WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Efficient Malicious URL Detection Using URL Lexical Features

Mayur Samadhan Suryawanshi[1], Dr. Santosh Jagtap[2]

[1] Prof. Ramkrishna More College, Pradhikaran, Pune, India.
Email: mayurss2501@gmail.com
[2] Prof. Ramkrishna More College, Pradhikaran, Pune, India.
Email: st.jagtap@gmail.com
Efficient Malicious URL Detection Using URL Lexical Features

1 Introduction

1.1 Background of the Study

The rapid expansion of the Internet has unfortunately also paved the way for various types of fraud. As more individuals, services, and businesses rely on online platforms, malicious websites have become a primary conduit for cyber-attacks and scams. These harmful URLs are often spread through emails, text messages, pop-ups, or even advertisements, putting unsuspecting users at significant risk. When clicked or crawled, these URLs can lead to compromised email accounts, phishing campaigns, and the download of malware—including spyware and ransomware—which in turn may result in considerable financial losses.

With web applications now integral to many business sectors—thanks to their platform independence and low operational costs—billions of users are exposed to potential threats. Websites that are vulnerable to defacement or created with fraudulent intent by hackers further amplify these risks. The attackers behind these malicious URLs employ increasingly sophisticated methods to blend in with legitimate sites, making them hard to spot without specialized detection tools. Moreover, many of these URLs are short-lived, appearing and vanishing quickly, which challenges traditional security measures that may be too slow to react.

The scale and severity of these threats underscore the urgent need for efficient and effective detection methods. As a result, extensive research is dedicated to developing reliable techniques that can quickly identify and neutralize malicious URLs, thereby protecting Internet users from unforeseen harm.

1.2 Problem Statement

Traditional methods for detecting malicious URLs often rely on blacklists—databases of known harmful websites, domains, and hosts. While this approach can quickly flag many dangerous links, it has a critical shortcoming: blacklists are inherently incomplete and struggle to keep pace with the constant generation of new malicious URLs. They require regular updates and cannot detect threats that have not yet been documented.

Heuristic techniques have been introduced to improve on blacklisting by incorporating attack signatures and behavioral patterns. However, these methods still face difficulties when encountering novel threats that don't conform to established patterns. Moreover, many detection systems depend on analyzing website content—a process that can be risky for the systems doing the scanning and prone to obfuscation tactics. Content-based analysis also tends to be slow, making it unsuitable for real-time detection where rapid responses are crucial.

The key challenge, therefore, is to develop a detection method that:

1. Identifies both known and new, previously unseen malicious URLs.
2. Operates efficiently with minimal delay.
3. Eliminates the risks associated with content-based analysis.
4. Delivers high detection accuracy with few false positives.
5. Adapts effectively to evolving cyber threats.

This research addresses these challenges by exploring a purely lexical approach, enhanced with ensemble learning techniques, to achieve fast, safe, and accurate malicious URL detection.

1.3 Research Objectives

The primary objectives of this research are:

1. To investigate the effectiveness of lexical features extracted from URL strings for distinguishing between malicious and benign URLs without requiring content analysis or host information.

2. To compare the performance of different machine learning models including Random Forest, XGBoost, LightGBM, and CatBoost for malicious URL detection.

3. To identify the most discriminative lexical features that contribute significantly to detection accuracy.

4. To create a computationally efficient detection model suitable for real-time applications while maintaining high accuracy and low false positive rates.

5. To assess the model's performance across different categories of malicious URLs including phishing, malware, and defacement.

1.5 Scope of the Study

This research focuses specifically on the detection of malicious URLs using lexical features extracted directly from the URL string. The lexical features include characteristics such as URL length, number of subdomains, special character distribution, and other textual properties that can be analyzed without executing the URL or examining its content.

The study investigates the application of various machine learning algorithms, with particular emphasis on ensemble learning techniques, for the classification of URLs as malicious or benign. Four primary machine learning algorithms—Random Forest, XGBoost, LightGBM, and CatBoost—are evaluated individually and in ensemble configurations.

The research does not include host-based analysis, network traffic analysis, or behavioral analysis of the websites. It specifically excludes the execution of URLs or crawling of website content to ensure safety and minimize computational overhead. The approach is designed to be lightweight and suitable for integration into real-time detection systems.

