DETECTING PHISHING URLS USING DATAMINING TECHNIQUES

Mr. Htin Linn, Ms. May Khine Soe, Mr. Nyan Lin Aung

*Corresponding author: htin.l65@rsu.ac.th

Abstract

This paper explores the application of data mining techniques for detecting phishing URLs, with a focus on improving the classification accuracy of malicious and safe URLs. A dataset consisting of URLs and their corresponding features, such as length, domain information, and HTTP/HTTPS status, was used to train and evaluate various machine learning models. The research employs an iterative approach, progressively incorporating data from 2019 to 2023 to enhance model performance. The results demonstrate significant improvements in precision, recall, and F1-scores, with the final model achieving an accuracy of 94% on the 2024 dataset. The model is evaluated on both training and benchmark data, providing insights into its ability to generalize to new, unseen data. This study highlights the effectiveness of data mining techniques in developing robust models for phishing URL detection, which can be deployed in real-world cybersecurity applications.

Keywords: Phishing Detection, Data Mining, Machine Learning, URL Classification, Cybersecurity

1. INTRODUCTION

Phishing is a deceptive and sophisticated form of cybercrime in which attackers manipulate individuals and organizations into revealing sensitive information, such as login credentials, financial details, or personal data. This is typically done by redirecting victims to fraudulent websites that closely mimic legitimate ones, creating a false sense of trust. Phishing has emerged as a significant threat in the digital era, targeting individuals, businesses, and governments alike. The consequences of phishing are far-reaching, often leading to financial losses, identity theft, data breaches, and reputational damage (Alkhalil et al., 2021). In today's interconnected world, where online transactions and communications have become a cornerstone of personal and professional life, phishing represents one of the most pressing cybersecurity challenges.

The rapid evolution of phishing techniques has made detection increasingly challenging. Attackers frequently change the appearance, structure, and domain of phishing websites to bypass conventional security systems. This adaptability allows campaigns to evade traditional detection methods. The sheer volume of web traffic further complicates manual analysis, which is time-consuming and impractical at scale. Rule-based systems, relying on predefined patterns, often fail to detect subtle variations in modern phishing attacks, highlighting the need for scalable, automated solutions with high accuracy (Kadlak & Sharma, 2018).

Machine learning has emerged as a powerful tool in the fight against phishing. By leveraging data-driven techniques, machine learning models can analyze patterns in URLs and classify them as phishing or legitimate based on their features. Among these models, logistic regression has proven to be particularly effective for binary classification problems, such as phishing detection. Logistic regression is a statistical method that predicts the probability of a given input belonging to one of two categories. Its simplicity, efficiency, and interpretability make it an ideal choice for phishing detection tasks, especially in real-time applications where speed and reliability are critical (Sperandei, 2014).

One of the key advantages of logistic regression is its ability to provide probabilistic outputs. Rather than merely labeling a URL as phishing or legitimate, logistic regression assigns a probability score, which reflects the likelihood of the URL being malicious. This feature enables more nuanced decision-making, allowing cybersecurity systems to take actions based on the level of risk. For example, URLs with high probabilities can be automatically blocked, while those with moderate risk can be flagged for further analysis. Additionally, logistic regression's interpretability allows cybersecurity professionals to understand how specific features, such as URL length or the presence of suspicious keywords, contribute to classification outcomes. This transparency not only builds trust in the model but also aids in refining detection strategies to address emerging threats (Starbuck, 2023).

Logistic regression is particularly well-suited for environments where computational resources are limited. Unlike more complex machine learning models, such as neural networks, logistic regression requires minimal processing power, making it accessible for organizations of all sizes. Its straightforward implementation and low computational cost make it a practical choice for real-time phishing detection, even in scenarios where high volumes of data need to be processed quickly. Moreover, its robustness and reliability have been widely validated in various binary classification tasks, further solidifying its value in cybersecurity applications.

The objective of this study is to explore the use of logistic regression as a tool for detecting phishing URLs. By analyzing a dataset of URLs and extracting relevant features, such as domain structure, protocol type, and character patterns, the research aims to build a model capable of accurately

classifying URLs as phishing or legitimate. The study also highlights the limitations of traditional detection methods and demonstrates how logistic regression can address these challenges effectively. Through this work, we aim to showcase the potential of logistic regression as a scalable, interpretable, and efficient solution for combating phishing attacks. Ultimately, this research contributes to the broader effort of enhancing cybersecurity defenses and protecting individuals and organizations from the growing threat of phishing.

