

Phishing Website Detection Using Machine Learning

Neha Nandkishor Kotkar¹, Prachi Vijay Pathare²

¹ Assistant Professor, Sir Parashurambhau College, Pune, Maharashtra, India

² Student, Sir Parashurambhau College, Pune, Maharashtra, India

Email: nkotkar.2611@gmail.com, prachipathare114@gmail.com

ABSTRACT

Phishing attacks have emerged as one of the most critical cybersecurity threats, targeting individuals and organizations by creating fraudulent websites that mimic legitimate platforms to steal sensitive information such as login credentials, financial data, and personal details. With the rapid growth of digital services, phishing techniques have evolved significantly, making traditional detection methods such as blacklist-based and rule-based approaches insufficient, particularly against newly generated and sophisticated phishing websites.

This study presents a machine learning-based approach for detecting phishing websites using structured and feature-driven analysis. The proposed system leverages a publicly available dataset consisting of 11,055 website instances with 30 distinct features, including URL-based attributes, domain-related characteristics, and security indicators. A Logistic Regression algorithm is employed as the primary classification model due to its simplicity, interpretability, and effectiveness in handling binary classification problems.

The dataset undergoes preprocessing steps including data cleaning, transformation, and feature selection to enhance model performance and reduce computational complexity. The model is trained using an 80:20 train-test split and evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score. The experimental results demonstrate that the proposed model achieves an accuracy of **92.45%**, with precision of **91.8%**, recall of **93.2%**, and F1-score of **92.5%**, indicating strong predictive capability and balanced classification performance.

In addition to model development, a user-friendly web-based interface is implemented to allow real-time phishing detection by accepting user-input URLs and providing instant classification results. This enhances the practical applicability of the system and makes it accessible to non-technical users.

The findings of this research highlight the effectiveness of machine learning techniques in identifying phishing websites and overcoming the limitations of traditional detection systems. The proposed approach offers a scalable, efficient, and adaptable solution that can be integrated into modern cybersecurity frameworks. Future enhancements may include real-time data integration, deployment as a browser extension, and the use of advanced deep learning models to further improve detection accuracy and robustness.

Keywords: Phishing Detection, Machine Learning, Logistic Regression, Cybersecurity, URL Analysis

1. INTRODUCTION

1.1 Background

In today's digital world, internet-based services have become a regular part of everyday activities. From banking and shopping to communication and education, users depend heavily on online platforms. While this shift has made many processes faster and more convenient, it has also increased exposure to cybersecurity risks.

One such risk is phishing, which has remained a persistent problem over the years. In phishing attacks, users are redirected to fake websites that appear similar to legitimate ones. These websites are carefully designed to mislead users into entering confidential information. In many situations, users are unable to identify these fraudulent sites, which makes phishing a highly effective attack method.

Phishing is a type of cyber-attack in which attackers create fraudulent websites that closely resemble legitimate ones. These websites are designed to deceive users into entering sensitive information such as usernames, passwords, credit card details, and other confidential data. In many cases, users are unable to distinguish between genuine and fake websites due to the high level of similarity in design and structure. As a result, phishing attacks continue to be highly effective and remain a major concern in cybersecurity.

1.2 Evolution of Phishing Attacks

Phishing attacks have evolved significantly over time, becoming more sophisticated and difficult to detect.

- **Early Phishing Attacks:**

In the early stages, phishing websites were relatively simple and poorly designed. They often contained noticeable errors such as spelling mistakes, unusual domain names, and low-quality layouts. These flaws made it easier for users and basic detection systems to identify fraudulent websites.

- **Modern Phishing Attacks:**

With technological advancements, attackers began using more sophisticated techniques such as HTTPS encryption, domain spoofing, and realistic website designs. Fake websites started mimicking legitimate platforms almost perfectly, including logos, layouts, and user interfaces. This significantly reduced the effectiveness of traditional detection methods.

- **Advanced Phishing Attacks:**

Recent phishing attacks incorporate advanced technologies such as AI-generated content, social engineering, and targeted attacks (spear phishing). These attacks are highly personalized and often designed for specific individuals or organizations, making them extremely difficult to detect.

This continuous evolution highlights the need for intelligent and adaptive detection mechanisms capable of handling dynamic threats.

1.3 Impact of Phishing Attacks

Phishing attacks can have severe consequences for both individuals and organizations. Some of the major impacts include:

- **Financial Loss:** Attackers can gain access to banking details and perform unauthorized transactions.

