representation with Multilevel Vectors:
    - ◇ Word Embeddings: Convert words into dense vector representations using pre-trained embeddings like Word2Vec, GloVe, or BERT to capture semantic meaning.
    - ◇ Contextual Vectors: Utilize embeddings that capture context within the text, such as BERT or GPT, to represent words in relation to surrounding words.
  - RCNN Architecture:
    - ◇ Embedding Layer: Initialize with pre-trained word embeddings or contextual vectors.
    - ◇ Convolutional Layers: Apply convolutional filters to capture local features and patterns in the text.
    - ◇ Recurrent Layers: Use RNN variants (like LSTM or GRU) to capture long-term dependencies and sequential information in the text.
  - Attention Mechanism:

    - ◇ Attention Layer: Implement an attention mechanism to focus on important words or phrases in the email, enhancing the model's ability to identify critical parts of the text related to phishing.
    - ◇ This mechanism assigns weights to different parts of the input, allowing the model to prioritize more relevant information.
  - – Classification Layer:
    - ◇ Fully Connected Layer: Pass the combined features from the RCNN and attention mechanism through fully connected layers for classification.
    - ◇ Output Layer: Use a softmax activation function to output probabilities for phishing and legitimate classes.
  - – Model Training:
    - ◇ Train the model on a labeled dataset using a suitable loss function (e.g., cross-entropy loss) and an optimizer (e.g., Adam).
    - ◇ Validate the model on a separate validation set to tune hyperparameters and prevent overfitting.
  - – Evaluation:
    - ◇ Evaluate the model's performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC score on a test dataset. Perform cross-validation to ensure the model's robustness and generalizability.

- **Hybrid ensemble feature selection framework for enhancing machine learning-based phishing detection systems(2019)**

  Kang Leng Chiew, Choon Lin Tan, Kok Sheik Wong, Kelvin S.C. Yong, Wei King Tiong[5] Machine Learning Models: Hybrid Ensemble Feature Selection (HEFS), Random Forest Classifier.

  The study uses a real browser for[12]feature extraction, improving robustness. However, detailed accuracy results for baseline features with classifiers other than Random Forest are not provided. The phishing dataset used for benchmarking HEFS is not specified.

- **Phishing detection using random forest with NLP-features(2017)** Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, Banu Diri [3] Machine Learning Models: Random Forest with NLP-Based Features.

  The study presents a high-accuracy real-time anti-phishing system that uses a variety of machine learning (ML) algorithms and features for URL-based phishing detection. Nevertheless, [11]issues pertaining to NLP-based features are not examined, and the performances of the seven classification algorithms are not thoroughly analyzed.

  – Working

    ◇ Data Collection:Collect a dataset containing phishing and legitimate emails or websites.

    ◇ Feature Extraction Using NLP:

      · Text Processing: Clean and preprocess the text by removing stop words, punctuation, and performing tokenization.

      · Feature Engineering:[11]Feature engineering is the process of creating new features or modifying existing ones from raw data to improve the performance of machine learning models.

    ◇ Feature Preprocessing:Normalize and standardize the extracted features. Handle missing values if any.

    ◇ Model Training with Random Forest:

      · Random Forest Classifier: Train the Random Forest model using the extracted NLP features. Random Forest is an ensemble method that builds multiple decision trees and merges them to get a more accurate and stable prediction.

      · Parameter Tuning: Optimize hyperparameters like the number of trees, max depth, and feature subset size using techniques like grid search or random search.

    ◇ Model Evaluation:Evaluate the model's performance using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC score on a validation dataset.Perform cross-validation to ensure the model's robustness and generalizability.

Table 2.1: Literature Survey

Sr.no	Author and Year	Techniques/ Methodology	Advantages	Gaps
1	ABDUL KARIM,M.SHAHROZ, KHABIB (2023)[7]	Machine Learning Models, Hybrid models (LR+SVC+DT)	It multiple machine learning algorithm which uses canopy feature selection and Grid Search Hyper parameter Optimization for enhancing accuracy and efficiency.	No discussion on false positive/negative rates and No insights on scalability for large datasets.
2	V.Shahrivari, M.MahdiDarab (2020)[13]	Logistic regression, Ada booster, randomforest,KNN, neural networks, SVM, Gradient boosting, and XGBoost	Phishing techniques include email filtering, user education, and real-time machine learning detection for prevention.	Dataset details omitted.
3	O.Koray,E.Buber, BanuDiri(2017)[3]	Using Random Forest with NLP-based features	A real-time anti-phishing system achieving 97.98 and is used Real-time.	Paper lacks detailed analysis of the seven classification algorithm.

4	Y.FANG,C.ZHANG (2019)[5]	Deep learning model named THEMIS(advance dversion of RCNN)	Enhanced Embedding in THEMIS.	Some phishing emails may not have an email header due to which efficiency and accuracy decreases.
5	K.Chiew,C.Tan, K.Tionga(2019)[4]	Hybrid Ensemble Feature Selection(HEFS),Random Forest Classifier	Using a real browser for feature extraction. .	Fails to specify the phishing dataset used to benchmark HEFS.

Chapter 3

Methodology

3.1 Problem Definition

The problem statement outlines the challenge of effectively addressing the increasing threat posed by phishing attacks on the internet. To tackle this issue, the proposed solution involves creating a robust phishing detection system specifically focused on URLs. This system is designed to leverage a variety of machine learning algorithms, making it comprehensive and adaptable.

In simpler terms,[15] the goal is to develop a sophisticated defense mechanism that can accurately identify and block phishing attempts, particularly those involving deceptive URLs. The use of diverse machine learning algorithms enhances the system's ability to stay ahead of evolving phishing tactics. The overarching aim is to create a resilient solution that safeguards users and organizations from the growing sophistication of phishing attacks on the internet.

3.2 Project Objectives

- To create ongoing monitoring and reporting to assess system effectiveness and spot phishing URLs.
- To introduce a hybrid model (DT+RF+XGB) for enhanced phishing detection.
- Discuss and compare evaluation parameters to demonstrate the superiority of the

proposed approach.

3.3 Scope of Project

- User Interface Design: Create an intuitive and user-friendly interface for system administrators, facilitating easy management, monitoring, and response to potential phishing threats.
- Security Collaboration: Foster collaboration with the wider cybersecurity community to stay informed about the latest threats and best practices, ensuring the system's relevance and effectiveness.
- Continuous Improvement: Establish procedures for continuous monitoring, evaluation, and updating of the system to remain effective against evolving phishing tactics over time.

3.4 Project Constraints

3.4.1 Data Availability

The quality and quantity of phishing website datasets may pose limitations on the system's accuracy and performance.The dataset was gathered and saved as a CSV file from the well-known Kaggle dataset repository.

3.4.2 Computational Resources

The effective training of hybrid machine learning requires substantial computational resources,including powerful GPUs or TPUs. Constraints on hardware resources may impact the scalability and speed of model development and experimentation.

3.4.3 Time Constraints

With respect to project timelines and deadlines is critical, as the project may be subject to specific submission or completion dates.

3.4.4 Domain Knowledge

The successful implementation of hybrid machine learning model demands expertise in the field of artificial intelligence and machine learning. Constraints on the team's expertise may influence the project's outcomes and the chosen approaches.

3.4.5 Regulatory and Ethical constraints

Compliance with data privacy, copyright, and other legal and ethical considerations can impact data usage and system deployment.

3.5 Proposed Architecture

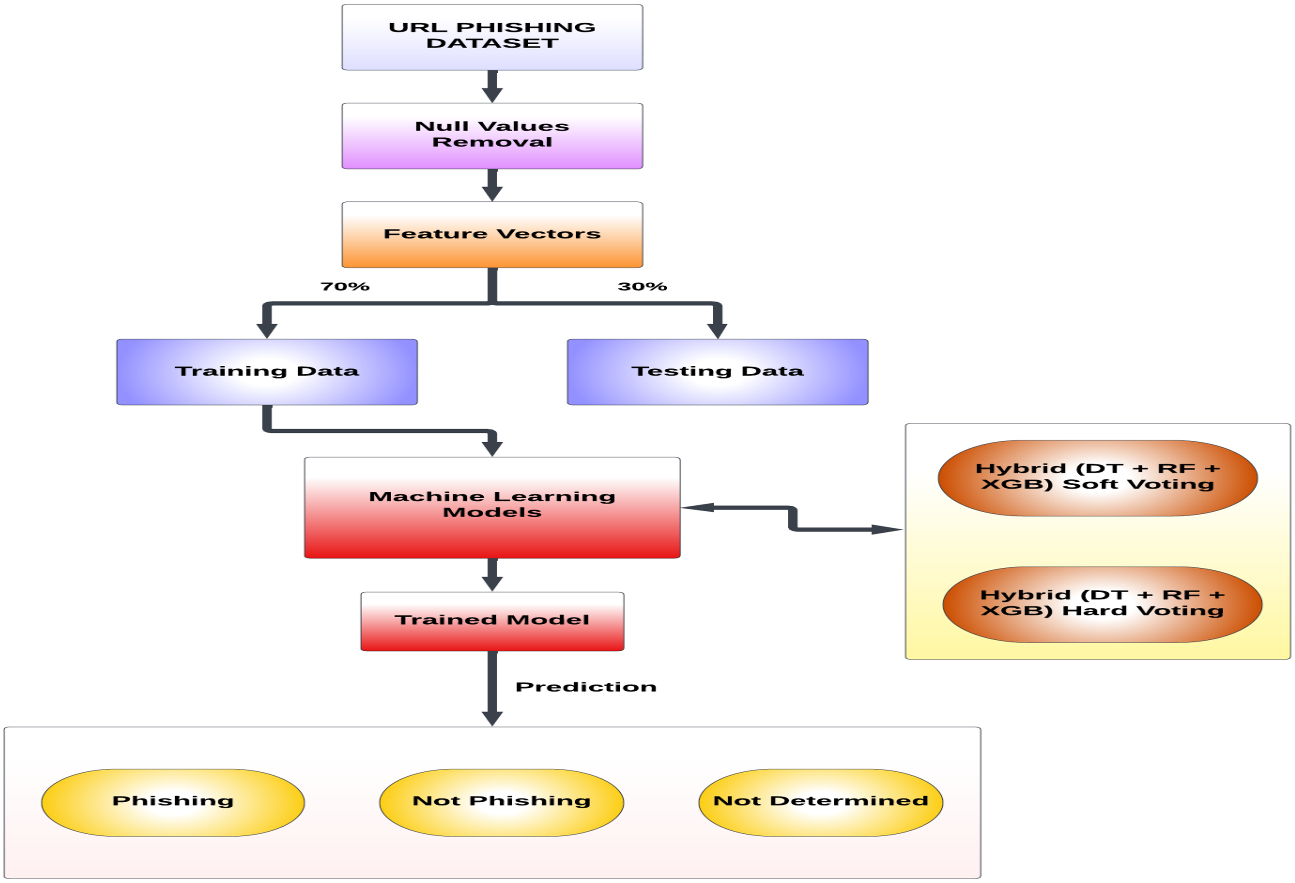


Figure 3.1: System Architecture

The machine learning model development process for identifying phishing URLs begins with the "URL Phishing" dataset, potentially containing flagged phishing or authentic URLs. Null values are removed, and the data is transformed into feature vectors, likely encompassing URL characteristics. Subsequently, training and testing datasets are split (70 and 30, respectively) for model training and evaluation. Various machine learning models, including XGBoost, Random Forest, and Decision Tree ensembles, are trained using the training data. Two voting methods, "Soft Voting" and "Hard Voting," integrate predictions from these models. Model performance on the testing data is not shown. Finally, a new URL undergoes classification as "Phishing," "Not Phishing," or "Not Determined" by the trained model.