2. LITERATURE REVIEW

The detection of phishing URLs has emerged as a critical area of research in cybersecurity, with numerous studies exploring various data mining and machine learning approaches. This review examines the evolution of phishing detection techniques, with a particular focus on logistic regression and related methodologies, while identifying current gaps in the research landscape.

i. Evolution of Phishing Detection Approaches

Early research in phishing URL detection primarily relied on blacklist-based approaches and simple rule-based systems. Khonji et al. (2013) conducted a comprehensive survey of phishing detection techniques, highlighting the limitations of these traditional methods, particularly their inability to detect zero-day phishing attacks. Their work emphasized the need for more sophisticated, machine learning-based approaches that could adapt to evolving threats.

ii. Feature Engineering in URL Analysis

Jeeva and Rajsingh (2016) made significant contributions to the field by introducing association rule mining for phishing detection. Their research demonstrated the importance of proper feature selection in URL analysis, identifying key characteristics such as URL length, special character frequency, and domain attributes as crucial indicators of malicious intent. This work established a foundation for feature engineering in phishing detection systems, achieving accuracy rates of 89.7% using their proposed methodology.

iii. Application of Logistic Regression

The application of logistic regression in classification problems has been extensively studied across various domains. Green et al. (1998) demonstrated the effectiveness of logistic regression in binary classification tasks, particularly highlighting its interpretability and computational efficiency. While their work wasn't specifically focused on phishing detection, their findings regarding model optimization and feature selection have been influential in cybersecurity applications.

Feroz and Mengel (2014) were among the first to specifically apply online logistic regression to phishing URL detection. Their research achieved notable success, with accuracy rates of 92% in identifying malicious URLs. They emphasized the advantage of logistic regression's ability to handle real-time classification tasks, making it particularly suitable for cybersecurity applications where rapid detection is crucial.

iv. Recent Developments in Machine Learning Approaches

More recent studies have explored various machine learning techniques for phishing detection.

Chiramdasu et al. (2021) conducted a comparative analysis of different algorithms, including logistic regression, support vector machines (SVM), and random forests. Their research indicated that while more complex models sometimes achieved marginally better results, logistic regression offered the best balance between accuracy and computational efficiency, particularly in real-world applications. Zouina and Outtaj (2017) proposed a novel approach combining SVM with similarity index measurements, achieving accuracy rates of 95.8%. While their results were impressive, their methodology required significant computational resources, potentially limiting its practical application in real-time detection systems.

v. Limitations in Current Research

Despite these advancements, several limitations persist in current phishing detection research:

Dataset Currency: Many studies rely on older datasets that may not reflect current phishing techniques. Shahrivari et al. (2020) noted this limitation, emphasizing the need for continuously updated datasets to maintain model relevance.

Feature Evolution: Limited research exists on how URL features evolve over time and how this evolution affects model performance. Most studies focus on static feature sets, potentially missing emerging patterns in phishing techniques.

Model Adaptability: Few studies have examined how models maintain their effectiveness as phishing techniques evolve. The lack of longitudinal studies tracking model performance over extended periods represents a significant gap in current research.

Real-world Application: While many studies report high accuracy rates in controlled environments, there is limited research on how these models perform in real-world scenarios with evolving threats and varying network conditions.

vi. Research Gap and Current Study

This study aims to address these limitations by:

- Utilizing a comprehensive dataset spanning multiple years (2019-2024)
- Implementing an iterative training approach to evaluate model adaptability
- Focusing on logistic regression's practical applicability in real-world scenarios
- Examining how feature importance evolves over time

By addressing these aspects, this research contributes to the existing body of knowledge while providing practical insights for implementing effective phishing detection systems in real-world applications.

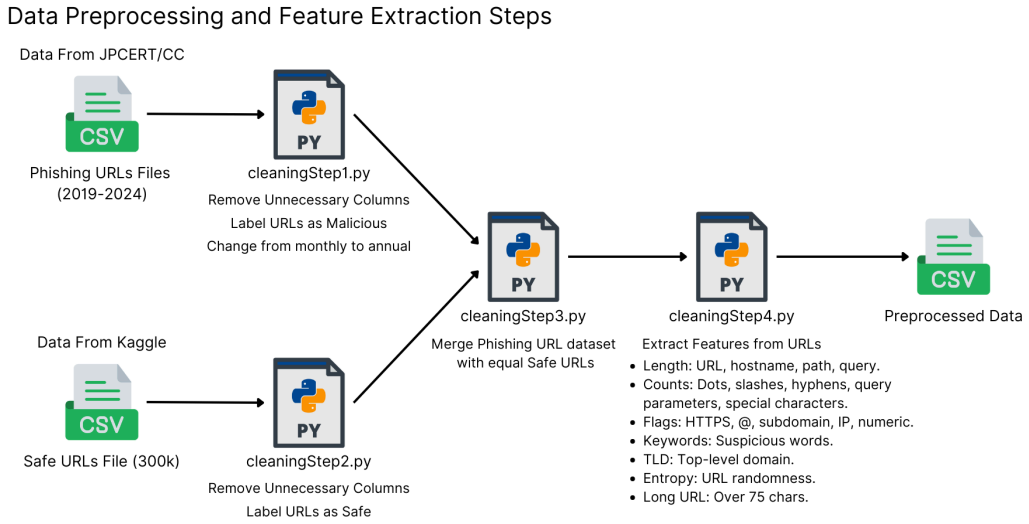
3. METHODOLOGY

In this study, we aim to identify phishing URLs by analyzing various features that can distinguish them from benign (safe) URLs. The dataset used for this analysis is sourced from two locations: Kaggle and JPCERT/CC. The safe URLs are obtained from Kaggle's dataset, **Malicious and Benign URLs (Kumar, 2019)**, while the phishing URLs are sourced from JPCERT/CC's **PhishURL List (JPCERT/CC, n.d.)**. The phish url data spans several years, with the safe URLs dataset containing approximately 300,000 entries, while the phishing URLs dataset includes around 200,000 entries.