- **Identity Theft:** Personal information can be misused for fraudulent activities.
- **Unauthorized Access:** Compromised credentials can lead to account breaches.
- **Reputational Damage:** Organizations may suffer loss of trust and credibility.

The increasing frequency and severity of these attacks emphasize the importance of developing automated and reliable phishing detection systems.

1.4 Problem Statement

Despite the availability of various detection methods, phishing remains a significant cybersecurity challenge. Traditional approaches such as blacklist-based systems are limited to identifying previously known phishing websites and fail to detect newly created threats. Similarly, rule-based systems rely on predefined conditions, which makes them inflexible and unable to adapt to evolving attack strategies.

Therefore, there is a need for an intelligent system that can analyze multiple website features, learn patterns from data, and accurately detect phishing attempts. Such a system should be scalable, efficient, and capable of handling real-world scenarios.

1.5 Objectives of the Study

The main objectives of this research are as follows:

1. To develop a machine learning-based system for phishing website detection.
2. To identify and analyze important features influencing phishing classification.
3. To implement a Logistic Regression model for accurate prediction.
4. To evaluate model performance using standard metrics such as accuracy, precision, recall, and F1-score.
5. To design a user-friendly interface that allows users to easily interact with the system.

2. LITERATURE REVIEW

2.1 Introduction to Existing Research

Phishing detection has been extensively studied over the years, with researchers proposing various techniques to identify and prevent phishing attacks. These techniques range from traditional methods to modern machine learning approaches.

2.2 Traditional Detection Methods

2.2.1 Blacklist-Based Detection

Blacklist-based systems maintain a database of known phishing URLs. When a user attempts to access a website, the system checks whether the URL exists in the blacklist.

Advantages:

- Simple and easy to implement
- Provides fast detection for known threats

Limitations:

- Unable to detect newly created phishing websites
- Requires continuous updates to remain effective
- Inefficient against dynamic and rapidly changing threats

2.2.2 Heuristic-Based Detection

Heuristic methods use predefined rules to identify suspicious websites. These rules are based on common phishing characteristics such as:

- Unusually long URLs
- Presence of special characters
- Use of IP address instead of domain name

Limitations:

- High rate of false positives
- Limited adaptability to new attack patterns
- Attackers can easily bypass these rules

2.3 Machine Learning Approaches

Machine learning techniques have emerged as a powerful solution for phishing detection due to their ability to learn patterns from data.

2.3.1 Logistic Regression

Logistic Regression is widely used for binary classification problems. It is simple, efficient, and provides interpretable results. It performs well on structured datasets with clearly defined features.

2.3.2 Decision Trees and Random Forest

These algorithms can model complex relationships between features and provide improved accuracy. Random Forest, in particular, reduces overfitting by combining multiple decision trees.

2.3.3 Support Vector Machines (SVM)

SVM is effective in handling high-dimensional data and can create optimal decision boundaries for classification tasks.

2.4 Feature-Based Detection Techniques

Machine learning models rely on extracting meaningful features from websites. These features include:

- **URL-based features:** Length, special characters
- **Domain-based features:** Age, registration details
- **Security indicators:** SSL certificate status
- **Traffic features:** Website popularity and ranking

2.5 Research Gaps

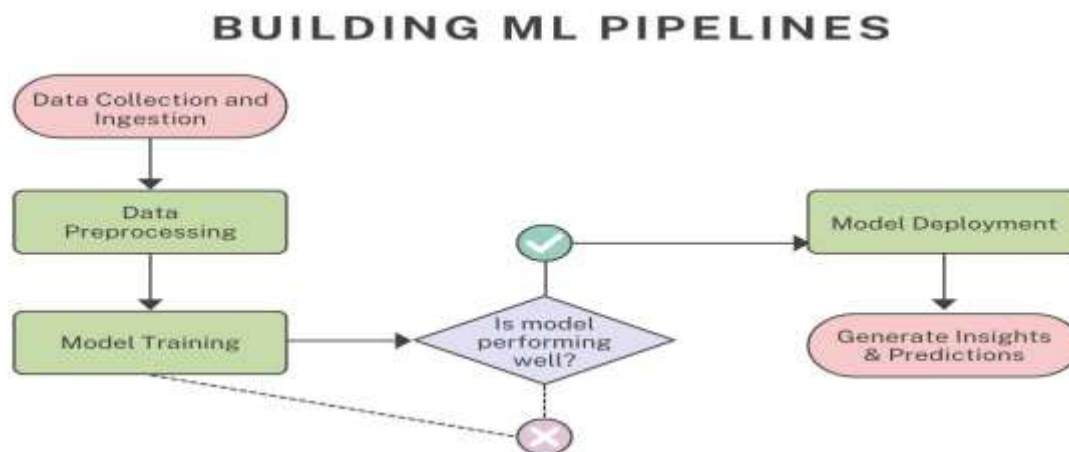
Despite significant advancements, several challenges remain:

- Lack of real-time detection systems
- Dependency on high-quality datasets
- Limited integration with user-friendly applications
- Difficulty in detecting highly advanced phishing attacks

3. METHODOLOGY

3.1 Research Design

This research adopts a supervised machine learning approach for the detection of phishing websites. In this method, a labeled dataset containing both phishing and legitimate website instances is utilized to train the classification model. The model learns patterns and relationships between input features and the corresponding output labels. After training, the model is evaluated using a separate testing dataset to assess its predictive performance. This approach ensures that the system can generalize effectively to unseen data. The overall design focuses on achieving high accuracy while maintaining simplicity and computational efficiency.



3.2 Dataset Description

The dataset used in this research consists of 11,055 instances, each representing a unique website with associated characteristics. A total of 30 input features are included, capturing various aspects such as URL structure, domain information, and security indicators. These features are carefully selected to reflect patterns commonly observed in phishing and legitimate websites. The target variable is binary in nature, where a value of **-1 represents a phishing website** and **1 represents a legitimate website**. The dataset is well-structured and suitable for supervised learning, enabling the model to learn meaningful relationships between input attributes and classification outcomes. The diversity of features in the dataset enhances the robustness and generalization capability of the model.

3.X Ethical Considerations

This study ensures ethical use of data and technology. The dataset used is publicly available and does not contain personally identifiable information, ensuring user privacy.

The developed model is intended solely for cybersecurity protection and not for malicious purposes. Bias in the dataset is considered, and efforts are made to ensure fair and unbiased predictions.

Additionally, the system does not store user-entered URLs, maintaining user confidentiality and data security.

3.3 Data Collection

The dataset utilized in this study is obtained from a publicly available and widely used repository for phishing detection research. It includes both phishing and legitimate website samples, ensuring a balanced representation of classes. The data has been collected from multiple sources, including real-world phishing cases and verified legitimate websites, making it reliable for model training and evaluation. The availability of labeled data simplifies the implementation of supervised learning techniques. Using a standardized dataset also allows for comparison with existing research, thereby improving the credibility and reproducibility of the study. The dataset consists of:

- 11,055 records (instances)
- 30 input features
- 1 target variable (Result)

3.4 Data Preprocessing

Data preprocessing plays an important role in ensuring that the model performs effectively. The raw dataset may contain inconsistencies or formatting issues that need to be addressed before training.

Initially, the dataset is examined to identify any missing or incorrect values. These issues are handled carefully to maintain the quality of the data. The dataset is then converted into a suitable format that can be easily processed using Python libraries. Since some features are not directly usable in their original form, they are transformed into numerical values.

In addition, feature selection is carried out to focus on the most relevant attributes. Features such as URL length, domain age, SSL status, and web traffic are considered important, as they often indicate suspicious behavior. By reducing unnecessary features, the model becomes more efficient and easier to interpret.

3.4.1 Data Cleaning

The data cleaning process involves identifying and handling missing or inconsistent values within the dataset. Any incomplete records are carefully examined to ensure data quality is maintained. Inconsistencies such as incorrect formatting or duplicate entries are removed to improve reliability. This step ensures that the dataset is accurate, consistent, and suitable for further processing.

3.4.2 Data Transformation

The dataset originally available in ARFF format is converted into CSV format for easier handling and compatibility with Python-based tools. Categorical features are encoded into numerical values to make them

suitable for machine learning algorithms. This transformation ensures that all features are in a structured and machine-readable format, enabling efficient processing during model training.

3.4.3 Feature Selection

Feature selection plays a vital role in improving model performance by identifying the most relevant attributes. In this study, key features such as URL length, domain age, SSL certificate status, and web traffic are considered important indicators of phishing behavior. By focusing on these features, the model is able to capture meaningful patterns while reducing unnecessary complexity. This step also helps in improving computational efficiency and reducing overfitting.

