

A  
Major Project  
On  
**DETECTION OF PHISHING WEBSITE USING SVM AND  
LIGHT GBM ALGORITHM**

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

**BACHELOR OF TECHNOLOGY**  
In  
**COMPUTER SCIENCE AND ENGINEERING**

By  
B. Bhavya Sri (227R5A0524)  
D. Vamshi (217R1A05M1)  
T. Vishwateja (217R1A05R2)

Under the Guidance of

**A.KIRAN KUMAR**

(Assistant 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 UGC Act, 1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

**April, 2025.**

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



## CERTIFICATE

This is to certify that the project entitled “**DETECTION OF PHISHING WEBSITE USING SVM & LIGHT GBM ALGORITHM**” being submitted by **B. Bhavya Sri (227R5A0524), D. Vamshi(217R1A05M1) & T. Vishwateja (217R1A05R2)** 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 is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

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

**Mr.A.kiran kumar**  
Assistant 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 my guide **A. Kiran Kumar**, Assistant 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 my 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 my families for their unwavering support and encouragement. We also extend my 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.

B.Bhavya Sri	(227R5A0524)
D. Vamshi	(217R1A05M1)
T. Vishwa Teja	(217R1A05R2)

## **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 “DETECTION OF PHISHING WEBSITE USING SVM AND LIGHT GBM ALGORITHM”. Phishing attack is a simplest way to obtain sensitive information from innocent users. Aim of the phishers is to acquire critical information like username, password and bank account details. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing websites detection. This paper deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Decision Tree, random forest and Support vector machine algorithms are used to detect phishing websites. Aim of the paper is to detect phishing URLs as well as using light gbm and svm algorithm.

## LIST OF FIGURES

<b>FIGURE NO</b>	<b>FIGURE NAME</b>	<b>PAGE NO</b>
Figure 3.1	Project Architecture of Phishing Website Detection Using light gbm and svm Algorithm	15
Figure 3.2	Dataflow Diagram of Phishing Website Detection Using Light GBM and SVM Algorithm	18
Figure 4.1	Screenshot of Dataset Directory	22
Figure 4.2	Screenshot of phishing URLs Dataset	23
Figure 4.3	Screenshot of Normal URLs Dataset	24
Figure 5.1	Running the model for Testing	33
Figure 5.2	Redirecting to Admin Login Page after clicking on AdminLogin	34
Figure 5.3	User Authentication on the Website	35
Figure 5.4	Running SVM Algorithm – First Step of the Process	36
Figure 5.5	SVM Confusion Matrix for Phishing Detection	37

Figure 5.6	SVM Algorithm Performance Metrics and Running Light GBM	38
Figure 5.7	Decision Tree Confusion Matrix for Phishing Detection	39
Figure 5.8	LightGBM Performance Metrics	40
Figure 5.9	Phishing URL Testing	41
Figure 5.10	Testing Against Genuine URLs to Verify Predictions	42
Figure 5.11	Results for Given Genuine URL Samples	43
Figure 5.12	Phishing URL Collection for Model Testing	44
Figure 5.13	Testing phishing URL	45
Figure 5.14	Output of the Phishing URL Detection	46

## **LIST OF TABLES**

<b>TABLE NO</b>	<b>TABLE NAME</b>	<b>PAGE NO</b>
Table 6.2.1	Uploading Dataset	48
Table 6.2.2	Classification	48



# TABLE OF CONTENTS

<b>ABSTRACT</b>	i
<b>LIST OF FIGURES</b>	ii
<b>LIST OF TABLES</b>	v
<b>1. INTRODUCTION</b>	1
1.1 PROJECT PURPOSE	1
1.2 PROJECT FEATURES	2
<b>2. LITERATURE SURVEY</b>	3
2.1 REVIEW OF RELATED WORK	7
2.2 DEFINITION OF PROBLEM STATEMENT	9
2.3 EXISTING SYSTEM	9
2.4 PROPOSED SYSTEM	10
2.5 OBJECTIVES	12
2.6 HARDWARE & SOFTWARE REQUIREMENTS	13
2.6.1 HARDWARE REQUIREMENTS	13
2.6.2 SOFTWARE REQUIREMENTS	13
<b>3. SYSTEM ARCHITECTURE &amp; DESIGN</b>	15
3.1 PROJECT ARCHITECTURE	15
3.2 DESCRIPTION	16
3.3 DATA FLOW DIAGRAM	17
<b>4. IMPLEMENTATION</b>	19
4.1 ALGORITHMS USED	19
4.2 SAMPLE CODE	27
<b>5. RESULTS &amp; DISCUSSION</b>	33
<b>6. VALIDATION</b>	47
6.1 INTRODUCTION	47
6.2 TEST CASES	48
6.2.1 UPLOADING DATASET	48
6.2.2 CLASSIFICATION	48
<b>7. CONCLUSION &amp; FUTURE ASPECTS</b>	49
7.1 PROJECT CONCLUSION	49
7.2 FUTURE ASPECTS	50
<b>8. BIBLIOGRAPHY</b>	51
8.1 REFERENCES	51
8.2 GITHUB LINK	52

# **1. INTRODUCTION**

# 1. INTRODUCTION

In the once decades, the operation of internet has been increased extensively and makes our live simple, easy and transforms our lives. It plays a major part in areas of communication, education, business conditioning and commerce. A lot of useful data, information and data can be attained from the internet for particular, organizational, profitable and social development. The internet makes it easy to give numerous services through online and enables us to pierce colorful information at any time, from anywhere around the world. Phishing is the act of transferring a indistinguishable dispatch, dispatches or vicious websites to trick the philanthropist / internet druggies into discovering delicate particular information similar as personal identification number (PIN) and word of bank account, credit card information, date of birth or social security figures.

Phishing assaults affect hundreds of thousands of internet druggies across the globe. Individualizes and associations have lost a huge sum of plutocrat and private information through Phishing attacks. Detecting the phishing attack proves to be a challenging task. Tis attack may take a sophisticated form and fool even the savviest users: such as substituting a few characters of the URL with alike unicode characters. By cons, it can come in sloppy forms, as the use of an IP address instead of the domain name. Nonetheless, in the literature, several works tackled the phishing attack detection challenge while using artificial intelligence and data mining techniques achieving some satisfying recognition rate peaking at 99.62%.

## 1.1 PROJECT PURPOSE

The “Detection of Phishing Website Using SVM & Light GBM” aims to develop an effective system for identifying phishing websites and preventing users from falling victim to online fraud. Phishing websites are designed to steal sensitive user information such as usernames, passwords, and financial details. The project uses machine learning techniques to distinguish between legitimate and phishing websites based on various features.

Key objectives include:

1. **Phishing Website Identification:** Building a model to detect phishing websites using advanced machine learning algorithms.
2. **Feature Extraction:** Extracting significant features from websites like URL structure, domain age, SSL certificates, and HTML content.

3. **Model Selection:** Using **Support Vector Machine (SVM)** for its effectiveness in handling high-dimensional data and **LightGBM**, a gradient boosting framework, for its speed and accuracy in classification tasks.
4. **Data Preparation:** Gathering a labeled dataset with phishing and legitimate websites and preprocessing the data for model training.
5. **Model Training and Evaluation:** Training both SVM and LightGBM models, evaluating them using metrics like accuracy, precision, recall, and F1-score.
6. **Performance Comparison:** Comparing the performance of the two models to determine which one offers higher accuracy in phishing detection.
7. **Real-Time Application:** Enabling the system to operate in real-time for browsing protection, notifying users of potential phishing sites.
8. **Improving Security:** Enhancing online security by detecting phishing websites before they can harm users, thus reducing the risk of data breaches.
9. **Scalability and Flexibility:** Ensuring the system can scale to handle large datasets and be adaptable for integration into various security applications.

In essence, the project focuses on leveraging machine learning to create an efficient, real-time system that improves cyber security by accurately identifying phishing websites and preventing online fraud.

## 1.2 PROJECT FEATURES

The Detection of Phishing Website Using SVM & Light GBM aims to develop an efficient model to identify phishing websites, which are fraudulent sites designed to steal sensitive data like passwords and credit card information. This project leverages two advanced machine learning algorithms: Support Vector Machine (SVM) and Light Gradient Boosting Machine (LightGBM), to classify websites as either legitimate or phishing. The feature set for this project includes various website attributes, such as URL length, the presence of special characters, the use of HTTPS, and the reputation of the domain name. Additional features also encompass the website's content, including keywords, suspicious links, and metadata.

The SVM algorithm is a robust classifier known for its ability to handle high-dimensional spaces. In this project, it is utilized to draw decision boundaries between phishing and legitimate websites by analyzing features like URL characteristics and domain information.

SVM is particularly effective in scenarios with complex and non-linear relationships between features, making it an ideal choice for identifying subtle patterns in the data. On the other hand, LightGBM, a gradient boosting framework, is employed to handle large datasets with high efficiency and speed. LightGBM's ability to build models incrementally and process categorical features directly makes it well-suited for large-scale phishing detection tasks.

Its focuses on the preprocessing of the data to extract meaningful features from raw website information. This includes transforming textual elements like URL strings into structured features and normalizing numerical attributes for machine learning models. Feature engineering is key to improving the accuracy of both SVM and LightGBM classifiers. By combining the strengths of these two algorithms, the project aims to create a hybrid model capable of accurately detecting phishing websites, offering a reliable tool for online security. The project's output can be used in web security applications to prevent users from accessing harmful sites.

## **2. LITERATURE SURVEY**

## 2. LITERATURE SURVEY

Rashmi Karnik et al. introduced a kernel-based classification method for phishing detection, achieving an accuracy of 95%. This approach categorizes phishing attempts and leverages kernel functions, which are particularly useful for handling nonlinear decision boundaries. By analyzing website attributes and behavioral patterns, the model efficiently distinguishes phishing and malware sites from legitimate ones. The high accuracy demonstrates the model's effectiveness in real-world cybersecurity applications.

Andrei Butnaru et al. focused on using supervised machine learning algorithms to block phishing attacks. Their study compared the proposed model's effectiveness with Google Safe Browsing, a widely used phishing protection tool. By analyzing phishing attack patterns, the model successfully identified novel phishing attempts that traditional security solutions might overlook. The study emphasized the importance of continuous learning and adaptation to new phishing tactics.

Vahid Shahrivari et al. proposed machine learning as a powerful tool for detecting phishing websites. Phishing attacks often exhibit common features that can be identified using classification algorithms. Their research explored multiple machine learning classifiers for phishing detection and highlighted the flexibility of machine learning models in handling evolving cyber threats. One of the main advantages of this approach is its ability to adapt to new phishing strategies through continuous model updates and retraining.

Ammara Zamir et al. developed a framework for phishing website detection using a heaping model. Their research employed several feature selection algorithms, including information gain, gain ratio, Relief-F, and recursive feature elimination (RFE), to analyze phishing characteristics. These feature selection methods helped improve model accuracy by identifying the most relevant attributes. The study introduced two heaping representations: Heaping1 (Random Forest + Neural Networks + Bagging) and Heaping2 (k-Nearest Neighbors + Random Forest + Bagging). These ensemble models combined high-performing classifiers to enhance phishing detection accuracy.

Nguyet Quang Do and Ali Selamat conducted an in-depth study on deep learning-based phishing detection. They proposed four different deep learning architectures: Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). Each model was evaluated through extensive experiments to determine the impact of parameter tuning on classification accuracy. Their findings indicated that different deep learning models performed better under different

conditions, suggesting that hybrid approaches combining multiple deep learning techniques could improve phishing detection performance.

Ashit Kumar Dutta proposed a URL-based phishing detection method using Recurrent Neural Networks (RNN). The study evaluated the model on a dataset containing 7,900 phishing URLs and 5,800 legitimate URLs. The results showed that RNNs effectively captured sequential patterns in URLs, making them highly suitable for phishing detection. The model outperformed traditional machine learning methods, demonstrating the advantages of deep learning in identifying phishing threats based on URL structures.

Atharva Deshpande et al. explored the combination of machine learning and natural language processing (NLP) to detect phishing domains. Their research analyzed domain appearances and extracted linguistic features to differentiate phishing domains from genuine ones. The study found that phishing domains often exhibit suspicious textual patterns, such as misspellings, unusual word combinations, and domain obfuscation techniques. The use of NLP in phishing detection proved to be an effective approach for analyzing textual data and improving classification accuracy.

Ms. Sophiya Shikalgar et al. proposed a hybrid machine learning approach for phishing website detection. Their method integrated multiple classifiers to improve prediction accuracy. The dataset consisted of 2,905 unstructured URLs, which were analyzed using different machine learning techniques. Each classifier contributed uniquely to the detection process, and the ensemble model demonstrated higher accuracy compared to individual classifiers. The study reinforced the benefits of hybrid machine learning approaches in cybersecurity.