The methodology for preprocessing the data involves combining the datasets, balancing them, and extracting meaningful features from the URLs that can be used for further analysis or model training. The phishing URLs from JPCERT/CC are organized by months, and as part of the preprocessing steps, we will consolidate them into annual data from 2019 to 2024. This separation of data by year is crucial, as it allows us to train the model year after year and review the results over time to evaluate the model's performance and improvements. The tools used to perform this analysis are primarily Python, along with necessary libraries such as Pandas for data manipulation and Math for various computations.

i. Data Preprocessing

Figure 1. **Data Preprocessing and Feature Extraction Steps**



Data preprocessing is a crucial step in any machine learning or data analysis task, as it ensures that the raw data is transformed into a clean, usable format that is ready for modeling. In this study, the data preprocessing steps are divided into several stages: data cleaning, data balancing, feature extraction, and dataset finalization.

a. Data Cleaning

The first step in data preprocessing was to load the datasets containing the safe and phishing URLs. The safe URLs dataset, sourced from Kaggle, contains benign URLs that are assumed to be safe. The phishing URLs dataset, sourced from JPCERT/CC, contains URLs labeled as phishing, which are intended to deceive or harm users.

Both datasets were loaded into memory using Python's **Pandas** library, which allows for efficient handling of large datasets. Since the phishing URL dataset from JPCERT/CC is divided into multiple

monthly CSV files for each year, the first preprocessing task was to consolidate these files into a single annual dataset for each year between 2019 and 2024. This step involved reading the monthly files for each year and combining them into a unified dataset. After consolidating the data, we ensured that the data contained no missing or duplicate entries. If any URLs were found to be invalid or improperly formatted, they were removed from the dataset.

b. Data Balancing

Once the datasets were cleaned, the next task was to balance the number of phishing and safe URLs. Given that the safe URLs dataset contained approximately 300,000 entries and the phishing URLs dataset contained about 200,000 entries, the goal was to ensure that the final dataset for model training or analysis would have an equal number of phishing and safe URLs.

To achieve this, we selected a random subset of safe URLs from the Kaggle dataset to match the number of phishing URLs. This step was critical because having an imbalanced dataset, where one class (e.g., phishing URLs) vastly outnumbers the other (e.g., safe URLs), could lead to biased results when building a machine learning model. By ensuring that both classes are equally represented, we help the model learn to identify phishing URLs more effectively.

c. Feature Extraction

After consolidating and balancing the datasets, the next stage of preprocessing involved extracting relevant features from the URLs. Feature extraction is the process of transforming raw data into a set of measurable attributes that can be used by machine learning algorithms.

Several features were extracted from each URL in both the phishing and safe datasets. These features were chosen based on their relevance to distinguishing phishing URLs from benign ones. Some of the features extracted include:

- **Length of URL**: The total number of characters in the URL, which can indicate whether a URL is suspiciously long (a common characteristic of phishing URLs).
- **Hostname Length**: The length of the hostname (e.g., "example.com"), which may be an indicator of whether a URL is trying to impersonate a legitimate website.
- **Length of Path and Query**: The number of characters in the path and query sections of the URL.
- **Presence of Suspicious Keywords**: Certain keywords like "account", "login", "secure", "verify", and "password" are commonly found in phishing URLs. A list of suspicious keywords was used to check for their presence in the URL.
- **Special Characters**: Count of non-alphanumeric characters in the URL.
- **Top-Level Domain (TLD)**: The domain extension (e.g., ".com", ".org") that could reveal whether the URL belongs to a legitimate or suspicious domain.
- **Entropy**: A measure of randomness in the URL. Phishing URLs often exhibit higher entropy due to their obfuscation techniques.
- **IP Address**: A check for the presence of IP addresses in the URL, which could suggest that the URL is not associated with a typical domain name system (DNS).

These features were extracted using custom Python functions and stored in a new dataframe that was then combined with the original dataset. The resulting dataset contained both the original URL and the newly extracted features.