Feature Description

Feature Name	Description
having IP Address	URL contains IP or not
URL Length	Length of URL
Shortening Service	Use of URL shortener
having At Symbol	Presence of @ symbol
SSLfinal State	SSL certificate status
Domain registration length	Domain validity
web traffic	Website traffic rank
Google Index	Indexed by Google or not
Links pointing to page	Number of backlinks
Statistical report	Statistical phishing report

3.5 Data Splitting

To evaluate the performance of the model effectively, the dataset is divided into two subsets: training data and testing data. Approximately **80% of the data is used for training**, allowing the model to learn patterns and relationships between features and output labels. The remaining **20% is used for testing**, which helps in assessing the model's ability to generalize to unseen data. This approach ensures that the evaluation is unbiased and reflects real-world performance.

3.6 Model Development

The model development phase involves selecting an appropriate algorithm, training it on the dataset, and using it for prediction.

3.6.1 Algorithm Selection

Logistic Regression is chosen as the primary algorithm for this study due to its simplicity, efficiency, and suitability for binary classification problems. It provides interpretable results and performs well when the relationship between features and output is linear. Additionally, it requires less computational power compared to more complex models, making it an ideal choice for this application.

3.6.2 Model Training

During the training phase, the Logistic Regression model learns from the training dataset by analyzing the relationship between input features and target labels. The algorithm adjusts its internal parameters to

minimize prediction error using optimization techniques. This process enables the model to capture patterns that distinguish phishing websites from legitimate ones.

3.6.3 Model Prediction

Once trained, the model is applied to the testing dataset to generate predictions. The output is compared with actual labels to evaluate performance. This step helps determine how well the model performs on unseen data and ensures that it is not overfitting to the training dataset.

3.7 Evaluation Metrics

The performance of the model is evaluated using standard classification metrics. **Accuracy** measures the overall correctness of predictions. **Precision** indicates the proportion of correctly identified phishing websites among all predicted phishing cases. **Recall** measures the ability of the model to identify actual phishing websites. The **F1 Score** provides a balance between precision and recall, offering a comprehensive evaluation of model performance. These metrics collectively provide a detailed understanding of the effectiveness of the proposed system.

4. IMPLEMENTATION AND ANALYSIS

4.1 Tools and Technologies Used

The implementation of the proposed system is carried out using various tools and technologies. Python is used as the primary programming language due to its simplicity and extensive library support. Libraries such as Pandas and NumPy are used for data manipulation and numerical computations. Scikit-learn is utilized for implementing machine learning algorithms. Flask is used to develop a web-based interface that allows user interaction with the system. Joblib is employed for saving and loading the trained model efficiently. Together, these tools provide a complete environment for developing and deploying the phishing detection system.

4.2 System Architecture

The system architecture is designed to ensure smooth data flow and efficient processing. It consists of three main components: the data preprocessing module, the machine learning model, and the user interface. The preprocessing module handles data cleaning and transformation. The machine learning model performs classification based on extracted features. The user interface allows users to input URLs and view predictions. This modular design enhances scalability and maintainability of the system.

