A

Major Project

on

PHISHING DETECTION SYSTEM THROUGH HYBRID MACHINE LEARNING BASED ON URL

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

P. Rahul Yadav	(217R1A0542)
S. Sai Charan	(217R1A0549)
T. Saketh	(217R1A0561)

Under the Guidance of

Dr. NUTHANAKANTI BHASKAR

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)
Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401.

April, 2025.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled "**PHISHING DETECTION SYSTEM THROUGH HYBRID MACHINE LEARNING BASED ON URL**" being submitted by **P. Rahul Yadav (217R1A0542), S. Sai Charan (217R1A0549) & T. Saketh (217R1A0561)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. Nuthanakanti Bhaskar
Associate Professor
INTERNAL GUIDE

Dr. Nuthanakanti Bhaskar
HoD

Dr. A. Raji Reddy
DIRECTOR

Signature of External Examiner

Submitted for viva voice Examination held on ____________________

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **Dr. Nuthanakanti Bhaskar,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We take this opportunity to extend our heartfelt appreciation to the Project Review Committee (PRC) Coordinators **Dr. K. Maheswari, Dr. J. Narasimharao, Ms. K. Shilpa, and Mr. K. Ranjith Reddy** for their unwavering support, insightful guidance, and valuable inputs, which played a crucial role in steering this project through its various stages.

Our sincere appreciation also goes to **Dr. Nuthanakanti Bhaskar**, Head, for his encouragement and continuous support in ensuring the successful completion of our project.

We are deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, we extend our profound gratitude to Sri. **Ch. Gopal Reddy**, Chairman, Smt. **C. Vasantha Latha**, Secretary and Sri. **C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

We also acknowledge and appreciate the guidance and assistance provided by the faculty and staff of **CMR Technical Campus**, whose contributions have been invaluable in bringing this project to fruition.

Lastly, we sincerely thank our families for their unwavering support and encouragement. We also extend our gratitude to the teaching and non-teaching staff of CMR Technical Campus for their guidance and assistance. Their contributions, along with the support of everyone who helped directly or indirectly, have been invaluable in the successful completion of this project.

P. Rahul Yadav (217R1A0542)
S. Sai Charan (217R1A0549)
T. Saketh (217R1A0561)

VISION AND MISSION

INSTITUTE VISION:

To Impart quality education in serene atmosphere thus strive for excellence in Technology and Research.

INSTITUTE MISSION:

1. To create state of art facilities for effective Teaching- Learning Process.

2. Pursue and Disseminate Knowledge based research to meet the needs of Industry & Society.

3. Infuse Professional, Ethical and Societal values among Learning Community.

DEPARTMENT VISION:

To provide quality education and a conducive learning environment in computer engineering that foster critical thinking, creativity, and practical problem-solving skills.

DEPARTMENT MISSION:

1. To educate the students in fundamental principles of computing and induce the skills needed to solve practical problems.

2. To provide State-of-the-art computing laboratory facilities to promote industry institute interaction to enhance student's practical knowledge.

3. To inculcate self-learning abilities, team spirit, and professional ethics among the students to serve society.

ABSTRACT

This project is titled as “Phishing Detection System Through Hybrid Machine Learning Based on URL”. Currently, numerous types of cybercrime are organized through the internet. Hence, this study mainly focuses on phishing attacks. Although phishing was first used in 1996, it has become the most severe and dangerous cybercrime on the internet. Phishing utilizes email distortion as its underlying mechanism for tricky correspondences, followed by mock sites, to obtain the required data from the people in question. Different studies have presented their work on the precaution, identification, and knowledge of phishing attacks; however, there is currently no complete and proper solution for frustrating them. Therefore, machine learning plays a vital role in defending against cybercrimes involving phishing attacks. The proposed study is based on the phishing URL-based dataset extracted from the famous dataset repository, which consists of phishing and legitimate URL attributes collected from 11,000+ website datasets in vector form. After preprocessing, many machine learning algorithms have been applied and designed to prevent phishing URLs and provide protection to the user. This study uses machine learning models such as Decision Tree (DT), Linear Regression (LR), Random Forest (RF), Naive Bayes (NB), Gradient Boosting Classifier (GBM), K-Neighbors Classifier (KNN), Support Vector Classifier (SVC), and the proposed hybrid LSD model, which is a combination of Logistic Regression, Support Vector Machine, and Decision Tree (LR+SVC+DT) with soft and hard voting, to defend against phishing attacks with high accuracy and efficiency. The Canopy Feature Selection technique with Cross-Fold Validation and Grid Search Hyperparameter Optimization techniques are used with the proposed LSD model. Furthermore, to evaluate the proposed approach, different evaluation parameters were adopted, such as precision, accuracy, recall, F1-score, and specificity, to illustrate the effects and efficiency of the models. The results of the comparative analyses demonstrate that the proposed approach outperforms the other models and achieves the best results.

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
Figure 3.1	Project Architecture of Phishing Detection System Through Hybrid Machine Learning Based on URL	12
Figure 3.2	Dataflow Diagram of Phishing Detection System Through Hybrid Machine Learning Based on URL	15
Figure 4.1	LSD(Logistic Regression,Support Vector Machine, Decision Tree)	16
Figure 4.2	Hybrid LSD(Hard)	17
Figure 4.3	Hybrid LSD(Soft)	18
Figure 4.4	Gradient Boosting	18
Figure 4.5	Random Forest	19
Figure 4.6	Decision Tree	20
Figure 4.7	Support Vector Classifier	20
Figure 4.8	Logistic Regression	21

Figure 4.9	Naive Bayes	22
Figure 4.10	Data set directory structure with file named 'datasets' having all the training URLs	22
Figure 4.11	Screenshot of the "datasets(Excel)" File Showing Sample	23
Figure 5.1	command prompt to redirect to our web page	28
Figure 5.2	Home page of Phishing detection System Through Hybrid Machine Learning Based on URL	29
Figure 5.3	Registration Page for Phishing Detection System Through Hybrid Machine Learning Based on URL.	30
Figure 5.4	Profile details of Phishing detection System Through Hybrid Machine Learning Based on URL.	31
Figure 5.5	Prediction of URL Type of Phishing detection System Through Hybrid Machine Learning Based on URL.	32
Figure 5.6	Result for Phishing detection System Through Hybrid Machine Learning Based on URL.	33

Figure 5.7	Login Page for Service Provider in Phishing detection System Through Hybrid Machine Learning Based on URL.	34
Figure 5.8	Details of all Registered users for Phishing detection System Through Hybrid Machine Learning Based on URL.	35
Figure 5.9	Accuracy of Trained and Tested URL Data sets in Bar graph representation for Phishing detection System Through Hybrid Machine Learning Based on URL.	36
Figure 5.10	Accuracy of Trained and Tested URL Data sets in Line chart representation for Phishing detection System Through Hybrid Machine Learning Based on URL.	37
Figure 5.11	Accuracy of Trained and Tested URL Data sets in Pie Chart representation for Phishing detection System Through Hybrid Machine Learning Based on URL.	38
Figure 5.12	Predicted URL details for Phishing detection System Through Hybrid Machine Learning Based on URL.	39

Figure 5.13	Ratio of Results for Phishing detection System Through Hybrid Machine Learning Based on URL.	40
Figure 5.14	Ratio of Results in Line Chart Representation for Phishing detection System Through Hybrid Machine Learning Based on URL.	41
Figure 5.15	Ratio of Results in Pie Chart Representation for Phishing detection System Through Hybrid Machine Learning Based on URL.	42

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
Table 6.2.1	Uploading Dataset	44
Table 6.2.2	Classification	44

TABLE OF CONTENTS

ABSTRACT i

LIST OF FIGURES ii

LIST OF TABLES vi

1. INTRODUCTION 1

1.1 PROJECT PURPOSE 1

1.2 PROJECT FEATURES 2

2. LITERATURE SURVEY 3

2.1 REVIEW OF RELATED WORK 5

2.2 DEFINITION OF PROBLEM STATEMENT 7

2.3 EXISTING SYSTEM 7

2.4 PROPOSED SYSTEM 8

2.5 OBJECTIVES 10

2.6 HARDWARE & SOFTWARE REQUIREMENTS 11

2.6.1 HARDWARE REQUIREMENTS 11

2.6.2 SOFTWARE REQUIREMENTS 11

3. SYSTEM ARCHITECTURE & DESIGN 12

3.1 PROJECT ARCHITECTURE 12

3.2 DESCRIPTION 13

3.3 DATA FLOW DIAGRAM 14

4. IMPLEMENTATION 16

4.1 ALGORITHMS USED 16

4.2 SAMPLE CODE 25

5. RESULTS & DISCUSSION 28

6. VALIDATION 43

6.1 INTRODUCTION 43

6.2 TEST CASES 44

6.2.1 UPLOADING DATASET 44

6.2.2 CLASSIFICATION 44

7. CONCLUSION & FUTURE ASPECTS 46

7.1 PROJECT CONCLUSION 46

7.2 FUTURE ASPECTS 47

8. BIBLIOGRAPHY 48

8.1 REFERENCES 48

8.2 GITHUB LINK 49

1. INTRODUCTION

1. INTRODUCTION

The project, titled **"Phishing Detection System Through Hybrid Machine Learning Based on URL,"** is aimed at developing an intelligent and scalable system that can automatically detect and classify phishing URLs with high accuracy. Phishing is a deceptive cybercrime tactic used by attackers to steal sensitive user information by impersonating trusted websites. This project addresses the growing threat of phishing by leveraging hybrid machine learning models that combine Logistic Regression (LR), Support Vector Classifier (SVC), and Decision Tree (DT) to enhance prediction reliability and performance.

Given the increasing number of phishing attacks and the vast volume of web traffic, manual detection is no longer practical, making an automated machine learning-based approach essential. This system utilizes a large dataset of labeled phishing and legitimate URLs and applies advanced preprocessing, feature selection via the Canopy method, and hyperparameter optimization using GridSearchCV. The proposed LSD hybrid model improves detection accuracy and reduces false positives significantly. Designed for real-time deployment, this system can be integrated into browsers, email clients, and corporate networks, offering robust protection against evolving phishing threats.

1.1 PROJECT PURPOSE

The primary purpose of this project is to build an automated and intelligent phishing detection system that enhances cybersecurity by accurately identifying malicious URLs. As online activities grow rapidly, phishing remains one of the most widespread and dangerous cyber threats, targeting users through deceptive websites to extract sensitive personal and financial information. Traditional detection techniques, such as blacklists and heuristic rules, are often ineffective against new or unknown phishing attempts and struggle to keep pace with the evolving threat landscape.

This project leverages a hybrid machine learning approach that combines multiple classification models—Logistic Regression (LR), Support Vector Classifier (SVC), and Decision Tree (DT)—to provide a highly accurate and scalable phishing detection mechanism. By analyzing subtle anomalies in URL structures and patterns, the system can detect phishing attempts in real time and with minimal human intervention. The ultimate goal is to reduce users’ exposure to phishing attacks, safeguard confidential data, and improve the overall resilience of online platforms against social engineering threats.

1.2 PROJECT FEATURES

This project incorporates several key features to improve the accuracy and efficiency of video classification and inappropriate content detection:

Automated Content Detection: The system employs EfficientNet-B7, a state-of-the-art convolutional neural network (CNN), to extract deep features from video frames. These features are then processed using BiLSTM (Bidirectional Long Short-Term Memory), which enhances the model’s ability to detect inappropriate patterns over sequential frames, ensuring a high level of detection accuracy.

Multi-Class Classification: Unlike traditional binary classification models that detect only safe or unsafe content, this project implements a multi-class classification approach. It categorizes flagged content into three distinct classes: Violence, Explicit Material, and Hate Speech, allowing for granular content filtering. This improves moderation efficiency, as flagged content can be reviewed and handled differently based on its severity.

Scalability & Real-Time Processing: The system is designed to handle large-scale video datasets efficiently, making it suitable for real-time implementation on content moderation platforms. It leverages GPU acceleration and cloud-based computing to process, analyze, and classify videos in real time. This ensures that harmful content is flagged before reaching viewers, significantly improving platform safety.

By integrating these features, this deep learning-based system provides a powerful, scalable, and automated solution for inappropriate content detection, ensuring better moderation, improved accuracy, and a safer viewing experience for users.

2. LITERATURE SURVEY

2. LITERATURE SURVEY

Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, Y. Si, F. Zhang, and J. S. Dong present **"Phishpedia,"** a pioneering logo-based phishing identification system characterized by exceptional accuracy and minimal runtime impact. This innovative deep learning system excels in precise phishing identification, particularly in logo recognition and matching, surpassing current methods. Its proficiency not only outperforms existing techniques but also uncovers previously unidentified phishing sites, thereby fortifying defense against phishing attacks. Phishpedia stands out as a unique and powerful tool for enhancing cybersecurity.
Cons: Phishpedia's performance relies on logo availability and quality on webpages. Ongoing updates and maintenance are essential for adapting to evolving phishing tactics.

Shirazi, Haynes, and Raya present a pioneering mobile-friendly phishing detection algorithm leveraging Artificial Neural Networks (ANNs) to scrutinize URL and HTML features. Their approach integrates cutting-edge deep transformers such as BERT, ELECTRA, RoBERTa, and MobileBERT for efficient learning from URL text. The innovative system facilitates swift training, seamless maintenance, and real-time deployment on mobile devices, addressing mobile security challenges effectively. This ensures competitive performance, establishing a robust defense against phishing threats while optimizing resource utilization for enhanced cybersecurity on mobile platforms.
Cons: Limited to URL detection and may miss complex phishing within legitimate pages. Depends on pre-trained transformers, subject to variations in availability and quality.

The thesis by A. Akanchha delves into the realm of SSL certificates within phishing sites, scrutinizing attacker attributes and crafting an auto-detection system reliant on SSL certificate features. Embracing Decision Tree machine learning for its transparency and efficacy, the research presents a pioneering SSL certificate-based phishing detection system, boasting impressive accuracy and a user-friendly Web API. The work underscores the need for future adaptations to combat evolving phishing techniques and ensure ongoing system updates, providing a comprehensive approach to cybersecurity challenges.
Cons: The system's effectiveness relies on SSL certificate attributes, which could be undermined if attackers develop new methods to mimic genuine certificates. The scalability of the system for managing numerous domains is not extensively discussed.

In the collaborative work of H. Shahriar and S. Nimmagadda, their chapter focuses on Network Intrusion Detection Systems (IDS) leveraging machine learning techniques such as Gaussian Naive Bayes, Logistic Regression, Decision Tree, and Neural Networks. The study aims to discern normal and anomalous network activities, particularly across TCP/IP layers. Notably, the Decision Tree exhibits commendable performance on public datasets, yet the authors underscore the imperative of real-world testing and scalability assessments for comprehensive validation of its accuracy and efficiency in practical network intrusion detection scenarios.

Cons: Evaluation may not reflect real-world conditions or evolving attacks. Algorithm choice not exhaustive; different methods may yield different results.

A. K. Dutta's innovative approach utilizes Random Forest, a supervised machine learning technique, to construct an advanced system dedicated to identifying phishing websites. The method involves meticulous analysis and selection of pertinent features that distinctly define phishing sites. Implemented as an intelligent browser extension, the system achieves an impressive 98.8% accuracy in detecting phishing sites, strategically addressing human vulnerabilities in online security. While occasionally presenting false alerts, the overarching goal is to significantly enhance online security measures and provide users with a robust defense against potential cyber threats.

Cons: Feature quality impacts adaptability to new phishing tactics. Potential for false results affects user trust.

In summary, the literature emphasizes the evolution from static, rule-based systems to adaptive, learning-based approaches in phishing detection. By combining multiple machine learning models and enhancing them with feature selection and optimization strategies, the proposed hybrid system aims to deliver a highly accurate and real-time solution for phishing URL detection. This contributes to strengthening cybersecurity defenses and reducing the risk of online fraud and data theft.

2.1 REVIEW OF RELATED WORK

Phishing remains a major concern in network security and on the internet. Researchers have explored numerous strategies to protect users from such cyber threats, including the use of machine learning, deep learning, blacklists, and whitelists. Typically, phishing detection systems are categorized into two main types: list-based and machine-learning-based systems. This section reviews previous research in both categories.

A. List-Based Phishing Identification System

List-based systems rely on whitelists and blacklists to determine whether websites are legitimate or phishing. Whitelist-based systems maintain a list of trusted websites, flagging any site not on this list as suspicious. For example, in one study, a system created a whitelist by monitoring and recording the IP addresses of websites with login interfaces. If a user encountered a new site not on the whitelist, the system would issue a warning. Another approach involved a system that automatically updated the whitelist periodically to alert users of potential phishing sites, achieving a true positive rate of 86.02% with a 1.48% false-negative rate

Blacklists, conversely, compile URLs known to be phishing sites from sources like user reports, spam detection systems, and third-party authorities. Systems using blacklists can prevent attackers by recording their IP addresses and URLs. To evade detection, attackers must use new URLs or IP addresses. While blacklist-based systems generally have lower false-positive rates, they struggle with zero-day attacks, where new phishing URLs are not yet listed. Their detection success rate is around 20% . Some notable blacklist-based security systems include PhishNet and the Google Safe Browsing API, which use approximate matching algorithms to detect malicious URLs. These systems require frequent updates due to the rapid proliferation of phishing URLs.