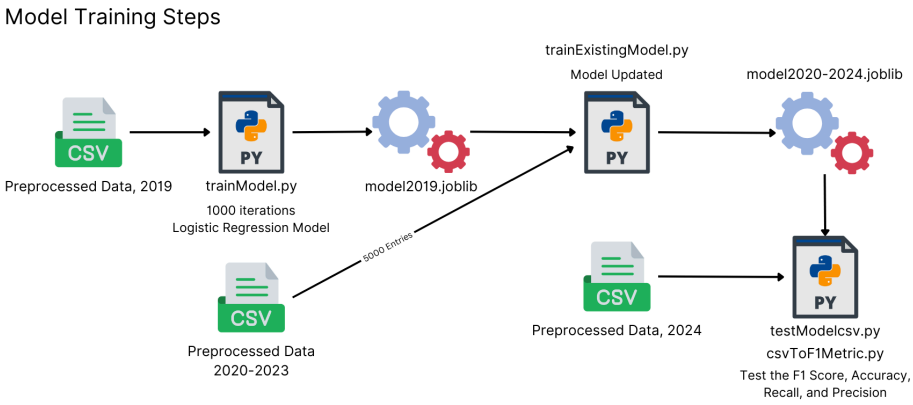
d. Final Dataset

At the end, the column containing the original URL is removed. The enriched dataset, which includes its status (phishing or safe), and the extracted features, was saved to a new CSV file. This file serves as the final dataset for further analysis, machine learning model training, or other downstream tasks.

By following the preprocessing steps, the dataset was transformed into a clean, balanced, and feature-enriched format that is suitable for the development of the logistic regression model. The preprocessing pipeline ensures that the dataset accurately reflects the characteristics of both phishing and safe URLs, and is structured for efficient and effective analysis.

ii. Model Development

Figure 2. **Data Preprocessing and Feature Extraction Steps**



After completing the data preprocessing step, the model development process proceeds with the following stages, which include model selection, training, evaluation, and updating. These steps ensure the trained model is accurate, reliable, and capable of generalizing well on new data.

a. Feature Preparation

At this stage, the data has already been cleaned and preprocessed. Now, the following steps are executed to prepare the data for model training:

- **Feature Selection**: The dataset is divided into feature variables (X) and the target variable (y). The target variable (status) indicates whether a URL is phishing (1) or benign (0), while the features consist of the attributes extracted from the URL during preprocessing (e.g., URL length, domain-related features, special character presence, etc.).
- **Categorical Encoding**: If the dataset includes any categorical variables, they are encoded using one-hot encoding (pd.get_dummies()) to convert them into numeric representations, making the data suitable for input into machine learning algorithms.
- **Handling Missing Values**: If there are any missing values in the feature set, they are handled (for example, by imputation or using default values). This ensures that the model is trained on a complete dataset, minimizing data quality issues.
- **Data Alignment**: The features are aligned to ensure consistency in column names, especially when training the model on previously seen data and new data. This step ensures that the model expects the same features in future data as it did in the original training data.

b. Model Selection

With the dataset properly prepared, a **Logistic Regression** model is selected for the classification task. Logistic regression is a robust and well-understood algorithm for binary classification problems, making it ideal for distinguishing between phishing and benign URLs. Its advantages include:

- **Interpretability**: It allows us to understand the contribution of each feature to the prediction.
- **Efficiency**: It is computationally efficient, especially for large datasets, which is important when handling URL data that can be large and varied.

The Logistic Regression model is configured with a higher iteration limit (max_iter=2000) to ensure that the model converges properly during training, especially with the complex feature set derived from URL data.

c. Model Training

Once the model is selected, the next step is to train the Logistic Regression model on the prepared dataset:

- **Initial Training (2019 Data Split):** To begin the training process, the Logistic Regression model will first be trained using the **2019 dataset**. An **80-20 split** will be applied to this dataset, where 80% of the data is used for training and 20% is used for testing. This initial split helps establish a baseline performance for the model.

- **Cross-Validation**: For the initial training, **10-fold cross-validation** will be applied. This technique splits the training data into 10 equal parts (folds), uses 9 parts for training, and tests the model on the remaining fold. This process is repeated 10 times so that each fold is used as the test set once. Cross-validation ensures that the model generalizes well and is not overfitting to any particular subset of the data.

- **Model Fitting**:After cross-validation, the Logistic Regression model will be **fit to the entire 80% of the training data** using the fit() method. This allows the model to learn the relationship between the features and the target variable (e.g., the URL classification status).

- **Incremental Training**: Once the initial model is trained on the 2019 data, the model will be incrementally fed new data year by year. Specifically, the model will be trained on **5,000 new data points per year** starting with the 2020 dataset. This incremental training ensures that the model adapts to trends and changes in the data over time without requiring a full retraining from scratch. For example, after training on the 2019 data, the model will be trained on the first **5,000 entries from the 2020 dataset**, then the next **5,000 entries from the 2021 dataset**, and so on, for each subsequent year.