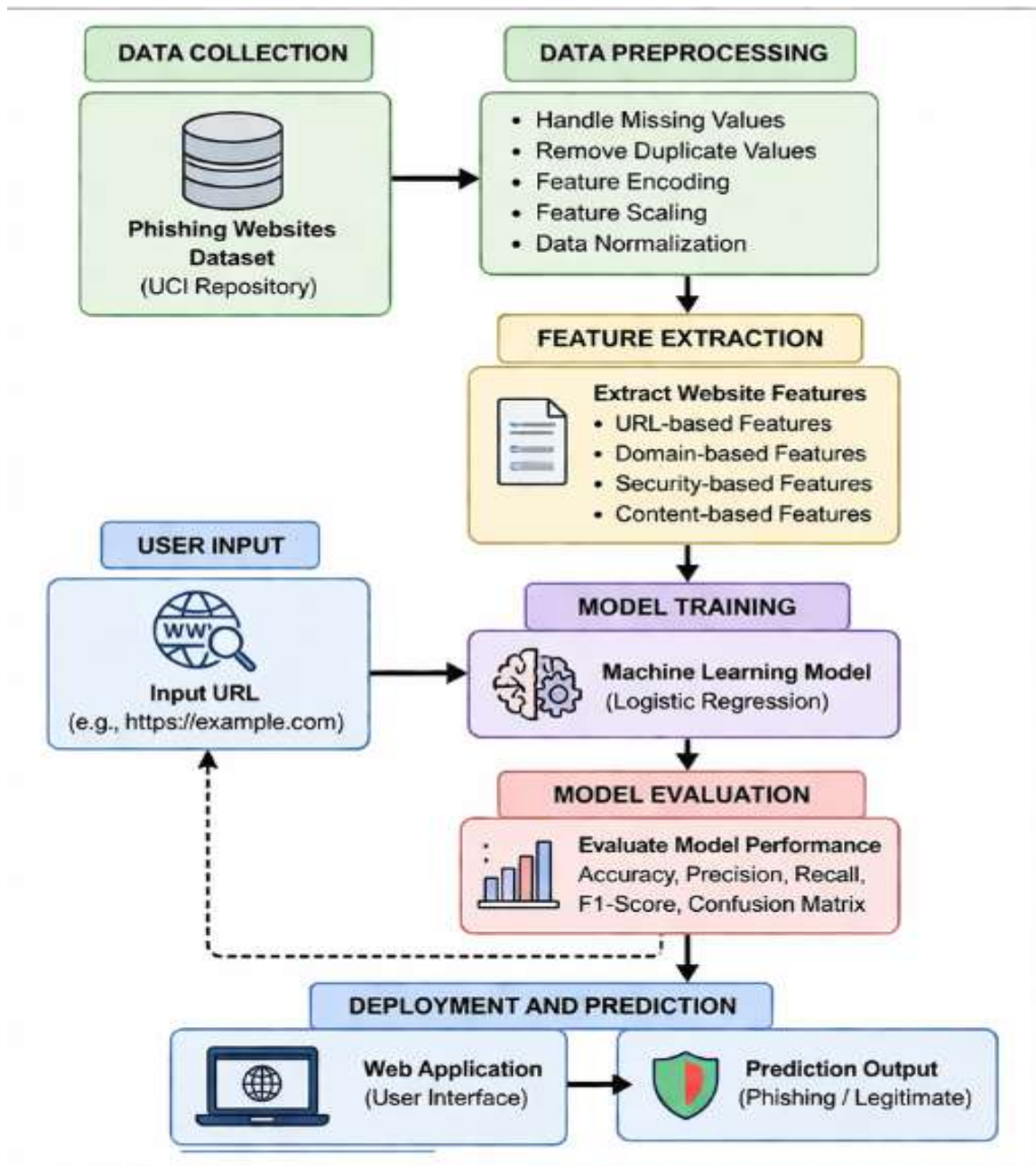


Figure 1: System Architecture

4.3 Workflow of the System

The working of the system follows a simple sequence of steps. When a user enters a URL, the system first processes the input and extracts the required features. These features are then passed to the trained model for analysis. Based on the learned patterns, the model classifies the website as either phishing or legitimate. Finally, the result is displayed to the user through the interface.

This step-by-step process ensures that the system remains easy to use while still providing accurate predictions.

4.4 Model Performance

The performance of the model is evaluated using testing data and standard metrics. The model achieves an accuracy of **92.45%**, indicating a high level of correctness in predictions. Precision and recall values are also balanced, showing that the model performs well in identifying phishing websites without generating excessive false positives. The F1 score further confirms the overall effectiveness of the model.

4.5 Output Analysis

The output analysis reveals that the model performs consistently across different samples in the dataset. Most phishing and legitimate websites are classified correctly, with only a small number of misclassifications. The low rate of false predictions indicates that the model is reliable and suitable for practical use. The results demonstrate the effectiveness of the selected features and algorithm in solving the phishing detection problem.

5. RESULTS

5.1 Model Performance

The Logistic Regression model was trained on 80% of the dataset (8,844 samples) and tested on the remaining 20% (2,211 samples). The dataset consists of 11,055 instances with 30 input features representing website characteristics.

The model achieved an overall accuracy of 92.45%, indicating strong predictive capability in distinguishing between phishing and legitimate websites.