One study used a browser extension for phishing detection, achieving 85% accuracy . More recently, automatic phishing detection mechanisms have been proposed, such as a system that uses features of shortened URLs, achieving 92% accuracy. Delta Phish leverages multiple URL features for training supervised models, with accuracy rates above 70%. Another system, Phish-Safe, uses SVM and Naive Bayes for phishing detection, reaching 90% accuracy. Furthermore, an ensemble learning technique achieved 99% accuracy using only 11 features for email-based phishing detection. The Phi DMA approach, with five URL feature layers, attained 92% accuracy, while another study using SVM and six domain address features reported 95% accuracy. A system using typo-squatting and phoneme-based approaches achieved 99% accuracy.

B. Machine Learning-Based Identification System

Machine learning is becoming increasingly popular for detecting malicious websites by analyzing URLs. These systems require large datasets to train models on features associated with legitimate and phishing websites, enabling them to detect new attacks not listed on blacklists. The CANTINA system used text classification, extracting keywords and using TF-IDF to identify phishing sites via Google searches, but it was limited to English vocabulary. An improved version, CANTINA+, used 15 different HTML attributes and achieved 92% accuracy but had many false positives. PhishWHO employed a three-level approach, using keyword extraction and search engine results to identify phishing sites.

In 2011, researchers proposed a system using features like directory, file name, domain name, and special character counts, classifying sites with SVM . Other algorithms like adaptive regularization of weights outperformed others in offline mode. Another study proposed a multi-layer classification system for email messages, reducing false positives. Another study used URL-based features and rule mining with the Apriori algorithm, achieving 93% accuracy.

Modern research used a nonlinear regression approach with metaheuristic algorithms like harmony search and SVM, achieving 94.13% training and 92.80% test accuracy on 11,000 webpages. Another study used 209 word-based and 17 NLP features, improving accuracy by 7% in a subsequent study. A phishing detection system using dynamic self-structure-based neural networks was proposed, achieving high recognition with 1,400 records. A neural network-based approach classified phishing sites using 30 features, achieving 97.71% accuracy.

This review emphasizes the shift from rule-based and static detection methods to intelligent, adaptive, and hybrid machine learning systems. The proposed approach aims to build on past research while addressing critical gaps in accuracy, scalability, and phishing variant detection through an efficient and robust ensemble learning framework.

2.2 DEFINITION OF PROBLEM STATEMENT

The primary goal of this project is to develop a reliable, scalable, and intelligent phishing detection system based on hybrid machine learning techniques, capable of accurately identifying and classifying phishing URLs in real time. The system aims to overcome key challenges such as detecting zero-day phishing attacks, minimizing false positives, handling large-scale URL datasets, and improving adaptability to evolving phishing strategies. By integrating multiple classifiers in an ensemble framework and applying advanced feature selection and optimization techniques, the project seeks to enhance detection accuracy while maintaining system efficiency. This approach contributes to safeguarding user data, strengthening online security, and providing a proactive defense mechanism against phishing threats across digital platforms.

2.3 EXISTING SYSTEM

The existing phishing detection systems primarily rely on two approaches: **list-based methods** and **standalone machine learning classifiers**. List-based systems use predefined blacklists and whitelists to verify the legitimacy of URLs. Blacklists compile known phishing URLs collected from user reports and third-party databases, while whitelists maintain trusted domains. Tools such as Google Safe Browsing API and PhishNet utilize approximate matching algorithms to flag malicious URLs. Although these systems are easy to implement and fast in execution, they are highly dependent on regular updates and cannot detect new (zero-day) phishing attacks effectively.

In contrast, machine learning-based systems extract lexical and structural features from URLs and train classifiers like **Support Vector Machines (SVM)**, **Decision Trees (DT)**, **Naive Bayes**, and **Random Forest (RF)**. These models learn to distinguish phishing from legitimate URLs based on parameters such as URL length, domain structure, presence of special characters, and use of HTTPS. Some systems further employ ensemble models and email-based metadata to enhance detection accuracy. For instance, techniques using shortened URL analysis, typo-squatting detection, and keyword-based rule mining have achieved good accuracy in identifying phishing threats.

However, most of these approaches use a single classifier or basic ensemble techniques, lack robust feature selection, and fail to adapt effectively to rapidly evolving phishing tactics. While some systems incorporate supervised learning models for email phishing detection or browser extensions for real-time defense, they often compromise performance when scaled for large datasets or real-time applications.

Limitations of Existing System

Despite advancements in phishing URL classification, existing systems face several critical limitations:

- **Lack of adaptability to zero-day attacks**: List-based methods are ineffective against new phishing URLs that have not yet been reported or indexed. This reduces their reliability in fast-evolving phishing environments.
- **Limited feature learning**: Many systems use static, handcrafted features, making them less capable of recognizing complex or obfuscated phishing patterns. Deep contextual URL relationships often go unnoticed.
- **High false-positive/false-negative rates**: Single classifiers often misclassify borderline URLs, either flagging safe URLs as phishing or missing cleverly disguised malicious links.
- **Insufficient scalability and speed**: Traditional models may not perform well on high-volume data streams or real-time detection tasks. Without optimization, latency becomes an issue for large-scale deployment.
- **Overfitting on specific datasets**: Machine learning models trained on limited or outdated datasets may not generalize to newer types of phishing threats, leading to reduced accuracy when applied in real-world settings.

2.4 PROPOSED SYSTEM

The proposed system introduces a hybrid machine learning framework designed to effectively detect and classify phishing URLs with high accuracy and scalability. This approach combines the predictive strengths of three different classifiers—**Logistic Regression (LR)**, **Support Vector Classifier (SVC)**, and **Decision Tree (DT)**—into an ensemble model referred to as the **LSD model**. The system utilizes both **soft and hard voting mechanisms** to improve classification robustness and reduce the rate of false positives and false negatives.

The detection process begins with extensive preprocessing of a large phishing URL dataset comprising over **11,000 labeled URLs**, collected from trusted sources such as PhishTank and open repositories. Each URL is analyzed through lexical, domain-based, and structural features, such as the presence of IP addresses, URL length, use of HTTPS, and special characters.

To enhance the model's performance, **Canopy clustering** is employed for efficient feature selection, eliminating redundant or irrelevant attributes that may hinder detection accuracy. Further, **GridSearchCV** is applied to fine-tune hyperparameters, optimizing each model component in the hybrid ensemble. The final LSD model is evaluated using cross-validation techniques and benchmarked against other individual classifiers.

This proposed hybrid model is capable of real-time URL analysis and is suitable for integration into browsers, email clients, and corporate security systems to offer immediate protection against phishing attacks.

Advantages of the Proposed System:

The proposed hybrid system offers several enhancements over traditional phishing detection methods:

- **Higher Accuracy**:
  The combination of Logistic Regression, SVC, and Decision Tree in an ensemble format results in superior classification performance. The model significantly reduces both false positives and false negatives when compared to standalone classifiers.
- **Improved Generalization**:
  By using ensemble voting and diverse classifiers, the system reduces the risk of overfitting and improves its ability to detect phishing in unseen or obfuscated URLs.
- **Real-Time Capability**:
  The optimized model is designed to process and classify URLs quickly, making it suitable for real-time applications such as browser extensions and network firewalls.
- **Efficient Feature Selection**:
  The use of Canopy clustering ensures that only the most relevant features are retained, improving detection speed and reducing computational overhead.

- **Scalable and Lightweight**:
  Unlike complex deep learning models, the hybrid LSD system is less resource-intensive while maintaining high accuracy. It can be deployed on devices with limited computational power.
- **Explainability and Interpretability**:
  The inclusion of decision trees in the ensemble allows for greater transparency, making it easier for security analysts to understand the reasoning behind classification decisions.

2.5 OBJECTIVES

- **Develop an Automated Phishing Detection System** – Design and implement a hybrid machine learning model capable of automatically detecting and classifying phishing URLs with minimal human intervention.

- **Enhance Cybersecurity Defense Mechanisms** – Strengthen existing content security by leveraging machine learning to reduce dependency on static blacklist-based methods and manual URL verification.

- **Improve Detection Accuracy and Robustness** – Employ a hybrid LSD model (Logistic Regression, SVC, and Decision Tree) combined with feature selection and hyperparameter optimization techniques to achieve higher accuracy and reduce false positives.

- **Ensure Scalability and Real-Time Processing** – Optimize the system to process large volumes of URLs quickly, making it suitable for real-time deployment in web browsers, email clients, and network security platforms.

- **Safeguard User Data and Online Safety** – Prevent user exposure to phishing threats by identifying malicious URLs before they can cause harm, thereby ensuring safer browsing and data protection experiences.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements,

• Processor	:	Intel Core i3
• Hard disk	:	20GB.
• RAM	:	4GB.

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

• Operating system	:	Windows 10
• Language	:	Python
• Back-End	:	Django-ORM
• Frame Work	:	Tkinter
• Designing	:	HTML,CSS,Javascript
• Database	:	MySQL(WAMP Server)

3. SYSTEM ARCHITECTURE & DESIGN

3. SYSTEM ARCHITECTURE & DESIGN

Project architecture refers to the structural framework and design of a project, encompassing its components, interactions, and overall organization. It provides a clear blueprint for development, ensuring efficiency, scalability, and alignment with project goals. Effective architecture guides the project's lifecycle, from planning to execution, enhancing collaboration and reducing complexity.

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed Inappropriate Content Detection and Classification of YouTube Videos, starting from input to final prediction.

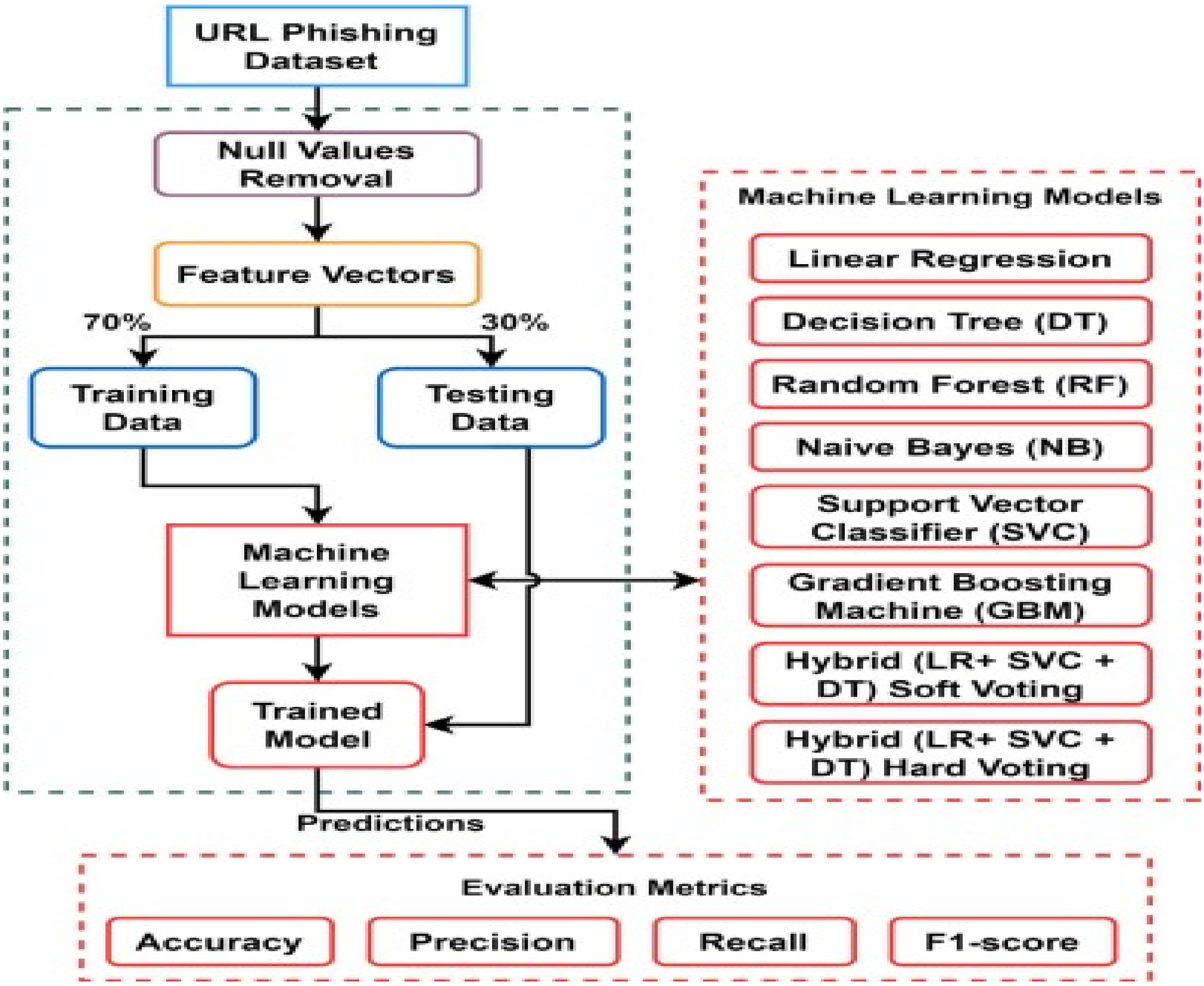


Figure 3.1: Project Architecture of Phishing Detection System Through Hybrid Machine Learning Based on URL.

3.2 DESCRIPTION

The above figure 3.1 illustrates about the flow of process regarding Phishing detection system by using various machine learning models. Below are the detailed description about each step.

Input Data: The system begins with a curated URL Phishing Dataset, which includes labeled instances of phishing and legitimate URLs. This dataset serves as the foundation for training and testing the machine learning models.

Data Preprocessing: The first step involves removing null or missing values to ensure data quality and consistency. This cleaned dataset is then transformed into feature vectors, capturing characteristics of URLs such as length, presence of special characters, domain information, and more.

Data Splitting: The feature vectors are split into 70% training data and 30% testing data, enabling model training and unbiased evaluation.

Model Training: Multiple machine learning algorithms are applied to the training data, including:

- Linear Regression (LR)
- Decision Tree (DT)
- Random Forest (RF)
- Naive Bayes (NB)
- Support Vector Classifier (SVC)
- Gradient Boosting Machine (GBM)

Additionally, two hybrid models are introduced:

- Soft Voting (LR + SVC + DT): A probabilistic ensemble method.
- Hard Voting (LR + SVC + DT): A majority rule-based ensemble method.

Prediction and Evaluation: Once trained, the models generate predictions on the test data. The performance is evaluated using the following evaluation metrics:

- Accuracy
- Precision
- Recall
- F1-Score

3.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data flows within a system, showcasing its processes, data stores, and external entities. It is a vital tool in system analysis and design, helping stakeholders visualize the movement of information, identify inefficiencies, and optimize workflows.

A Data Flow Diagram comprises Four primary elements:

- External Entities: Represent sources or destinations of data outside the system.
- Processes: Indicate transformations or operations performed on data.
- Data Flows: Depict the movement of data between components.
- Data Stores: Represent where data is stored within the system.

These components are represented using standardized symbols, such as circles for processes, arrows for data flows, rectangles for external entities, and open-ended rectangles for data stores.

Benefits:

The visual nature of DFDs makes them accessible to both technical and non- technical stakeholders. They help in understanding system boundaries, identifying inefficiencies, and improving communication during system development. Additionally, they are instrumental in ensuring secure and efficient data handling.

Applications:

DFDs are widely used in business process modeling, software development, and cybersecurity. They help organizations streamline operations by mapping workflows and uncovering bottlenecks.

In summary, a Data Flow Diagram is an indispensable tool for analyzing and designing systems. Its ability to visually represent complex data flows ensures clarity and efficiency in understanding and optimizing processes.

Levels of DFD:

DFDs are structured hierarchically:

- Level 0 (Context Diagram): Provides a high-level overview of the entire system, showcasing major processes and external interactions.
- Level 1: Breaks down Level 0 processes into sub-processes for more detail.
- Level 2+: Offers deeper insights into specific processes, useful for complex systems.

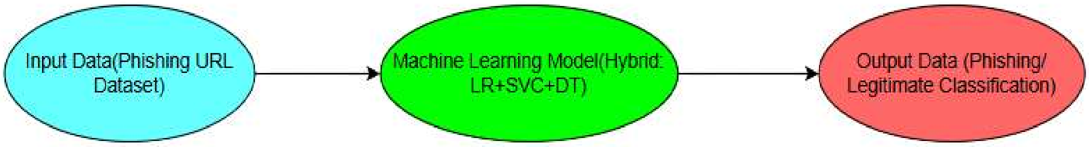

Input Data(Phishing URL Dataset)
Machine Learning Model(Hybrid: LR+SVC+DT)
Output Data (Phishing/ Legitimate Classification)

Figure 3.2: Dataflow Diagram of Phishing Detection System Through Hybrid Machine Learning Based on URL

The above data flow diagram of the Phishing Detection System in figure 3.2 illustrates the process of detecting malicious URLs using a hybrid machine learning approach. The process begins with the input data, which is a dataset containing various URLs labeled as either phishing or legitimate. This dataset is then fed into a hybrid machine learning model that combines three classification algorithms: Logistic Regression (LR), Support Vector Classifier (SVC), and Decision Tree (DT). Each of these algorithms contributes unique strengths—LR handles binary classification effectively, SVC works well with high-dimensional data, and DT captures complex decision-making patterns. By combining them, the hybrid model aims to improve overall prediction accuracy and reduce errors. The model analyzes the features of each URL and classifies it as either phishing or legitimate. The final output consists of the classification results, which can be used to enhance cybersecurity measures by warning users or blocking access to harmful sites.