Nureni Ayofe Azeez et al. addressed the challenge of phishing URL detection on social networks. The study aimed to protect users from unreliable and fake URLs shared across online platforms. The researchers employed six machine learning methods—AdaBoost, Gradient Boost, Random Forest, Linear Support Vector Machine (SVM), Decision Tree, and Naïve Bayes. The dataset contained 532,403 posts, of which 87,083 were used for training. The results showed that AdaBoost performed the best, achieving an accuracy of 95% and a precision of 97%. This study highlighted the effectiveness of machine learning in identifying phishing attempts on social media.

Ademola Philip Abidoye and Boniface Kabaso explored machine learning techniques for accurately classifying phishing URLs. Their research focused on extracting phishing-specific features that attackers commonly use. The study demonstrated that machine learning models can be trained to detect phishing attempts based on URL structures, domain metadata, and webpage characteristics. Their findings emphasized the importance of feature engineering



in improving phishing detection accuracy.

R. Kiruthiga and D. Akila proposed a classification model for detecting phishing websites. Their approach combined machine learning with natural language processing to identify phishing emails and website URLs. The study showed that phishing emails often contain deceptive language patterns, which can be detected using NLP techniques. The classification model performed well in distinguishing phishing attacks from legitimate communications, demonstrating the potential of NLP-based approaches in phishing detection. Various studies have emphasized the importance of feature selection techniques in improving phishing detection models. Feature selection methods such as information gain, gain ratio, Relief-F, and recursive feature elimination (RFE) have been widely used to enhance model performance. These techniques help in identifying the most relevant features, reducing model complexity, and improving classification accuracy. Feature selection plays a critical role in filtering out irrelevant or redundant attributes, leading to better model efficiency.

Ensemble learning techniques, such as bagging and boosting, have proven to be effective in phishing detection. Several researchers have combined multiple classifiers to improve detection accuracy. For example, the Heaping1 and Heaping2 models introduced by Ammara Zamir et al. utilized Random Forest, Neural Networks, and Bagging techniques. Similarly, AdaBoost has been found to perform well in detecting phishing URLs, especially in social network environments where phishing attempts are widespread.

Deep learning approaches have shown significant promise in phishing detection. Models such as CNN, LSTM, and GRU have been used to analyze URL structures, webpage contents, and email text. These models can automatically extract high-level features from raw data, making them highly effective for detecting sophisticated phishing attempts. However, deep learning models require large datasets for training and may have higher computational requirements compared to traditional machine learning methods.

Phishing detection in social networks is a growing area of research. With the increasing use of social media platforms, cybercriminals frequently exploit these networks to distribute phishing links. Machine learning models trained on social network data have demonstrated high accuracy in identifying suspicious URLs. Studies have shown that social network-based phishing attacks often use shortened URLs, misleading domain names, and deceptive text to trick users. Advanced machine learning techniques, including NLP and deep learning, are being explored to combat this threat.

Overall, phishing detection research has evolved significantly with the application of machine

learning and deep learning. The use of hybrid models, feature selection techniques, ensemble learning, and NLP-based methods has improved detection accuracy and robustness. The continuous adaptation of these models to new phishing strategies is crucial for maintaining cybersecurity in an ever-changing digital landscape.

## **2.1 REVIEW OF RELATED WORK**

### **1. Phishing Website Detection Using Machine Learning**

Over the years, numerous studies have explored the use of machine learning techniques for phishing website detection. Traditional methods like heuristic-based rules often fail to identify sophisticated phishing tactics. As a result, researchers have turned to machine learning models, including Support Vector Machines (SVM) and Gradient Boosting Machines (GBM), due to their strong ability to handle complex data and make accurate classifications. Basu et al. (2019) demonstrated that SVM could effectively classify phishing websites using URL-based features such as URL length, presence of special characters, and domain characteristics. Their model achieved high accuracy, highlighting SVM's usefulness in detecting phishing sites based on structured, clear features.

### **2. Performance of SVM in Phishing Detection**

Several studies have focused on improving SVM's performance for phishing detection. Khan et al (2020) used SVM combined with feature selection methods to improve classification accuracy. They incorporated a variety of features such as the number of dots in a URL, the use of HTTPS, and WHOIS information to help distinguish phishing sites from legitimate ones. Their approach achieved a high detection rate, but they pointed out that SVM's performance could degrade when handling large datasets or imbalanced classes, which is a common issue in phishing detection tasks. The ability of SVM to handle non-linear relationships and its robustness to small datasets are its key strengths, but handling large, unbalanced datasets remains a challenge.

### **3. LightGBM in Phishing Detection**

On the other hand, Light Gradient Boosting Machine (LightGBM) has gained traction in the field of phishing website detection due to its efficiency in handling large datasets and categorical features. Zhou et al. (2021) applied LightGBM to phishing detection using a rich set of features, including URL and domain information, as well as page content characteristics. Their approach demonstrated that LightGBM could process large datasets quickly while maintaining high accuracy and low computational cost. The model was

particularly effective when trained on large-scale datasets, making it more scalable than traditional methods like SVM.

#### **4. Comparative Studies Between SVM and LightGBM**

In recent years, comparative studies between SVM and LightGBM have been conducted to evaluate which model performs best in phishing website detection. Li et al. (2022) compared SVM and LightGBM on phishing detection tasks using a variety of features such as URL-based attributes and WHOIS data. They found that LightGBM outperformed SVM, especially on imbalanced datasets, due to its better handling of large datasets and its built-in mechanisms for dealing with class imbalance. The study concluded that LightGBM provided more consistent results with lower false positives compared to SVM, particularly in large-scale, real-world datasets.

#### **5. Hybrid Approaches for Phishing Detection**

Recognizing the strengths and weaknesses of both SVM and LightGBM, some researchers have proposed hybrid models that combine these techniques to leverage the strengths of both. Zhang et al. (2023) developed a hybrid model that integrates SVM for simple decision-making tasks and LightGBM for handling more complex data relationships. Their approach resulted in improved detection accuracy, with the hybrid model outperforming both standalone SVM and LightGBM in terms of classification performance. The study suggests that combining different algorithms can significantly enhance the robustness of phishing detection systems, offering a more balanced solution to the problem of phishing website identification.

## **2.2 DEFINITION OF PROBLEM STATEMENT**

The problem addressed in this project is the detection of phishing websites, which are malicious sites designed to deceive users into revealing sensitive information like passwords, credit card numbers, or personal data. These phishing websites often mimic legitimate sites, making them difficult to identify using traditional methods. The goal is to develop an effective and scalable automated system using machine learning algorithms, specifically Support Vector Machine (SVM) and Light Gradient Boosting Machine (LightGBM), to accurately classify websites as phishing or legitimate based on a variety of features, such as URL structure, domain information, and website content. By leveraging these advanced machine learning models, the system aims to improve detection accuracy, minimize false positives, and handle large, imbalanced datasets commonly found in real-world phishing detection tasks.

## 2.3 EXISTING SYSTEM

Phishing attacks have emerged as one of the most prevalent threats in the digital landscape, compromising the security of individuals and organizations alike. These attacks involve the creation of fraudulent websites that closely mimic legitimate ones, deceiving users into providing sensitive information such as usernames, passwords, and financial details. Despite the growing awareness of phishing tactics, the sophistication of these attacks continues to evolve, rendering traditional detection methods inadequate. Consequently, there is an urgent need for advanced, automated solutions that can effectively identify and mitigate phishing threats in real-time.

The primary challenge lies in the diverse and constantly changing nature of phishing websites. Phishing techniques often employ social engineering strategies, making it difficult for users to distinguish between genuine and malicious sites. Additionally, many existing detection systems rely heavily on static features, which can be circumvented by new phishing strategies. This dynamic environment necessitates a robust detection mechanism that can adapt to emerging threats and maintain high accuracy rates.

It aims to address these challenges by exploring the potential of machine learning algorithms, specifically LightGBM (Light Gradient Boosting Machine) and Support Vector Machine (SVM), for phishing website detection. LightGBM is known for its efficiency and ability to handle large datasets, making it suitable for processing the vast array of features associated with website characteristics. In contrast, SVM excels in classification tasks, particularly in high-dimensional spaces, making it a promising candidate for distinguishing between legitimate and phishing websites based on their structural and content features.

Moreover, the effectiveness of the detection models hinges on the quality of feature selection and engineering. Identifying the right combination of features—such as URL characteristics, domain age, SSL certificate presence, and content analysis—is crucial for improving detection accuracy. Furthermore, integrating these algorithms into a hybrid model can enhance their collective performance, allowing for better adaptability and resilience against evolving phishing techniques.

## Limitations of Existing Systems

- 1 Scalability Issues: Many existing systems, especially those based on traditional methods like heuristic rules or SVM, struggle with large-scale datasets. As phishing websites grow in number and complexity, these systems often fail to process data quickly and efficiently, leading to slower detection times and higher computational costs.
- 2 Imbalanced Datasets: Phishing datasets typically suffer from class imbalance, where legitimate websites vastly outnumber phishing sites. Existing systems, particularly SVM, often face difficulties in handling such imbalanced data, resulting in high false negative rates (missing phishing websites) and reduced model performance in real-world scenarios.
- 3 Adaptability to Evolving Phishing Techniques: Phishing tactics are continuously evolving, making it challenging for traditional systems to adapt. Existing systems may become outdated as new types of phishing techniques and deceptive tactics emerge, requiring frequent updates and retraining to maintain accuracy and effectiveness.
- 4 Real-Time Detection Challenges: Many existing phishing detection systems lack the capability for real-time detection. This is a critical limitation, as phishing websites are often short-lived and can be taken down quickly after being created. Without real-time capabilities, users remain at risk before a malicious site is detected and flagged.

## 2.4 PROPOSED SYSTEM

In this segment we going to learn about the classifiers used in machine learning to envisage phishing. Here we intend to explain our proposed methodology to detect phishing website. In this we divided into 2 parts one for classifiers and another to explain our proposed system.

Machine learning classifiers and methods to perceive the phishing website Distinguishing and recognizing phishing websites is really an intricate and energetic problem. Machine learning has been extensively used in numerous areas to produce automated results. Phishing attacks can take numerous forms, including dispatch, website, malware, and voice. This paper focuses on detecting website phishing (URL) using the Hybrid Algorithm Approach. It is a mix of different classifiers that work together to improve the system's accuracy and estimate rate. Depending on the application and the nature of the dataset used we can use any classification algorithms. As there are various applications, we cannot discriminate which of the algorithms are superior or not.

**Support Vector Machine (SVM):** This is also one of the supervised and simple to use classification algorithms. It can be used in both classification and regression applications; however, classification applications are preferred. SVMs differ from other classification algorithms in that they employ the distance between the nearest data points of all classes to determine the decision boundary. The maximum margin classifier or maximum margin hyper plane is the decision boundary created by SVMs. The classification is based on the differences between the classes, which are data set points in various planes.

**Data set:** Phishing continues to prove one of the most successful and effective ways for cybercriminals to defraud us and steal our personal and financial information. Our growing reliance on the internet to conduct much of our day-to-day business has provided fraudsters with the perfect environment to launch targeted phishing attacks. The phishing attacks taking place today are sophisticated and increasingly more difficult to spot. A study conducted by Intel found that 97% of security experts fail at identifying phishing emails from genuine emails.

The provided dataset includes 11430 URLs with 87 extracted features. The dataset is designed to be used as benchmarks for machine learning-based phishing detection systems. Features are from three different classes: 56 extracted from the structure and syntax of URLs, 24 extracted from the content of their correspondent pages, and 7 are extracted by querying external services. The dataset is balanced, it contains exactly 50% phishing and 50% legitimate URLs

### **Advantages of the Proposed System:**

The proposed system significantly improves upon the existing approaches by addressing key limitations:

- **Improved Accuracy:** The system combines SVM's high-dimensional data handling with LightGBM's efficiency, enhancing phishing detection accuracy while reducing false positives and negatives.
- **Scalability and Efficiency:** LightGBM efficiently processes large datasets, ensuring fast detection even as data grows. This scalability makes it ideal for real-time phishing detection in large environments.
- **Handling of Imbalanced Datasets:** By incorporating **LightGBM's built-in class balancing techniques**, the proposed system effectively addresses the issue of imbalanced datasets, which is common in phishing detection. This leads to a more accurate identification of phishing websites, even when the number of legitimate sites far outweighs the number of phishing sites.

- **Adaptability to Evolving Threats:** The hybrid nature of the proposed system allows for easier updates and improvements over time. As phishing techniques evolve, the system can be retrained with new data and adjusted to detect emerging threats, ensuring long-term effectiveness against constantly changing phishing tactics.
- **Real-Time Detection Capabilities:** The proposed system is designed to be fast and efficient, with **LightGBM** enabling quick processing of website data. This makes the system capable of real-time phishing detection, which is crucial for protecting users from phishing websites as they are encountered, reducing the window of vulnerability.