3.6 Dataset

The dataset was gathered and saved as a CSV file from the [7] well-known Kaggle dataset repository, which offers benchmark datasets for academic use. The collection included 33 attributes and 11054 items that were taken from over 11,000 websites. Some characteristics that help distinguish between phishing and legitimate website URLs are UsingIP, LongURL, ShortURL, Symbol@, Redirecting//, PrefixSuffix-, SubDomains, HTTPS, NonStdPort, DomainRegLen, HTTPSDomainURL, Favicon, RequestURL, AnchorURL, LinksIn-ScriptTags, and ServerFormHandler.
Dataset link: `https://www.kaggle.com/code/eswarchandt/websitephishing/input?select=phishing.csv`

3.7 Proposed Method With Mathematical Model

.

- **Random Forest:** Random Forest is an ensemble learning method widely used for classification and regression tasks.[10] It comprises decision trees trained on random subsets of the data, with randomness introduced both in bootstrapped sampling and feature [2] selection. During prediction, individual tree outputs are either averaged (for regression) or combined through majority voting (for classification). This randomness helps prevent overfitting and improves robustness. Random Forest handles high-dimensional data well and provides feature importance scores[7].

- Ensemble Method

$$F(x_t) = \frac{1}{B} \sum_{i=0}^{B} F_i(x_t) \tag{3.1}$$

  - ◇ $F(x_t)$ is the output of the random forest for the input x_t,
  - ◇ B is the number of trees in the random forest,
  - ◇ $F_i(x_t)$ is the output of the i-th tree in the forest for the input x_t,
  - ◇ $\sum$ denotes the summation over the specified range.

.

- **XGB:**Extreme Gradient Boosting, or XGBoost for short, is a sophisticated version of gradient boosting machines.[9] It is well known for its effectiveness in supervised learning tasks as well as its scalability and efficiency. XGBoost sequentially constructs an ensemble of weak prediction models, usually decision trees. By optimizing a predetermined objective function typically a mix of the loss function and regularization term—each new model fixes the flaws of its predecessors.

  - **Objective Function**

$$\text{Objective} = \sum_{i=1}^{n} \text{Loss}(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.2}$$

  - ◇ n is the number of samples.
  - ◇ $\hat{y}_i$ is the predicted value for the i-th sample.
  - ◇ y_i is the true label for the i-th sample.
  - ◇ K is a parameter representing some set of values.

- **Gradient Boosting**

$$\hat{y}(t) = \sum_{k=1}^{t} f_k(x) \tag{3.3}$$

◇ t is the iteration or boosting round.

◇ $f_k(x)$ is the prediction of the k-th model at input x.

◇ **Regularization**

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 + \alpha \sum_{j=1}^{T} |w_j| \tag{3.4}$$

◇ T is the number of leaves in the tree.

◇ w_j is the weight assigned to the j-th leaf.

◇ γ, λ, and α are regularization parameters.

- **Decision Tree:**A decision tree is a tree-like model used in data analysis and machine learning. It recursively splits data into subsets based on the most significant attribute, creating a predictive model resembling a flowchart. It's easy to interpret, handles both categorical and numerical data, and is effective for classification and regression tasks.

  - **Entropy (E(S)):**

$$E(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{3.5}$$

  - **Conditional Entropy (E(T, X)):**

$$E(T, X) = -\sum_{c \in X} p(c) \cdot E(c) \tag{3.6}$$

  - **Information Gain (IG(T, X)):**

$$IG(T, X) = E(T) - E(T, X) \tag{3.7}$$

◇ S is a set,

◇ c represents the classes or values in the set,

- ◇ p_i is the probability of occurrence of element i in set S,
- ◇ T is a set, and X is an attribute or feature,
- ◇ $p(c)$ is the probability of occurrence of value c in set X,
- ◇ $E(c)$ represents the entropy of the subset of T associated with value c, and
- ◇ $\sum$ denotes the summation over the specified range.

3.8 Proposed Algorithm

Algorithm 1: Hybrid Machine Learning Model (RF + DT + XGB)

```
Input: Training data $X_{train}$, $y_{train}$, Test data $X_{test}$, $y_{test}$
Output: Predicted class label
train_and_test_hybrid_model($X_{train}$, $y_{train}$, $X_{test}$, $y_{test}$, num_of_epochs,
 param_grid):
    initialize rf_model ← RandomForestClassifier()
    initialize xgb_model ← XGBClassifier()
    initialize dt_model ← DecisionTreeClassifier()
    initialize estimators ← [('rf', rf_model), ('xgb', xgb_model),
 ('dt', dt_model)]
    initialize best_model ← None
    for epoch ← 1 to num_of_epochs do
        initialize grid_search ←
 GridSearchCV(estimator=VotingClassifier(estimators),
 param_grid=param_grid, cv=5)
        grid_search.fit($X_{train}$, $y_{train}$)
        best_model ← grid_search.best_estimator_
        best_model.fit($X_{train}$, $y_{train}$)
        // Evaluate on the test set and print results
        validation_results ← best_model.evaluate($X_{test}$, $y_{test}$)
        print(Accuracy: validation_results)
    end
return best_model.predict($X_{test}$)
```

- This algorithm implements a hybrid machine learning model combining Random Forest (RF), Decision Tree (DT), and XGBoost (XGB) classifiers. It iteratively trains the model for a specified number of epochs, optimizing hyperparameters through grid search cross-validation. The algorithm selects the best-performing model based on validation results and evaluates it on the test set. Finally, it returns the predicted class labels for the test data using the selected best model.

Chapter 4

Project Requirement

4.1 Hardware Requirements

- Disk Space: Min 120 GB.
- Processor: 10th Gen Intel Core i5.
- RAM: Min 8 GB.

4.2 Software Requirements

- Operating System: Linux/Windows.
- OS Type: 64-bit.
- Python Version: 3.11.4.
- Tools: Google Colaboratory/ Jupyter Notebook,Streamlit.
- Library:Pandas,Numpy, Matplotlib, sklearn,xgboost.

4.3 Performance Requirement

The model should achieve a high level of accuracy in detecting phishing website. Define a specific target accuracy rate that the model should meet or exceed.Ensure that the training process is efficient and doesn't take an excessively long time, especially if this model is intended for realtime.

4.4 Software Requirements

Software quality attributes, or requirements, define the desired characteristics of a system. They include performance, reliability, usability, and security. Performance requirements ensure efficient system response times. Reliability demands consistent and error-free functionality. Usability emphasizes user-friendly interfaces. Security requirements protect against unauthorized access and data breaches. These attributes collectively shape a system's functionality, user experience, and resilience, which are crucial for delivering high-quality software solutions. we should also ensure that the dataset quality of be maintained.

4.5 Security Requirements

Security is paramount in our project for cardiovascular disease detection from retinal fundus images. To safeguard sensitive medical data, we've adopted a robust security framework that includes data encryption for storage and transmission, access control with role-based permissions, and strong authentication methods, such as multi-factor authentication. Data privacy and compliance with healthcare regulations are diligently upheld, with practices like data anonymization and audit trails to ensure patient confidentiality and traceability. We have implemented security measures like intrusion detection, secure APIs, and regular security assessments to protect against unauthorized access and cyber threats. Our commitment to data security extends to physical access control, secure coding practices, and incident response planning. These measures collectively ensure the highest level of security for our healthcare application.

4.6 Other Requirements

Define a clear project timeline with milestones and deadlines to track progress and ensure the project stays on schedule.Maintain data quality and integrity through data cleansing, validation, and error-checking procedures to prevent inaccurate or incomplete patient records.

Chapter 5

Project Planning

This plan is the basis for the execution for the tracking of all the project activities. It shall be used throughout the life of the project and shall be kept up to date to reflect the actual accomplishments and plans of the project.

5.1 Project Estimates

1.Calculation of Total Effort, Time, and Cost:
– Effort (Data Preprocessing) = 6 person-months
– Effort (Model Development) = 4 person-months
– Effort (Model Deployment) = 5 person-months
– Total Effort (COCOMO) = 6 + 5 + 5 = 16 person-months
– Total Time (COCOMO) = Total Effort (COCOMO) / Team Size
– Total Time (COCOMO) = 16 / 4 = 4 months
– Total Cost (COCOMO) = Total Effort (person-months) * Monthly Labor Cost + Functioning Cost(i.e Cloud services cost)
– Total Cost (COCOMO) = 4 * Rs.8000 + Rs. 10,000 = Rs. 42000

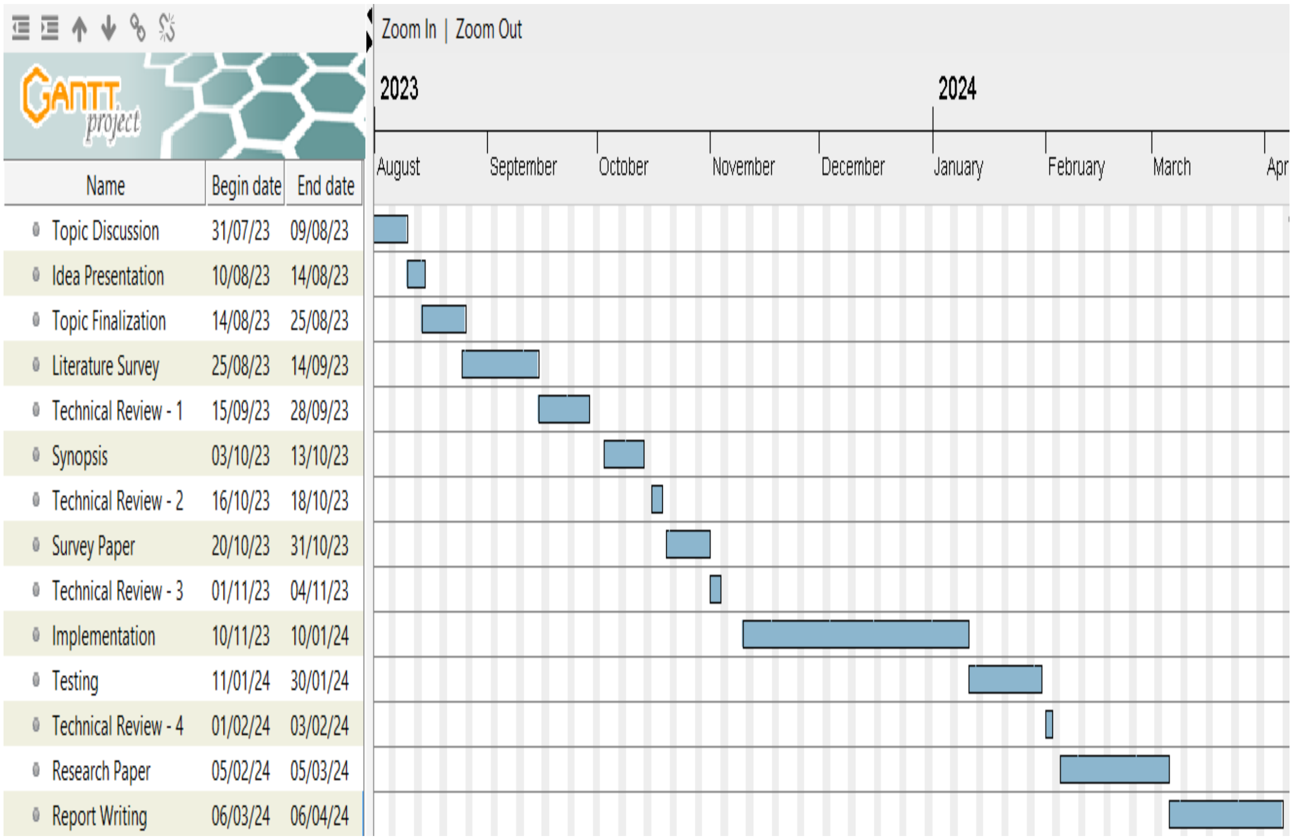
5.2 Team Structure

Name	**Role**	**Responsibilities**
Mr.Siddheshwar Patil	Database Specialist, Documentation	Database Design, Diary Maintenance
Mr.Rohit Mehetre	Project Manager, Developer	Database Design and Implementation, Coding
Mr.Abhishek Kubde	Developer,Database Specialist	Coding , Coordination,
Mr.Shubham Gaikwad	Developer, Documentation	Project planning, Report writing