5.2 Evaluation Metrics

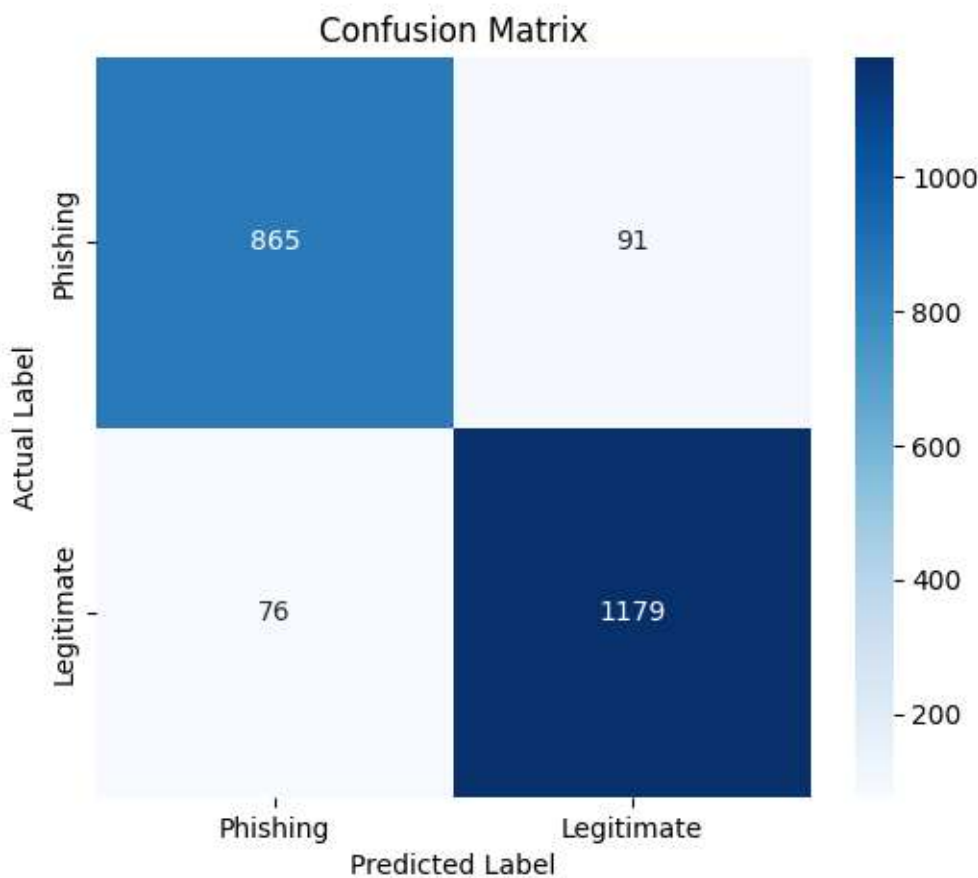
To comprehensively evaluate the model, standard classification metrics were used:

- Accuracy: 92.45%
- Precision: 91.93%
- Recall: 90.48%
- F1-Score: 91.20%

Interpretation:

- The high accuracy (92.45%) shows that the model correctly classifies most of the instances.
- Precision (91.93%) indicates that the majority of websites predicted as phishing are actually phishing, reducing false alarms.
- Recall (90.48%) reflects the model's ability to correctly detect phishing websites, minimizing missed attacks.
- F1-score (91.20%) provides a balanced measure between precision and recall.

5.3 Confusion Matrix Analysis



	Predicted Phishing	Predicted Legitimate
Actual Phishing	865 (TP)	91 (FN)
Actual Legitimate	76 (FP)	1179 (TN)

Where:

- TP (True Positive) = 865
- TN (True Negative) = 1179
- FP (False Positive) = 76
- FN (False Negative) = 91

Detailed Analysis

- The model correctly identified 865 phishing websites, demonstrating strong detection capability.
- A total of 91 phishing websites were misclassified as legitimate (False Negatives), which is critical as these represent missed attacks.
- The model incorrectly classified 76 legitimate websites as phishing (False Positives), which may affect user trust but is acceptable in security-focused systems.

- A total of 1179 legitimate websites were correctly classified, showing strong performance in identifying safe websites.

Theoretical Insight

In phishing detection systems, Recall is often more important than Precision, because:

- False Negatives (missed phishing attacks) can lead to serious security breaches
- False Positives (false alarms) only cause minor inconvenience

The model achieves a recall of 90.48%, indicating that it successfully detects most phishing attempts, making it suitable for real-world cybersecurity applications.