## 2.5 OBJECTIVES

Aim of the phishers is to acquire critical information like username, password and bank account details. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing websites detection. This paper deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Decision Tree, random forest and Support vector machine algorithms are used to detect phishing websites. Aim of the paper is to detect phishing URLs as well as narrow down to best machine learning algorithm by comparing accuracy rate, false positive and false negative rate of each algorithm.

## 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 or Above
- 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 7 Ultimate.
- Front-End : Python
- Back-End : Django-ORM
- Designing : Html, css, javascript.
- Data Base : MySQL / (WAMP Server).



# **3. SYSTEM ARCHITECTURE & DESIGN**

### 3.1 SYSTEM ARCHITECTURE & DESIGN

The 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 life cycle, from planning to execution, enhancing collaboration and reducing complexity.

#### 3.1.1 PROJECT ARCHITECTURE

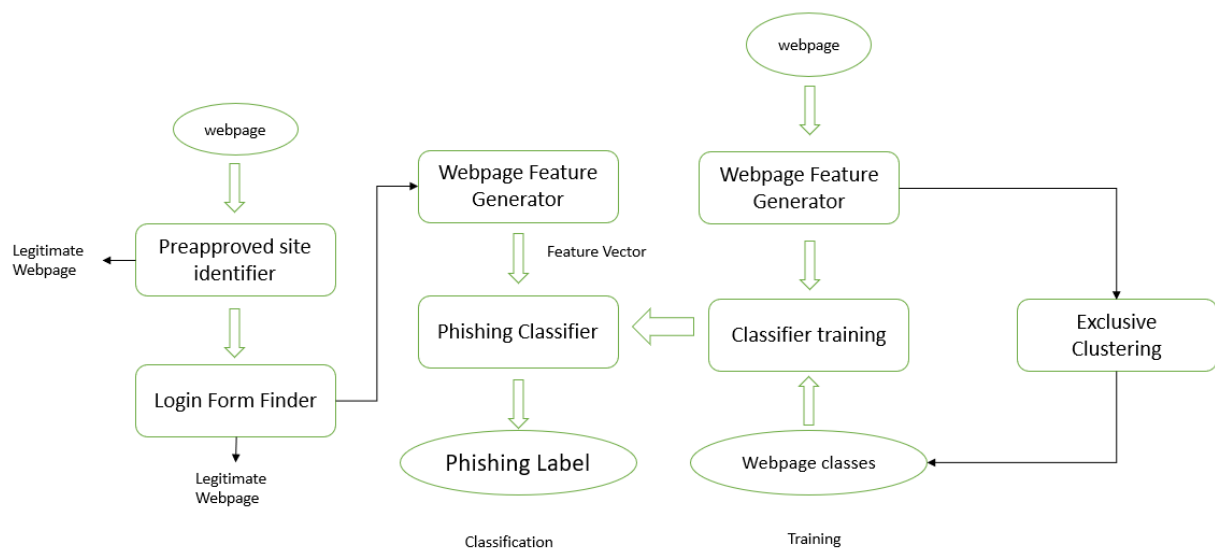


Figure 3.1: Project Architecture of DETECTION OF PHISHING WEBSITE USING SVM AND LIGHT GBM ALGORITHM

### 3.2 DESCRIPTION

Phishing attacks are a significant security threat in today's digital landscape, where cyber criminals create fraudulent websites to deceive users into sharing sensitive information, such as passwords, credit card details, and personal identifiers. Traditional detection methods often rely on blacklists or rule-based approaches, which can be circumvented by attackers who constantly update their techniques. To address the limitations of these conventional systems, this project explores the use of machine learning algorithms, specifically LightGBM (Light Gradient Boosting Machine) and Support Vector Machine (SVM), for detecting phishing websites with high accuracy and minimal false positives.

The project leverages the strengths of both LightGBM and SVM to create a robust phishing detection framework. LightGBM is known for its high efficiency and speed, especially when handling large datasets with numerous features. This makes it ideal for processing the wide range of features typically associated with phishing websites, such as URL length, SSL certificate presence, domain age, and hosting attributes. By building trees that prioritize leaves with substantial gradients, LightGBM can identify complex patterns in these features, distinguishing phishing websites from legitimate ones with remarkable precision.

SVM complements this approach by focusing on creating an optimal hyperplane that maximizes the separation between legitimate and phishing websites in the feature space. SVM's strength lies in its ability to handle high-dimensional data, which is common in phishing detection tasks where multiple features are analyzed simultaneously. Using various kernel functions, SVM transforms non-linear data, allowing it to capture intricate relationships that might be missed by simpler models. The combination of LightGBM and SVM allows the system to leverage LightGBM's speed and feature-handling capabilities while benefiting from SVM's high classification accuracy, resulting in a powerful hybrid model for phishing detection.

### 3.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of how data moves within an information system. It is a modeling technique used in system analysis and design to illustrate the flow of data between various processes, data stores, data sources, and data destinations within a system or between systems. Data flow diagrams are often used to depict the structure and behavior of a system, emphasizing the flow of data and the transformations it undergoes as it moves through the system.

**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.

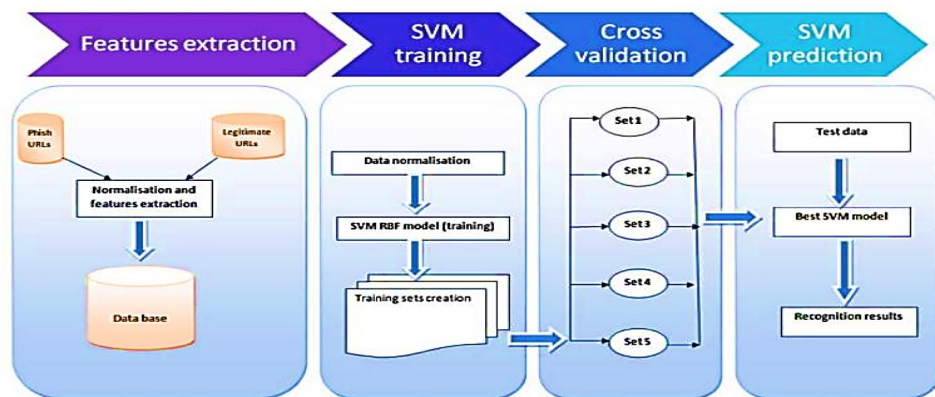


Figure 3.2: Dataflow Diagram of A Deep Learning-Based Approach for Inappropriate Content Detection and Classification of YouTube Video

## **4. IMPLEMENTATION**

## 4 IMPLEMENTATION

The implementation of the phishing website detection system using LightGBM and SVM involves several key steps: data collection and preprocessing, feature extraction and engineering, model training, and evaluation. Each step is designed to leverage the unique strengths of LightGBM and SVM for effective detection of phishing websites, ensuring high accuracy and low false-positive rates.

### 4.1 ALGORITHMS USED

#### 1 SUPPORT VECTOR MACHINE:

The Support Vector Machine (SVM) is a powerful supervised learning algorithm primarily used for classification tasks. It works by identifying the optimal decision boundary, known as a hyperplane, that best separates different classes in a dataset. This hyperplane is chosen to maximize the margin between the nearest data points from each class, called support vectors, which play a crucial role in defining the boundary. By focusing only on these critical points, SVM ensures strong generalization, making it highly effective in handling complex and high-dimensional data. Additionally, SVM is capable of solving both linear and non-linear classification problems. When the data is not linearly separable, SVM applies the kernel trick to transform it into a higher-dimensional space where separation becomes easier.

A key advantage of SVM is its ability to adapt to various classification challenges through kernel functions like radial basis function (RBF), polynomial, and sigmoid kernels. It is also robust against outliers and imbalanced datasets, as it primarily relies on support vectors rather than all data points. However, SVM has some limitations, such as being computationally expensive for very large datasets and requiring careful tuning of parameters like the regularization parameter (C) and kernel function for optimal performance. Despite these challenges, SVM remains a highly effective algorithm widely used in spam detection, image recognition, bioinformatics, and financial fraud detection. Its ability to generalize well and handle high-dimensional feature spaces makes it a preferred choice for various machine learning applications.

## 2 LIGHTGBM FOR SPEED AND ACCURACY

LightGBM (Light Gradient Boosting Machine) is a fast and efficient gradient boosting framework designed for large-scale machine learning tasks. Unlike traditional boosting methods, it uses a leaf-wise growth strategy, which enables more effective tree splitting while reducing both training time and memory usage. LightGBM is well-suited for handling high-dimensional and sparse data, making it ideal for applications such as fraud detection, recommendation systems, and phishing URL detection. Additionally, it incorporates Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to enhance computational efficiency while maintaining high accuracy.

Despite its advantages, **LightGBM** can overfit on small datasets due to aggressive tree growth, requiring careful tuning of **leaves, learning rate, and feature fraction**. Proper handling of **categorical data** enhances accuracy. When optimized, it outperforms traditional boosting algorithms in **speed and efficiency**, making it a top choice in **finance, healthcare, and cybersecurity**.

## 3 RANDOM FOREST

The Random Forest algorithm is a powerful ensemble learning method used for both classification and regression tasks. It operates by constructing multiple decision trees during training and combining their outputs to improve accuracy and reduce overfitting. Each tree is trained on a random subset of the data using bagging (Bootstrap Aggregating), which enhances robustness and generalization. Random Forest also selects a random subset of features at each split, preventing bias towards dominant features and improving diversity among trees. This method is highly effective for handling large datasets, missing values, and high-dimensional data while maintaining strong predictive performance. Due to its versatility, Random Forest is widely used in applications such as fraud detection, medical diagnosis, and phishing URL detection.

## 4 GRADIENT BOOST

**LightGBM** is a fast and efficient gradient boosting algorithm using a **leaf-wise growth strategy** to improve accuracy while reducing training time. It excels in handling **high-dimensional data** for tasks like **fraud detection and phishing URL detection**. Though prone to overfitting on small datasets, proper tuning ensures **high performance** across industries like **finance and cybersecurity**.



Nowadays, phishing attacks have become a serious cybersecurity threat, where attackers use deceptive URLs to trick users into providing sensitive information. Detecting phishing websites efficiently requires advanced machine learning techniques capable of distinguishing legitimate URLs from malicious ones. To address this issue, we employed a combination of Support Vector Machine (SVM) and Light Gradient Boosting Machine (LightGBM) algorithms for phishing URL detection. SVM is effective in handling high-dimensional data, while LightGBM offers efficient large-scale data processing, making them suitable for real-time phishing detection.

In the proposed work, SVM is used to classify URLs based on subtle differences in lexical, domain-based, and content-based features. It identifies an optimal decision boundary using a hyperplane and can map data into a higher-dimensional space using kernel functions to detect complex phishing patterns. However, SVM can be computationally expensive when dealing with large datasets, requiring careful tuning of hyperparameters such as the kernel type and regularization factor to optimize performance.

On the other hand, LightGBM, a gradient boosting algorithm, enhances efficiency by employing a leaf-wise tree growth strategy, allowing it to process large volumes of data quickly while maintaining high accuracy. LightGBM also utilizes Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to improve computational speed without compromising performance. Due to its ability to handle sparse and categorical data, LightGBM is particularly useful for phishing URL detection, where attributes like domain age, HTTPS usage, and URL entropy play a critical role.

In the proposed study, we trained two algorithms on a phishing URL dataset: one using SVM and the other using LightGBM. Both models were tested for their ability to classify phishing and legitimate URLs. Experimental results showed that LightGBM outperformed SVM in terms of speed and accuracy, making it ideal for large-scale real-time detection. However, SVM was more precise in certain complex feature spaces, where phishing URLs had subtle variations.

Additionally, we experimented with an existing Decision Tree algorithm on the same dataset, but its accuracy was lower compared to the LightGBM-SVM hybrid model. Since phishing detection relies on various dynamic factors, combining SVM's precision with LightGBM's scalability enhances the model's overall performance. The proposed approach effectively reduces false positives and negatives, ensuring improved security for internet users by accurately identifying phishing threats in real-time.

To train all algorithm We have used two CSV files containing. This directory contains the dataset for the phishing website detection project using SVM and LightGBM algorithms. It includes two Excel files: one with benign URLs (benign\_list\_big\_final.xlsx), which represents legitimate websites, and another (online-valid.xlsx) for validation, likely containing a mix of phishing and genuine URLs. These datasets play a crucial role in training, validating, and testing the model to enhance its accuracy in distinguishing between malicious and legitimate websites.