- **Benchmarking with 2024 Data**: To evaluate the model's performance over time, the **2024 data** will be used as a benchmark. After each incremental training step, the model will be tested on the **2024 dataset** to measure its accuracy and ensure that it continues to generalize well on the most recent data. This allows for continuous monitoring of the model's effectiveness as it learns from more recent data. By repeatedly testing the model on the 2024 data, we can track how well the model adapts to newer trends and whether its accuracy improves, stagnates, or degrades as more data is added for training.

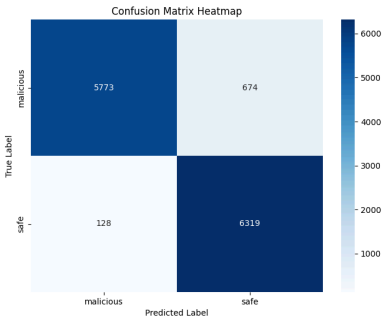
4. RESULT AND DISCUSSION

i. Initial Model Results

Table 1. **Initial Results for the Model**

Metric	Malicious	Safe	Overall
Precision	0.98	0.90	0.94
Recall	0.90	0.98	0.94
F1-Score	0.94	0.94	0.94
Accuracy	-	-	0.94

Figure 3. **Confusion Matrix for the Initial Data**

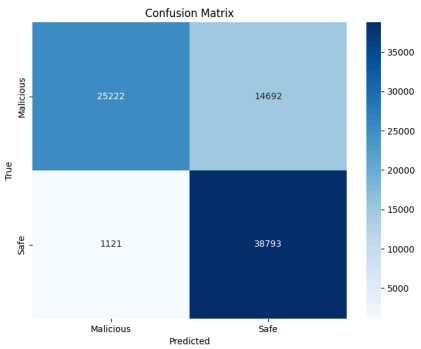


The model was initially trained on the 2019 dataset, achieving an accuracy of 94%. The performance metrics indicated strong results across both classes. Specifically, the model achieved a precision of 98% for malicious URLs and 90% for safe URLs. The recall for malicious URLs was 90%, while for safe URLs, it was 98%. The F1 score, which balances precision and recall, was 0.94 for both categories.

Table 2. **Results Tested on Benchmark Data, Initial Training**

Metric	Malicious	Safe	Overall
Precision	0.96	0.73	0.84
Recall	0.63	0.97	0.80
F1-Score	0.76	0.83	0.80
Accuracy	-	-	0.80

Figure 4. **Confusion Matrix for the Benckmark Data, Initial Training**



When the model was evaluated on the 2024 dataset, the accuracy dropped to 80%. Despite the drop in accuracy, the model still showed reasonable performance. The precision for malicious URLs remained high at 96%, but the precision for safe URLs was lower, at 73%. Recall was notably lower for malicious URLs at 63%, while the recall for safe URLs was strong at 97%. The F1 score for the 2024 data was 0.80, with a score of 0.76 for malicious URLs and 0.83 for safe URLs.

The confusion matrix for the 2024 dataset revealed that the model correctly identified 25,310 malicious URLs and 10,604 safe URLs. However, it misclassified 14,610 malicious URLs as safe, and 10,604 safe URLs were incorrectly identified as malicious.

These results highlight the model's strong initial performance, but also indicate the challenges of generalizing to newer data, particularly for malicious URLs, which saw a notable drop in recall.

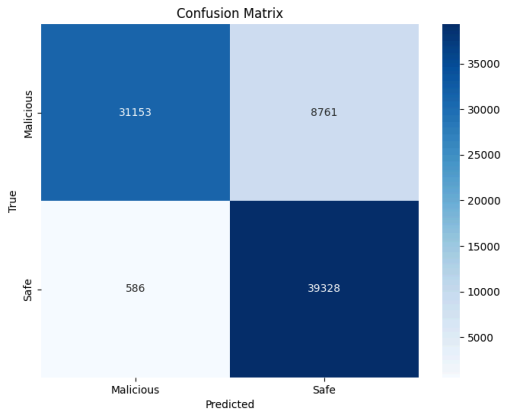
ii. Iterative Model Results

In the iterative step, data from subsequent years were incorporated and used as training data. 5000 data entries were used for each year.

a. Model's Performance After 2020 Data

After incorporating data from 2020, the model's performance slightly decreased, with accuracy dropping to 88% and F1-score to 0.88. Despite the drop, the model maintained good precision and recall for malicious URLs, though recall for malicious URLs dropped to 0.78. Performance on the 2024 dataset remained consistent, with an accuracy of 0.88 and an F1-score of 0.80.

Figure 5. **Confusion Matrix for the Benckmark Data, 2020 Iteration**

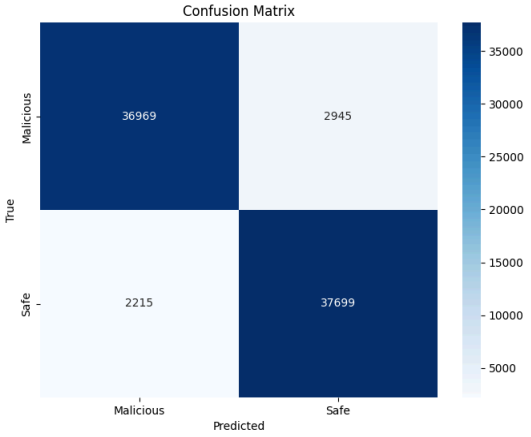


b. Model's Performance After 2021 Data

After incorporating data from 2021, the model's performance improved significantly. The accuracy increased to 0.95, and the F1-score reached 0.95. Precision and recall for both the malicious and safe classes also saw improvements, with F1-scores for both classes around 0.94.