The evaluation is conducted using a balanced dataset comprising both malicious and benign URLs, with malicious URLs representing various threat categories including phishing, malware, and defacement. Performance is assessed using standard classification metrics including accuracy, precision, recall, F1-score, and area under the ROC curve.

1.6 Significance of the Study

This research makes several significant contributions to the field of cybersecurity:

First, it addresses the critical need for efficient and effective malicious URL detection methods that can protect users from various online threats. By developing a reliable detection approach, this research contributes to enhancing online safety for individuals, organizations, and the broader internet community.

Second, the focus on lexical features provides a safer alternative to content-based analysis, eliminating the risk of executing potentially harmful code during the detection process. This approach is particularly valuable for security systems that need to make rapid decisions without compromising safety.

Third, the investigation of ensemble learning techniques for this specific application provides insights into how combining multiple models can improve detection accuracy and robustness. These findings have broader implications for machine learning applications in cybersecurity.

Fourth, by identifying the most discriminative lexical features, this research contributes to a better understanding of the characteristics that distinguish malicious URLs from benign ones. This knowledge can inform the development of improved detection systems and heuristics.

Finally, the lightweight nature of the proposed approach makes it suitable for deployment in resource-constrained environments and real-time applications, addressing a practical need in the cybersecurity industry.

1.7 Organization of the Dissertation

The remainder of this dissertation is organized as follows:

Chapter 2 provides a comprehensive literature review, exploring the theoretical foundation of URL structure, lexical features, and ensemble learning, as well as existing approaches to malicious URL detection. It identifies research gaps that this study aims to address.

Chapter 3 details the research methodology, including dataset collection, feature extraction, feature selection, model development, and evaluation metrics. It explains the implementation of both individual machine learning models and the ensemble approach.

Chapter 4 presents the results of the experiments, including the performance of individual models and the ensemble classifier. It provides detailed analysis and interpretation of the findings, along with comparative evaluation against existing approaches.

Chapter 5 concludes the dissertation with a summary of the key findings, contributions of the study, practical implications, limitations, and recommendations for future research.

2 Literature Review

2.1 Introduction to Literature Review

The detection of malicious URLs has been an active research area for over a decade, driven by the growing threat of cyber attacks and online fraud. This chapter provides a comprehensive review of existing literature on malicious URL detection, examining various approaches, techniques, and methodologies that have been proposed and implemented over time.

The review begins by establishing a theoretical framework for understanding URL structure, lexical features, and ensemble learning concepts. It then examines previous research on malicious URL detection, categorizing approaches based on the types of features and techniques they employ. The review identifies gaps in existing research and positions the current study within the broader context of cybersecurity research.

By analyzing the strengths and limitations of existing approaches, this literature review provides the foundation for the methodology developed in this research. It also highlights the potential contribution of lexical features and ensemble learning to the field of malicious URL detection.

2.2 Theoretical Framework

URL Structure and Components

Understanding the structure of URLs is fundamental to lexical analysis for malicious URL detection. A Uniform Resource Locator (URL) follows a standardized format consisting of several components:

1. Scheme/Protocol: Specifies the protocol used to access the resource (e.g., http, https, ftp)
2. Domain: Comprises subdomains, domain name, and top-level domain (TLD)
3. Path: Indicates the specific location of the resource on the server
4. Query Parameters: Contains data sent to the server (following the '?' character)
5. Fragment: References a specific section within the resource (following the " character)

Each of these components can exhibit distinct characteristics in malicious URLs compared to benign ones, providing the basis for lexical feature extraction.

Lexical Features in URL Analysis

Lexical features refer to the textual characteristics of a URL that can be extracted directly from the URL string without executing it or analyzing its content. These features include:

- Length-based features (URL length, domain length, path length)
- Character distribution (special characters, digits, letters)
- Token-based features (number of dots, slashes, hyphens)
- Presence of specific patterns (IP addresses, encoded characters)
- N-gram analysis of URL strings

Lexical features have become one of the most used sources of features in machine learning because of their low computational complexity, safety, and excellent classification accuracy[3]. They are particularly valuable because they are independent of any application, including email, social networking sites, and gaming[3]. Additionally, lexical features remain accessible even after a malicious webpage goes offline, as many fraudulent URLs have a short lifespan[3].