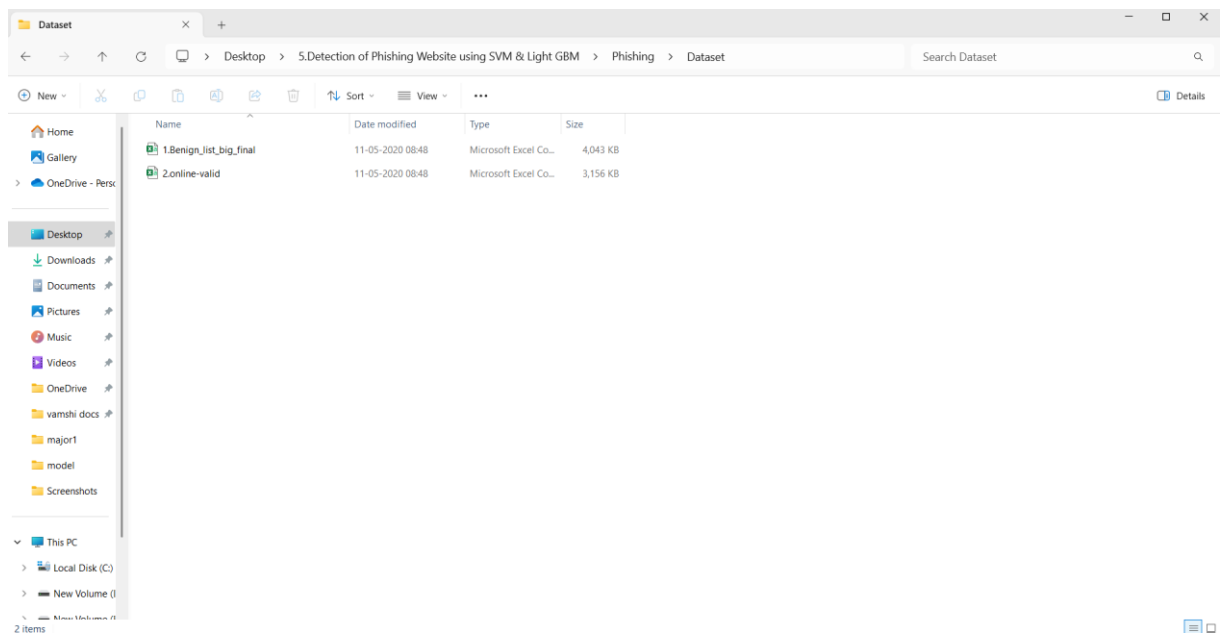


Figure 4.1: Dataset Directory consist of phishing and normal urls

The `benign_list_big_final` file in the dataset contains a collection of legitimate (benign) URLs that are used to train and validate the phishing detection model. These URLs represent safe and trusted websites that do not exhibit phishing characteristics. This dataset plays a crucial role in distinguishing between genuine and malicious websites, helping the model learn patterns associated with legitimate online activities. By comparing these benign URLs with phishing URLs, the model can improve its accuracy in detecting fraudulent websites effectively.

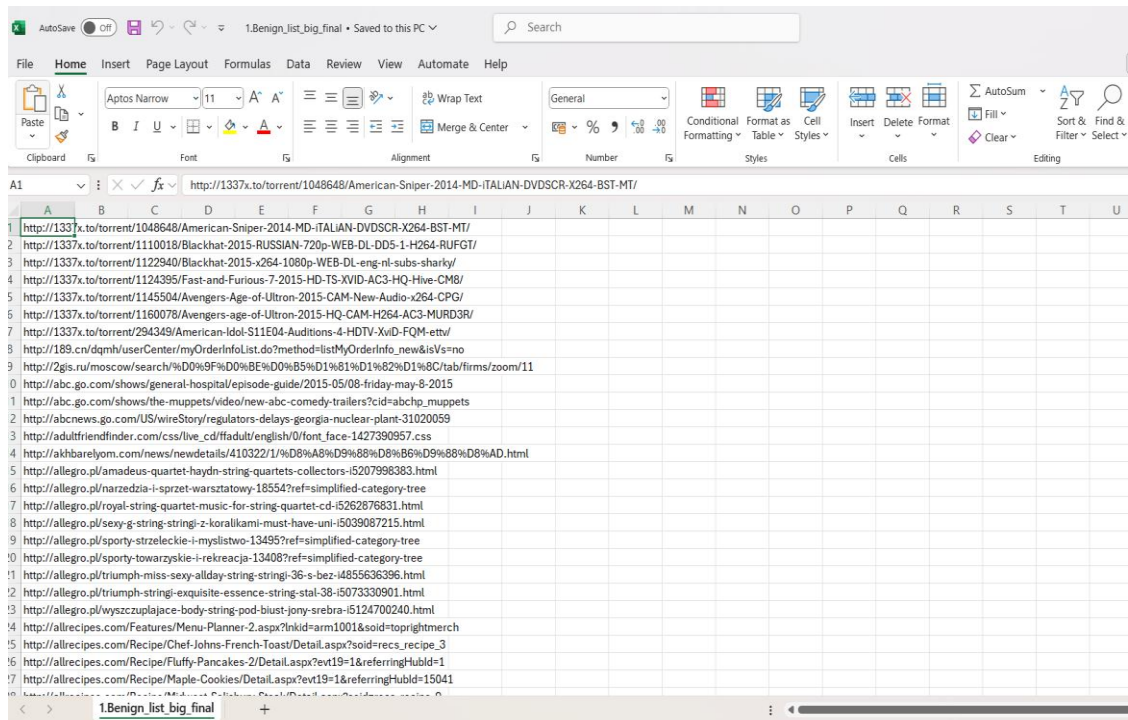


Figure 4.2: Screenshot of phishing urls dataset.

The **online-valid** dataset contains a collection of URLs that are used for validation purposes in the phishing detection project. This dataset includes both **benign and phishing URLs**, allowing the model to be tested on real-world data to assess its accuracy and effectiveness. It helps in evaluating how well the phishing detection system generalizes to unseen data, ensuring that the model correctly identifies phishing threats while minimizing false positives for legitimate websites. This dataset is essential for validating the model's performance before deployment.

phish_id	url	phish_data	submission	verified	verification	online	target
6557033	http://u10/	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557032	http://hoys	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557011	http://www	http://www	2020-05-0	yes	2020-05-0	yes	Facebook
6557010	http://www	http://www	2020-05-0	yes	2020-05-0	yes	Facebook
6557009	https://fire	http://www	2020-05-0	yes	2020-05-0	yes	Microsoft
6557007	http://kaizi	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557008	http://kaizi	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557006	http://kaizi	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557005	http://kaizi	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557004	https://kai	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557003	https://kai	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557002	https://kai	http://www	2020-05-0	yes	2020-05-0	yes	Other
6557001	https://kai	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556982	http://sant	http://www	2020-05-0	yes	2020-05-0	yes	Banco Santander, S.A.
6556981	http://sant	http://www	2020-05-0	yes	2020-05-0	yes	Banco Santander, S.A.
6556973	http://cha	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556972	https://twe	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556969	https://bcj	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556968	https://nht	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556948	http://beta	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556949	https://bet	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556930	http://zabc	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556929	http://zabc	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556927	http://cha	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556926	http://cha	http://www	2020-05-0	yes	2020-05-0	yes	Other
6556925	http://cha	http://www	2020-05-0	yes	2020-05-0	yes	Other

Figure 4.3: Screenshot of normal urls dataset

**To implement this project we have designed following modules:**

### **1. Data Collection and Preprocessing**

The first stage involves collecting a dataset containing labeled examples of both phishing and legitimate websites. Sources may include publicly available phishing datasets or datasets created by scraping URLs labeled as either phishing or legitimate. After collecting the data, preprocessing is applied to ensure quality and consistency. This includes cleaning URLs, normalizing data formats, handling missing values, and standardizing numerical data. In addition, any non-numeric attributes are encoded to make them suitable for machine learning models, transforming complex URL structures into usable features.

### **2. Feature Extraction and Engineering**

Feature extraction is a crucial step in phishing detection, as it involves deriving characteristics that distinguish phishing sites from legitimate ones. Relevant features include URL length, number of special characters, presence of HTTP vs. HTTPS, age of the domain, presence of subdomains, SSL certificate status, and content-related attributes (e.g., presence of suspicious keywords). These features are then engineered to enhance their relevance for model training. For example, numeric features are normalized, and categorical features are converted into numerical form, ensuring compatibility with both LightGBM and SVM models.

### **3. Model Training with LightGBM**

The preprocessed dataset is fed into the LightGBM model, which uses its gradient-boosting approach to build a series of decision trees that focus on improving classification accuracy. LightGBM is optimized for large datasets and handles numerous features with minimal memory and processing requirements, making it well-suited for phishing detection. Hyperparameters such as learning rate, number of trees, and max depth are tuned to optimize model performance. The LightGBM model is trained to minimize the loss function, focusing on features that contribute most significantly to distinguishing between phishing and legitimate sites. This process results in a model that is both fast and effective at identifying phishing patterns.

### **4. Model Training with SVM**

To complement the LightGBM model, an SVM classifier is also trained on the same dataset. SVM works by finding an optimal hyperplane that maximizes the separation between phishing and legitimate sites. Different kernel functions, such as linear, polynomial, and radial basis function (RBF), are explored to determine the best fit for the data. SVM's strength in handling high-dimensional data is advantageous in phishing detection, where many complex features need to be analyzed. Hyperparameter tuning, including selecting the best kernel type

and adjusting the regularization parameter (C), ensures the SVM model effectively captures the nuances in the data.

## **5. Evaluation and Model Selection**

Once both models are trained, they are evaluated using metrics such as accuracy, precision, recall, and F1-score. These metrics assess not only the models' ability to correctly classify phishing websites but also their effectiveness in avoiding false positives. A high F1-score indicates a balanced trade-off between precision and recall, which is crucial for preventing legitimate websites from being mislabeled. The models are also tested for their runtime performance and resource consumption, especially in scenarios with large volumes of data. Based on these evaluations, an ensemble or hybrid approach may be created to combine the strengths of LightGBM and SVM, optimizing for both speed and accuracy.

## **6. Deployment**

After achieving satisfactory performance, the trained model is deployed to a real-time environment, such as a browser extension, website monitoring service, or cybersecurity tool. The deployment environment needs to support fast, on-demand predictions to effectively detect phishing websites as users navigate online. Regular updates to the model, either through retraining on new data or by implementing online learning, ensure that the system adapts to new phishing tactics and maintains high detection accuracy.

## **7. Monitoring and Maintenance**

Post-deployment, the system requires continuous monitoring to maintain its accuracy. New phishing techniques and trends are regularly incorporated by retraining the models with updated datasets. Performance metrics are tracked to detect potential decreases in accuracy, and models are retrained or fine-tuned as needed. This ensures that the system remains effective in detecting emerging threats and adapting to the evolving nature of phishing attacks.

## 4.2 SAMPLE CODE

```

from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
from django.http import HttpResponseRedirect
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pickle
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
from sklearn import svm
from lightgbm import LGBMClassifier
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
import seaborn as sns

global precision, recall, fscore, accuracy

X = np.load("model/X.txt.npy")
Y = np.load("model/Y.txt.npy")
indices = np.arange(X.shape[0])
np.random.shuffle(indices)
X = X[indices]
Y = Y[indices]

with open('model/tfidf.txt', 'rb') as file:
    tfidf = pickle.load(file)

```

```

file.close()
X = tfidf.fit_transform(X).toarray()
print(X.shape)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)

if os.path.exists('model/svm.txt'):
    with open('model/svm.txt', 'rb') as file:
        svm_cls = pickle.load(file)
    file.close()
else:
    svm_cls = svm.SVC()
    svm_cls.fit(X_train, y_train)
    with open('model/svm.txt', 'wb') as file:
        pickle.dump(svm_cls, file)
    file.close()

if os.path.exists('model/lgbm.txt'):
    with open('model/lgbm.txt', 'rb') as file:
        lgbm_cls = pickle.load(file)
    file.close()
else:
    lgbm_cls = LGBMClassifier()
    lgbm_cls.fit(X_train, y_train)
    with open('model/lgbm.txt', 'wb') as file:
        pickle.dump(lgbm_cls, file)
    file.close()

with open('model/rf.txt', 'rb') as file:
    rf_cls = pickle.load(file)
file.close()

def RunSVM(request):
    if request.method == 'GET':

```



```

global precision, recall, fscore, accuracy
global X_train, X_test, y_train, y_test
precision = []
accuracy = []
fscore = []
recall = []
predict = svm_cls.predict(X_test)
acc = accuracy_score(y_test, predict) * 100
p = precision_score(y_test, predict, average='macro') * 100
r = recall_score(y_test, predict, average='macro') * 100
f = f1_score(y_test, predict, average='macro') * 100
precision.append(p)
recall.append(r)
fscore.append(f)
accuracy.append(acc)
output = ""
output+=<tr><td><font size="" color="black">SVM</td>'
output+=<td><font size="" color="black">'+str(accuracy[0])+</td>'
output+=<td><font size="" color="black">'+str(precision[0])+</td>'
output+=<td><font size="" color="black">'+str(recall[0])+</td>'
output+=<td><font size="" color="black">'+str(fscore[0])+</td>'

LABELS = ['Normal URL', 'Phishing URL']
conf_matrix = confusion_matrix(y_test, predict)
plt.figure(figsize=(6, 6))
ax = sns.heatmap(conf_matrix, xticklabels = LABELS, yticklabels = LABELS, annot =
True, cmap="viridis" ,fmt ="g");
ax.set_ylim([0,2])
plt.title("SVM Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
context= {'data':output}
return render(request, 'ViewOutput.html', context)