Chapter 6

Project Schedule

6.1 Project Breakdown Structure



Name	Begin date	End date
Topic Discussion	31/07/23	09/08/23
Idea Presentation	10/08/23	14/08/23
Topic Finalization	14/08/23	25/08/23
Literature Survey	25/08/23	14/09/23
Technical Review - 1	15/09/23	28/09/23
Synopsis	03/10/23	13/10/23
Technical Review - 2	16/10/23	18/10/23
Survey Paper	20/10/23	31/10/23
Technical Review - 3	01/11/23	04/11/23
Implementation	10/11/23	10/01/24
Testing	11/01/24	30/01/24
Technical Review - 4	01/02/24	03/02/24
Research Paper	05/02/24	05/03/24
Report Writing	06/03/24	06/04/24

Figure 6.1: Timeline chart

Chapter 7

Project Design

7.1 UML Diagrams

7.1.1 Activity Diagram

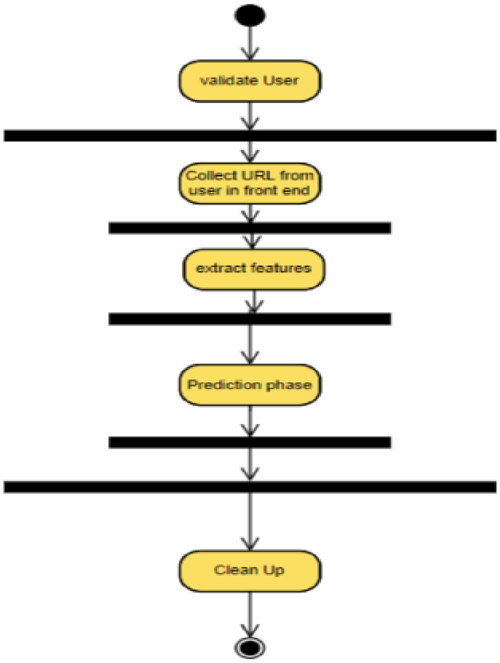


Figure 7.1: Activity Diagram