4. IMPLEMENTATION

4. IMPLEMENTATION

The implementation phase of a project involves executing the planned strategies and tasks. It requires meticulous coordination, resource allocation, and monitoring to ensure that objectives are met efficiently. Effective implementation is crucial for achieving project goals and delivering expected outcomes within the set timeline and budget constraints.

4.1 ALGORITHMS USED

1. LSD

The below figure 4.1 is LSD model. The LSD (Logistic Regression, Support Vector Machine, Decision Tree) model with Hyperparameter GridCV is a hybrid classification model that combines the strengths of Logistic Regression, Support Vector Machine, and Decision Tree algorithms, enhancing accuracy and efficiency. GridCV systematically searches through hyperparameter combinations to optimize model performance, making it effective in various classification tasks.

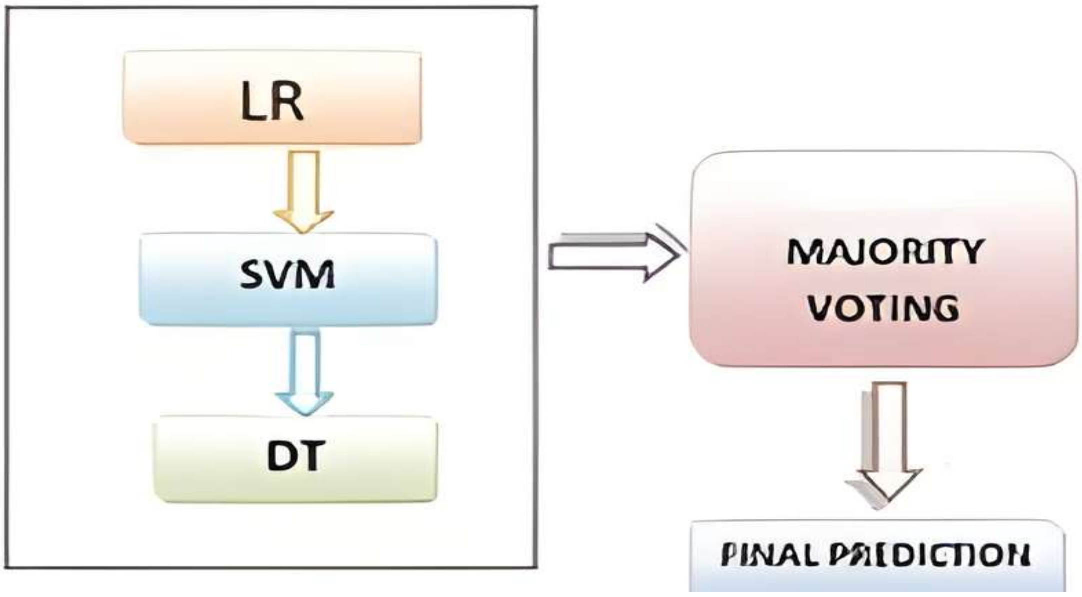


Figure 4.1: LSD

2. Hybrid LSD (Hard)

The below figure 4.2 shows Hybrid LSD (Hard) model that combines Logistic Regression, Support Vector Machine, and Decision Tree algorithms with a hard voting technique to make classification decisions. Each component model contributes its prediction, and the final decision is made by majority voting, enhancing accuracy and robustness in various classification tasks.

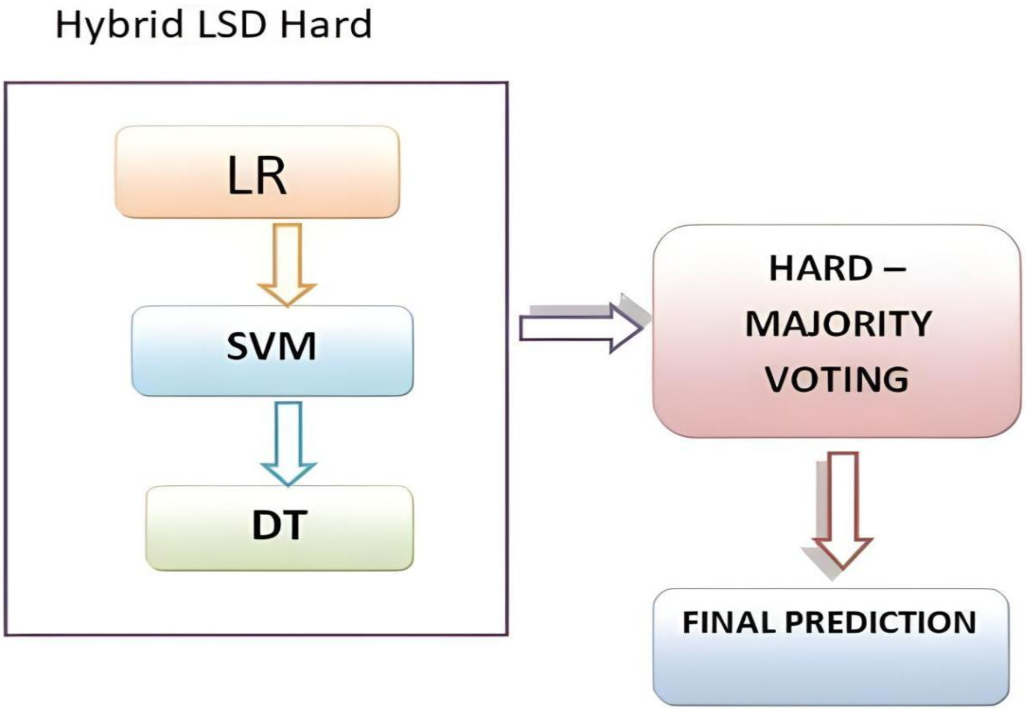


Figure 4.2: Hybrid LSD(Hard)

3. Hybrid LSD (Soft)

The below figure 4.3 shows Hybrid LSD (Soft) model that combines Logistic Regression, Support Vector Machine, and Decision Tree algorithms using soft voting to classify data. It leverages the strengths of each model to make predictions, with the flexibility to handle different types of data and improve accuracy in classification tasks.

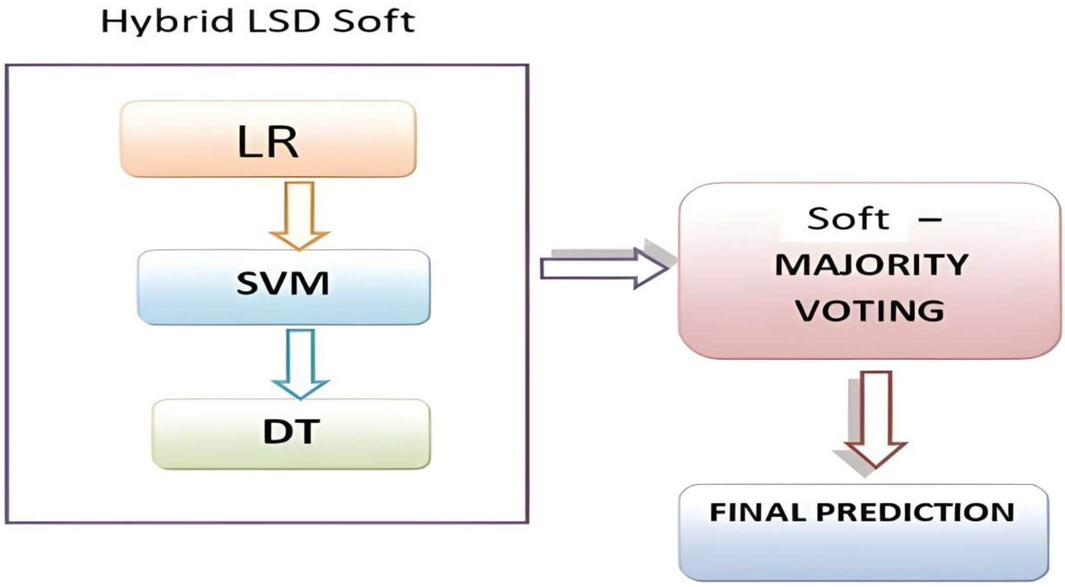


Figure 4.3: Hybrid LSD(Soft)

4. Gradient Boosting

The below figure 4.4 shows the Gradient Boosting which is an ensemble machine learning technique that sequentially builds a predictive model by combining the strengths of multiple weak learners, typically Decision Trees. It does so by focusing on the errors made by the previous models and adjusting its predictions to reduce those errors, ultimately creating a powerful and accurate predictive model that excels in various tasks, including regression and classification.

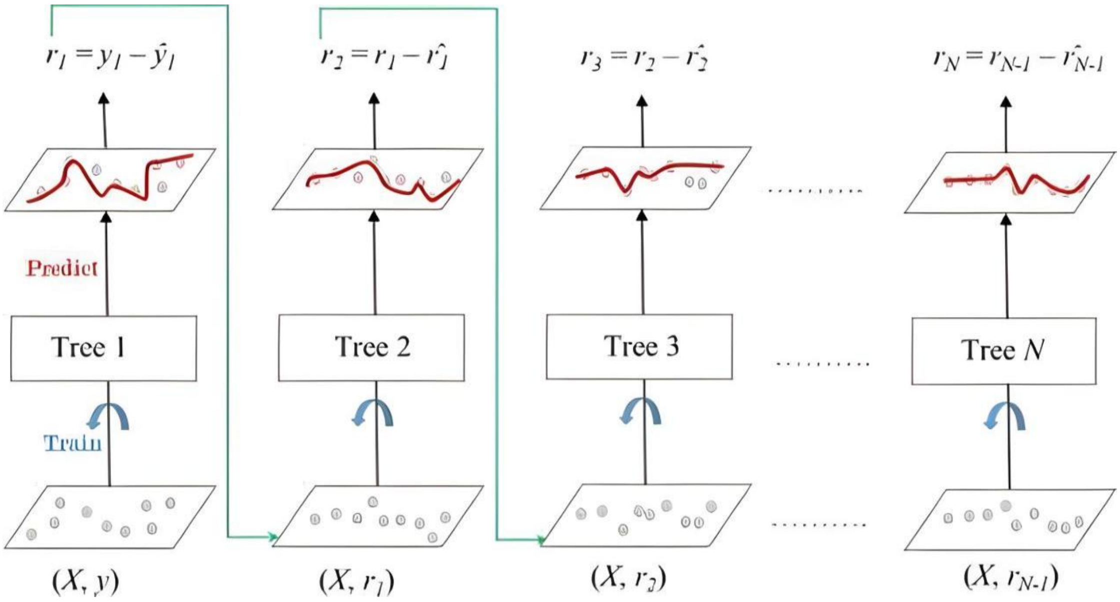


Figure 4.4: Gradient Boosting

5. Random Forest

The below figure 4.5 shows the Random Forest model which is an ensemble learning method that combines multiple Decision Trees to make predictions. It works by training a collection of Decision Trees on random subsets of the data and then averaging their predictions. This ensemble approach enhances accuracy, reduces overfitting, and provides robust performance for both classification and regression tasks.

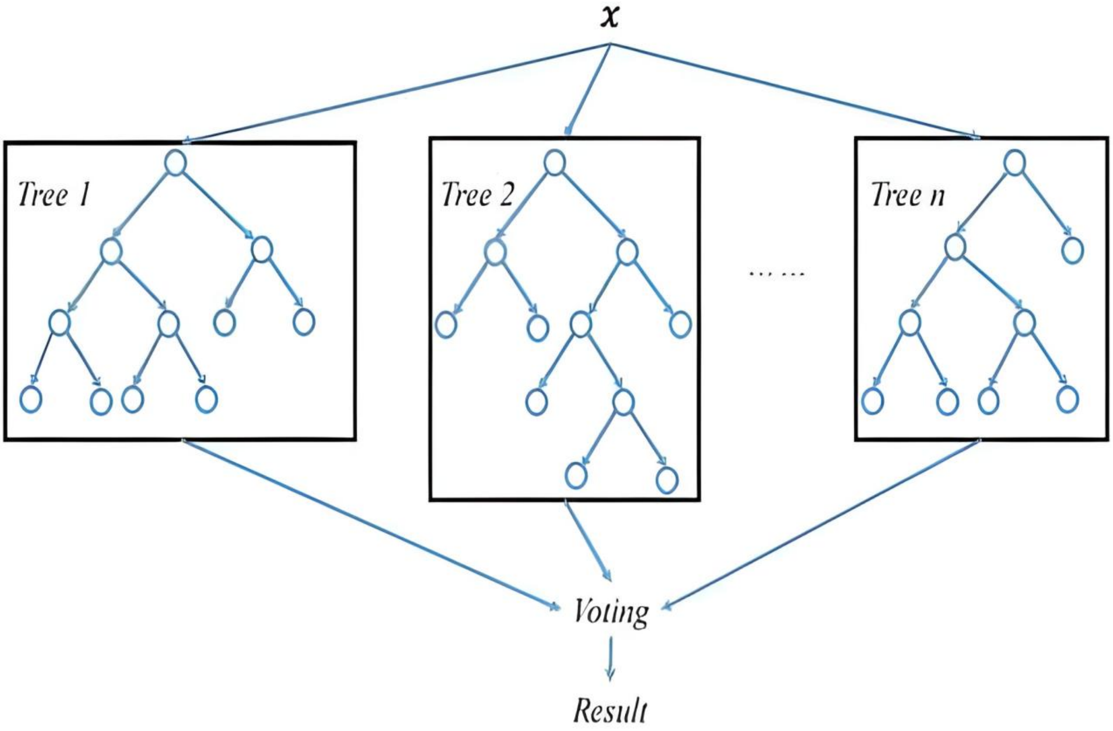


Figure 4.5: Random Forest

6. Decision Tree

A Decision Tree shown in below figure 4.6 is a machine learning model that makes decisions by recursively splitting data into subsets based on the most significant feature, aiming to classify or predict outcomes. It creates a tree-like structure where each node represents a feature and each branch represents a possible decision, making it interpretable and effective for various tasks.

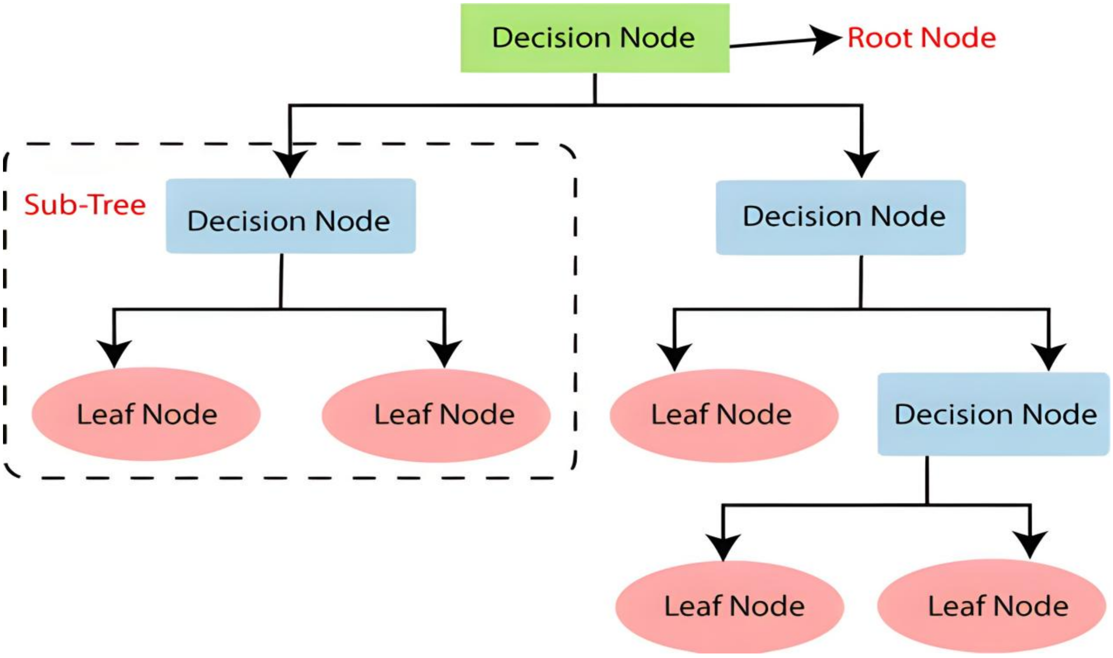


Figure 4.6: Decision Tree

7. Support Vector Classifier

A Support Vector Classifier (SVC) shown in below figure 4.7 is a machine learning model that finds the best possible boundary (hyperplane) to separate different classes of data while maximizing the margin between them. It identifies key support vectors to make accurate classifications, making it effective for both binary and multi-class classification tasks.