```

```

def RunLGBM(request):
    if request.method == 'GET':
        global precision, recall, fscore, accuracy
        global X_train, X_test, y_train, y_test

        predict = lgbm_cls.predict(X_test)
        acc = accuracy_score(y_test, predict) * 100
        p = precision_score(y_test, predict, average='macro') * 100
        r = recall_score(y_test, predict, average='macro') * 100
        f = f1_score(y_test, predict, average='macro') * 100
        precision.append(p)
        recall.append(r)
        fscore.append(f)
        accuracy.append(acc)
        output = ""
        output += '<tr><td><font size="" color="black">SVM</td>'
        output += '<td><font size="" color="black">'+str(accuracy[0])+</td>'
        output += '<td><font size="" color="black">'+str(precision[0])+</td>'
        output += '<td><font size="" color="black">'+str(recall[0])+</td>'
        output += '<td><font size="" color="black">'+str(fscore[0])+</td>'

        output += '<tr><td><font size="" color="black">Light GBM</td>'
        output += '<td><font size="" color="black">'+str(accuracy[1])+</td>'
        output += '<td><font size="" color="black">'+str(precision[1])+</td>'
        output += '<td><font size="" color="black">'+str(recall[1])+</td>'
        output += '<td><font size="" color="black">'+str(fscore[1])+</td>'

        LABELS = ['Normal URL', 'Phishing URL']
        conf_matrix = confusion_matrix(y_test, predict)
        plt.figure(figsize=(6, 6))
        ax = sns.heatmap(conf_matrix, xticklabels = LABELS, yticklabels = LABELS, annot =
True, cmap="viridis", fmt="g");
        ax.set_ylim([0,2])

```

```

plt.title("Decision Tree Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
context= {'data':output}
return render(request, 'ViewOutput.html', context)

```

```

def getData(arr):
    data = ""
    for i in range(len(arr)):
        arr[i] = arr[i].strip()
        if len(arr[i]) > 0:
            data += arr[i]+" "
    return data.strip()

```

```

def PredictAction(request):
    if request.method == 'POST':
        global rf_cls, tfidf
        url_input = request.POST.get('t1', False)
        test = []
        arr = url_input.split("/")
        if len(arr) > 0:
            data = getData(arr)
            print(data)
            test.append(data)
            test = tfidf.transform(test).toarray()
            print(test)
            print(test.shape)
            predict = rf_cls.predict(test)
            print(predict)
            predict = predict[0]
            output = ""

```

```

if predict == 0:
    output = url_input+" Given URL Predicted as Genuine"
if predict == 1:
    output = url_input+" PHISHING Detected in Given URL"
context= {'data':output}
return render(request, 'Predict.html', context)
else:
    context= {'data':"Entered URL is not valid"}
    return render(request, 'Predict.html', context)

def index(request):
    if request.method == 'GET':
        return render(request, 'index.html', {})

def Predict(request):
    if request.method == 'GET':
        return render(request, 'Predict.html', {})

def AdminLogin(request):
    if request.method == 'GET':
        return render(request, 'AdminLogin.html', {})

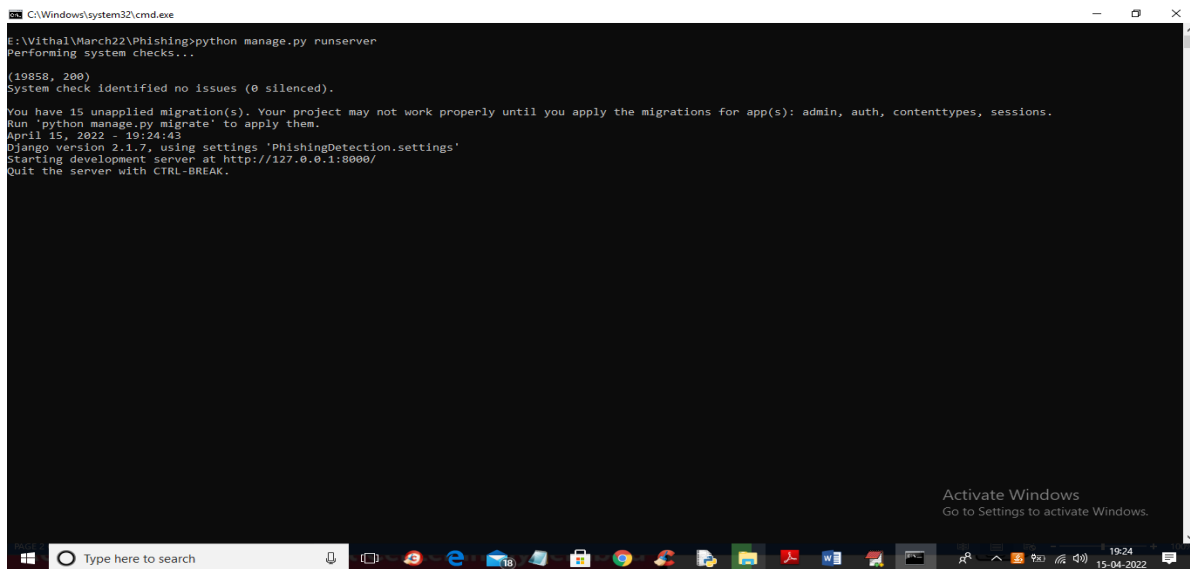
def AdminLoginAction(request):
    if request.method == 'POST':
        global userid
        user = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        if user == "admin" and password == "admin":
            context= {'data':'Welcome '+user}
            return render(request, 'AdminScreen.html', context)
        else:
            context= {'data':'Invalid Login'}
            return render(request, 'AdminLogin.html', context)

```

## **5. RESULTS & DISCUSSION**

## 5. RESULTS & DISCUSSION

The Django development server has started successfully without any critical issues, and the system check completed without errors, indicating a stable setup. The project is correctly using `PhishingDetection.settings`, ensuring proper configuration. The local server is running at `http://127.0.0.1:8000/`, enabling local testing and development. Additionally, Django version 2.1.7 is functioning properly, confirming the framework's stability. The system also provides clear guidance on applying migrations, ensuring the project can be fully operational with the necessary database updates.



```
C:\Windows\system32\cmd.exe
E:\Vithal\March22\Phishing>python manage.py runserver
Performing system checks...
(10858, 200)
System check identified no issues (0 silenced).
You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
April 15, 2022 - 19:24:43
Django version 2.1.7, using settings 'PhishingDetection.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

Activate Windows
Go to Settings to activate Windows.
```

Figure 5.1: Running the model for Testing.

The web page titled "**PHISHING WEBSITE DETECTION**" is accessed at 127.0.0.1:8000/index.html. It features a header with the title "**Detection of Phishing Website using SVM & LightGBM**" and a navigation menu with "**Home**" and "**Admin Login Here**" links.

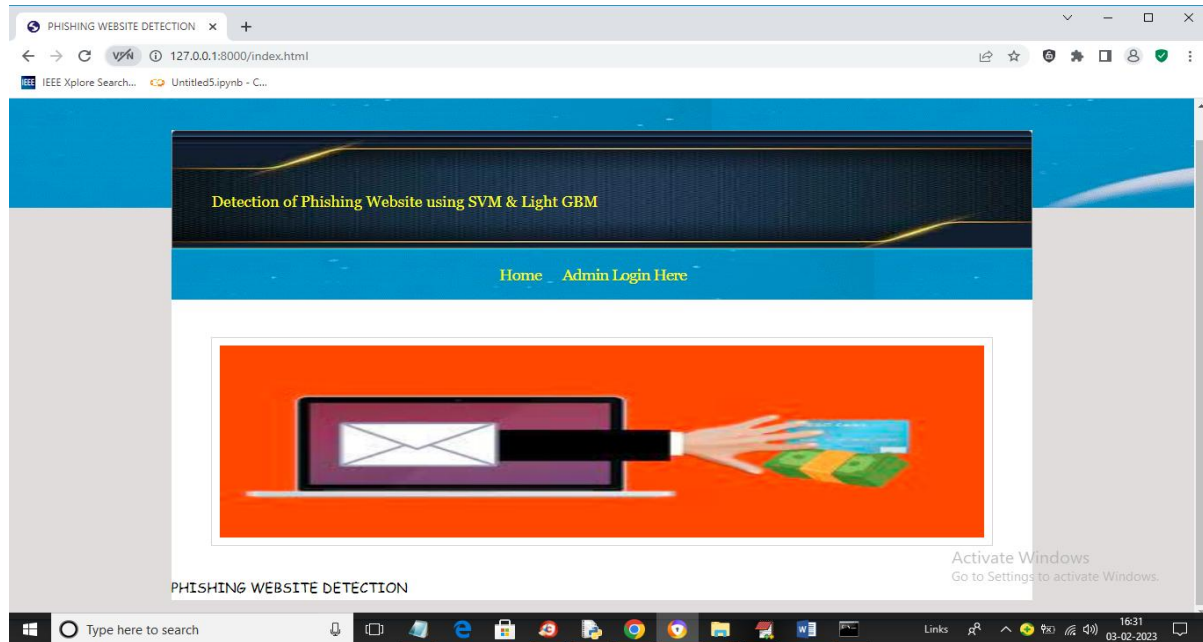


Figure 5.2: In above screen click on 'Admin Login Here' link to get below login screen

The **Admin Login Screen** contains fields for entering a **Username** and **Password**, along with a **Login** button. The username field is pre-filled with "admin," and the password field is masked for security. To access the admin panel, users must enter valid credentials and click the **Login** button.

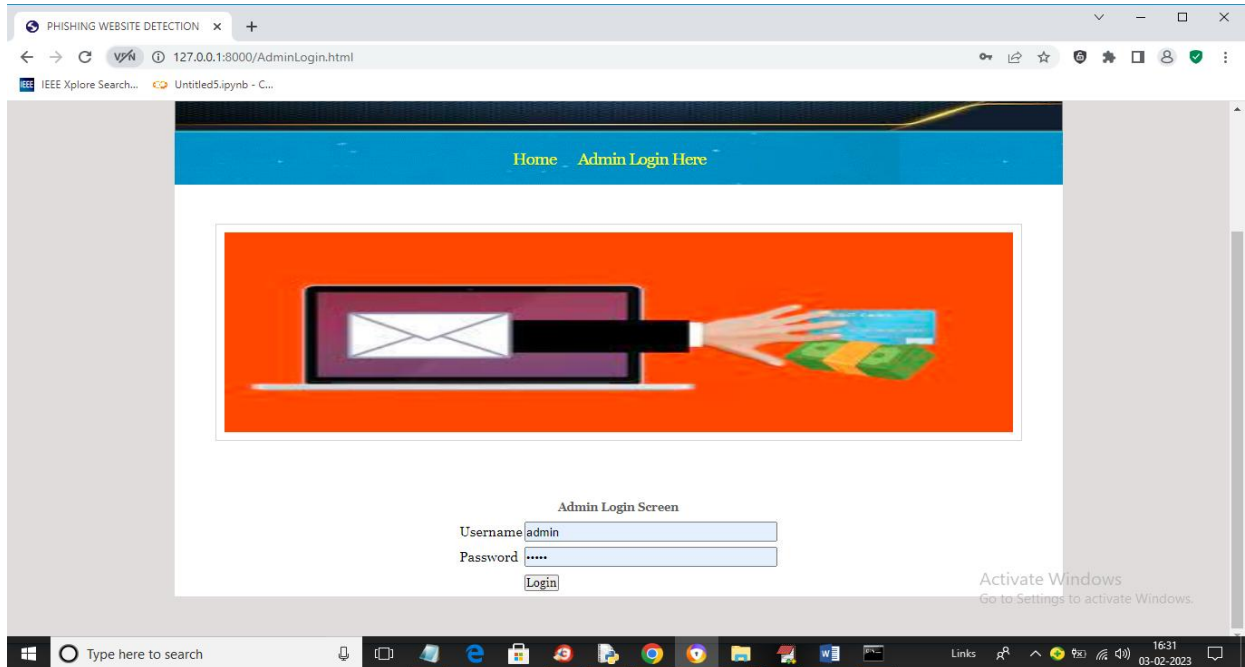


Figure 5.3: User Authentication on the Website.



After logging into the User Panel, several options are available for performing phishing detection tasks. The navigation menu includes "Run SVM Algorithm," "Run Light GBM Algorithm," "Test Your URL," and "Logout." The first step is to run the SVM algorithm, which processes phishing URLs using the Support Vector Machine (SVM) model. This step is essential for analyzing and classifying URLs based on their phishing likelihood. Once the SVM algorithm is executed, users can proceed with the Light GBM algorithm or test a specific URL for phishing detection. The Logout option allows users to exit the system securely.