7.1.2 Use Case Diagram

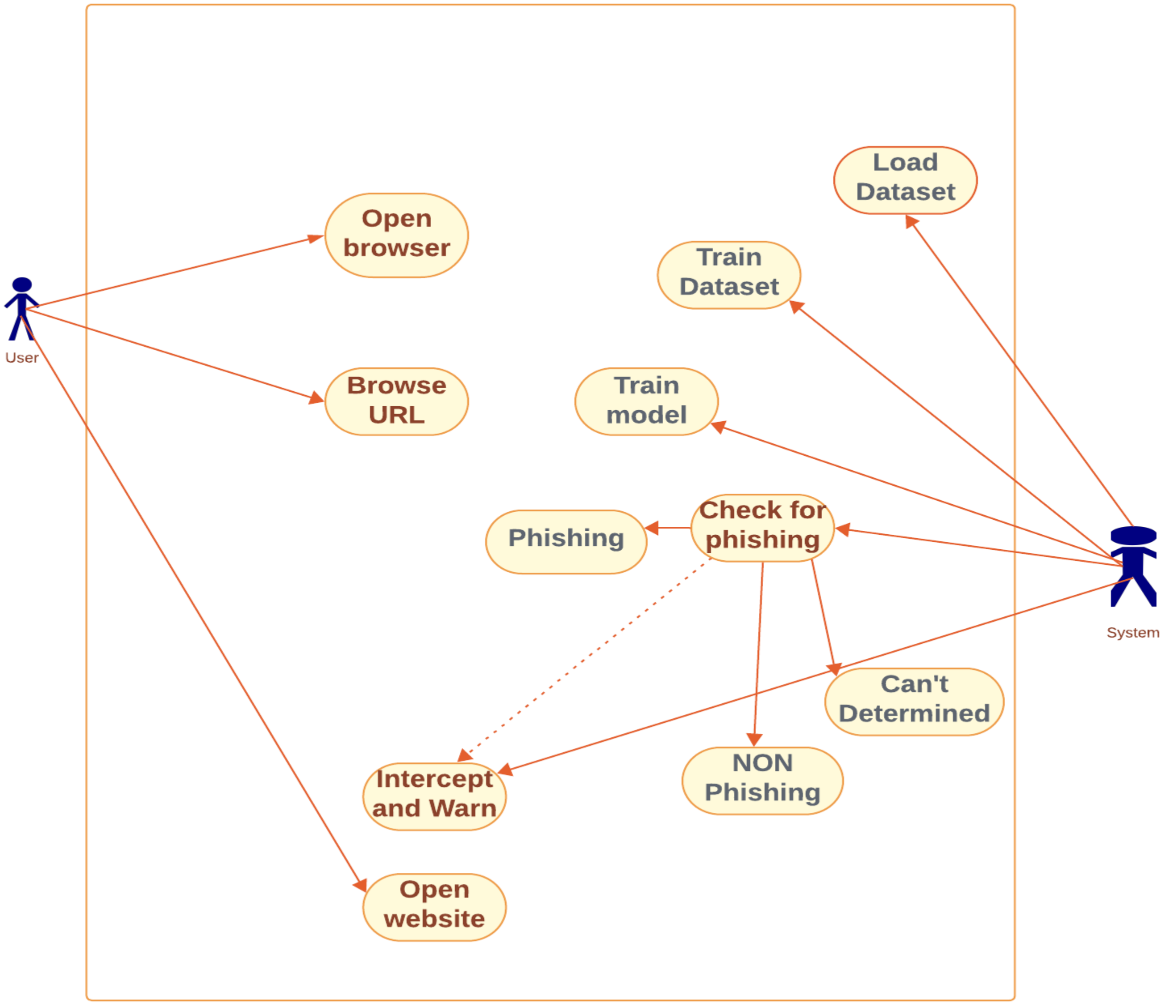


Figure 7.2: Use Case Diagram

7.1.3 Class Diagram

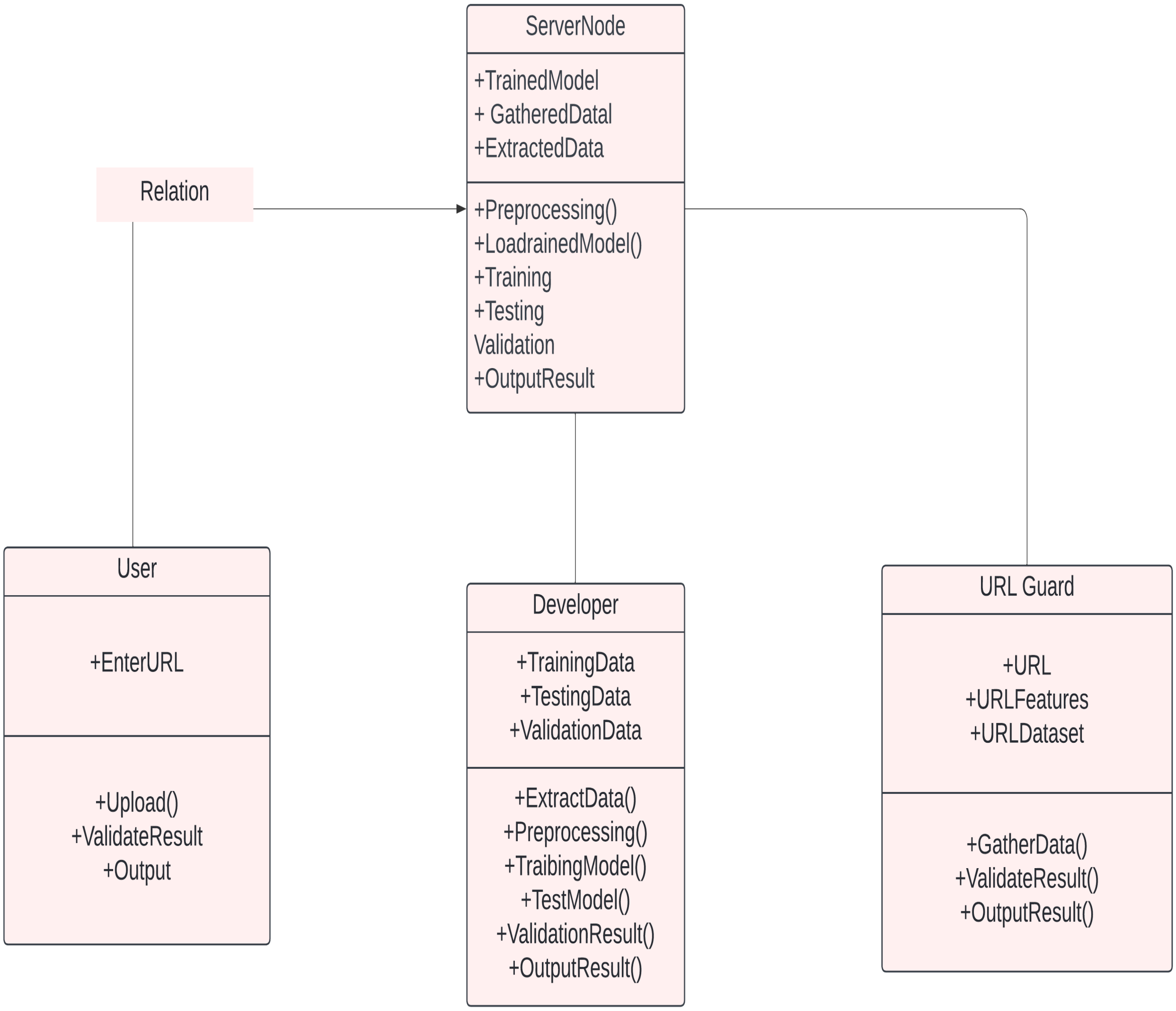


Figure 7.3: Class Diagram

7.1.4 Sequence Diagram

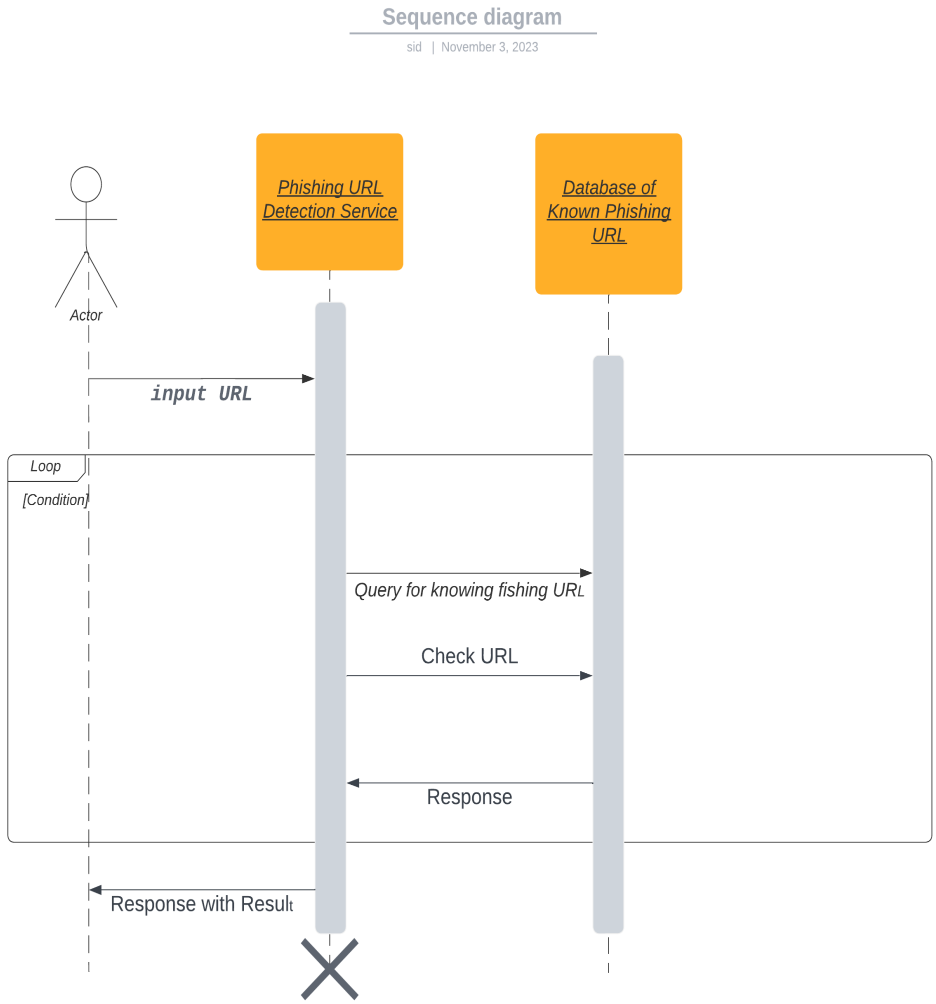


Figure 7.4: Sequence Diagram

7.1.5 Data Flow Diagrams-(Level 0,1)

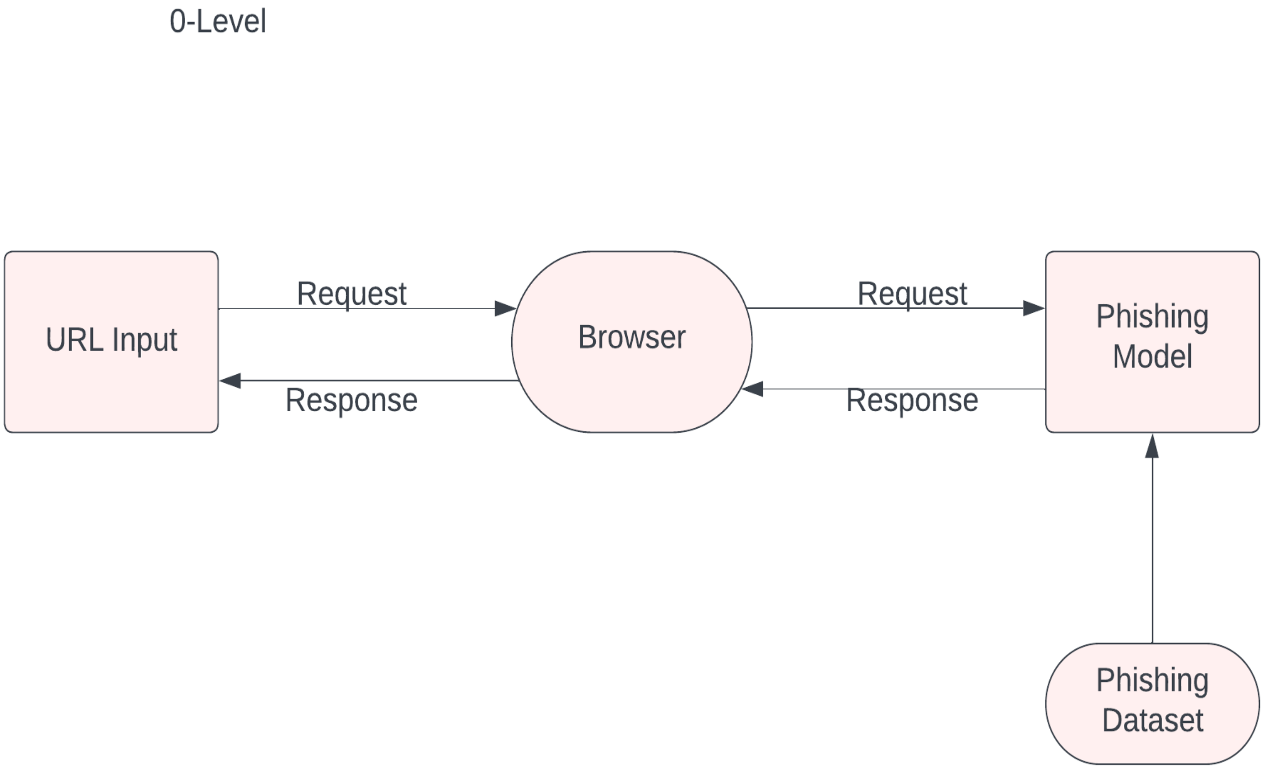


Figure 7.5: DFD Level 0

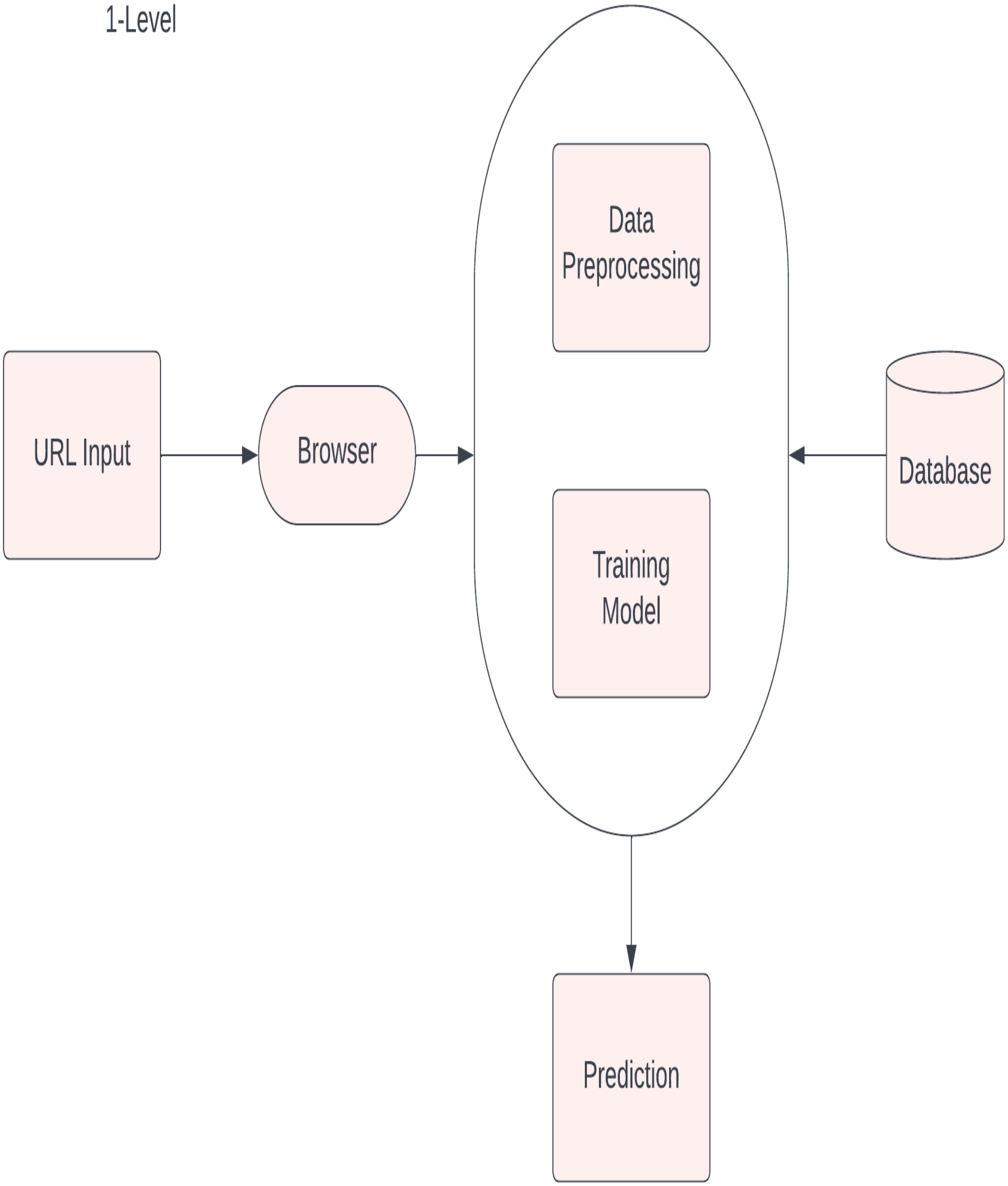


Figure 7.6: DFD Level 1

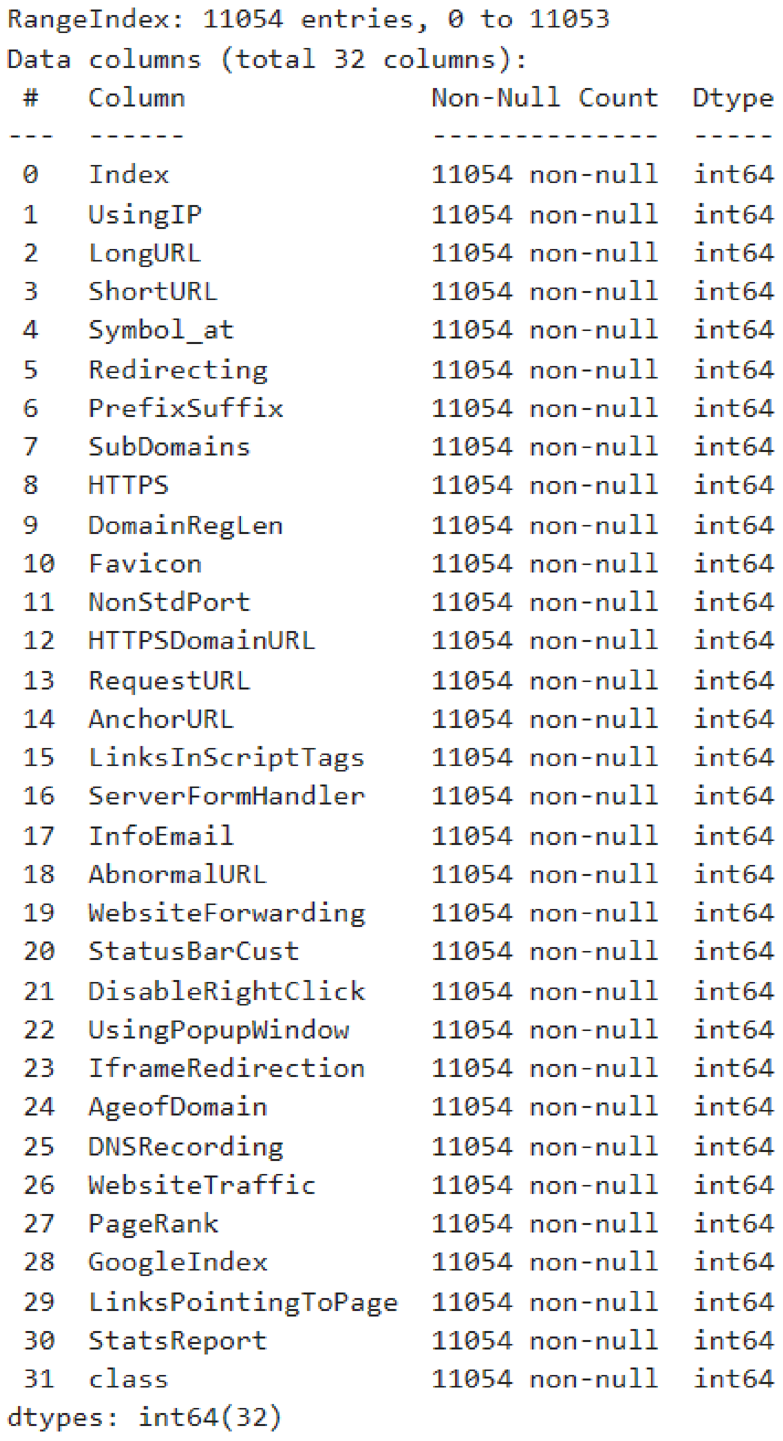
Chapter 8

Results and Experimentation

8.1 Experimental Setup

For the purpose of the experiment we have prepared a suitable dataset by removing the null values, performing cross fold valadation and grid search hyperparameter tuning.