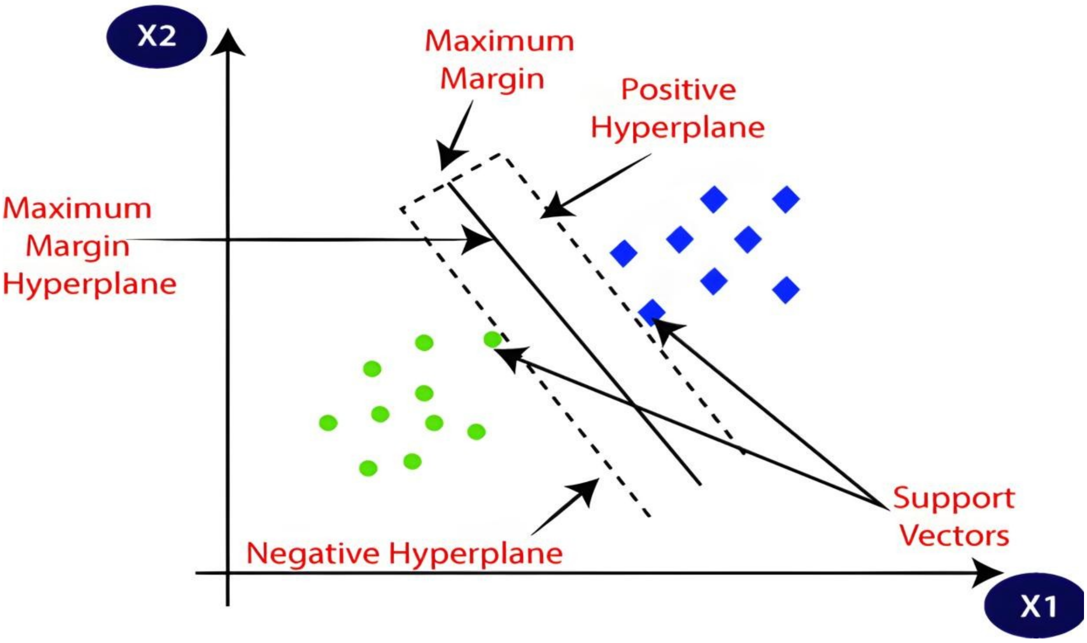


Figure 4.7: Support Vector Classifier

8. Logistic Regression

The below figure shows Logistic Regression model which is a classification algorithm that predicts the probability of an input belonging to a specific category. It employs the sigmoid function to map the input features to a probability score between 0 and 1, and a threshold is applied to classify the input into one of two or more categories based on this probability. The model learns coefficients during training to best fit the data and make accurate classifications.

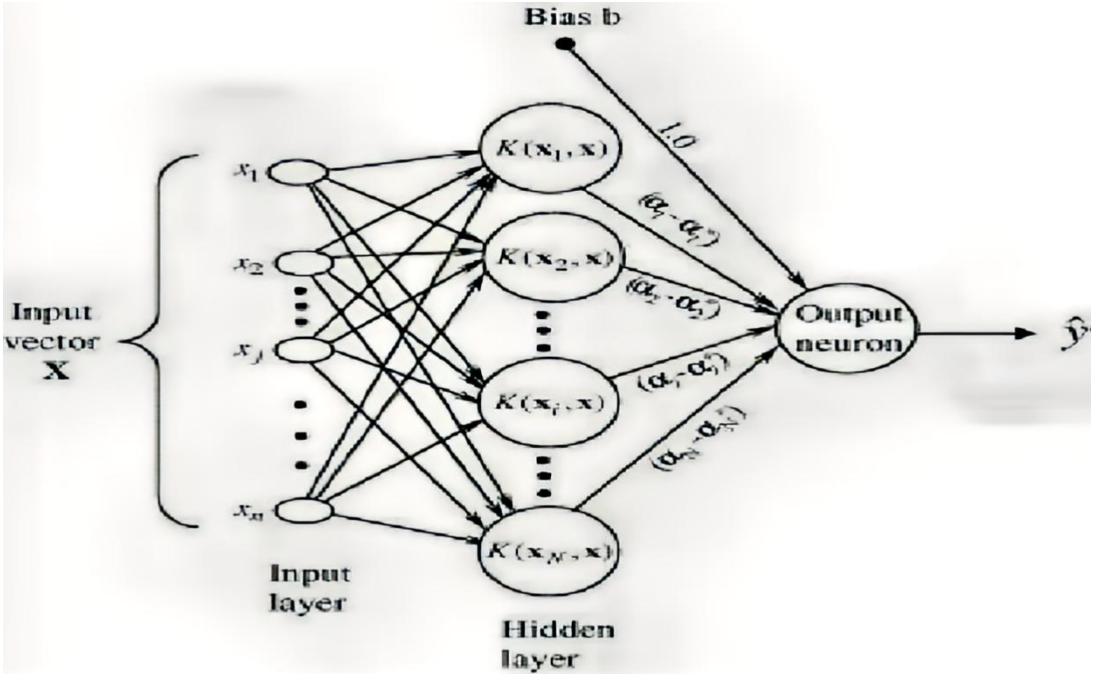

Bias b
1.0
Input vector X
x_1
x_2
x_j
x_n
Input layer
$K(\mathbf{x}_1, \mathbf{x})$
$K(\mathbf{x}_2, \mathbf{x})$
$K(\mathbf{x}_i, \mathbf{x})$
$K(\mathbf{x}_N, \mathbf{x})$
Hidden layer
$(\alpha_1 - \alpha_1^*)$
$(\alpha_2 - \alpha_2^*)$
$(\alpha_i - \alpha_i^*)$
$(\alpha_N - \alpha_N^*)$
Output neuron
$\hat{y}$

Figure 4.8: Logistic Regression

9. Naive Bayes

Naive Bayes shown in below figure 4.9 is a probabilistic classification algorithm that works by applying Bayes' theorem with the “naive” assumption of feature independence. It calculates the probability of a data point belonging to a particular class based on the probabilities of its individual features. Naive Bayes is particularly efficient for text classification tasks, spam detection, and other situations where feature independence is a reasonable approximation.

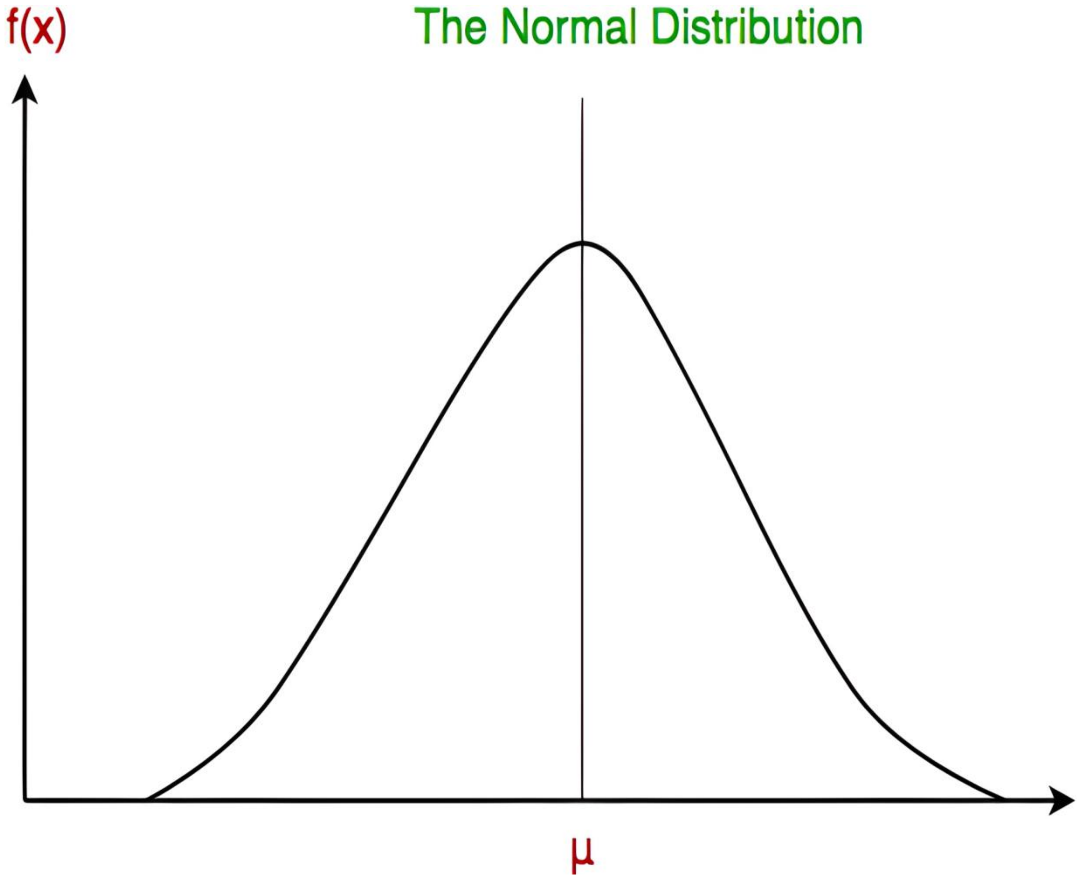
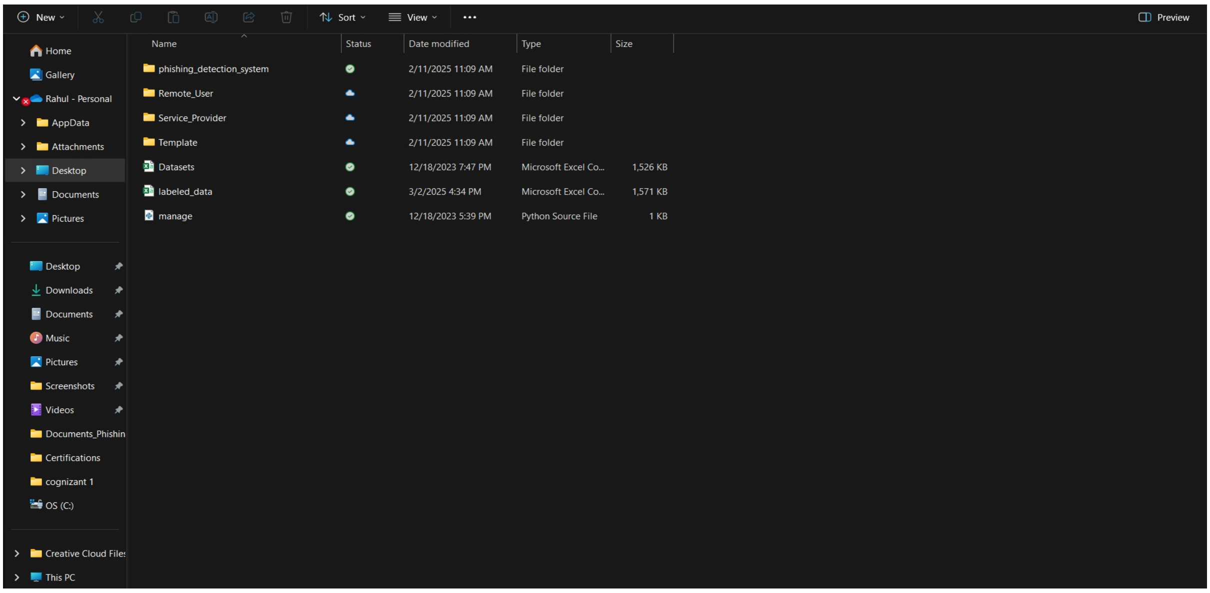


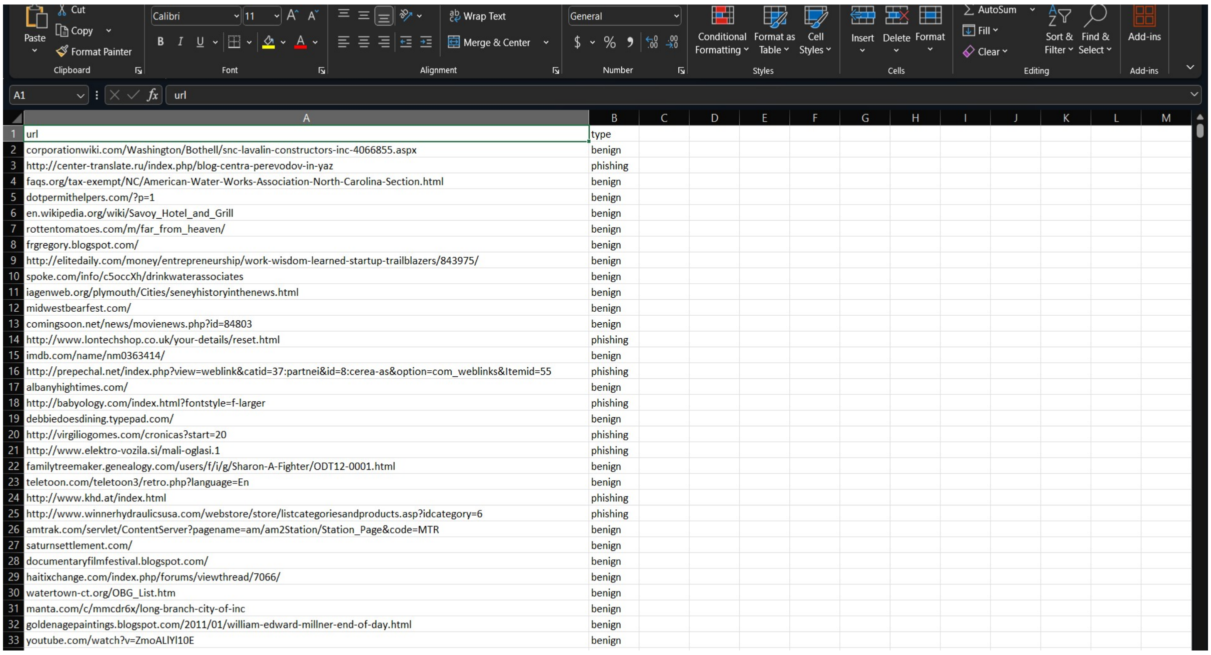
Figure 4.9: Naive Bayes

To train all algorithms we have used below dataset file consisting of training URLs shown in below figure 4.10 and figure 4.11.



Name	Status	Date modified	Type	Size
phishing_detection_system	✓	2/11/2025 11:09 AM	File folder	
Remote_User	☁	2/11/2025 11:09 AM	File folder	
Service_Provider	☁	2/11/2025 11:09 AM	File folder	
Template	☁	2/11/2025 11:09 AM	File folder	
Datasets	✓	12/18/2023 7:47 PM	Microsoft Excel Co...	1,526 KB
labeled_data	✓	3/2/2025 4:34 PM	Microsoft Excel Co...	1,571 KB
manage	✓	12/18/2023 5:39 PM	Python Source File	1 KB

Figure 4.10: Dataset directory with file named 'datasets' having all the training URLs



A1 url

	A	B
1	url	type
2	corporationwiki.com/Washington/Bothell/snc-lavalin-constructors-inc-4066855.aspx	benign
3	http://center-translate.ru/index.php/blog-centra-perevodov-in-yaz	phishing
4	faqs.org/tax-exempt/NC/American-Water-Works-Association-North-Carolina-Section.html	benign
5	dotpermithelpers.com/?p=1	benign
6	en.wikipedia.org/wiki/Savoy_Hotel_and_Grill	benign
7	rottentomatoes.com/m/far_from_heaven/	benign
8	frgregory.blogspot.com/	benign
9	http://elitedaily.com/money/entrepreneurship/work-wisdom-learned-startup-trailblazers/843975/	benign
10	spoke.com/info/c5occXh/drinkwaterassociates	benign
11	iagenweb.org/plymouth/Cities/seneyhistoryinthenews.html	benign
12	midwestbearfest.com/	benign
13	comingsoon.net/news/movienews.php?id=84803	benign
14	http://www.lontechshop.co.uk/your-details/reset.html	phishing
15	imdb.com/name/nm0363414/	benign
16	http://prepechal.net/index.php?view=weblink&catid=37:partnei&id=8:cerea-as&option=com_weblinks&Itemid=55	phishing
17	albanyhightimes.com/	benign
18	http://babyology.com/index.html?fontstyle=f-larger	phishing
19	debbiedoesdining.typepad.com/	benign
20	http://virgiliogomes.com/cronicas?start=20	phishing
21	http://www.elektro-vozila.si/mali-oglasi.1	phishing
22	familytreemaker.genealogy.com/users/f/i/g/Sharon-A-Fighter/ODT12-0001.html	benign
23	teletoon.com/teletoon3/retro.php?language=En	benign
24	http://www.khd.at/index.html	phishing
25	http://www.winnerhydraulicsusa.com/webstore/store/listcategoriesandproducts.asp?idcategory=6	phishing
26	amtrak.com/servlet/ContentServer?pagename=am/am2Station/Station_Page&code=MTR	benign
27	saturnsettlement.com/	benign
28	documentaryfilmfestival.blogspot.com/	benign
29	haitixchange.com/index.php/forums/viewthread/7066/	benign
30	watertown-ct.org/OBG_List.htm	benign
31	manta.com/c/mmcdr6x/long-branch-city-of-inc	benign
32	goldenagepaintings.blogspot.com/2011/01/william-edward-millner-end-of-day.html	benign
33	youtube.com/watch?v=ZmoALlYl10E	benign

Figure 4.11: Screenshot of the "datasets(Excel)" File Showing Sample URLs

To implement this project, we have designed the following modules:

1. **Upload Phishing URL Dataset**:

This module is used to upload the dataset containing phishing and legitimate URLs to the application for further preprocessing and model training**(Figure 4.1)**.