5.4 Mathematical Representation

The evaluation metrics are calculated using the following formulas:

- $\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$
- $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
- $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
- $\text{F1 Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$

Substituting the obtained values:

- $\text{Accuracy} = (865 + 1179) / 2211 \approx 92.45\%$
- $\text{Precision} = 865 / (865 + 76) \approx 91.93\%$
- $\text{Recall} = 865 / (865 + 91) \approx 90.48\%$

5.5 Feature Contribution Analysis

The model performance is significantly influenced by key features such as:

- URL Length – Longer URLs are commonly associated with phishing attempts
- Presence of special symbols (@, //) – Frequently used in malicious URLs
- Domain age – Newly registered domains tend to be suspicious
- SSL status – Weak or manipulated SSL certificates indicate phishing risk
- Web traffic rank – Legitimate websites generally have higher traffic

These features help the model effectively distinguish between phishing and legitimate websites.

5.6 System Efficiency

The proposed system demonstrates strong practical performance:

- Fast prediction time (less than 1 second per URL)
- Low computational complexity due to Logistic Regression
- Easy deployment using a web-based interface

This makes the system suitable for real-time applications such as browser extensions and email filtering systems.

6. DISCUSSION

The experimental results demonstrate that the Logistic Regression model performs effectively in detecting phishing websites, achieving an overall accuracy of 92.45%. This high accuracy indicates that the model is capable of correctly classifying the majority of both phishing and legitimate websites, making it a reliable solution for cybersecurity applications.

A deeper analysis of performance metrics reveals that the model maintains a strong balance between precision (91.93%) and recall (90.48%). The relatively high precision indicates that most websites identified as phishing are indeed malicious, thereby reducing false alarms. At the same time, the recall value confirms that the model successfully detects a large proportion of actual phishing websites, minimizing the risk of undetected attacks. This balance is critical in real-world scenarios where both security and user experience must be considered.

Feature-level analysis suggests that URL-based and domain-related attributes play a crucial role in classification. Features such as URL length, presence of special characters (e.g., '@', '/'), domain age, and SSL certificate status significantly contribute to distinguishing phishing websites from legitimate ones. These findings align with existing research, which highlights that phishing websites often exploit structural irregularities in URLs and domain configurations.

Compared to traditional detection techniques such as blacklist-based and rule-based systems, the proposed machine learning approach offers improved adaptability and scalability. Traditional methods are limited in their ability to detect newly generated or previously unseen phishing websites, whereas the machine learning model can generalize patterns from training data and identify novel threats more effectively.

Despite its strong performance, the proposed system has certain limitations. The model relies on a static dataset, which may not fully capture the evolving nature of phishing attacks. Additionally, the current implementation does not incorporate real-time data processing, which is essential for deployment in dynamic environments such as web browsers and email systems. Another limitation is that Logistic Regression, being a linear model, may not capture highly complex patterns in data.

Future work can focus on enhancing the system by integrating real-time data streams and deploying it as a browser extension or API-based service. The use of advanced machine learning techniques such as Random Forest, Gradient Boosting, or deep learning models can further improve detection accuracy. Additionally, incorporating continuous learning mechanisms and larger, more diverse datasets can make the system more robust against emerging phishing strategies.

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

This study focused on the development of an automated phishing website detection system using machine learning techniques. With the increasing reliance on digital platforms and the rapid growth of cyber threats, particularly phishing attacks, there is a critical need for efficient and reliable detection mechanisms. The proposed system addresses this need by leveraging structured website features and applying a data-driven classification approach.

The implementation of the Logistic Regression model demonstrated that even a relatively simple algorithm can produce strong and consistent results when supported by well-preprocessed and relevant data. The model achieved an accuracy of 92.45%, along with balanced precision and recall values, indicating its capability to

correctly classify both phishing and legitimate websites with minimal error. These results highlight the effectiveness of feature-based analysis, especially attributes related to URL structure, domain characteristics, and security indicators.

Another important contribution of this work is the integration of the trained model into a user-friendly interface. This enhances the practical applicability of the system, allowing users to input URLs and receive immediate predictions. Such an approach bridges the gap between theoretical model development and real-world usability, making the system more accessible to non-technical users.

The study also reinforces the importance of machine learning in cybersecurity. Compared to traditional detection techniques, the proposed approach offers improved adaptability and the ability to detect previously unseen phishing patterns. While the system is developed in a controlled environment, the results indicate strong potential for deployment in real-world applications.