- removing the null values: Removing null values from a dataset entails identifying and eliminating any entries that lack data. This process helps ensure data quality by preventing inaccuracies in analysis or modeling caused by missing information.
- cross fold validation: Cross-fold validation divides a dataset into multiple subsets, training the model on all but one and validating on the omitted subset. This process iterates, ensuring each subset serves as both training and validation data, providing robust model evaluation.
- Grid search hyperparameter tuning: Grid search hyperparameter tuning involves systematically searching through a specified subset of hyperparameter combinations to find the optimal configuration for a machine learning model. It exhaustively evaluates each combination to identify the best-performing set.



```
RangeIndex: 11054 entries, 0 to 11053
Data columns (total 32 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Index                11054 non-null  int64
 1   UsingIP              11054 non-null  int64
 2   LongURL              11054 non-null  int64
 3   ShortURL             11054 non-null  int64
 4   Symbol_at            11054 non-null  int64
 5   Redirecting          11054 non-null  int64
 6   PrefixSuffix         11054 non-null  int64
 7   SubDomains           11054 non-null  int64
 8   HTTPS                11054 non-null  int64
 9   DomainRegLen         11054 non-null  int64
 10  Favicon              11054 non-null  int64
 11  NonStdPort           11054 non-null  int64
 12  HTTPSDomainURL       11054 non-null  int64
 13  RequestURL           11054 non-null  int64
 14  AnchorURL            11054 non-null  int64
 15  LinksInScriptTags    11054 non-null  int64
 16  ServerFormHandler    11054 non-null  int64
 17  InfoEmail            11054 non-null  int64
 18  AbnormalURL          11054 non-null  int64
 19  WebsiteForwarding    11054 non-null  int64
 20  StatusBarCust        11054 non-null  int64
 21  DisableRightClick    11054 non-null  int64
 22  UsingPopupWindow     11054 non-null  int64
 23  IframeRedirection    11054 non-null  int64
 24  AgeofDomain          11054 non-null  int64
 25  DNSRecording         11054 non-null  int64
 26  WebsiteTraffic       11054 non-null  int64
 27  PageRank             11054 non-null  int64
 28  GoogleIndex          11054 non-null  int64
 29  LinksPointingToPage  11054 non-null  int64
 30  StatsReport          11054 non-null  int64
 31  class                11054 non-null  int64
dtypes: int64(32)
```

Figure 8.1: Features

8.2 Test Specifications

- Test Dataset for model: The model was evaluated on a separate test dataset, ensuring unbiased performance assessment.
- Evaluation Metrics: The test set performance was evaluated in terms of both loss and accuracy.
- Test dataset for system: The system was evaluated on the testing data which split to 30 percent of dataset.

8.2.1 Assumptions and Dependencies

Assumptions:

- Availability of Internet Connectivity: It is assumed that users will have access to reliable internet connectivity to interact with the system.
- Compliance with Regulatory Standards:Compliance with regulatory standards for running a phishing detection website includes adherence to data protection laws (e.g., GDPR), implementing strong security measures, such as encryption, regular audits, and following industry best practices to ensure the confidentiality, integrity, and availability of user data while effectively detecting and mitigating phishing threats.

Dependencies:

- Availability of Dataset: Correct and use-able dataset should be available
- Server Infrastructure: Dependence on the availability and reliability of server infrastructure to host the system's backend components and support real-time interactions with users.

8.3 Performance Measures

8.3.1 Accuracy

The accuracy metrics provide insights into the overall performance of the model on the test set.The test accuracy indicates the percentage of correctly classified samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8.1}$$

- Test Accuracy(hard voting):96.92.
- Test Accuracy(soft voting):96.89.

8.3.2 Precision

Precision expresses the degree to which the model classifies the phishing URLs and gauges the positive rate, or the degree to which the model predicts the positive values. With regard to precision, each classifier fared well.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8.2}$$

8.3.3 Recall

Ametric for analyzing classification models that indicates how many times the model correctly identified out of all the possible positive labels. The classifier accurately identifies the classifier for both sorts of URLs in order to predict both phishing and authentic URLs.

$$\text{Precision} = \frac{TP}{TP + FN} \tag{8.3}$$

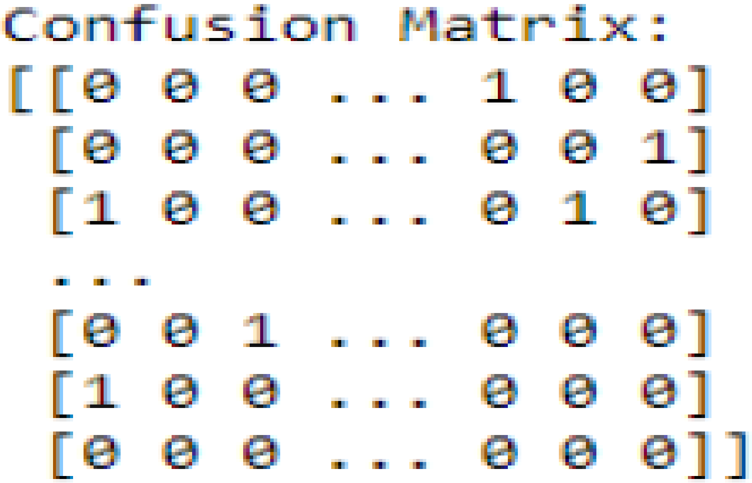
8.3.4 F1-Score

The F1score is the harmonic mean of precision and recall,where the F1 score reaches its best value (perfect precision and recall).

$$\text{F1-Score} = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \tag{8.4}$$