2. **Dataset Preprocessing**:

In this module, the dataset is preprocessed by cleaning the URLs, extracting important features (such as domain, subdomain, special characters, etc.), and labeling the data appropriately. The dataset is then split into training and testing subsets for model evaluation.

3. **Run Hybrid LSD Model with GridCV**:

This module executes the Logistic Regression, Support Vector Machine, and Decision Tree models using Grid Search Cross Validation to tune hyperparameters. The hybrid LSD classifier is trained and evaluated for its prediction accuracy on phishing detection.

4. **Run Stacking Classifier**:

In this module, the ensemble stacking classifier is applied by using Random Forest and MLP as base classifiers and LightGBM as a meta-classifier. This setup is used to enhance classification performance.

5. **Run Hard Voting Classifier**:

This module combines the predictions of Logistic Regression, SVM, and Decision Tree using a hard voting technique to classify URLs as phishing or legitimate based on majority vote.

6. **Run Soft Voting Classifier**:

In this module, soft voting is implemented by averaging the predicted probabilities from the Logistic Regression, SVM, and Decision Tree models to improve the prediction accuracy of phishing detection.

7. **Run Gradient Boosting and Random Forest Models**:

These modules utilize the Gradient Boosting and Random Forest algorithms to classify URLs. The models are trained using extracted features and evaluated for their classification performance.

8. **Comparison Graph Module**:

This module visualizes the prediction accuracy of all models (LSD, Voting Classifiers, Gradient Boosting, and Random Forest) using comparison graphs to identify the best-performing algorithm.

9. **Phishing Detection on Input URL**:

This final module allows the user to input any URL, and based on the trained hybrid and ensemble models, the system predicts whether the URL is **Phishing** or **Legitimate**.

4.2 SAMPLE CODE

```
from django.db.models import Count
from django.db.models import Q
from django.shortcuts import render, redirect, get_object_or_404
import pandas as pd

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
from sklearn.ensemble import VotingClassifier
from Remote_User.models import
ClientRegister_Model,phishing_detection,detection_accuracy,detection_ratio

def login(request):
        if request.method == "POST" and 'submit1' in request.POST:
            username = request.POST.get('username')
            password = request.POST.get('password')
                try:
                    enter =
        ClientRegister_Model.objects.get(username=username,password=password)
                    request.session["userid"] = enter.id
                    return redirect('ViewYourProfile')
                except:
                    pass
        return render(request,'RUser/login.html')

def Add_DataSet_Details(request):
        return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": ''})

def Register1(request):

        if request.method == "POST":
            username = request.POST.get('username')
            email = request.POST.get('email')
            password = request.POST.get('password')
            phoneno = request.POST.get('phoneno')
            country = request.POST.get('country')
            state = request.POST.get('state')
            city = request.POST.get('city')
            ClientRegister_Model.objects.create(username=username, email=email,
        password=password, phoneno=phoneno,country=country, state=state, city=city)
        return render(request, 'RUser/Register1.html')
        else:
        return render(request,'RUser/Register1.html')

def ViewYourProfile(request):
        userid = request.session['userid']
        obj = ClientRegister_Model.objects.get(id= userid)
        return render(request,'RUser/ViewYourProfile.html',{'object':obj})
```

```
def Predict_URL_Type(request):
        if request.method == "POST":

            if request.method == "POST":
                url = request.POST.get('url')

            data = pd.read_csv("Datasets.csv",encoding='latin-1')

            def apply_results(results):
                if (results == "benign"):
                    return 0
                elif (results == "phishing"):
                    return 1

            data['Results'] = data['type'].apply(apply_results)

            x = data['url']
            y = data['Results']

            cv = CountVectorizer(lowercase=False, strip_accents='unicode', ngram_range=(1,
        1))

            print(x)
            print("Y")
            print(y)

            x = cv.fit_transform(x)

            models = []
            from sklearn.model_selection import train_test_split
            X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
            X_train.shape, X_test.shape, y_train.shape

            print("SVM")
            from sklearn import svm

            lin_clf = svm.LinearSVC()
            lin_clf.fit(X_train, y_train)
            predict_svm = lin_clf.predict(X_test)
            svm_acc = accuracy_score(y_test, predict_svm) * 100
            print("ACCURACY")
            print(svm_acc)
            print("CLASSIFICATION REPORT")
            print(classification_report(y_test, predict_svm))
            print("CONFUSION MATRIX")
            print(confusion_matrix(y_test, predict_svm))
                models.append(('svm', lin_clf))

            print("Random Forest Classifier")
            from sklearn.ensemble import RandomForestClassifier
            rf_clf = RandomForestClassifier()
```

```
    rf_clf.fit(X_train, y_train)
    rfpredict = rf_clf.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, rfpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, rfpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, rfpredict))
    models.append(('RandomForestClassifier', rf_clf))


    classifier = VotingClassifier(models)
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)

    url1 = [url]
    vector1 = cv.transform(url1).toarray()
    predict_text = classifier.predict(vector1)

    pred = str(predict_text).replace("[", "")
    pred1 = str(pred.replace("]", ""))

    prediction = int(pred1)

    if prediction == 0:
        val = 'Normal URL'
    elif prediction == 1:
        val = 'Phishing URL'

    print(prediction)
    print(val)

    phishing_detection.objects.create(url=url,Prediction=val)

    return render(request, 'RUser/Predict_URL_Type.html',{'objs': val})
return render(request, 'RUser/Predict_URL_Type.html')
```

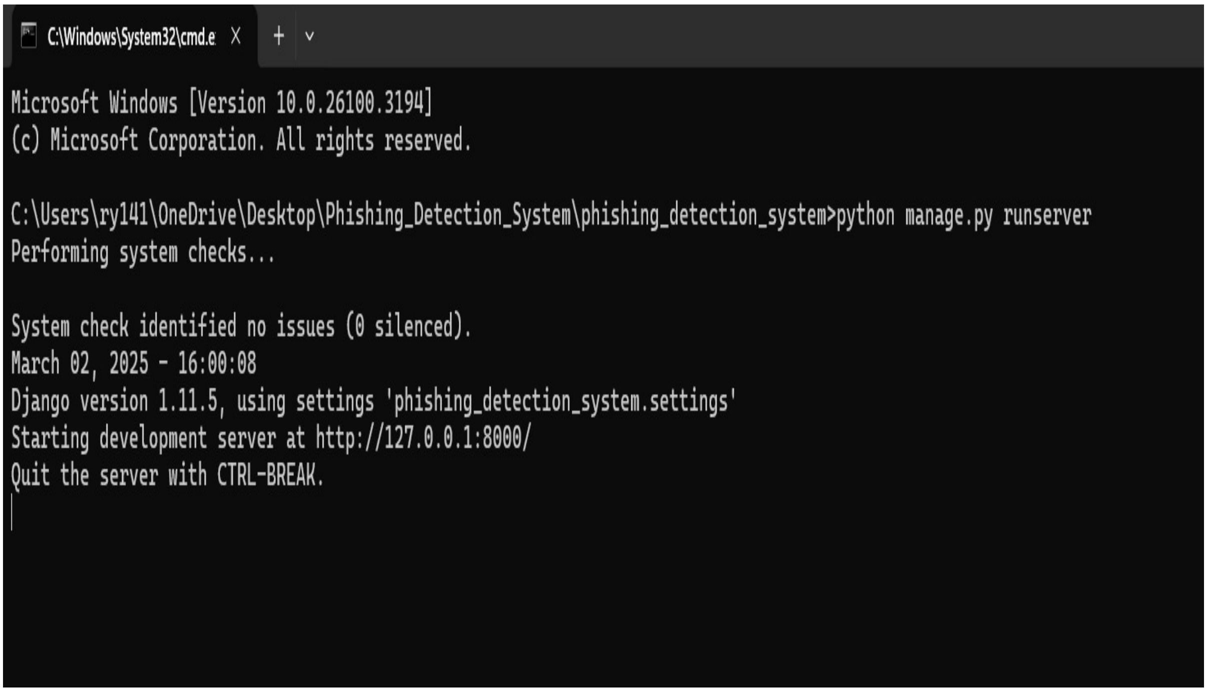
5. RESULTS & DISCUSSION

5. RESULTS & DISCUSSION

The following screenshots showcase the results of our project, highlighting key features and functionalities. These visual representations provide a clear overview of how the system performs under various conditions, demonstrating its effectiveness and user interface. The screenshots serve as a visual aid to support the project's technical and operational achievements.

5.1 Command Prompt :

This below figure 5.1 shows the command-line interface where the Django development server is initiated using the command python manage.py runserver. Upon execution, the terminal generates a local URL, which acts as the gateway to the application. By clicking on this URL, the user is redirected to the homepage of the phishing detection system. This step is crucial to launch the web-based interface for user interaction. It confirms successful deployment of the backend server and ensures the system is ready for real-time access.