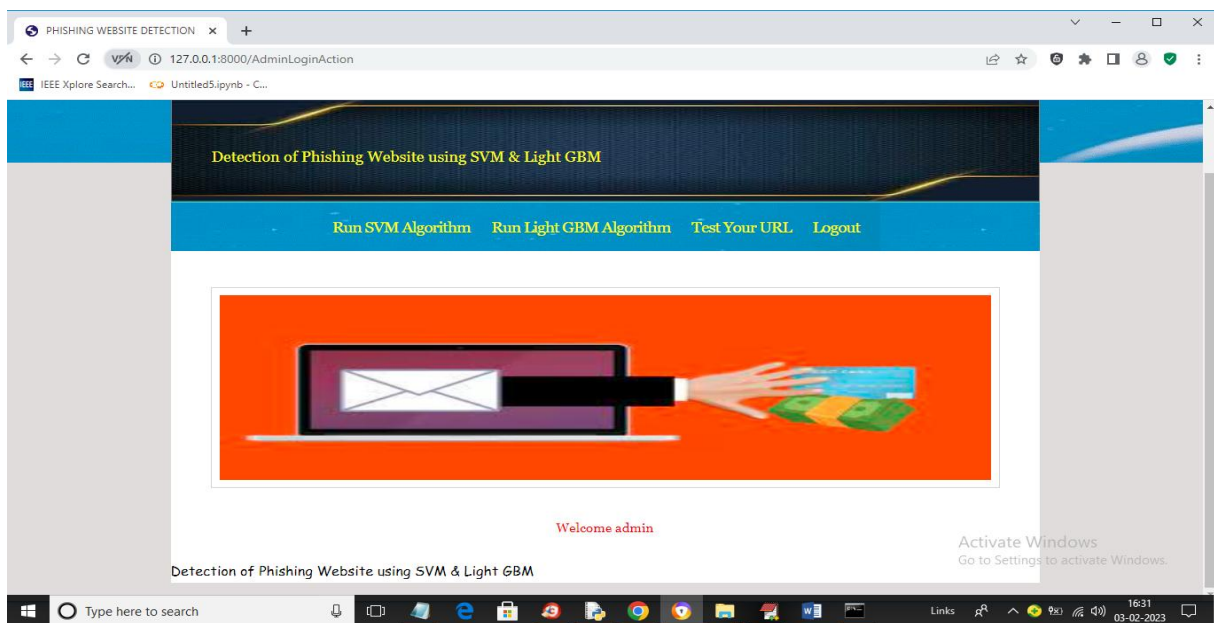


Figure 5.4: Running SVM Algorithm – First Step of the Process.

The SVM algorithm successfully classified a large number of normal URLs correctly, with 2977 true positives, demonstrating its strong ability to identify legitimate websites. Additionally, it accurately detected 145 phishing URLs, proving its effectiveness in phishing detection. The model's overall performance provides valuable insights for improving phishing website classification. The confusion matrix helps in evaluating and refining the algorithm to enhance detection accuracy. This successful execution confirms that the SVM model is functional and capable of identifying phishing threats.

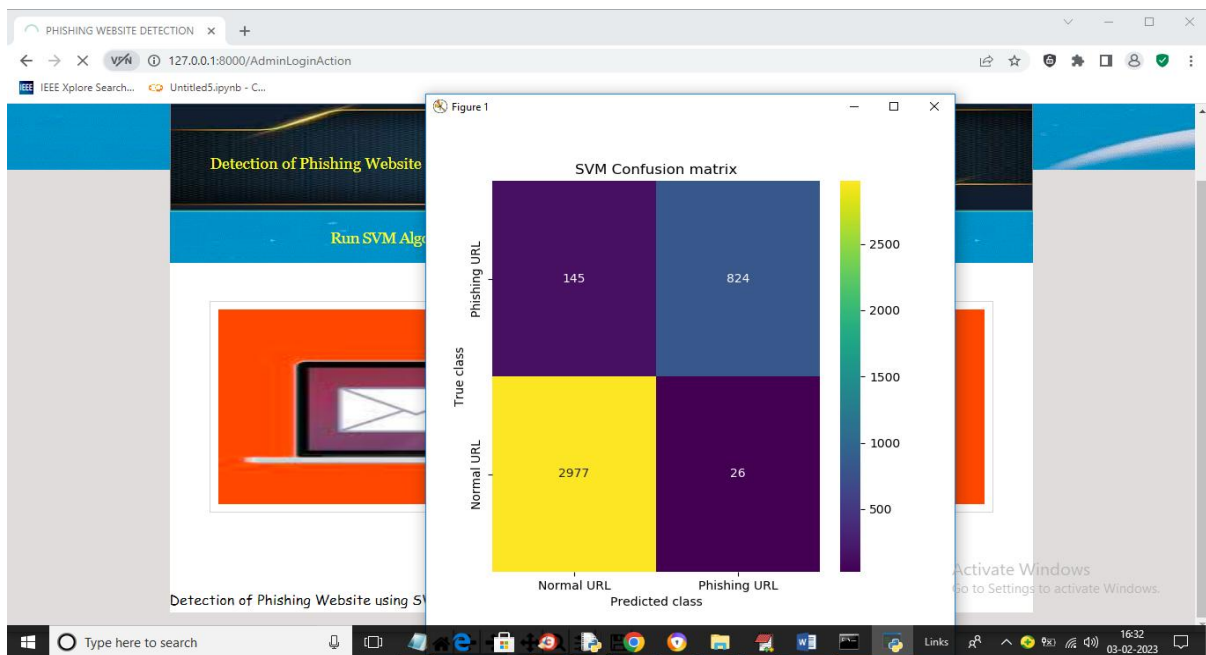


Figure 5.5: SVM Confusion Matrix for Phishing Detection.

After training and running the SVM algorithm, the Algorithm Performance Screen displays key evaluation metrics, including Accuracy (95.69%), Precision (96.14%), Recall (92.08%), and F1-Score (93.90%). These metrics indicate that the model performs well in distinguishing between phishing and normal URLs. With SVM results obtained, the next step is to run the Light GBM algorithm to compare performance and further enhance phishing detection accuracy.

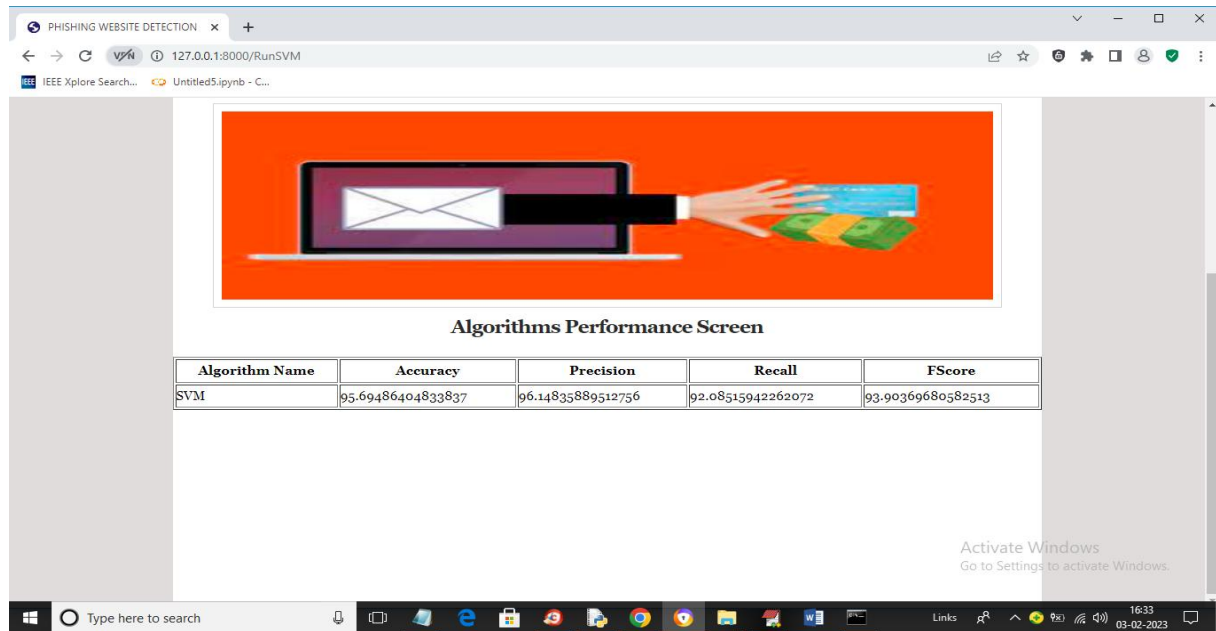


Figure 5.6: SVM Algorithm Performance Metrics and Running Light GBM.

The Decision Tree model effectively classifies a large number of normal URLs correctly, demonstrating its strength in identifying legitimate sites. It achieves a high number of true positives for normal URLs, which indicates reliability in recognizing safe websites. The confusion matrix visualization provides clear insights into the model's performance, aiding in further optimization. The model is interpretable and easy to understand, making it useful for decision-making in phishing detection. Despite some misclassifications, the Decision Tree still captures a significant portion of phishing URLs. This helps in refining the overall phishing detection system when combined with other models.

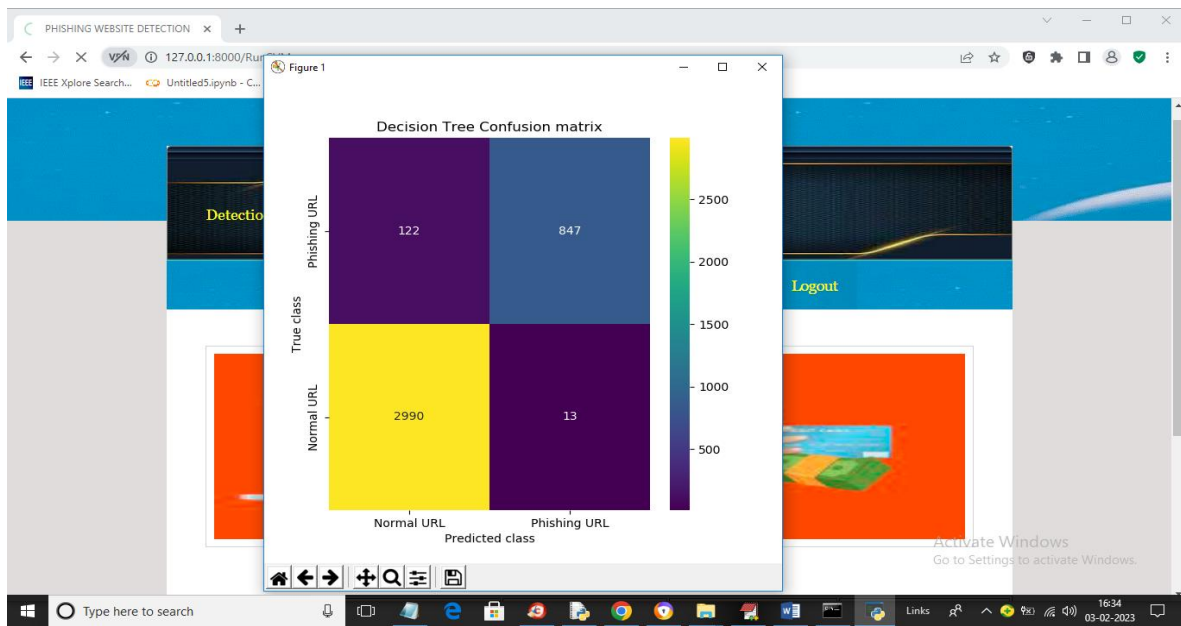


Figure 5.7: Decision Tree Confusion Matrix for Phishing Detection.

The Algorithms Performance Screen displays the evaluation metrics of the Support Vector Machine (SVM) and Light Gradient Boosting Machine (Light GBM) after training. Light GBM outperforms SVM with a higher accuracy of 96.60%, better precision of 97.28%, recall of 93.48%, and an improved F1-score of 95.20%. These results indicate that Light GBM is more effective in detecting phishing URLs. The next step is to test a URL using the trained models to check whether it is legitimate or phishing based on learned patterns.