8.3.5 Confusion Matrix

The confusion matrix provides a tabular representation of actual versus predicted class labels. It allows us to visualize the model's performance in terms of correctly and incorrectly classified instances for each class.



```
Confusion Matrix:
[[0 0 0 ... 1 0 0]
 [0 0 0 ... 0 0 1]
 [1 0 0 ... 0 1 0]
 ...
 [0 0 1 ... 0 0 0]
 [1 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

Figure 8.2: Confusion Matrix

8.3.6 Summary and Interpretation

The Accuracy and confusion [17] matrix offer a detailed breakdown of the model's performance across different classes. From the Accuracy, which provide insights into the overall performance of the model on the test set.The test accuracy indicates the percentage of correctly classified samples.Similarly, the confusion matrix reveals that the model's predictions are predominantly concentrated along the diagonal, indicating low predictive performance across multiple classes.Further investigation into these specific classes could provide insights into factors contributing to better predictive performance, such as class imbalance,data quality, or inherent separability.

8.4 Experimental Results

8.4.1 Exact Method

The experimental results showcase the functionality of the implemented hybrid model for phishing website.The below experimental results collectively demonstrate the efficiency of the implemented hybrid model in accurately detecting, segmenting, and recognizing phishing websites.steps for exact methods are:

- Data Collection and Preprocessing:Gather a dataset of labeled examples of phishing and legitimate websites. Features could include URL length, presence of HTTPS, domain age, etc.
- Feature Engineering:Extract relevant features from the dataset that could help distinguish between phishing and legitimate websites.
- Model Selection:Choose the individual models to include in the hybrid ensemble. In this case, Random Forest, Decision Trees, and XGBoost are selected.
- Ensemble Construction:Train each of the selected models (RF, DT, XGB) on the preprocessed dataset.Combine the predictions of the individual models using Voting.
- Model Evaluation:Evaluate the performance of the hybrid model using appropriate metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
- Model Deployment:Once satisfied with the model's performance, deploy it into a production environment for real-world phishing website detection.Continuously monitor the model's performance and update it as necessary with new data or improved techniques.

Upto this step we have the expected results that retrieves the status of the CNG tank maintenance. Further we have additionally made the system adaptable to minute errors. If the model fails to recognize some of the character of the plate, then the system checks for some similar plates if available. It prints the similar plates with the similarity percentage. Henceforth, the system retrieves the information accurately.

8.4.2 Result Analysis

- Based on the results of the research the following table is provided to compare the performance of different machine learning techniques used for detection of phishing websites:

Table 8.1: Result Analysis

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 score**
Decision Tree (DT)	0.94	0.94	0.94	0.94
Random Forest (RF)	0.96	0.92	0.93	0.92
Extreme Gradient Boosting (XGB)	0.96	0.96	0.96	0.96
DT + RF + XGB (Soft Voting)	0.96	0.96	0.96	0.96
DT + RF + XGB (Hard Voting)	0.96	0.97	096	0.96

- Above Table 8.1 depicts, proposed algorithm (DT+RF+XGB) having 0.97 precision, which is highest in the table.
- Above table 8.1 depicts, proposed algorithm (DT+RF+XGB) having 0.96 accuracy, recall, and F1-Score in the table.

8.4.3 Result visualization

- The below graphs shows the Accuracy, precision, Recall and F1-Score in an joined bar graph.

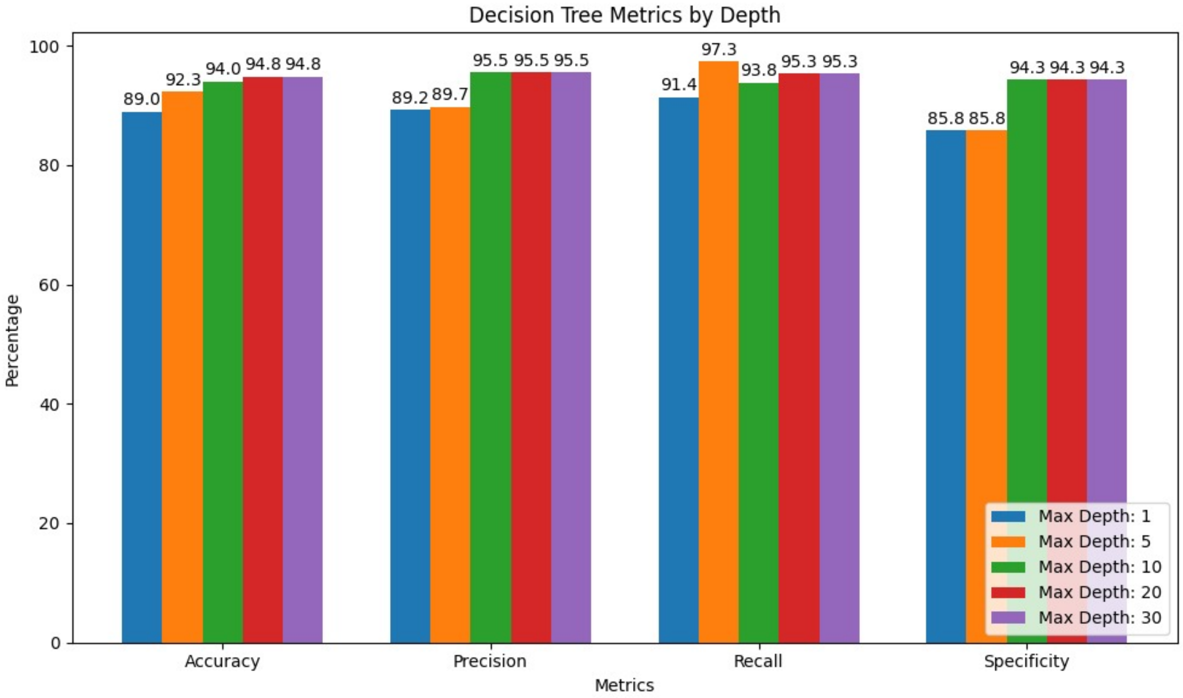


Figure 8.3: Graph for Decision Tree

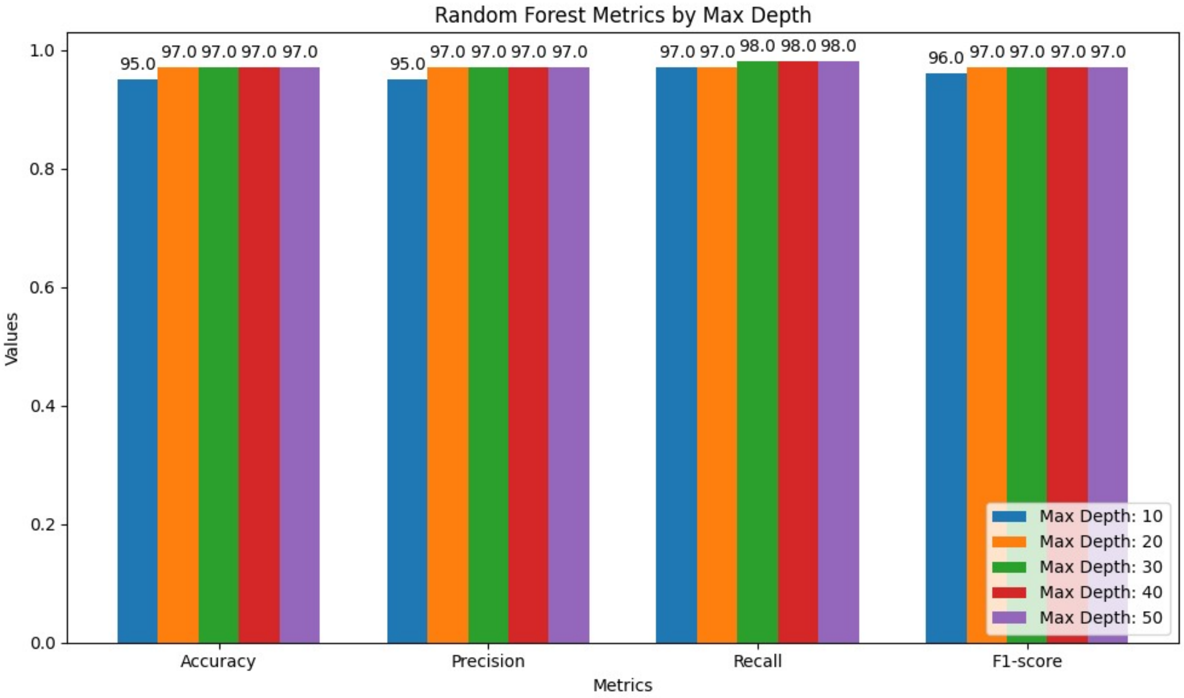


Figure 8.4: Graph for Random Forest

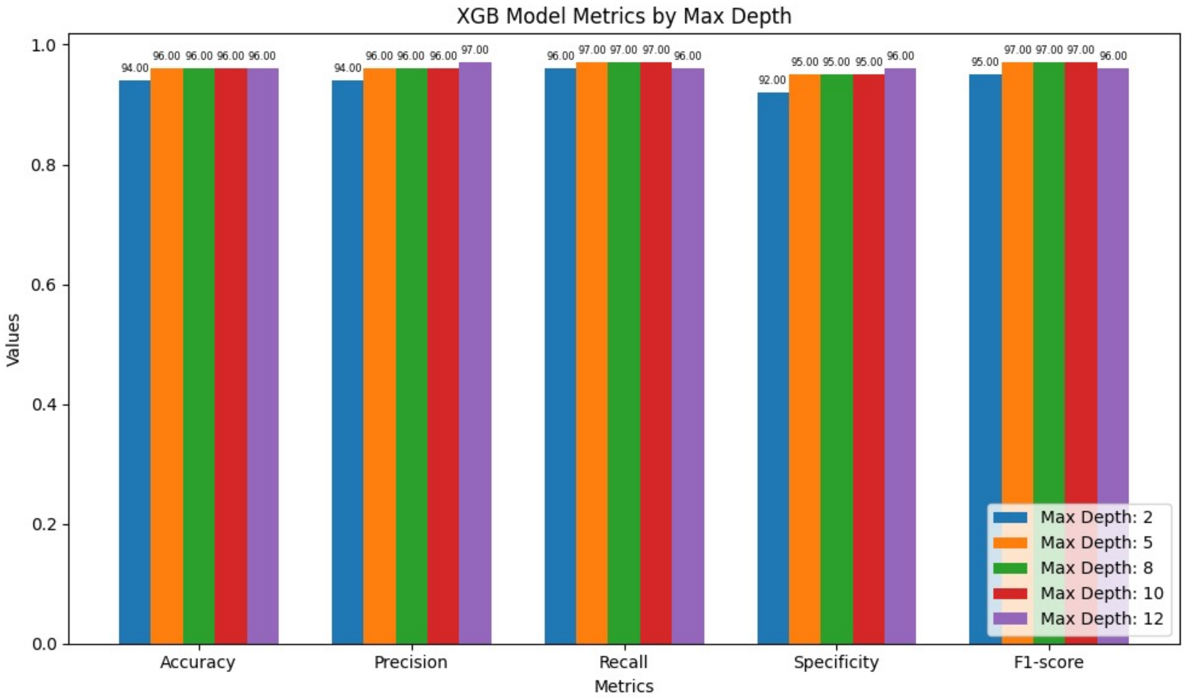


Figure 8.5: Graph for XGB

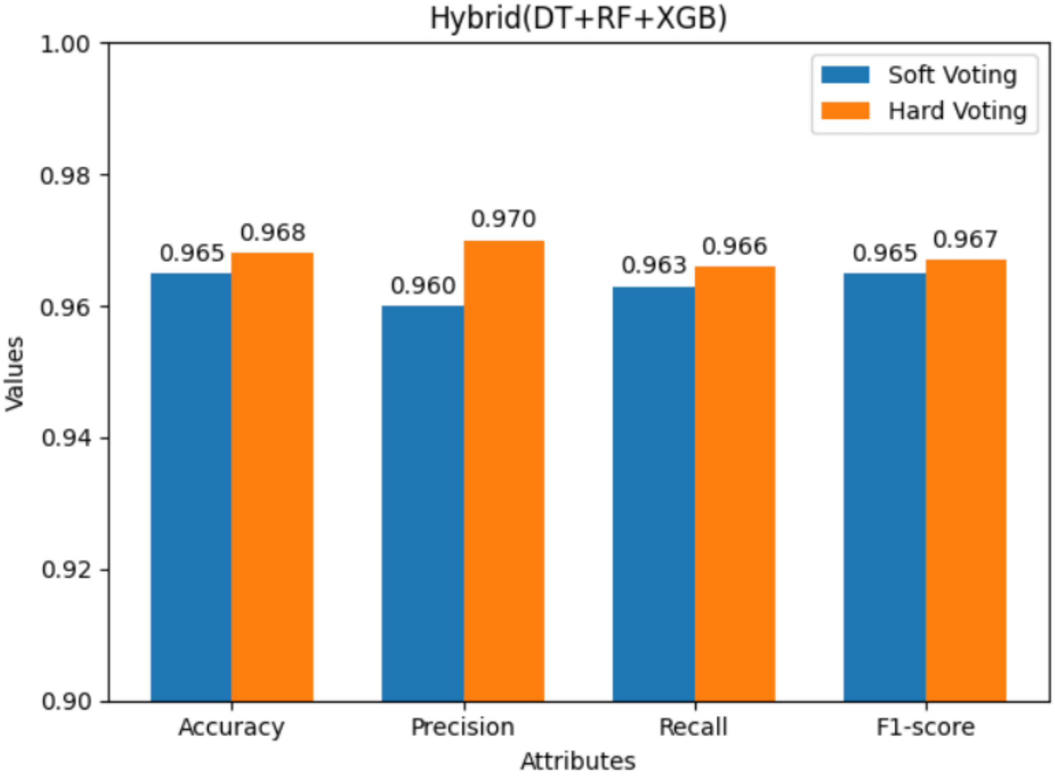


Figure 8.6: Graph for Hybrid Machine Learning Model(XGB+RF+DT)

8.5 Discussions

8.5.1 Machine Learning Algorithms for Phishing Detection

Ensemble Learning (combining multiple models) Training Datasets: Labeled datasets with phishing and legitimate URLs. Accuracy Measures: Accuracy: Overall correct classification rate. Precision: Ratio of true positives (correctly identified phishing URLs). Recall: Ratio of all phishing URLs correctly identified. F1 Score: Harmonic mean of precision and recall. Best Performing Algorithms: XGB: High accuracy and low false positive rate. DT + RF : Highest precision (few legitimate URLs misclassified). Ensemble methods often outperform single models.

8.5.2 Effectiveness of Proposed Ensemble

Improved performance compared to individual DT, RF and XGB models. High accuracy and F1 score with relatively low training time. Slightly high false positive rate (misclassifies some legitimate URLs).

8.5.3 Comparison with Other Methods

The proposed ensemble method significantly outperforms traditional machine learning techniques like Decision Trees (DT), Random Forests (RF), and Extreme Gradient Boosting (XGB). By integrating these methods into a cohesive ensemble, it achieves the lowest false positive rate, crucial for high-precision applications like phishing URL detection.

The ensemble excels in key performance metrics, boasting the highest accuracy, F1 score, and recall. This indicates its overall correctness and ability to effectively identify true phishing URLs. Additionally, the method is more efficient in training time compared to some sophisticated alternatives, making it suitable for large datasets and frequent updates.

Incorporating multi-head self-attention mechanisms and Convolutional Neural Networks (CNNs) enhances its performance by capturing intricate patterns and extracting hierarchical features. This combination not only reduces false positives but also improves robustness and generalization.

Overall, this ensemble method showcases the potential of advanced machine learning techniques in enhancing cybersecurity, particularly in phishing detection.

Table 8.2: Result Comparison

Sr.no	Author	Algorithm	Evaluation Parameters
1	Proposed algorithm	Hybrid Machine Learning Model(DT+RF+XGB)	• Accuracy= 0.96 • precision=0.96 • Recall= 0.97 • F1-Score=0.96
2	ABDULKARIM1, MOBEEN SHAHROZ 2, KHABIB MUSTOFA 1, SAMIR BRAHIM BELHAOUARI 3, ANDS. RAMANA KUMAR JOGA(2023)	Hybrid Machine Learning Model(SVC + DT +LR)r	• Accuracy= 0.95 • Precision=0.95 • Recall=0.96 • F1-Score=0.95
3	Sonowal, G., Kuppusamy, K(2020)	five-layer phishing detection model called PhiDMA	• Accuracy= 0.92 • Precision=0.91 • Recall=0.90 • F1-Score=0.90

4	KangLengChiew a, ChoonLinTan a,, KokSheik Wong b, Kelvin S.C.Yongc,Wei KingTionga(2019)	Hybrid Ensemble Feature Selection(HEFS),Random ForestClassifier	• Accuracy= 0.94
5	YONG FANG, CHENG ZHANG, CHENG HUANG, LIANG LIU,ANDYUE YANG (2019)	Deep learning model named THEMIS(advanced version of RCNN)	• Accuracy= 0.99 • Precision=0.99 • Recall=0.99 • F1-Score=0.99
6	Ozgur Koray Sahingoz, Ebubekir Buber b,OnderDemirb, BanuDiri(2017)	Using Random Forest with NLP-based features	• Accuracy= 0.97 • Precision=0.97 • Sensitivity=0.99 • F1-Score=0.98

Chapter 9

Proposed GUI / Backend specifications

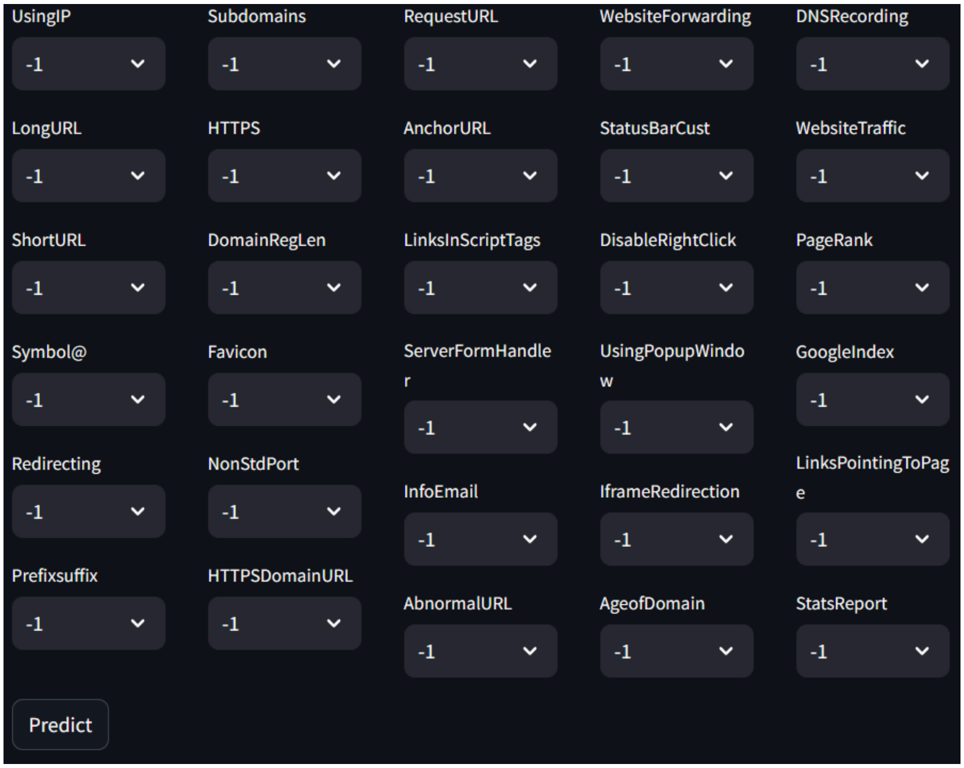
The system integrates with an extensive dataset containing pertinent information related to phishing websites. This dataset serves as a valuable resource for verifying phishing URLs information and providing additional context to users. This proactive approach enhances user experience and promotes adherence to maintenance schedules, contributing to safer and more efficient predictions.[17]By incorporating these advanced functionalities and additional features, phishing recognition and verification system not only addresses the core problem statement but also extends its capabilities to provide comprehensive solutions for phishing detection. Through intuitive GUI design, robust backend processing, and innovative features, The application sets a new standard for phishing detection system, offering unparalleled accuracy, reliability, and user satisfaction.

9.1 Proposed GUI

The proposed graphical user interface (GUI) of phishing detection system is developed using Streamlit, a powerful Python library for building interactive web applications. The GUI offers a user-friendly experience with intuitive functionalities, where Users can input URLs. The layout and features of the GUI:

- Upload functionality for URLs.