Performance on 2024 Data: When tested on the 2024 dataset, the model achieved an accuracy of 0.92, with an F1-score of 0.92. The malicious class had a precision of 0.94 and recall of 0.86, while the safe class had a recall of 0.99 and precision of 0.88.

Figure 6. **Confusion Matrix for the Benckmark Data, 2021 Iteration**

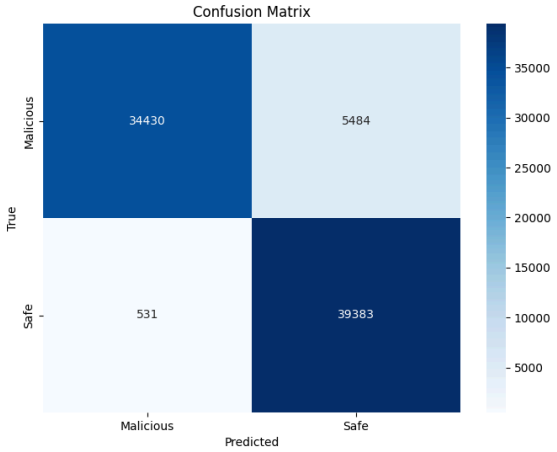


c. Model's Performance After 2022 Data

The inclusion of data from 2022 further improved the model's performance. Accuracy remained stable at 0.94, and the F1-score held steady at 0.94. The model continued to show strong precision and recall for both classes, with balanced performance across the malicious and safe classes.

Performance on 2024 Data: The model's performance on the 2024 dataset was again solid, maintaining an accuracy of 0.92 and an F1-score of 0.92. Both classes (malicious and safe) were correctly identified with high precision and recall. However, the recall for the malicious urls was a bit lower than the previous iteration.

Figure 7. **Confusion Matrix for the Benckmark Data, 2022 Iteration**

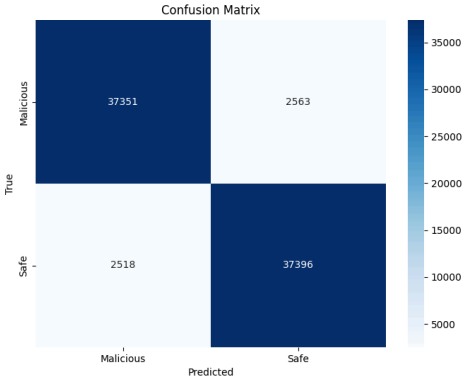


d. Model's Performance After 2023 Data

The final iteration incorporated data from 2023. This addition resulted in a slight improvement in model accuracy, which remained at 0.93, while the F1-score stayed at 0.93. The model's precision for the malicious class was 0.91, and its recall was 0.95. The safe class showed a precision of 0.95 and recall of 0.91.

Performance on 2024 Data: When tested on the 2024 dataset, the model demonstrated a precision of 0.94 and recall of 0.94 for both the malicious and safe classes. The overall performance was excellent, with an accuracy of 0.94 and an F1-score of 0.94.

Figure 8. **Confusion Matrix for the Benckmark Data, 2023 Iteration**



iii. Examination of the Top Features for Predictions

In addition to evaluating the model's performance on a benchmark dataset, we also examined the coefficients of each model iteration to identify the most influential features contributing to phishing predictions over the years.

Table 3. **Top Features indicating the Status of the URL from 2019-2023**

Year	**Feature**	**Coefficient**
2019	contains_https	8.337084
	contains_subdomain	7.116640
	unsafeKeywords_login	-4.834565
	unsafeKeywords_secure	-4.365772
	contains_ip_address	-4.268781
2020	contains_subdomain	6.093110
	contains_https	5.661759
	unsafeKeywords_login	-4.178875
	top_level_domain_edu	3.544091
	top_level_domain_org	3.326141
2021	contains_subdomain	7.077856
	contains_https	5.223093
	unsafeKeywords_login	-3.340850
	top_level_domain_edu	3.209867
	top_level_domain_org	3.178227
2022	contains_subdomain	7.078866
	contains_https	5.436827
	unsafeKeywords_login	-3.927167
	top_level_domain_cn	-3.404303
	top_level_domain_ca	3.337512
2023	contains_subdomain	5.418312
	contains_https	4.011888
	top_level_domain_uk	3.571878
	unsafeKeywords_login	-3.482007
	top_level_domain_ca	3.417930

The coefficients of these features indicate their relationship with the model's predicted outcome, with positive values suggesting that the feature contributes to the likelihood of the URL being legitimate, and negative values pointing to an association with phishing URLs.