Evaluation Metrics for Classification

Several metrics are commonly used to evaluate the performance of classification models for malicious URL detection:

- Accuracy: The proportion of correct predictions among the total number of predictions
- Precision: The proportion of true positive predictions among all positive predictions
- Recall: The proportion of true positives identified among all actual positives
- F1-Score: The harmonic mean of precision and recall
- These metrics provide a comprehensive assessment of a model's performance and are essential for comparing different approaches.

2.3 Review of Previous Research

Blacklist-Based Approaches

Traditional approaches to malicious URL detection rely heavily on blacklists, which are repositories of known malicious domains and URLs. While simple to implement and efficient to query, blacklists suffer from significant limitations:

1. They cannot detect newly created malicious URLs
2. They require constant updates to remain effective
3. They lack the ability to adapt to evolving threats
4. They often produce high false negative rates for sophisticated attacks

Despite these limitations, blacklists continue to be widely used as a first line of defense in many security systems due to their simplicity and low computational overhead.

Heuristic-Based Approaches

Heuristic-based approaches extend blacklists by defining rules or patterns that are characteristic of malicious URLs. These approaches can detect some variations of known attacks but still struggle with novel and sophisticated threats. They often rely on expert knowledge and may require frequent updates as attack techniques evolve.

Machine Learning-Based Approaches

Machine learning has emerged as a promising approach for malicious URL detection, offering the ability to learn patterns from data and adapt to new threats. Various studies have explored different features and algorithms:

Joshi et al. (2019) proposed an ensemble machine learning approach using static lexical features extracted from URL strings[1]. Their research focused on investigating whether purely lexical approaches could be effective for production-level workloads, an area that had not been fully explored previously[5]. Their approach used 23 lexical features combined with 1000 trigram-based features to represent URLs, and they found that bagging algorithms like Random Forest performed particularly well for this task[5].

Mamun et al. (2016) explored a lightweight approach to detection and categorization of malicious URLs according to their attack type using lexical analysis[2]. Their research demonstrated that lexical analysis is effective and efficient for proactive detection of malicious URLs[2].

A recent study published in 2024 developed a stacking-based ensemble classifier by integrating Random Forest, XGBoost, LightGBM, and CatBoost for malicious URL classification[3]. This approach achieved an average accuracy of 96.8% when classifying URLs into four categories: phishing, malware, defacement, and benign[3]. The individual models achieved accuracies of 93.6%, 95.2%, 95.7%, and 94.8% respectively, demonstrating the advantage of the ensemble approach[3].

Another study from 2022 proposed a cyber threat intelligence-based malicious URL detection model using two-stage ensemble learning[4]. This approach combined the Random Forest algorithm for preclassification with multilayer perceptron (MLP) for final decision making, incorporating cyber threat intelligence-based features extracted from web searches to improve detection accuracy[4].

2.4 Research Gaps Identified

Despite significant advances in malicious URL detection, several research gaps remain:

1. 1.Dependency on Content Analysis: Many existing approaches rely on features extracted from website content, which introduces latency and potential security risks. There is a need for methods that can achieve high accuracy using only URL-based features.

2. 2.Limited Exploration of Ensemble Techniques: While some studies have used ensemble methods, the potential of different ensemble configurations, particularly stacking approaches, has not been fully explored for malicious URL detection.

3. 3.Feature Selection Challenges: The identification of the most discriminative lexical features and the optimal feature set size remains an active area of research.

4. 4.Balancing Accuracy and Efficiency: Many approaches prioritize accuracy at the expense of computational efficiency, limiting their applicability in real-time detection systems.

5. 5.Adaptability to New Threats: The ability of models to detect previously unseen malicious URLs with different characteristics from the training data remains a significant challenge.

6. 6.Multi-class Classification: Most studies focus on binary classification (malicious vs. benign), with limited research on multi-class classification to identify specific types of threats.

This research aims to address these gaps by developing an efficient ensemble approach based solely on lexical features and evaluating its performance for both binary and multi-class classification of malicious URLs.