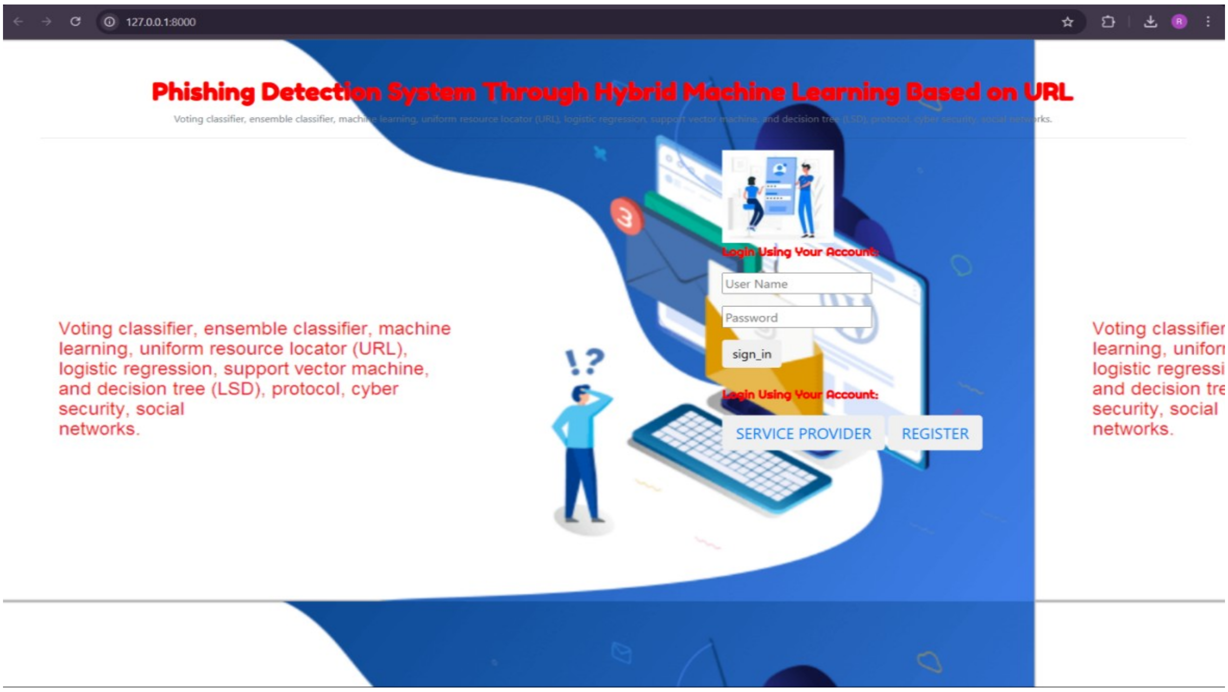


```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ry141\OneDrive\Desktop\Phishing_Detection_System\phishing_detection_system>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
March 02, 2025 - 16:00:08
Django version 1.11.5, using settings 'phishing_detection_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figure 5.1 : command prompt to redirect to our web page

5.2 Main interface/home page :

The home page as shown in below figure 5.2 serves as the starting point of the phishing detection system. It presents options for the user to either log in to their existing account or register as a new user. The interface is designed with simplicity and user-friendliness in mind. This screen connects the frontend with backend services and ensures proper navigation. The user cannot proceed to use detection features without accessing this page.



127.0.0.1:8000

Phishing Detection System Through Hybrid Machine Learning Based on URL

Voting classifier, ensemble classifier, machine learning, uniform resource locator (URL), logistic regression, support vector machine, and decision tree (LSD), protocol, cyber security, social networks.

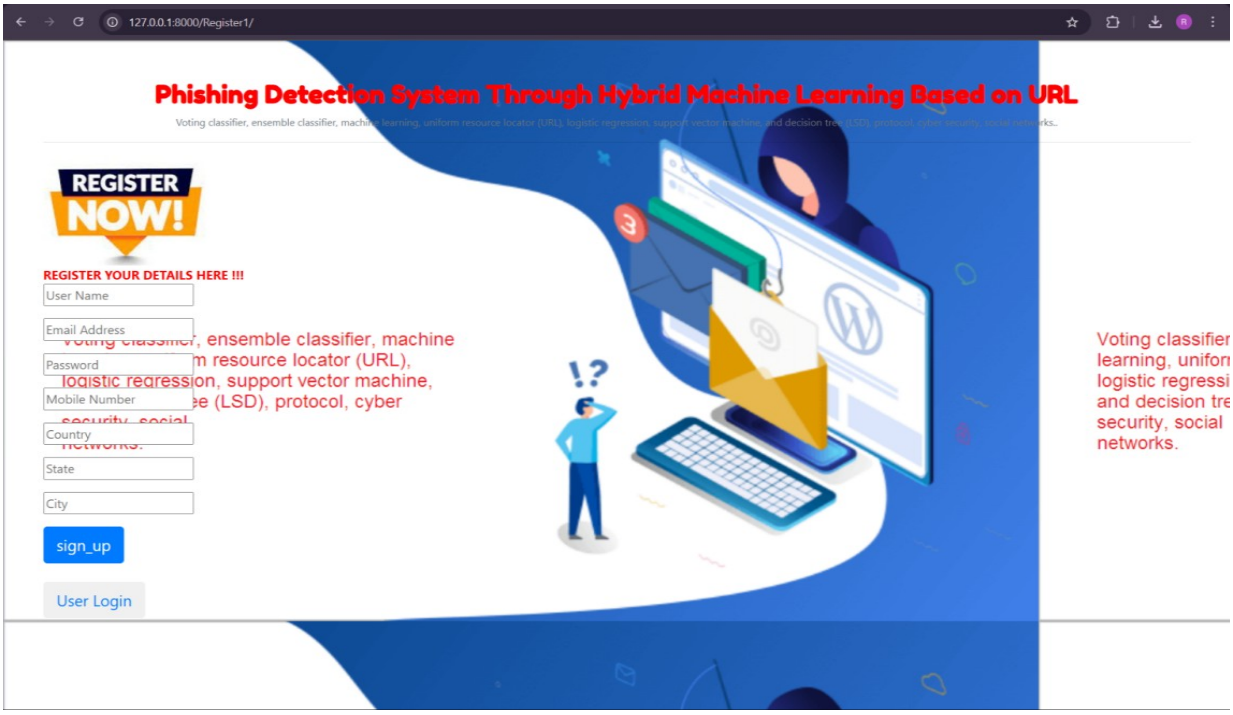
Voting classifier, ensemble classifier, machine learning, uniform resource locator (URL), logistic regression, support vector machine, and decision tree (LSD), protocol, cyber security, social networks.

Voting classifier
learning, unifor
logistic regressi
and decision tre
security, social
networks.

Figure 5.2 : Home page of Phishing detection System Through Hybrid Machine Learning Based on URL.

5.3 Registration Page :

The below figure 5.3 displays the registration form that captures essential user details such as username, password, email ID, contact number, and location information (country, state, city). It helps in maintaining user-specific logs and tracking activities. After successful registration, the user gains access to all the functionalities provided by the phishing detection system. The form also ensures that only authorized users can utilize the service.



127.0.0.1:8000/Register1/

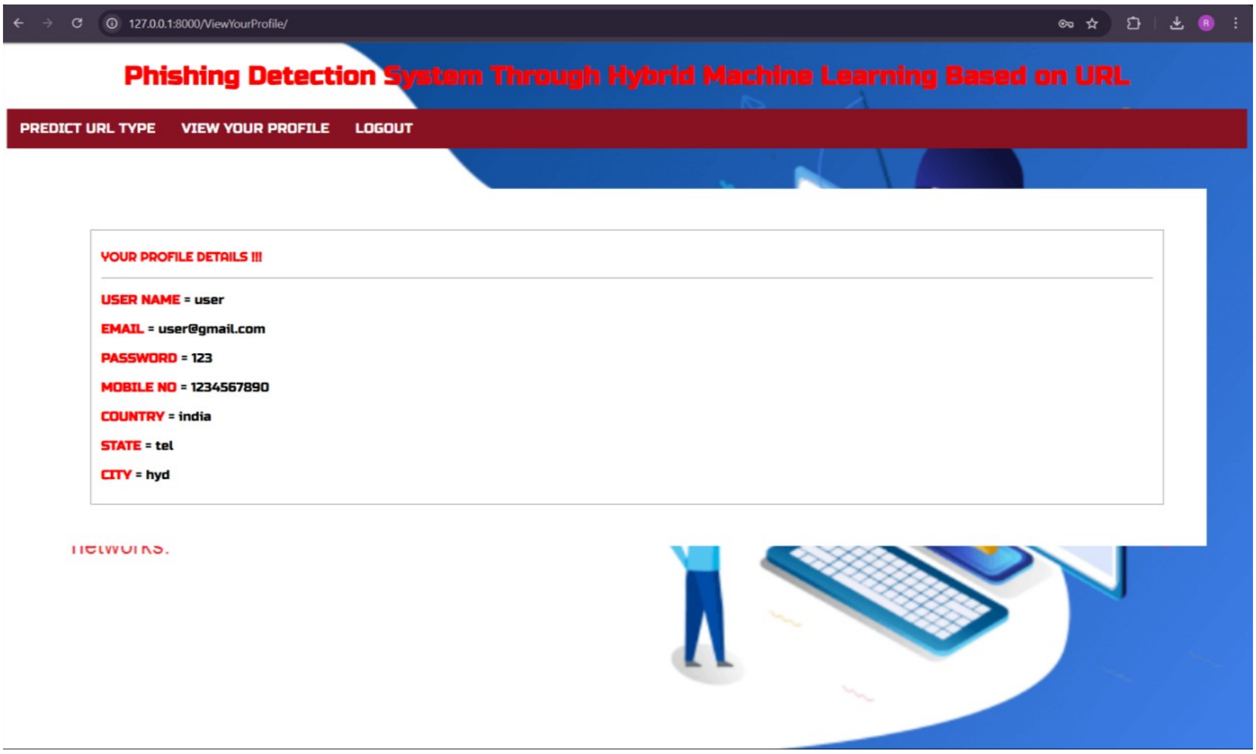
Phishing Detection System Through Hybrid Machine Learning Based on URL

Voting classifier, ensemble classifier, machine learning, uniform resource locator (URL), logistic regression, support vector machine, and decision tree (LSD), protocol, cyber security, social networks..

Figure 5.3 : Registration Page for Phishing Detection System Through Hybrid Machine Learning Based on URL.

5.4 Profile Details :

After logging in, the user is redirected to their profile page as shown in below figure 5.4, which displays all the personal information entered during registration. This ensures session integrity and helps the user confirm their account credentials. The profile page helps maintain a personalized experience and can be useful for logging activities or linking detection history to specific users. It also enables further user verification and account management features.



127.0.0.1:8000/ViewYourProfile/
Phishing Detection System Through Hybrid Machine Learning Based on URL
PREDICT URL TYPE
VIEW YOUR PROFILE
LOGOUT
networks.

YOUR PROFILE DETAILS !!!

USER NAME = user

EMAIL = user@gmail.com

PASSWORD = 123

MOBILE NO = 1234567890

COUNTRY = india

STATE = tel

CITY = hyd

Figure 5.4 : Profile details of Phishing detection System Through Hybrid Machine Learning Based on URL.

5.5 Predict URL Type:

The below figure 5.5 shows the screen which allows users to enter any URL they wish to test for phishing activity. It serves as the main interface for the phishing detection function. Upon submitting the URL, the backend model processes it and checks for suspicious patterns or characteristics. This module is powered by the trained hybrid machine learning model (LSD). The user receives instant feedback on whether the URL is safe or potentially harmful.

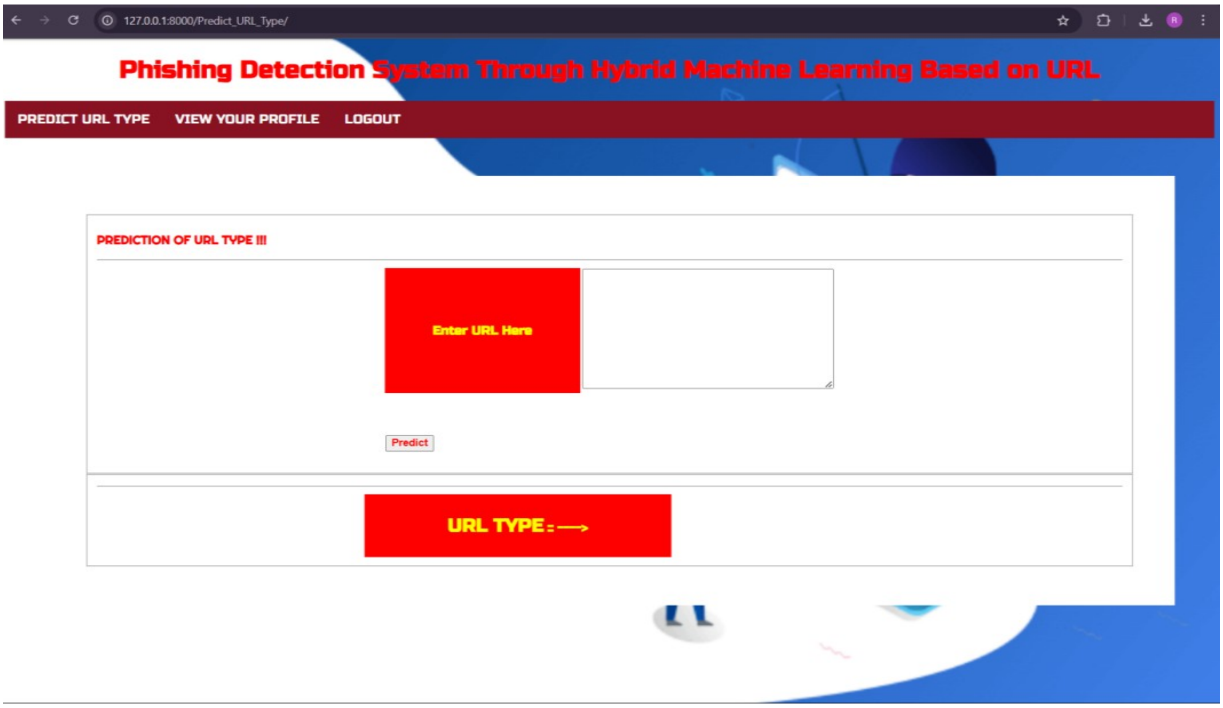
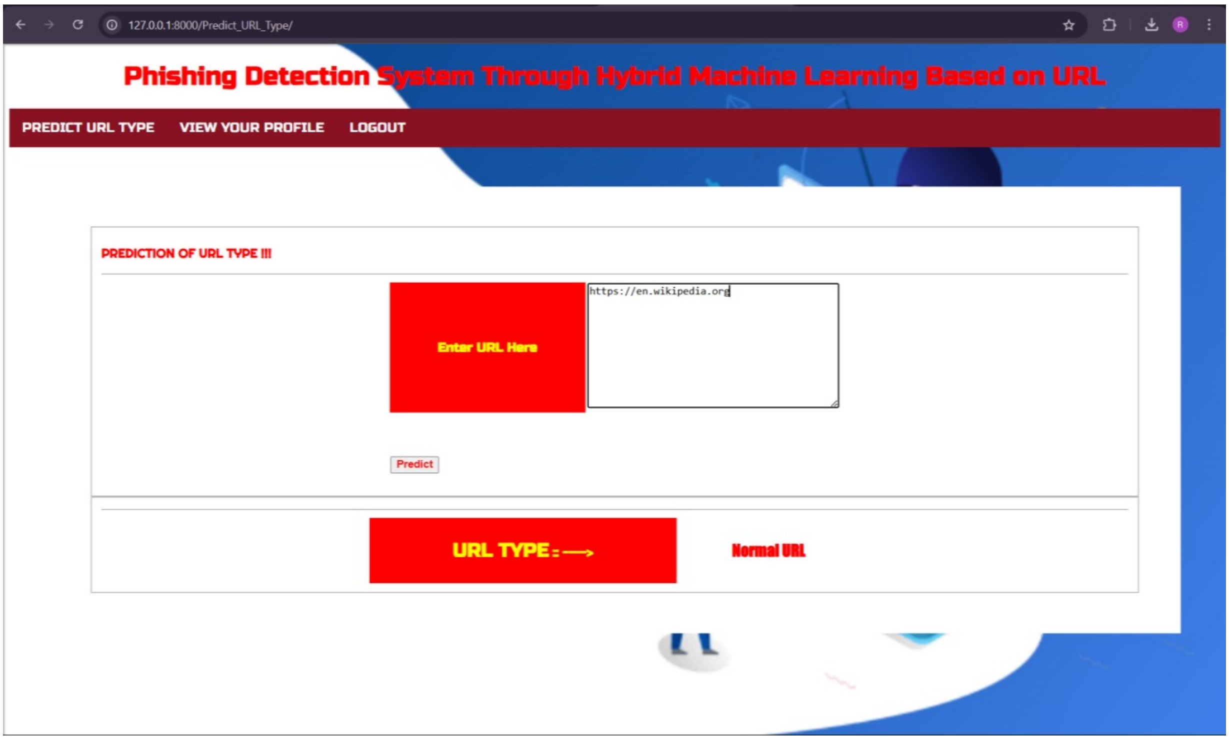

127.0.0.1:8000/Predict_URL_Type/
Phishing Detection System Through Hybrid Machine Learning Based on URL
PREDICT URL TYPE
VIEW YOUR PROFILE
LOGOUT
PREDICTION OF URL TYPE !!!
Enter URL Here
Predict
URL TYPE:⟶

Figure 5.5 : Prediction of URL Type of Phishing detection System Through Hybrid Machine Learning Based on URL.

5.6 Results

This below figure 5.6 shows the output result for the URL entered in the prediction page. Based on the model's evaluation, the system labels the input as either a **“Normal URL”** or **“Phishing URL.”** The decision is derived using voting techniques among Logistic Regression, Support Vector Classifier, and Decision Tree models. This module confirms the system’s purpose—helping users avoid malicious links effectively and in real-time.



127.0.0.1:8000/Predict_URL_Type/

Phishing Detection System Through Hybrid Machine Learning Based on URL

PREDICTION OF URL TYPE !!!

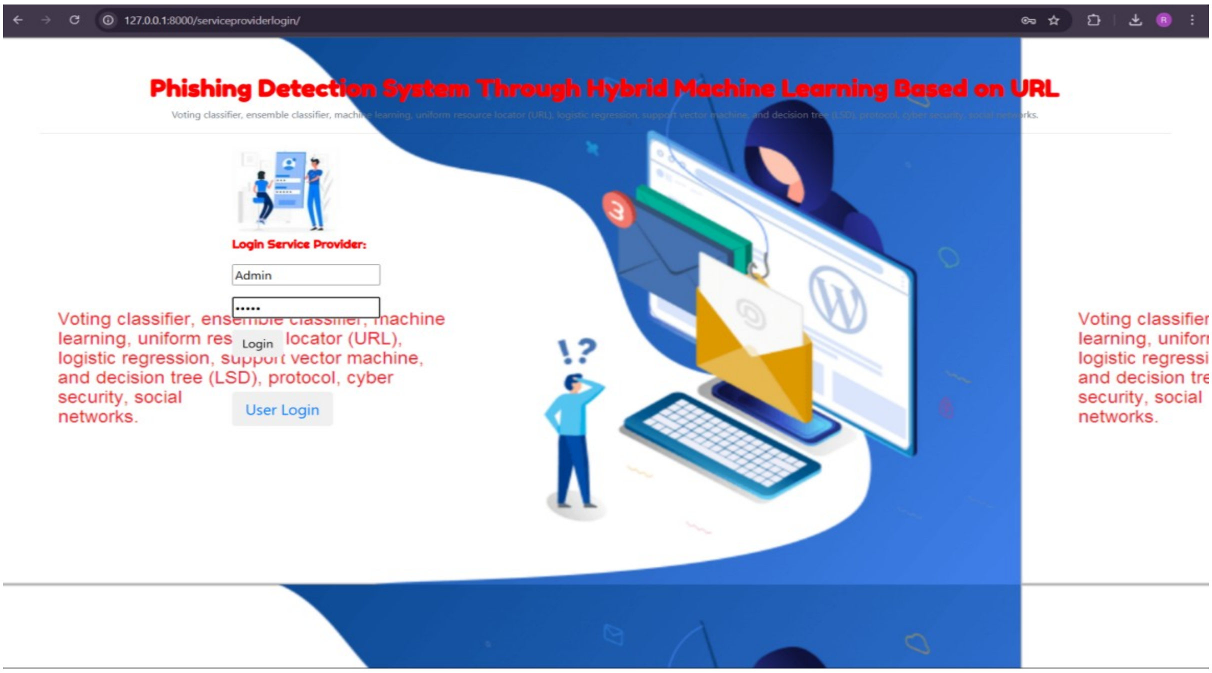
Enter URL Here: https://en.wikipedia.org

URL TYPE:⟶ Normal URL

Figure 5.6 : Result for Phishing detection System Through Hybrid Machine Learning Based on URL.

5.7 Login Page for Service Provider

The below figure 5.7 displays the login interface meant for the admin or service provider. With valid credentials, the service provider gains access to backend functionalities such as viewing user records, prediction logs, and system stats. This layer ensures system security and management control. It acts as a gateway for administrative operations, such as monitoring and managing datasets and model performance.


127.0.0.1:8000/serviceproviderlogin/

Phishing Detection System Through Hybrid Machine Learning Based on URL

Voting classifier, ensemble classifier, machine learning, uniform resource locator (URL), logistic regression, support vector machine, and decision tree (LSD), protocol, cyber security, social networks.

Login Service Provider:

Admin

•••••

Login

User Login

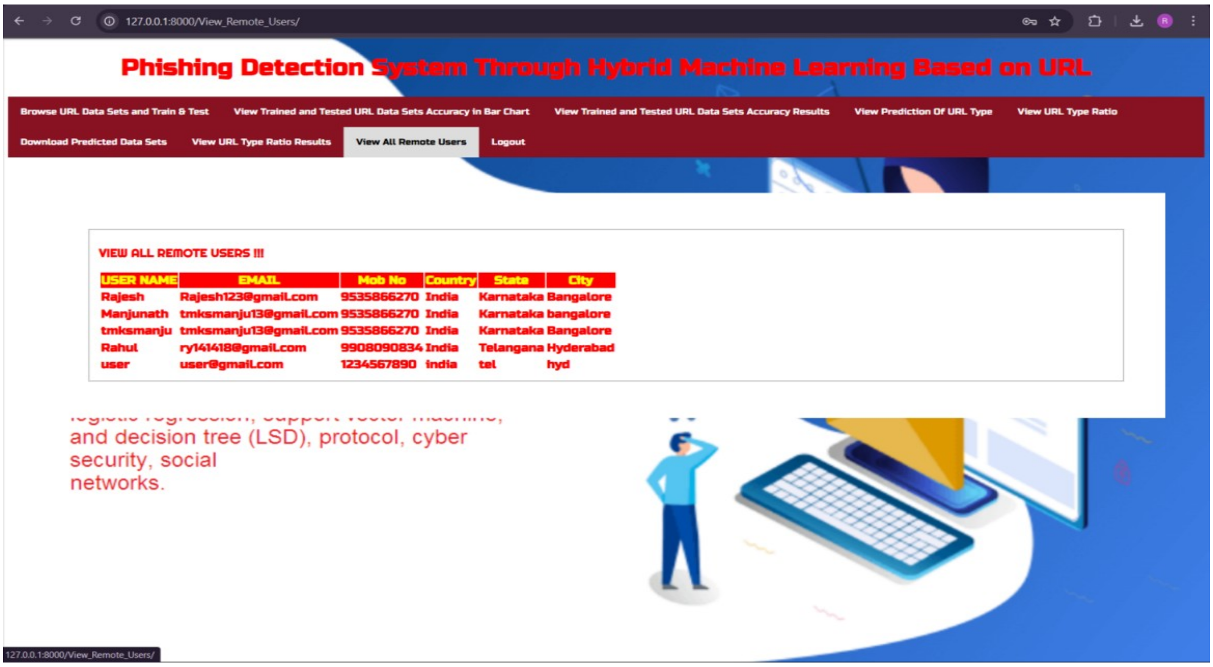
Voting classifier, ensemble classifier, machine learning, uniform resource locator (URL), logistic regression, support vector machine, and decision tree (LSD), protocol, cyber security, social networks.

Voting classifier
learning, unifor
logistic regressi
and decision tre
security, social
networks.

Figure 5.7 : Login Page for Service Provider in Phishing detection System Through Hybrid Machine Learning Based on URL.

5.8 Details of All Registered User

In the below figure 5.8 the system displays a list of all users who have registered on the platform. Each entry includes username, contact, and location information. This feature is typically accessible to service providers for administrative review. It helps in managing user base, tracking suspicious activities, and ensuring the platform is being used responsibly. It's a vital part of user management and system auditing.


127.0.0.1:8000/View_Remote_Users/
Phishing Detection System Through Hybrid Machine Learning Based on URL
Browse URL Data Sets and Train & Test
View Trained and Tested URL Data Sets Accuracy in Bar Chart
View Trained and Tested URL Data Sets Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout

VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	Mob No	Country	State	City
Rajesh	Rajesh123@gmail.com	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	9535866270	India	Karnataka	bangalore
tmksmanju	tmksmanju13@gmail.com	9535866270	India	Karnataka	Bangalore
Rahul	ry141418@gmail.com	9908090834	India	Telangana	Hyderabad
user	user@gmail.com	1234567890	india	tel	hyd

and decision tree (LSD), protocol, cyber security, social networks.

Figure 5.8 : Details of all Registered users for Phishing detection System Through Hybrid Machine Learning Based on URL.

5.9 Accuracy of Trained and Tested URL Data sets in Bar graph representation:

The below figure consists of a bar graph which compares the accuracy performance of different algorithms used in the system such as SVM, Decision Tree, Random Forest, and Naive Bayes. The chart highlights that SVM and Random Forest achieved the highest accuracy (95.93%) while Naive Bayes had the lowest (93.43%). This visual makes it easy to identify which model performs best for phishing detection. It supports model selection and optimization decisions.

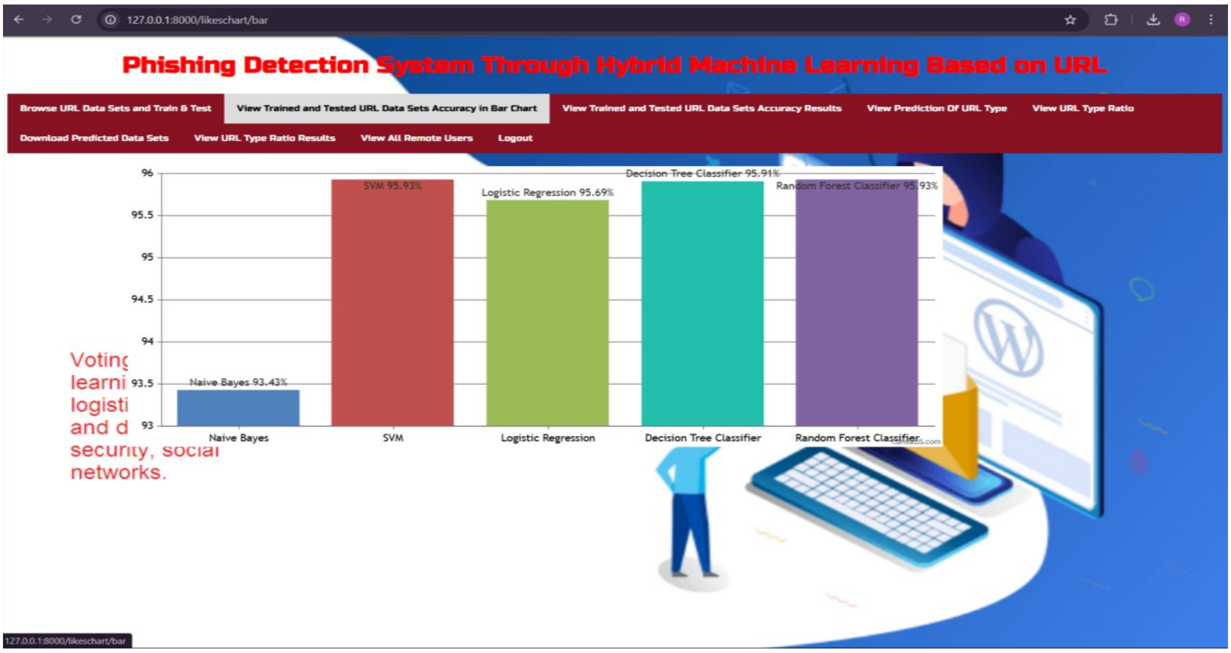

127.0.0.1:8000/likeschart/bar
Phishing Detection System Through Hybrid Machine Learning Based on URL
Browse URL Data Sets and Train & Test
View Trained and Tested URL Data Sets Accuracy in Bar Chart
View Trained and Tested URL Data Sets Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout
96
95.5
95
94.5
94
93.5
93
Naive Bayes 93.43%
SVM 95.93%
Logistic Regression 95.69%
Decision Tree Classifier 95.91%
Random Forest Classifier 95.93%
Naive Bayes
SVM