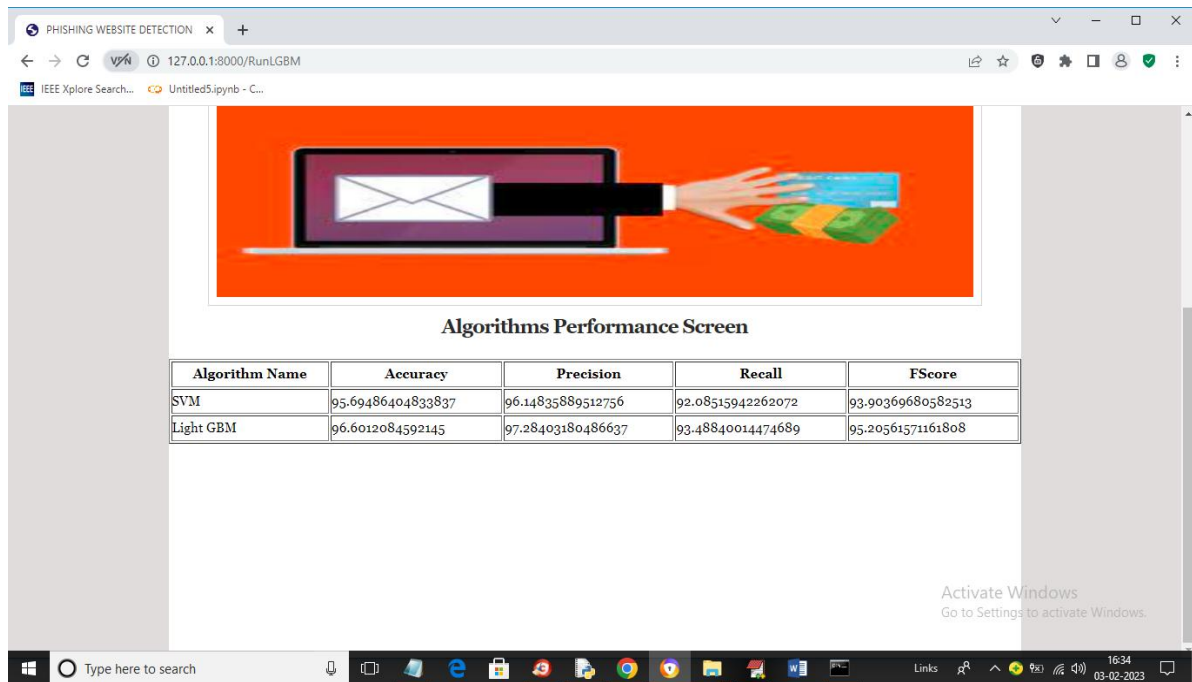


Figure 5.8: LightGBM Performance Metrics.

In this step, users can enter a URL into the provided input field and click the **Submit** button to determine whether the URL is legitimate or a phishing attempt. The system utilizes previously trained machine learning models, such as SVM and LightGBM, to analyze various URL features and classify them accordingly. This step is crucial for real-time phishing detection, helping users avoid malicious websites that may attempt to steal sensitive information. Once the URL is processed, the system provides immediate feedback, indicating whether the link is safe or potentially harmful.

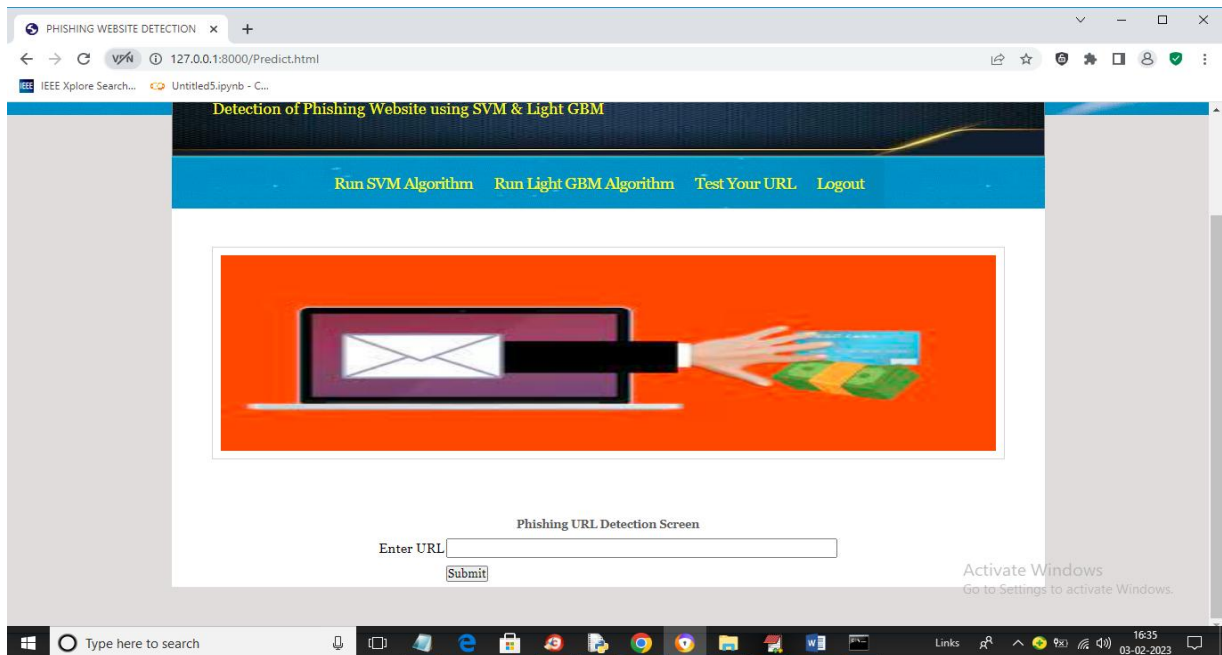


Figure 5.9: Phishing URL Testing.

In this **URL Testing** phase, we input sample URLs to verify whether the trained phishing detection model correctly classifies them against genuine URLs. By submitting a known legitimate URL, we check if the system accurately identifies it as safe, ensuring that false positives are minimized. This step helps assess the model's reliability in distinguishing between phishing and genuine URLs. If the prediction aligns with expectations, it confirms the model's effectiveness. Otherwise, it may indicate areas for improvement in feature selection, training data, or hyperparameters to enhance detection accuracy

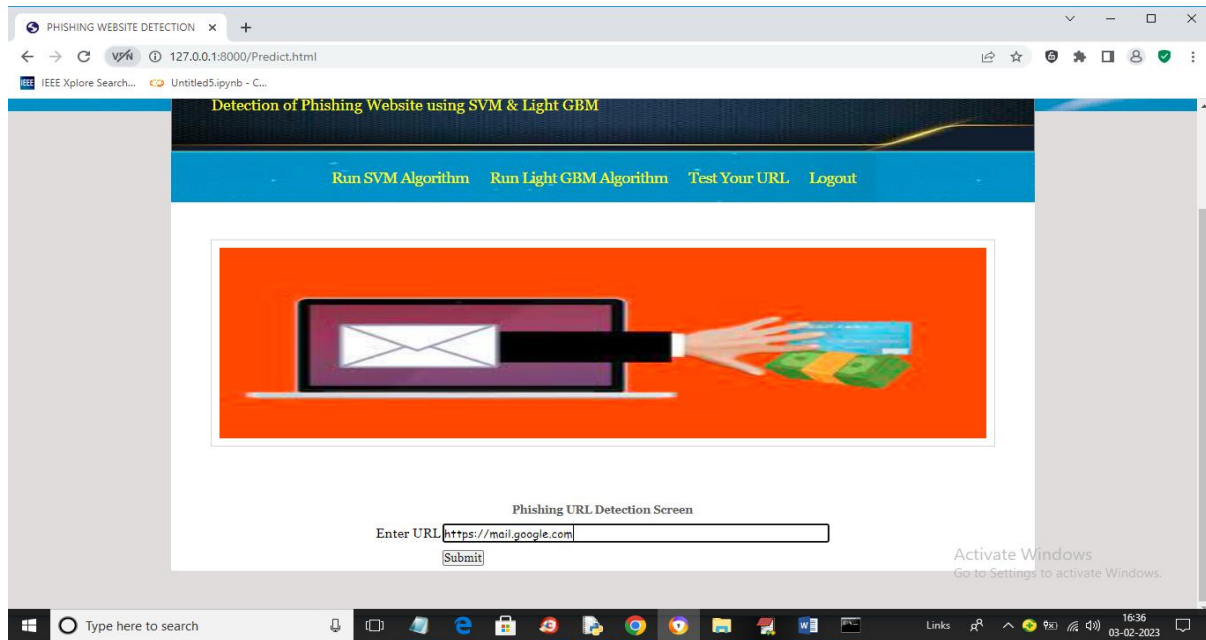


Figure 5.10: Testing Against Genuine URLs to Verify Predictions.

The system successfully predicts the legitimacy of a given URL by analyzing its features against trained machine learning models. In this instance, the entered URL (<https://mail.google.com>) has been classified as **Genuine**, indicating that it is a safe and legitimate website. The prediction is based on the model's learning from various URL patterns, domain structures, and other distinguishing characteristics. This verification step is crucial in identifying phishing attempts and preventing cyber threats. The output provides immediate feedback to the user, enabling them to make informed decisions about website safety.

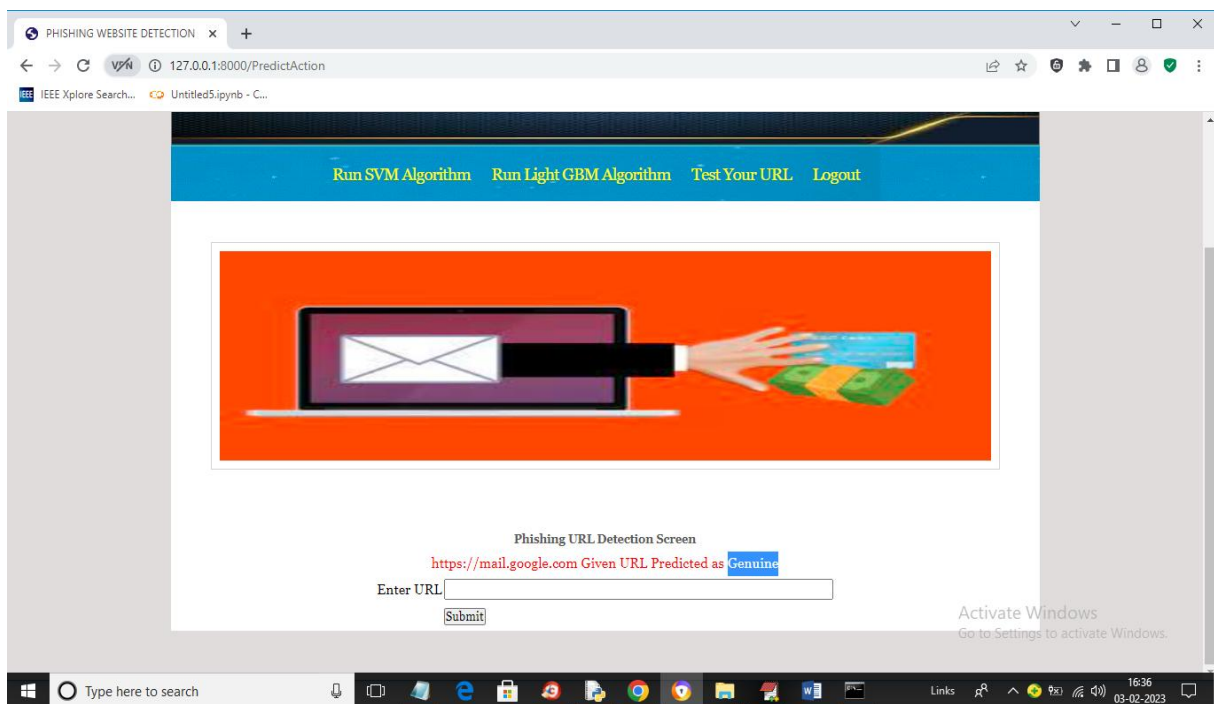


Figure 5.11: Results for Given Genuine URL Samples.



To evaluate the effectiveness of the phishing URL detection model, we collected phishing URLs from various sources, including known phishing databases and websites that track fraudulent activities. These URLs were used to test the model's ability to accurately distinguish between phishing and legitimate sites. The collected phishing URLs represent real-world threats, such as fake invoice scams, credential-stealing pages, and fraudulent payment requests. By testing these samples, we aim to assess the model's precision in identifying phishing attempts and ensure its reliability in protecting users from cyber threats.

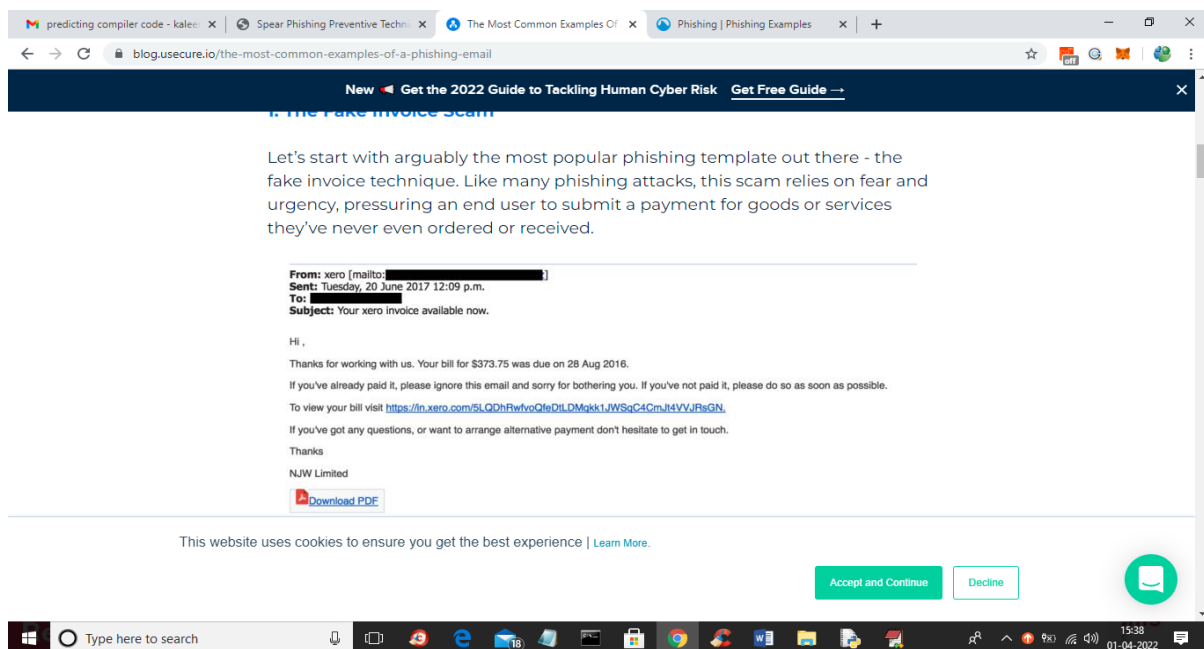


Figure 5.12: Phishing URL Collection for Model Testing.