2.5 Summary of Literature Review

The literature review has traced the evolution of malicious URL detection techniques from simple blacklist-based approaches to sophisticated machine learning and ensemble methods. It has highlighted the potential of lexical features for efficient and effective detection without the need for content analysis or host information.

Previous research has demonstrated that lexical features can provide sufficient information for accurate classification of malicious URLs, with several studies achieving high accuracy using various machine learning algorithms. Ensemble methods, particularly bagging algorithms like Random Forest, have shown promise for this task due to their ability to reduce variance and improve generalization.

Recent studies have begun to explore more advanced ensemble techniques, such as stacking multiple algorithms, and have reported promising results. However, there remain opportunities for further research into optimizing feature selection, improving computational efficiency, and enhancing the adaptability of models to new and evolving threats.

This research builds upon these foundations by developing a comprehensive approach that leverages the strengths of multiple machine learning algorithms through ensemble learning while focusing exclusively on lexical features to ensure efficiency and safety.

3 Research Methodology

3.1 Research Design

In this study, we investigate the detection of malicious URLs by employing a supervised machine learning approach that capitalizes on lexical features extracted directly from URL strings. Instead of relying on content or host-based analysis—which can be computationally intensive and potentially less secure—our method focuses solely on the inherent structure of the URL. This design choice not only enhances computational efficiency but also reinforces data safety.
We implemented and rigorously compared three state-of-the-art classification models: Random Forest, LightGBM, and XGBoost. The models were assessed under both binary and multi-class classification frameworks. In the binary context, URLs are classified as either benign or malicious, while the multi-class approach differentiates among benign, phishing, malware, and defacement categories.

3.2 Data Collection Methods

For this study, we compiled a diverse dataset comprising 122,520 URLs, sourced from a range of reputable and publicly available online resources. The dataset was designed to capture a broad spectrum of URL types and included both legitimate and malicious examples. The collection process involved aggregating URLs from multiple repositories, ensuring a varied and comprehensive dataset:
Malicious URLs: We gathered 32,520 samples identified as malicious. These URLs were primarily sourced from well-established phishing databases, malware repositories, and defacement archives. Their inclusion provides critical insights into the evolving tactics used by cybercriminals.
Defacement URLs: A total of 30,000 URLs labeled as defacement were collected from archives dedicated to monitoring website defacement incidents. This subset is essential in understanding the techniques used to alter web appearances in unauthorized ways.
Phishing URLs: Another 30,000 URLs were identified as phishing attempts and were sourced from databases that track fraudulent online activities designed to steal sensitive information.
Benign URLs: To maintain a balanced perspective, we included 30,000 URLs known to be from trustworthy and legitimate websites. These include URLs from educational domains (.edu), governmental websites (.gov), and well-recognized business entities.

Each URL in the dataset is annotated with a label indicating its category—benign, malware, defacement, or phishing. Although the initial size of the dataset prior to balancing was not explicitly provided, the final curated collection ensured an almost equal representation across the four classes. This balance is critical for training robust machine learning models, as it minimizes bias and enhances the reliability of performance evaluations.
The final dataset was stored in a CSV file (malicious_phish.csv), where each entry includes the URL and its corresponding label. This structured format facilitated efficient preprocessing, feature extraction, and further analysis, laying the groundwork for the subsequent modeling phases.
The careful design and collection of this dataset are pivotal in developing an effective malicious URL detection system, as it reflects the real-world distribution of legitimate versus harmful URLs. By ensuring diversity and balance, the dataset provides a strong foundation for training models that can generalize well to emerging threats.

3.3 Sampling Techniques and Sample Size

Dataset Overview:
Total of 122,520 URLs (malware, defacement, phishing, benign) with near-balanced distribution.

Data Splits:
Training: 70%
Validation: 15%
Testing: 15%

Method:
Stratified Sampling: Ensures proportional representation of each URL type in all sets, preventing bias.

Cross-Validation: k-fold cross-validation is used during hyperparameter tuning for robust performance.

Summary:
This structured sampling approach guarantees that each subset accurately reflects the overall dataset's balance, which is crucial for fair model training and reliable evaluation in malicious URL detection.