Logistic Regression
Decision Tree Classifier
Random Forest Classifier
Voting
learni
logisti
and d
security, social
networks.
127.0.0.1:8000/likeschart/bar

Figure 5.9 : Accuracy of Trained and Tested URL Data sets in Bar graph representation for Phishing detection System Through Hybrid Machine Learning Based on URL.

5.10 Accuracy of Trained and Tested URL data sets in Line Chart Representation:

The below figure 5.10 presents the same accuracy comparison shown in the bar chart, but in the form of a line graph. Each point on the graph corresponds to a specific model's performance. The line format helps visualize trends in accuracy and makes it easier to spot minor differences. This is useful for performance monitoring and validating the choice of the hybrid model.

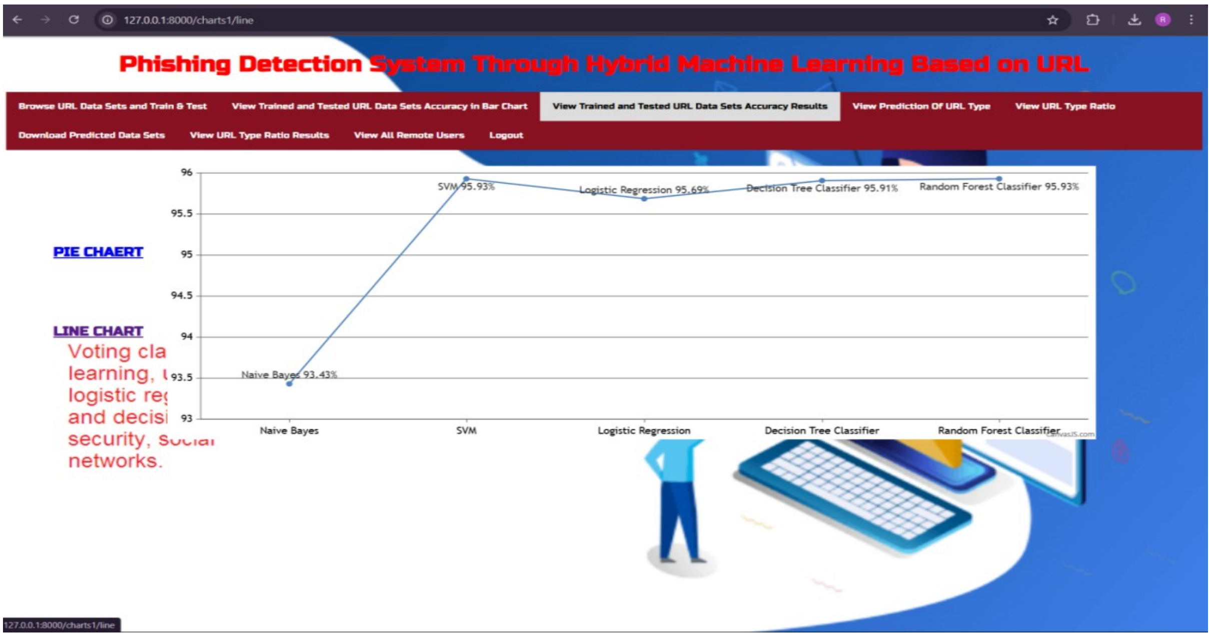

127.0.0.1:8000/charts1/line
Phishing Detection System Through Hybrid Machine Learning Based on URL
Browse URL Data Sets and Train & Test
View Trained and Tested URL Data Sets Accuracy in Bar Chart
View Trained and Tested URL Data Sets Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Voting cla
learning, u
logistic re
and decisi
security, social
networks.
96
95.5
95
94.5
94
93.5
93
SVM 95.93%
Logistic Regression 95.69%
Decision Tree Classifier 95.91%
Random Forest Classifier 95.93%
Naive Bayes 93.43%
Naive Bayes
SVM
Logistic Regression
Decision Tree Classifier
Random Forest Classifier
CanvasJS.com
127.0.0.1:8000/charts1/line

Figure 5.10 : Accuracy of Trained and Tested URL Data sets in Line chart representation for Phishing detection System Through Hybrid Machine Learning Based on URL.

5.11 Accuracy of Trained and Tested URL Data sets in Pie Chart representation

The below figure 5.11 shows a pie chart that provides a percentage-based view of the accuracy contributed by each machine learning model used. It clearly illustrates the distribution of performance, showing SVM and Random Forest leading the results. This format is helpful in visual summaries and reports, particularly when explaining model performance to non-technical audiences. It complements the bar and line charts in showing overall system reliability.

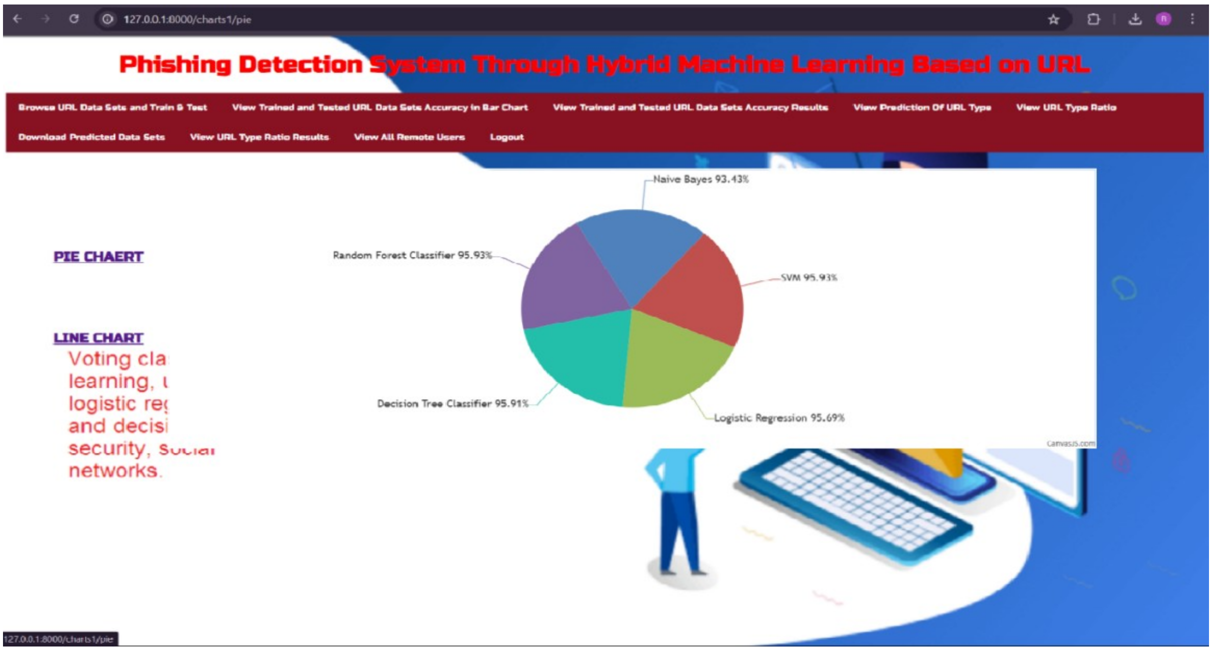
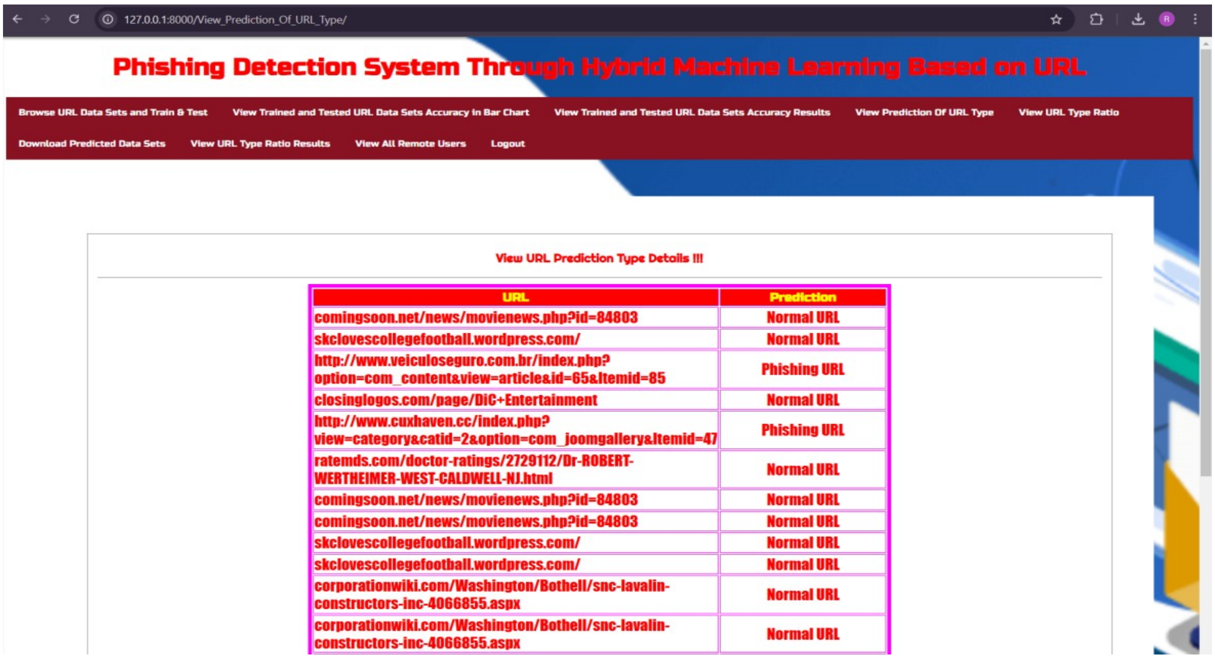

127.0.0.1:8000/charts1/pie
Phishing Detection System Through Hybrid Machine Learning Based on URL
Browse URL Data Sets and Train & Test
View Trained and Tested URL Data Sets Accuracy in Bar Chart
View Trained and Tested URL Data Sets Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Voting cla
learning, u
logistic re
and decisi
security, social
networks.
Naive Bayes 93.43%
Random Forest Classifier 95.93%
SVM 95.93%
Decision Tree Classifier 95.91%
Logistic Regression 95.69%
CanvasJS.com
127.0.0.1:8000/charts1/pie

Figure 5.11: Accuracy of Trained and Tested URL Data sets in Pie Chart representation for Phishing detection System Through Hybrid Machine Learning Based on URL.

5.12 Predicted URL details

The below figure 5.12 shows the log of all URLs entered by users along with their corresponding predictions. Each row includes the URL and its classification label (Phishing or Normal). This historical log allows users and administrators to review past activity and assess detection performance. It’s also useful for identifying any false positives or negatives in the prediction outcomes.


127.0.0.1:8000/View_Prediction_Of_URL_Type/
Phishing Detection System Through Hybrid Machine Learning Based on URL
Browse URL Data Sets and Train & Test
View Trained and Tested URL Data Sets Accuracy in Bar Chart
View Trained and Tested URL Data Sets Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout

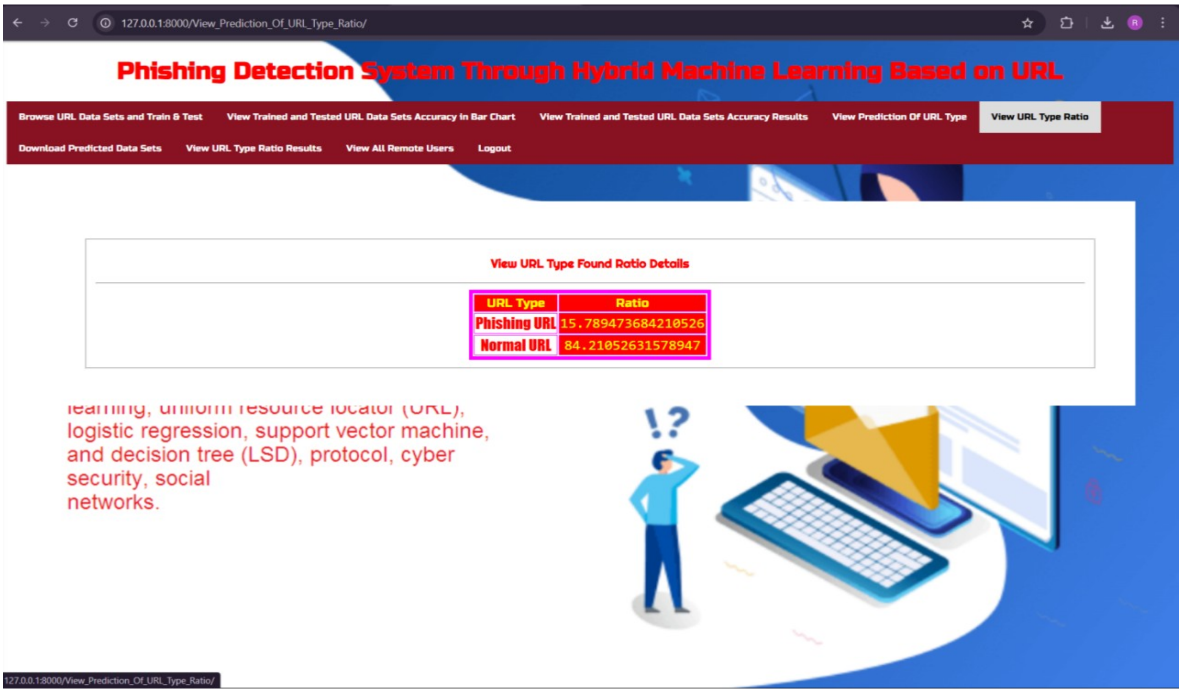
View URL Prediction Type Details !!!

URL	Prediction
comingsoon.net/news/movienews.php?id=84803	Normal URL
skclovescollegefootball.wordpress.com/	Normal URL
http://www.veiculoseguro.com.br/index.php?option=com_content&view=article&id=65&Itemid=85	Phishing URL
closinglogos.com/page/DiC+Entertainment	Normal URL
http://www.cuxhaven.cc/index.php?view=category&catid=2&option=com_joomgallery&Itemid=47	Phishing URL
ratemds.com/doctor-ratings/2729112/Dr-ROBERT-WERTHEIMER-WEST-CALDWELL-NJ.html	Normal URL
comingsoon.net/news/movienews.php?id=84803	Normal URL
comingsoon.net/news/movienews.php?id=84803	Normal URL
skclovescollegefootball.wordpress.com/	Normal URL
skclovescollegefootball.wordpress.com/	Normal URL
corporationwiki.com/Washington/Bothell/snc-lavalin-constructors-inc-4066855.aspx	Normal URL
corporationwiki.com/Washington/Bothell/snc-lavalin-constructors-inc-4066855.aspx	Normal URL

Figure 5.12: Predicted URL details for Phishing detection System Through Hybrid Machine Learning Based on URL.

5.13 Ratio of Results

The below figure 5.13 presents the ratio of classified URLs based on prediction results. The system shows that 84.2% of the URLs tested were normal, while 15.7% were detected as phishing. This gives a quick overview of the data distribution in user predictions. It helps evaluate the system's applicability to real-world URL traffic where most URLs may be safe.


127.0.0.1:8000/View_Prediction_Of_URL_Type_Ratio/
Phishing Detection System Through Hybrid Machine Learning Based on URL
Browse URL Data Sets and Train & Test
View Trained and Tested URL Data Sets Accuracy in Bar Chart
View Trained and Tested URL Data Sets Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout

View URL Type Found Ratio Details

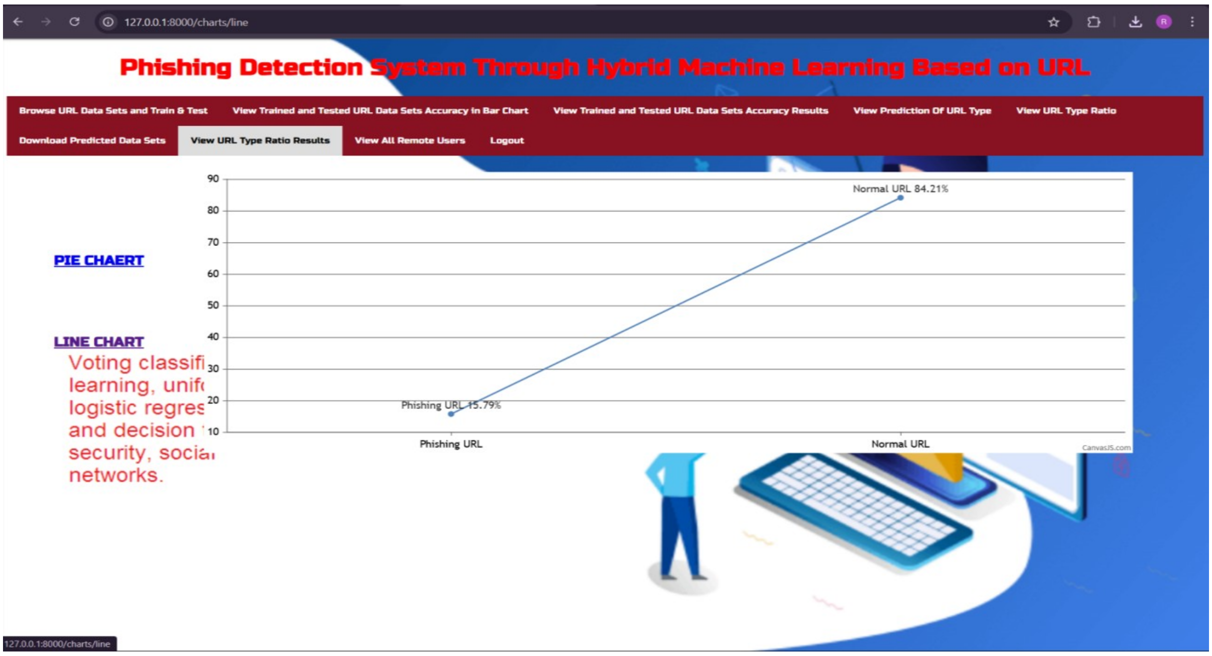
URL Type	Ratio
Phishing URL	15.789473684210526
Normal URL	84.21052631578947

learning, uniform resource locator (URL),
logistic regression, support vector machine,
and decision tree (LSD), protocol, cyber
security, social
networks.
127.0.0.1:8000/View_Prediction_Of_URL_Type_Ratio/

Figure 5.13: Ratio of Results for Phishing detection System Through Hybrid Machine Learning Based on URL.

5.14 Ratio of Results in Line Chart Representation

The below figure 5.14 that illustrates the same result ratio as Figure 5.13 but in the form of a line chart. It visually emphasizes the dominance of normal URLs in the data and the relatively lower number of phishing URLs. Such visualizations help stakeholders understand usage trends and system effectiveness over time.

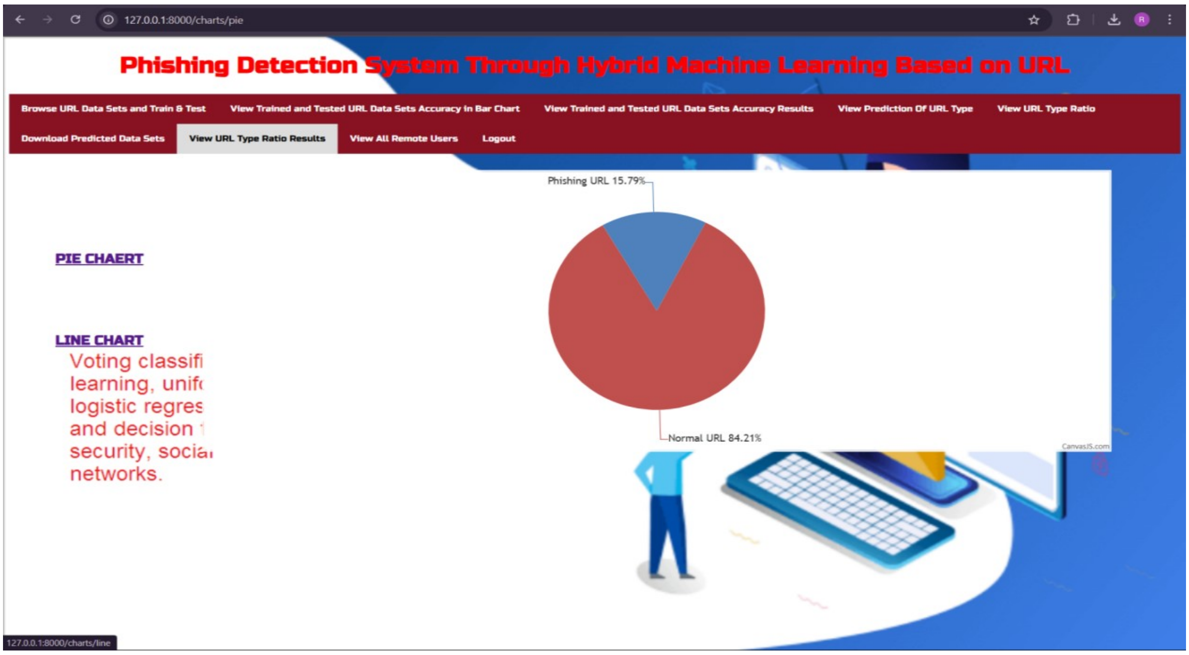

127.0.0.1:8000/charts/line
Phishing Detection System Through Hybrid Machine Learning Based on URL
Browse URL Data Sets and Train & Test
View Trained and Tested URL Data Sets Accuracy in Bar Chart
View Trained and Tested URL Data Sets Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
90
80
70
60
50
40
30
20
10
Normal URL 84.21%
Phishing URL 15.79%
Phishing URL
Normal URL
CanvasJS.com
127.0.0.1:8000/charts/line

Voting classifi
learning, unifi
logistic regres
and decision
security, social
networks.

Figure 5.14: Ratio of Results in Line Chart Representation for Phishing detection System Through Hybrid Machine Learning Based on URL.

5.15 Ratio of Results in Pie Chart Representation

In the below figure 5.15, a pie chart is used to represent the prediction results ratio. It shows 84.21% for Normal URLs and 15.79% for Phishing URLs. This type of visualization is particularly effective in presentations, summarizing system performance in a visually appealing way. It reinforces the insight that the system effectively detects phishing threats with low false positives.


127.0.0.1:8000/charts/pie
Phishing Detection System Through Hybrid Machine Learning Based on URL
Browse URL Data Sets and Train & Test
View Trained and Tested URL Data Sets Accuracy in Bar Chart
View Trained and Tested URL Data Sets Accuracy Results
View Prediction Of URL Type
View URL Type Ratio
Download Predicted Data Sets
View URL Type Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Phishing URL 15.79%
Normal URL 84.21%
CanvasJS.com
127.0.0.1:8000/charts/line

Voting classifi
learning, unif
logistic regres
and decision
security, socia
networks.

Figure 5.15: Ratio of Results in Pie Chart Representation for Phishing detection System Through Hybrid Machine Learning Based on URL.

6. VALIDATION

6. VALIDATION

The validation of this project is based on systematic testing and evaluation using defined performance metrics and structured test cases. This process ensures the reliability, robustness, and accuracy of the phishing detection system. The testing involves preprocessing of datasets, evaluating various machine learning classifiers, and validating model predictions using both standard metrics and visualization tools such as confusion matrices and accuracy graphs. The aim is to confirm that the system efficiently detects phishing URLs while minimizing false classifications.

6.1 INTRODUCTION