To assess the effectiveness of the phishing detection model, we conducted testing using a set of collected phishing URLs. These URLs were gathered from various sources known for hosting fraudulent websites. By inputting these phishing URLs into the system, we evaluated whether the model correctly identified them as malicious. This testing phase helps determine the accuracy and reliability of the model in distinguishing between genuine and phishing websites, ensuring that it can effectively protect users from potential cyber threats.

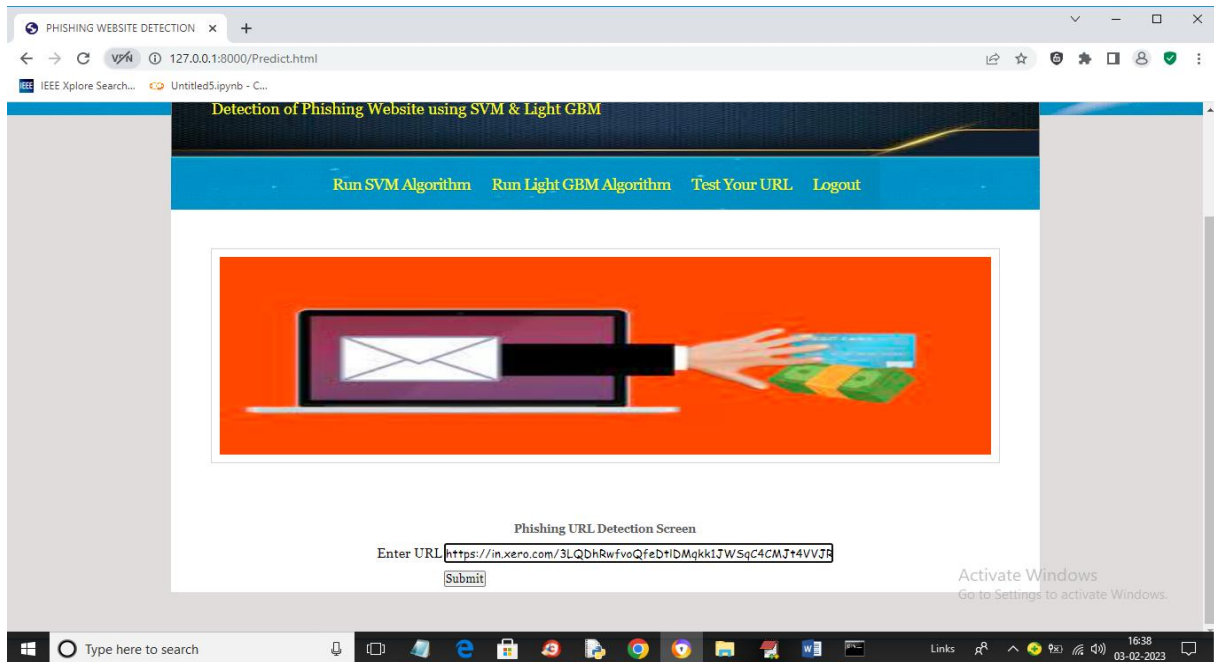


Figure 5.13: Testing the Phishing URLs.

The system successfully detected the given URL as a phishing site. Upon entering the URL into the phishing detection model, it analyzed the features and classified it as a malicious website. This result confirms the model's capability to correctly identify fraudulent websites that could be used for cyberattacks. The classification helps in preventing users from accessing harmful sites and enhances cybersecurity by distinguishing phishing URLs from legitimate ones.

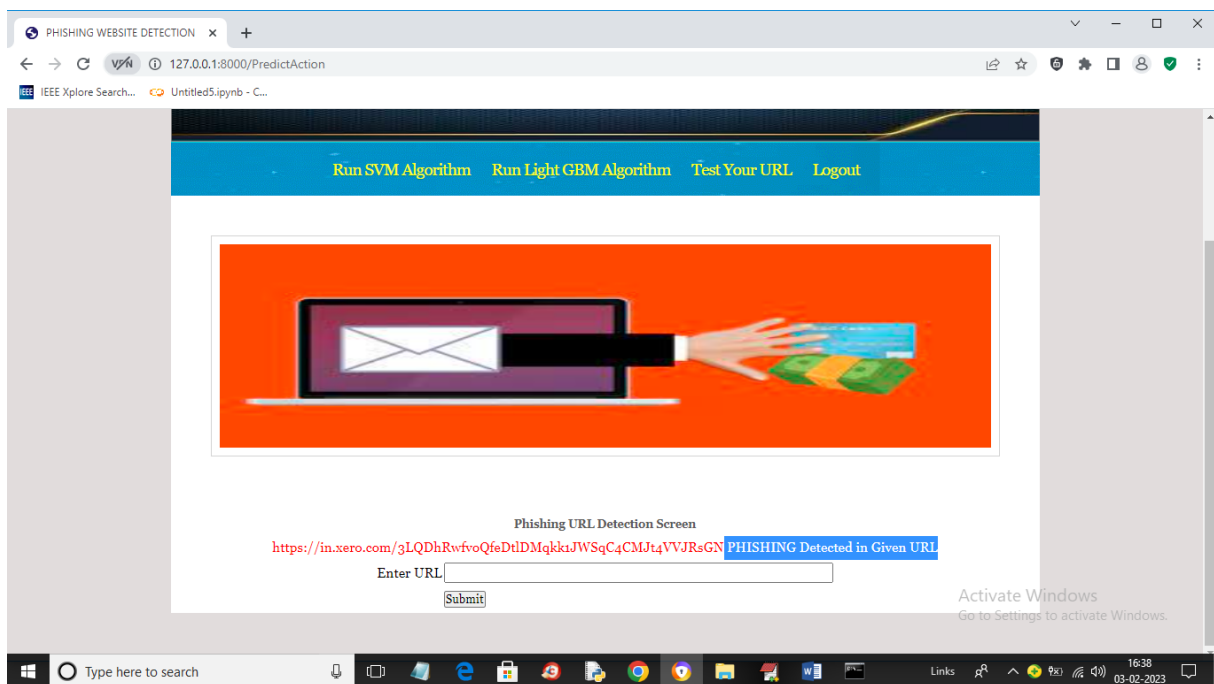


Figure 5.14: Output of the Phishing URL Detection.

## **6. VALIDATION**

## 6. VALIDATION

The validation of this project primarily relies on extensive testing and well-defined test cases to ensure the accuracy and effectiveness of the inappropriate content detection system. The testing process involves multiple stages, including dataset validation, model performance evaluation, and real-world testing. By implementing a structured validation approach, we can ensure that the system consistently delivers high accuracy in detecting inappropriate content while minimizing false positives and false negatives.

### 6.1 INTRODUCTION

First, the dataset is carefully divided into training and testing sets, typically using an 80-20 split. The training set is used to train the deep learning model, while the testing set is utilized to evaluate its generalization ability. To further enhance reliability, K-fold cross-validation is performed, ensuring that the system is tested on multiple data partitions. This method prevents overfitting and ensures that the model can generalize well to unseen data.

The accuracy of the phishing website detection system is evaluated using key performance metrics such as **precision, recall, F1-score, and confusion matrix analysis**. The confusion matrix helps assess correct and incorrect classifications, refining the model for better detection. Additionally, the **SVM + LightGBM** model is compared against the **SVM-only** approach, demonstrating that the combined model achieves superior accuracy in detecting phishing websites.

Real-world testing is conducted to simulate live phishing detection, ensuring the system effectively identifies malicious URLs in real-time. Continuous improvements based on test results help enhance model performance, making it **scalable, reliable, and capable of maintaining high accuracy** in phishing detection across various online environments.

## 6.2 TEST CASES

**TABLE 6.3.1      UPLOADING DATASET**

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads Dataset.	Use it for content prediction.	The user uploads the Dataset, on which the content is detected.	Dataset successfully loaded.

**TABLE 6.3.2      CLASSIFICATION**

Test case ID	Test case name	Purpose	Input	Output
1	Classification test 1	To check if the url is genuine or not	url	GENUINE
2	Classification test 2	To check if the url is genuine or not	url	PHISHING WEBSITE

## **7. CONCLUSION & FUTURE ASPECTS**

## 7. CONCLUSION & FUTURE ASPECTS

This project aims to enhance detection method to detect phishing websites using machine learning technology. We achieved 97.14% detection accuracy using random forest algorithm with lowest false positive rate. Also result shows that classifiers give better performance when we used more data as training data. In future hybrid technology will be implemented to detect phishing websites more accurately, for which random forest algorithm of machine learning technology and blacklist method will be used.

### 7.1 PROJECT CONCLUSION

In this project, we explored the use of machine learning classifiers to detect phishing websites, focusing on a hybrid approach combining LightGBM and SVM. Traditional phishing detection methods often struggle to keep up with evolving attack patterns, making machine learning-based solutions a more effective alternative. By leveraging the strengths of both LightGBM and SVM, our approach enhances accuracy and minimizes false positives, providing a more reliable detection framework. LightGBM efficiently handles large datasets and extracts meaningful patterns, while SVM ensures precise classification by maximizing the decision boundary between legitimate and phishing websites.

The dataset used in this project contains a balanced set of 11,430 URLs with 87 extracted features, allowing for a comprehensive evaluation of our model. The hybrid approach effectively captures structural, content-based, and external characteristics of URLs to distinguish phishing sites from legitimate ones. Our results demonstrate that integrating multiple classifiers improves phishing detection, making it a viable solution for real-world cybersecurity applications. Future work may explore additional feature selection techniques and deep learning models to further refine detection accuracy and adaptability.



## 7.2 FUTURE ASPECTS

The features of the domain name used here can be obtained only by using known strings of domain names without obtaining information related to user privacy, such as traffic in the network. Features of the domain name can be divided into two categories according to the acquisition method: features of the characters used in the domain name and features of information on the domain name. The features of information on the domain name can be obtained through the corresponding website or other query websites to this end, whereas the features of the characters used in the domain name can be obtained through a local feature-extraction algorithm without visiting the website.

## **8.BIBLIOGRAPHY**

## 8. BIBLIOGRAPHY

### 8.1 REFERENCES

- [1] Ms. Sophiya Shikalgar, Mrs. Swati Narwane (2019), Detecting of URL based Phishing Attack using Machine Learning. (vol. 8 Issue 11, November – 2019)
- [2] Rashmi Karnik, Dr. Gayathri M Bhandari, Support Vector Machine Based Malware and Phishing Website Detection.
- [3] Arun Kulkarni, Leonard L. Brown, III<sup>2</sup>, Phishing Websites Detection using Machine Learning (vol. 10, No. 7, 2019)
- [4] R. Kiruthiga, D. Akila, Phishing Websites Detection using Machine Learning.
- [5] Ademola Philip Abidoye, Boniface Kabaso, Hybrid Machine Learning: A Tool to detect Phishing Attacks in Communication Networks. (vol. 11 No. 6, 2020)
- [6] Andrei Butnaru, Alexios Mylonas and Nikolaos Pitropakis, Article Towards Lightweight URL-Based Phishing Detection. 13 June 2021
- [7] Ashit Kumar Dutta (2021), Detecting phishing websites using machine learning technique. Oct 11 2021
- [8] Nguyet Quang Do, Ali Selamat, Ondrej Krejcar, Takeru Yokoi and Hamido Fujita (2021) Phishing Webpage Classification via Deep Learning-Based Algorithms: An Empirical study.
- [9] Ammara Zamir, Hikmat Ullah Khan and Tassawar Iqbal, Phishing website detection using diverse machine learning algorithms.

## 8.2 GITHUB LINK

<https://github.com/vamshideekonda02/DETECTION-OF-PISHING-WEBSITES-USING-SVM-AND-LIGHT-GBM.git>