Consistently important features include contains_https and contains_subdomain, both showing strong positive coefficients, indicating a higher likelihood of legitimacy when these attributes are present. Features related to unsafe keywords, such as unsafeKeywords_login and unsafeKeywords_secure, consistently show negative coefficients, highlighting their role in identifying phishing URLs.

Over time, the importance of top-level domain (TLD) features, like .edu, .org, and .cn, increases, reflecting evolving domain trends and their association with URL legitimacy. These domain-specific patterns become more predictive in later years, alongside the stable importance of secure communication indicators like HTTPS.

iv. Comparison to Other Methods

In this section, we compare the performance of the current model (achieving 94% accuracy) against two other models. The purpose of this comparison is to evaluate the improvements made in the current model by considering different approaches used during model training, as well as to analyze the overall predictive capabilities of each model.

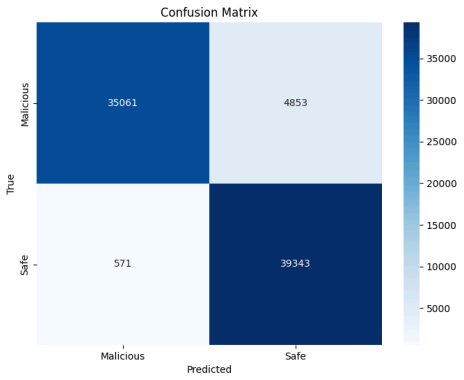
a. First Control Model: Combined Dataset Approach

The first model was trained using a combination of all available datasets into one unified training set. The model was trained without distinguishing between years or specific datasets. This approach provided a broad view of the data, allowing the model to learn patterns from the entire dataset. The performance metrics for this model are as follows:

- Accuracy: 0.93
- F1-Score: 0.93
- Malicious Class Precision: 0.98
- Malicious Class Recall: 0.88
- Safe Class Precision: 0.89
- Safe Class Recall: 0.99

Although the first model performed reasonably well with an overall accuracy of 93%, there were certain imbalances in its precision and recall for the malicious class. The malicious class had high precision (0.98), but relatively low recall (0.88), indicating that while the model was good at predicting malicious cases, it missed a significant number of them. The safe class, on the other hand, had excellent recall (0.99) but lower precision (0.89), meaning that the model predicted many safe cases but also incorrectly classified a fair number as safe when they were actually malicious.

Figure 9. **Confusion Matrix for the 1st Control Model**



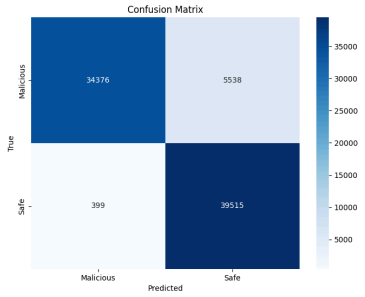
b. Second Control Model: 2022 Dataset Approach

The second model was trained exclusively on the 2022 dataset. This focused training approach aimed to capture the patterns and trends present in that specific year's data, under the assumption that 2022 data might hold distinctive characteristics that could boost performance. However, training on a single year of data limited the model's ability to generalize beyond the 2022 dataset.

- Accuracy: 0.93
- F1-Score: 0.93
- Malicious Class Precision: 0.99
- Malicious Class Recall: 0.86
- Safe Class Precision: 0.88
- Safe Class Recall: 0.99

Like the first model, the second model showed a balanced performance but again had issues with imbalanced precision and recall, especially in predicting the malicious class. Although it achieved high precision for the malicious class (0.99), the recall was still low (0.86), meaning it missed a significant number of malicious cases. The safe class continued to show very high recall (0.99), but the precision remained slightly lower at 0.88.

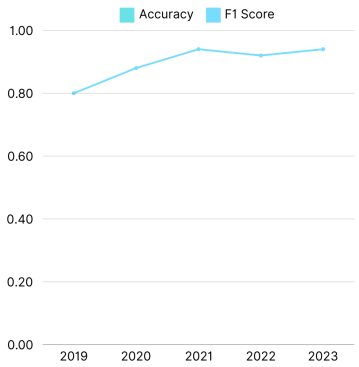
Figure 10. **Confusion Matrix for the 2nd Control Model**



v. Discussion

The model showed a similar level of accuracy and f1 score due to the balanced nature of the dataset.

Figure 9. **Overall Accuracy and F1 Score throught the Years**



The iterative training approach, where data from successive years (2020-2023) was progressively added, demonstrated clear improvements in the model's performance. Initially, the model performed strongly on the 2019 dataset, with a precision of 98% for malicious URLs and a recall of 90%. However, when tested on the 2024 dataset, the performance dropped slightly, particularly in recall for the malicious class, highlighting challenges in generalization to newer data.