- Display of the uploaded URLs.
- Recognition of characters in the URLs.
- Display of the predicted phishing non-phishing URLs.



UsingIP -1	Subdomains -1	RequestURL -1	WebsiteForwarding -1	DNSRecording -1
LongURL -1	HTTPS -1	AnchorURL -1	StatusBarCust -1	WebsiteTraffic -1
ShortURL -1	DomainRegLen -1	LinksInScriptTags -1	DisableRightClick -1	PageRank -1
Symbol@ -1	Favicon -1	ServerFormHandler -1	UsingPopupWindow -1	GoogleIndex -1
Redirecting -1	NonStdPort -1	InfoEmail -1	IframeRedirection -1	LinksPointingToPage -1
Prefixsuffix -1	HTTPSDomainURL -1	AbnormalURL -1	AgeofDomain -1	StatsReport -1

9.2 Backend Module

The backend module of the system encompasses the core functionalities that running of hybrid machine learning model. In hybrid machine learning model (DT+RF+XGB)This model integrates advanced computer vision algorithms, primarily utilizing the pickle library, for robust license plate detection. Upon receiving uploaded images from the GUI, the backend preprocesses the images to facilitate efficient detection and recognition. Furthermore, the backend incorporates a deep learning model trained for character recognition, enabling accurate identification of characters [7] on the detected license plate. Additionally, the backend manages the retrieval of supplementary information from a dataset based on the predicted license plate number.

Chapter 10

Conclusion

10.1 Conclusion

The proposed algorithm is successfully developed and implemented a comprehensive [3]URL-based phishing detection system. This system leverages a variety of machine learning algorithms to effectively identify and thwart phishing attempts, particularly those involving deceptive URLs. The project's objectives, including enhancing accuracy(0.96),recall(0.96),F1-score(0.96)and Precision(0.97) real-time monitoring, and [12]user-friendly interface design, have been achieved.Through continuous monitoring and collaboration with the cybersecurity community, the system remains adaptive and resilient against evolving phishing tactics.

In the future Create an intuitive and user-friendly interface for system administrators, facilitating easy management, monitoring, and response to potential phishing threat.Establish procedures for continuous monitoring, evaluation, and updating of the system to remain effective against evolving phishing tactics over time.

Bibliography

[1] Neda Abdelhamid, Aladdin Ayesh, and Fadi Thabtah. Phishing detection based associative classification data mining. *Expert Systems with Applications*, 41(13):5948–5959, 2014.

[2] Mohammed S Alam and Son T Vuong. Random forest classification for detecting android malware. In *2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing*, pages 663–669. IEEE, 2013.

[3] Ebubekir Buber, Banu Dırı, and Ozgur Koray Sahingoz. Detecting phishing attacks from url by using nlp techniques. In *2017 International conference on computer science and Engineering (UBMK)*, pages 337–342. IEEE, 2017.

[4] Kang Leng Chiew, Choon Lin Tan, KokSheik Wong, Kelvin SC Yong, and Wei King Tiong. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, 484:153–166, 2019.

[5] Yong Fang, Cheng Zhang, Cheng Huang, Liang Liu, and Yue Yang. Phishing email detection using improved rcnn model with multilevel vectors and attention mechanism. *IEEE Access*, 7:56329–56340, 2019.

[6] Fang Feng, Qingguo Zhou, Zebang Shen, Xuhui Yang, Lihong Han, and JinQiang Wang. The application of a novel neural network in the detection of phishing websites. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–15, 2018.

[7] Abdul Karim, Mobeen Shahroz, Khabib Mustofa, Samir Brahim Belhaouari, and

S Ramana Kumar Joga. Phishing detection system through hybrid machine learning based on url. *IEEE Access*, 11:36805–36822, 2023.

[8] Pradip Paithane and Sangeeta Kakarwal. Lmns-net: Lightweight multiscale novel semantic-net deep learning approach used for automatic pancreas image segmentation in ct scan images. *Expert Systems with Applications*, 234:121064, 2023.

[9] Pradip Mukundrao Paithane. Yoga posture detection using machine learning. In *Artificial Intelligence in Information and Communication Technologies, Healthcare and Education*, pages 27–33. Chapman and Hall/CRC, 2022.

[10] Pradip Mukundrao Paithane. Random forest algorithm use for crop recommendation. *ITEGAM-JETIA*, 9(43):34–41, 2023.

[11] Pradip Mukundrao Paithane and SN Kakarwal. Automatic pancreas segmentation using a novel modified semantic deep learning bottom-up approach. *International Journal of Intelligent Systems and Applications in Engineering*, 10(1):98–104, 2022.

[12] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, and Banu Diri. Machine learning based phishing detection from urls. *Expert Systems with Applications*, 117:345–357, 2019.

[13] Vahid Shahrivari, Mohammad Mahdi Darabi, and Mohammad Izadi. Phishing detection using machine learning techniques. *arXiv preprint arXiv:2009.11116*, 2020.

[14] Sami Smadi, Nauman Aslam, and Li Zhang. Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decision Support Systems*, 107:88–102, 2018.

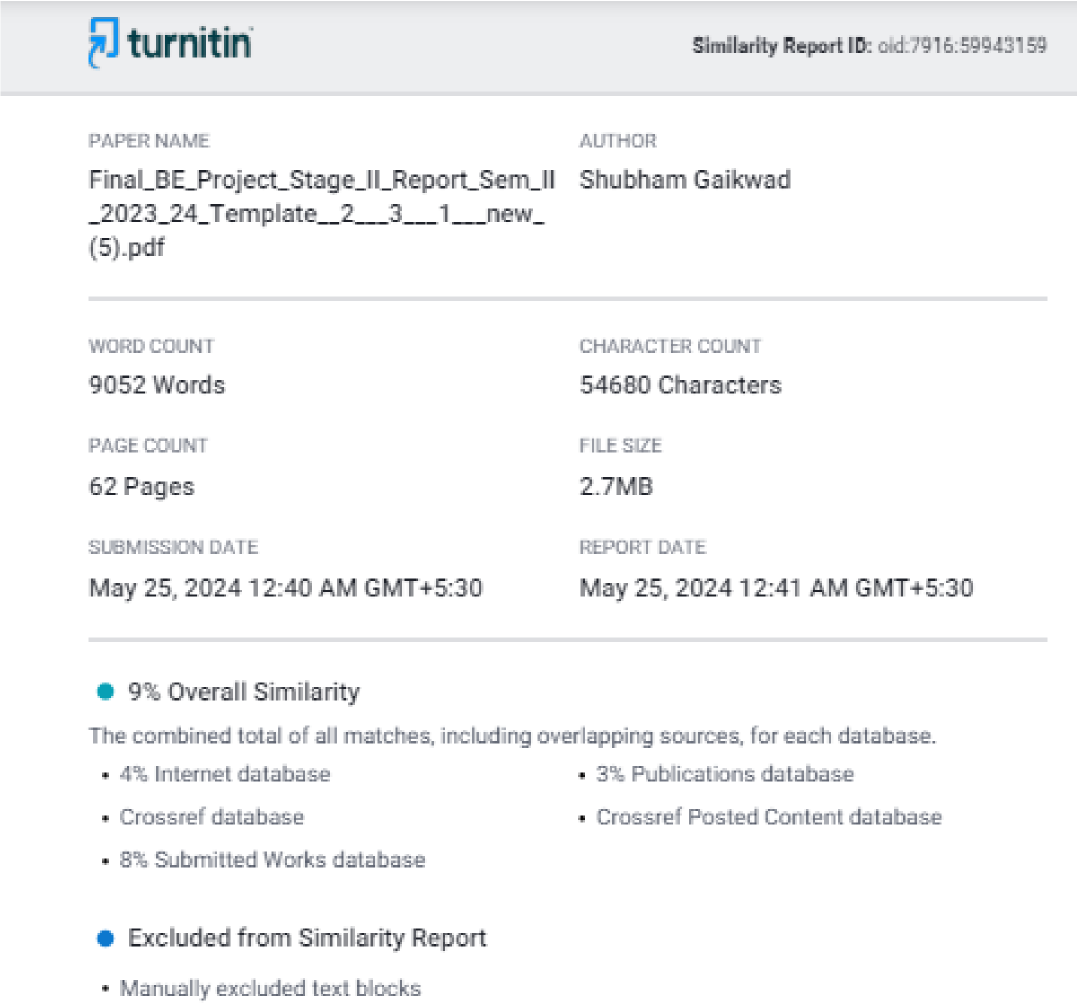
[15] Gunikhan Sonowal and KS Kuppusamy. Phidma–a phishing detection model with multi-filter approach. *Journal of King Saud University-Computer and Information Sciences*, 32(1):99–112, 2020.

[16] Shan Wang, Sulaiman Khan, Chuyi Xu, Shah Nazir, and Abdul Hafeez. Deep learning-based efficient model development for phishing detection using random forest and blstm classifiers. *Complexity*, 2020:1–7, 2020.

[17] Mouad Zouina and Benaceur Outtaj. A novel lightweight url phishing detection system using svm and similarity index. *Human-centric Computing and Information Sciences*, 7:1–13, 2017.

Appendix A

Plagiarism Report



turnitin

Similarity Report ID: oid:7916:59943159

PAPER NAME

Final_BE_Project_Stage_II_Report_Sem_II_2023_24_Template__2___3___1___new_(5).pdf

AUTHOR

Shubham Gaikwad

WORD COUNT

9052 Words

CHARACTER COUNT

54680 Characters

PAGE COUNT

62 Pages

FILE SIZE

2.7MB

SUBMISSION DATE

May 25, 2024 12:40 AM GMT+5:30

REPORT DATE

May 25, 2024 12:41 AM GMT+5:30

● 9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 4% Internet database
- Crossref database
- 8% Submitted Works database
- 3% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Manually excluded text blocks

Appendix B

Base Paper

A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari and S. R. K. Joga, ”Phishing Detection System Through Hybrid Machine Learning Based on URL,” in IEEE Access, vol. 11, pp. 36805-36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

paper link: `https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10058201`

Appendix C

Tools Used

1. Jupyter Notebook:

   - **Purpose:** Employed Jupyter Notebook for interactive data processing and algorithm development.
   - **Benefits:** Facilitated a dynamic and exploratory workflow, allowing for real-time code execution and visualization. Enhanced collaboration among team members during the development phase.

2. Gantt Chart:

   - **Tool:** Vizzlo's Gantt chart
   - **Purpose:** Developed a Gantt chart to visualize and manage project tasks, milestones, and Time requirements.
   - **Benefits:** Provided a clear timeline overview, aiding in task scheduling, resource allocation, and effective project management.

3. LaTeX:

   - **Purpose:** Employed LaTeX for document preparation and report writing.
   - **Benefits:** Ensured a professional and consistent document format, particularly valuable for technical reports. LaTeX is advantageous in producing high-quality and well-organized documentation.

4. Star UML:

    - **Purpose:** Used Star UML for designing and visualizing system architecture, including use case diagrams and other UML diagrams.
    - **Benefits:** Enabled the creation of clear and comprehensive visual representations of the system structure, improving communication among team members and stakeholders.

Appendix D

Papers Published/Certificates

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.176 ISSN: 2582- 3930

A Systematic Review of Evaluating the Efficiency of Hybrid Machine Learning Techniques in Unmasking Phishing URLs

Dr. Pradip Paithane[1], Siddheshwar Patil[2], Abhishek Kubde[3],
Rohit Mehetre[4], Shubham Gaikwad[5]

[1]*Department of Artificial Intelligence and Data Science, VPKBIET, Baramati*
[2]*Department of Artificial Intelligence and Data Science, VPKBIET, Baramati*
[3]*Department of Artificial Intelligence and Data Science, VPKBIET, Baramati*
[4]*Department of Artificial Intelligence and Data Science, VPKBIET, Baramati*