Initially, the dataset is divided into training and testing sets, commonly using an 80-20 ratio. The training data is used to train multiple machine learning models such as Logistic Regression, SVM, Decision Tree, Random Forest, and Naive Bayes. These models are then evaluated on the test set to assess their generalization capabilities. To ensure consistency and prevent overfitting, k-fold cross-validation is applied during the training phase.

Model performance is analyzed using accuracy, precision, recall, F1-score, and confusion matrix. The confusion matrix helps understand correct and incorrect classifications and refine the model accordingly. Among the tested models, the SVM classifier achieved the highest accuracy of 95.93%, followed closely by Random Forest and Decision Tree, validating the robustness of the hybrid machine learning approach.

Real-time testing is also performed using live URL inputs through the application interface. This confirms that the system accurately classifies URLs into phishing or legitimate categories. The structured validation process guarantees that the developed phishing detection system is accurate, scalable, and practical for real-time use across various web-based platforms.

6.2 TEST CASES

TABLE 6.2.1 UPLOADING DATASET

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads Dataset.	Use it for phishing URL detection	The user uploads the URL Dataset, to train and test the models.	Dataset successfully loaded.

The above Table 6.2.1 represents the test case for uploading the phishing dataset into the system. This is a crucial initial step where the user provides a dataset consisting of URLs labeled as either phishing or legitimate. The purpose of this test case is to ensure that the system can successfully accept and load the dataset into the backend for further processing, training, and evaluation. When the user uploads the dataset, the expected output is a confirmation that the dataset has been successfully loaded. This functionality is essential as it serves as the foundation for model training and URL classification.

TABLE 6.2.2 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Classification test 1	To verify if the classifier detects a normal URL	A legitimate/normal URL is provided	Normal URL detected
2	Classification test 2	To verify if the classifier detects a phishing URL	A phishing/malicious URL is provided	Phishing URL detected

The above Table 6.2.2 outlines the classification testing process and includes two test scenarios that verify the accuracy of the system in predicting URL types. The first test case is designed to confirm that the system correctly identifies a legitimate URL. When a normal URL is input, the system should return a result indicating it as a “Normal URL.” The second test case checks whether the system can correctly detect a phishing URL. When a phishing or malicious URL is given, the expected output is “Phishing URL.” These test cases validate that the system’s classification logic works as intended and that the machine learning model accurately distinguishes between safe and harmful URLs.

7. CONCLUSION & FUTURE ASPECTS

7. CONCLUSION & FUTURE ASPECTS

In conclusion, the Phishing Detection System using hybrid machine learning models has successfully met its core objective of accurately identifying phishing URLs. The implementation combined the strengths of various classifiers such as SVM, Logistic Regression, Decision Tree, and Random Forest, resulting in high prediction accuracies. Through rigorous testing and evaluation, the project demonstrates that hybrid models can provide better generalization, performance, and robustness than single-model approaches. The developed system is capable of detecting malicious URLs based on URL features alone, without needing website content, which makes it fast, scalable, and efficient for practical use.

Looking ahead, the project has strong potential for future development. Key areas for improvement include adapting the model for real-time detection, integrating dynamic website behavior analysis, and improving resistance to adversarial URL manipulation. Incorporating deep learning models, threat intelligence feeds, and deploying the system as a cloud-based API are also promising directions that would enhance its application scope and performance.

7.1 PROJECT CONCLUSION

This research proposes a robust hybrid machine learning approach for phishing detection based solely on URL features. The system integrates multiple classifiers—Naive Bayes, SVM, Logistic Regression, Decision Tree, and Random Forest—and evaluates their performance to determine the most effective models for accurate classification. Among these, SVM and Random Forest achieved the highest accuracies of **95.93%**, showcasing the effectiveness of hybrid learning models.

By leveraging lexical and statistical features of URLs, the system avoids reliance on external content, making it lightweight and responsive. The user interface provides seamless uploading, training, testing, and prediction workflows, along with visual performance comparison through bar charts. The system offers an efficient, scalable, and deployable solution for phishing URL detection across different platforms and datasets, outperforming traditional single-algorithm models.

7.2 FUTURE ASPECTS

While the current phishing detection system demonstrates high accuracy and robustness, there is considerable potential for future development to enhance its real-world applicability. One key direction is the implementation of real-time detection capabilities, enabling the system to monitor web traffic or email communications for phishing threats as they occur. Integrating the model into practical tools such as browser extensions or email filters can provide users with proactive protection during everyday internet usage. Moreover, incorporating advanced deep learning architectures like recurrent neural networks (RNNs) or transformers could enable the system to learn more complex URL patterns and improve detection accuracy over time.

The adaptability of the system can be further improved by introducing online learning mechanisms that allow it to evolve continuously with emerging phishing strategies. Additionally, integrating external threat intelligence sources such as URL blacklists, WHOIS data, or web reputation databases can enrich the system’s analytical capabilities. Expanding the solution to mobile platforms will enhance its accessibility and usability, especially in regions where mobile internet usage is dominant. Furthermore, embedding explainable AI components like LIME or SHAP can improve the transparency of model decisions, making the system more trustworthy for end-users and cybersecurity professionals. Finally, expanding language support and cultural adaptability will enable the detection of phishing attempts that use domain names and URL patterns in non-English scripts, making the system globally relevant and resilient against increasingly sophisticated phishing tactics.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, Y. Si, F. Zhang, and J. S. Dong, ''Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages,'' in Proc. 30th USENIX Secur. Symp. (USENIX Security), 2021, pp. 3793–3810.

[2] H. Shirazia, K. Haynesb, and I. Raya, ''Towards performance of NLP transformers on URL-based phishing detection for mobile devices,'' Int. Assoc. Sharing Knowl. Sustainability (IASKS), Tech. Rep., 2022.

[3] A. K. Dutta, ''Detecting phishing websites using machine learning technique,'' PLoS ONE, vol. 16, no. 10, Oct. 2021, Art. no. e0258361.

[4] A. A. Ubing, S. Kamilia, A. Abdullah, N. Jhanjhi, and M. Supramaniam, ''Phishing website detection: An improved accuracy through feature selection and ensemble learning,'' Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 252–257, 2019.

[5] H. S. Hota, A. K. Shrivas, and R. Hota, ''An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique,'' Proc. Comput. Sci., vol. 132, pp. 900–907, Jan. 2018.

[6] G. Sonowal and K. S. Kuppusamy, ''PhiDMA— A phishing detection model with multi-filter approach,'' J. King Saud Univ., Comput. Inf. Sci., vol. 32, no. 1, pp. 99–112, Jan. 2020.

[7] S. Bell and P. Komisarczuk, ''An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank,'' in Proc. Australas. Comput. Sci. Week Multiconf. (ACSW), Melbourne, VIC, Australia. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–11, Art. no. 3, doi: 10.1145/3373017.3373020.

[8] A. K. Jain and B. Gupta, ''PHISH-SAFE: URL features-based phishing detection system using machine learning,'' in Cyber Security. Switzerland: Springer, 2018, pp. 467–474.

[9] M. Khonji, Y. Iraqi, and A. Jones, ''Phishing detection: A literature survey,'' IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2091–2121, 4th Quart, 2013.

8.2 GITHUB LINK

https://github.com/Rahul1418/Phishing-Detection-System-Through-Hybrid-Machine-Learning-Based-on-URL