Incorporating data from 2020 resulted in a small decrease in performance, but recall for malicious URLs remained relatively stable. The model's performance further improved with the addition of 2021 data, as both precision and recall for the malicious class showed considerable gains. By 2022 and 2023, the model had reached a high level of stability and accuracy, with performance on the 2024 data maintaining solid results.

The final iteration, incorporating 2023 data, led to a significant improvement in recall for both classes, achieving an impressive balance with high precision. This iteration maintained an accuracy of 94% and an F1-score of 0.94 on the 2024 dataset, illustrating the model’s robustness in detecting both malicious and safe URLs across a variety of timeframes.

The confusion matrices provided insights into the model’s strengths and weaknesses, with misclassifications notably occurring for malicious URLs, especially in earlier iterations. These results highlight the importance of progressively incorporating newer data to enhance model generalization, reduce misclassifications, and improve overall performance.

5. CONCLUSION

In this study, we developed a machine learning model to detect phishing URLs using logistic regression. Through a systematic approach, we preprocessed the data, selected relevant features, and trained the model on phishing and legitimate URLs. Our model showed promising results, achieving high precision and recall scores, especially as we incorporated more years of data (2020–2023). This iterative process allowed the model to adapt to changes in phishing tactics over time, improving its accuracy in detecting phishing URLs.

The final model demonstrated robustness, balancing false positives and false negatives while consistently performing well across all years. The analysis of top features revealed critical patterns in URL characteristics that helped the model distinguish between legitimate and phishing websites.

Although the model shows strong performance, there is room for improvement. Future work could involve exploring other machine learning algorithms to further enhance performance, incorporating additional features, or updating the dataset with a larger set of recent phishing data to ensure the model remains relevant in the face of evolving phishing techniques.

In conclusion, this model holds significant potential for real-time phishing URL detection systems. It provides an effective method for identifying harmful websites and can be integrated into security tools to help protect users from phishing attacks.

Acknowledgement

We would like to express our gratitude to Dr. Kritsada Sriphaew, whose guidance was invaluable in applying logistic regression models to detect phishing URLs. We would also like to acknowledge the collective efforts of all the authors in conducting this research on phishing URL detection using logistic regression. This project would not have been possible without the collaborative contributions of each team member. Each author played a crucial role in the research design, data collection, feature extraction, model implementation, and analysis of results.

References

1. Alkhalil, Z., Hewage, C., Liqaa, N., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science, 3*. https://doi.org/10.3389/fcomp.2021.563060
2. Kadlak, A., & Sharma, S. (2018). Study on phishing attacks. *International Journal of Computer Applications, 182*, 27–29. https://doi.org/10.5120/ijca2018918286
3. Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia Medica, 24*(1), 12–18. https://doi.org/10.11613/BM.2014.003
4. Starbuck, C. (2023). Logistic regression. In *The fundamentals of people analytics*. Springer, Cham. https://doi.org/10.1007/978-3-031-28674-2_12
5. Kumar, S. (2019). *Malicious and Benign URLs* [Data set]. Kaggle. https://www.kaggle.com/datasets/siddharthkumar25/malicious-and-benign-urls
6. JPCERT/CC. (n.d.). *phishurl-list*. GitHub. https://github.com/JPCERTCC/phishurl-list/tree/main
7. Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: A literature survey. *IEEE Communications Surveys & Tutorials, 15*(4), 2091-2121. https://doi.org/10.1109/SURV.2013.032213.00009
8. Jeeva, S. C., & Rajsingh, E. B. (2016). Intelligent phishing URL detection using association rule mining. *Human-Centric Computing and Information Sciences, 6*(10). https://doi.org/10.1186/s13673-016-0064-3
9. Green, G. H., Boze, B. V., Choundhury, A. H., & Power, S. (1998). Using logistic regression in classification. *Marketing Research, 10*(3), 4-31. https://search.ebscohost.com/login.aspx?direct=true&db=bsu&AN=1303507&site=ehost-live
10. Feroz, M. N., & Mengel, S. (2014). Examination of data, rule generation and detection of phishing URLs using online logistic regression. *2014 IEEE International Conference on Big Data (Big Data)*, 241-250. https://doi.org/10.1109/BigData.2014.7004239
11. Chiramdasu, R., Srivastava, G., Bhattacharya, S., Reddy, P. K., & Gadekallu, T. R. (2021). Malicious URL detection using logistic regression. *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, 1-6. https://doi.org/10.1109/COINS51742.2021.9524269
12. Zouina, M., & Outtaj, B. (2017). A novel lightweight URL phishing detection system using SVM and similarity index. *Human-Centric Computing and Information Sciences, 7*(17). https://doi.org/10.1186/s13673-017-0098-1
13. Shahrivari, V., Darabi, M. M., & Izadi, M. (2020). *Phishing detection using machine learning techniques*. arXiv. https://arxiv.org/abs/2009.11116