[5]*Department of Artificial Intelligence and Data Science, VPKBIET, Baramati*

---***---

Abstract - Phishing URLs pose significant threats to individuals and organizations, exploiting vulnerabilities to perpetrate scams and fraud. The Indian Cybercrime Coordination Center (I4C) faces challenges in effectively addressing these threats, necessitating a systematic examination of phishing URL detection methods. This review article focuses on evaluating the efficacy of hybrid machine learning algorithms, particularly URL-based techniques, in combating phishing. Leveraging the unparalleled accuracy and performance of hybrid machine learning models, this research represents a groundbreaking approach to early detection and mitigation of phishing URLs, which are a prevalent cause of fraud and hacking globally. Recent advancements in hybrid machine learning have facilitated the integration of multiple methods to enhance accuracy and reliability in identifying and thwarting phishing attempts. This study contributes to the ongoing efforts in cybersecurity by shedding light on the potential of hybrid machine learning techniques in unmasking phishing URLs, thereby bolstering defenses against cyber threats.

Key Words: Phishing URLs, Hybrid Machine Learning, accuracy, URL-based techniques, Indian cybercrime coordination center (I4C)

1. INTRODUCTION

One of the core areas of computer science, computer security, is significantly impacted by criminal activities directed towards Internet users. Attacks and security problems of many kinds began to surface as the Internet and information technology applications developed. Beginning in the early 1990s, as the Internet gained worldwide popularity and accessibility, security risks additionally started to change. Attackers have targeted these sensitive data, and a particular type of assault known as phishing first surfaced in the mid-1990s.

One of the main topics covered in this thesis is phishing, which is a type of socially engineered online identity theft. In an ever-evolving cybersecurity landscape, the proliferation of phishing attacks poses a serious threat to both individuals and organizations. Multiple approaches that can identify phishing at various stages have been put forth by researchers looking at the issues surrounding phishing attacks. Some identify phishing attempts right down to the site page. Furthermore, certain models are designed to identify phishing attempts at an earlier email level, when the attacker is still attempting to persuade the recipient to visit the bogus website. This is a superior approach since, in the event that a phishing assault is discovered at the webpage level, it will first examine each Uniform Resource Locator (URL) the user tries to open before granting access, which will slow down website navigation.

Second, since the attack is discovered sooner, users are safer when phishing attacks are identified at the email level. For example, certain codes may be downloaded onto the user's device to infect it when the user views a web page. Furthermore, databases of phishing emails are always accessible, while a phony webpage only lasts for roughly 46 hours on average.

Fig -1: URL presentation based on HTTP

In fig1, label1 represents HTTP (Hypertext transfer protocol), which is used to acquire resources based on client requests. Label 2 denotes a hostname, and the hostname is further subdivided into three subdomains: top-level domain (also known as web address), and domain labeled 6 relates to a web server's directory. A label 7 "v" character with the value "AbcdEffGhIJ" and a label 6 "?" character in a URL initialize the parameter x. URLs are often used to represent website addresses.

© 2024, IJSREM | www.ijsrem.com DOI: 10.55041/IJSREM29671 | Page 1

URLGuard: A Holistic Hybrid Machine Learning Approach for Phishing Detection

First Author[1], Second Author[2], Third Author[3] Font Size 12

[1]Dr.Pradip Paithane
[2]Mr.Siddheshwar Patil
Mr.Abhishek Kubde
Mr.Shubham Gaikwad
Mr.Rohit Mhetre

---***---

Abstract - The fast growth of Internet technology has significantly changed online users' experiences, while security concerns are becoming increasingly overpowering. Among these concerns, phishing stands out as a prominent criminal activity that uses social engineering and technology to steal a victim's identification data and account information. According to the Anti-Phishing Working Group (APWG), the number of phishing detections increased by 46 in the first quarter of 2018 compared to the fourth quarter of 2017. So to overcome these situations below paper introduces a phishing detection system using a hybrid machine learning approach based on URL attributes. It addresses the growing threat of phishing attacks that exploit email manipulation and fake websites to deceive users and steal sensitive data. The study employs a phishing URL dataset with over 11,000 websites, extracted from a reputable repository. After pre-processing, a hybrid machine learning model, which includes Decision Tree, Random Forest, and XGB is employed to safeguard against phishing URLs. The proposed approach undergoes evaluation with key metrics such as precision, accuracy, recall, F1-score, and specificity. Results demonstrate that the proposed method surpasses other models, achieving superior accuracy and efficiency in detecting phishing attacks.

Key Words: Anti-Phishing Working Group (APWG), Decision Tree, and Random Forest, and XGB, hybrid machine learning.

1. INTRODUCTION *(Size 11, Times New roman)*

In the present interconnected digital environment, the internet functions not only as a conduit facilitating the vast exchange of information but also as a platform utilized for engaging in malicious activities. One of the most widespread forms of these malicious activities is phishing, which entails the use of deceptive strategies by cybercriminals with the aim of misleading individuals into revealing confidential information such as their login credentials, financial particulars, or personal data.

Phishing attacks commonly manifest in the guise of seemingly authentic emails, messages, or websites that are meticulously crafted to dupe users into believing they are interacting with a reputable and trustworthy source. Despite the fact that the concept of phishing is not novel, its level of sophistication and prevalence have undergone significant transformations in recent times, thereby presenting formidable obstacles to individuals, enterprises, and institutions on a global scale. As reported by the Anti-Phishing Working Group (APWG), a staggering number exceeding 200,000 distinct phishing websites were identified solely in the initial quarter of the year 2023, thereby underscoring the magnitude and gravity of this perilous threat.

Furthermore, it is imperative to highlight the significant financial ramifications associated with phishing attacks, which are truly astounding, leading to the annual loss of billions of dollars as a result of deceitful activities carried out through these cunning methods. The primary objective of this academic inquiry is to thoroughly investigate the complex realm of phishing websites, delving into their intricate operations, distinguishing features, and profound effects on the realm of cybersecurity. Through a comprehensive analysis of the strategies utilized by malicious actors to fabricate and disseminate phishing websites, along with the approaches for identifying and containing these risks, the primary aim of this scholarly endeavor is to enrich the comprehension of the dynamic landscape of online fraudulence.

Through empirical analysis, case studies, and scholarly insights, this paper will provide valuable insights into the following key areas:

1. The anatomy of phishing websites: Understanding the design, structure, and functionality of phishing websites, including common tactics used to mimic legitimate entities and exploit human psychology.

2. Detection and classification techniques: Exploring the methodologies and technologies utilized to identify and categorize phishing websites, from heuristic analysis to machine learning algorithms.

3. Impacts and consequences: Assessing the multifaceted repercussions of phishing attacks on individuals, businesses, and society at large, including financial losses, reputational damage, and erosion of trust in online platforms.

4. Countermeasures and best practices: Examining proactive measures and defensive strategies aimed at combating phishing threats, ranging from user education and awareness campaigns to technical solutions such as email filtering and website authentication protocols.