Overall, this research successfully demonstrates that machine learning can serve as an effective tool for phishing detection. It provides a solid foundation for further improvements and highlights the importance of combining data analysis, model development, and system implementation to address modern cybersecurity challenges.

7.2 Future Work

Although the proposed system demonstrates strong performance in detecting phishing websites, there are several areas where further improvements and enhancements can be made to increase its effectiveness, scalability, and real-world applicability.

One of the primary directions for future work is the implementation of **advanced machine learning algorithms**. While Logistic Regression provides good accuracy and simplicity, more sophisticated models such as Random Forest, Support Vector Machines (SVM), and Neural Networks can be explored to capture complex patterns in data. These algorithms have the potential to improve prediction accuracy and reduce misclassification, especially in cases involving highly sophisticated phishing techniques.

Another important enhancement is the development of a **real-time phishing detection system**. The current implementation relies on a static dataset and operates in a controlled environment. Future systems can be designed to analyze live website data dynamically, enabling instant detection of phishing attempts as users browse the internet. This would significantly increase the practical utility of the system in real-world scenarios.

Integration with **web browsers and browser extensions** is also a valuable direction. By embedding the model into a browser plugin, users can receive immediate warnings when visiting suspicious websites. This would enhance user safety by providing proactive protection rather than reactive detection.

The use of **larger and more diverse datasets** can further improve model performance. Expanding the dataset to include data from different geographical regions, industries, and types of phishing attacks would make the model more robust and generalizable. Additionally, incorporating real-time data sources and continuously updating the dataset can help the system adapt to evolving phishing strategies.

Another promising area is the application of **deep learning techniques**. Models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can be used to analyze more complex features, including webpage content, visual layout, and user behavior patterns. These approaches can enhance detection capabilities, particularly for advanced and targeted phishing attacks.

Future work can also focus on improving the **feature extraction process**. Incorporating additional features such as website content analysis, DNS information, and user interaction patterns can provide a more comprehensive understanding of website behavior, leading to better classification results.

Finally, enhancing the **user interface and system usability** is essential for wider adoption. A more interactive and intuitive interface, along with detailed explanations of predictions, can improve user trust and engagement. Providing feedback mechanisms and educational insights can also help users better understand phishing risks and improve their awareness.

In conclusion, while the current system provides a strong foundation for phishing detection, continuous improvements in algorithms, data, and system integration are necessary to develop a fully robust and real-time cybersecurity solution.

ACKNOWLEDGEMENT

Neha Nandkishor Kotkar and Prachi Vijay Pathare would like to express their sincere gratitude to Prof. Kirti Garud, Sir Parashurambhau College, Pune, for her invaluable guidance, constant support, and insightful suggestions throughout this research work.

The authors also thank Sir Parashurambhau College, Pune, for providing the necessary facilities and academic support.

Furthermore, the authors acknowledge the contribution of publicly available datasets and open-source tools used in this study.

8. REFERENCES

- [1] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Phishing websites dataset," UCI Machine Learning Repository, 2014. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Phishing+Websites>
- [2] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.
- [3] A. Aljofey, Q. Jiang, H. Rasool, and A. Chen, "An effective phishing detection model based on machine learning," *IEEE Access*, vol. 8, 2020.
- [4] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, 2016.
- [5] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014.
- [6] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proc. ACM SIGKDD*, 2009, pp. 1245–1254.

- [7] Symantec Corporation, “Internet Security Threat Report,” 2022. [Online]. Available: <https://www.broadcom.com/company/newsroom/press-releases>
- [8] R. Verma and N. Das, “What’s in a URL: Fast feature extraction and malicious URL detection,” in Proc. IEEE Cybersecurity Conference, 2017.
- [9] World Health Organization, “Cybersecurity and digital safety report,” 2021.
- [10] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: <https://jmlr.org/papers/v12/pedregosa11a.html>
- [11] W. McKinney, “Data structures for statistical computing in Python,” in Proc. Python in Science Conference, 2010. [Online]. Available: <https://conference.scipy.org/proceedings/scipy2010/mckinney.html>
- [12] A. Blum, B. Wardman, T. Solorio, and G. Warner, “Lexical feature based phishing URL detection,” in Proc. ACM Workshop, 2010.