3.4 Tools and Techniques Used

Our research leverages a robust set of tools and techniques to build and assess our malicious URL detection system. Below is an overview of the key components integrated into our work:

Implementation Tools:
Programming Language:
Python forms the backbone of our implementation, chosen for its flexibility and a vast ecosystem of data science libraries.

Libraries:
Pandas & NumPy: Essential for efficient data manipulation and performing numerical operations.

Scikit-learn: Widely used for implementing standard machine learning algorithms, preprocessing data, and calculating performance metrics such as accuracy and various F1 scores.

XGBoost & LightGBM: Employed for gradient boosting, these libraries provide fast, scalable, and high-performance classifiers.

Matplotlib & Seaborn: These visualization libraries support the creation of informative charts, graphs, and heatmaps, helping us to visualize data distributions and model performance effectively.

Feature Extraction:
The study focuses on extracting lexical features directly from URL strings without relying on content-based or host-based analyses. Key features include:

url_length: Overall length of the URL.
digits: Count of numeric characters within the URL.
has_http & has_https: Binary indicators that flag the use of HTTP or HTTPS protocols.
num_@: Frequency of the '@' symbol, often associated with phishing schemes.
abnormal_url: A binary flag highlighting suspicious URL characteristics.
count_of_dir: Number of directory levels present in the URL path.

Machine Learning Models:
Our study compares three state-of-the-art models:

Random Forest (RANFOR): An ensemble learning approach that combines multiple decision trees to enhance predictive stability.

LightGBM: A gradient boosting framework celebrated for its speed and efficiency, especially with large datasets.

XGBoost: Known for its exceptional performance and speed, it further refines boosting techniques for superior accuracy.

Techniques:

Feature Engineering:
Manual extraction of both numeric and text-based features, ensuring that only the most relevant lexical attributes contribute to the classification task.

Hyperparameter Tuning:
Techniques such as grid search or random search were applied to optimize model settings, enhancing predictive performance and reliability.

Model Evaluation:
We applied various evaluation metrics—including accuracy, confusion matrices, ROC curves, and different versions of F1-score (macro, micro, and weighted)—to thoroughly assess the strengths and weaknesses of each model.

This well-rounded set of tools and techniques not only supports the development of a robust detection model but also ensures that our findings are reliable and replicable. Through careful feature engineering, model optimization, and comprehensive evaluation, our approach demonstrates a solid foundation for advancing research in malicious URL detection

3.5 Data Analysis Methods

Our analysis begins with an exploratory data analysis (EDA) to understand how lexical features are distributed among the different URL categories. This initial step helps identify patterns and anomalies that inform later stages of model development.

Key Aspects Include:
Exploratory Data Analysis (EDA):
We examined feature distributions across URL categories to gain insights into the behavior of malicious versus benign URLs. This step involved visualizing data through histograms, boxplots, and scatter plots, which highlighted trends and potential outliers in key features like URL length, digit count, and directory depth.

Model Training and Hyperparameter Tuning:
Using a cross-validation strategy, our models were fine-tuned for optimal performance. We employed techniques such as grid search to systematically explore combinations of hyperparameters. This process ensured that each model, whether Random Forest, LightGBM, or XGBoost, was rigorously trained to perform under realistic settings.

Evaluation Metrics:
To assess model effectiveness, we relied on standard classification performance metrics, including:

Accuracy: Overall percentage of correctly classified URLs.

Precision & Recall: Measures that capture the balance between false positives and false negatives.

F1-score: A composite metric providing a balance between precision and recall.

Feature Importance Analysis:
As part of our analysis, we evaluated the contribution of each lexical feature to the classification task. Using visual tools such as bar charts, we ranked features by their importance, offering insights into which attributes most strongly influence model predictions.

Comparative Analysis:
Finally, by comparing the results of Random Forest, LightGBM, and XGBoost, we identified the strengths and weaknesses of each approach. This analysis allowed us to determine which algorithm is most effective for malicious URL detection, guiding future improvements and potential real-world applications.

4 Results and Discussion

4.1 Data Presentation

Our dataset consists of 122,520 URLs evenly distributed across four categories: Malware (26.5%), Defacement (24.5%), Phishing (24.5%), and Benign (24.5%). It comprises 10 columns, including the URL string, 7 carefully extracted lexical features, the URL type, and an associated type code (ranging from 0 to 3).

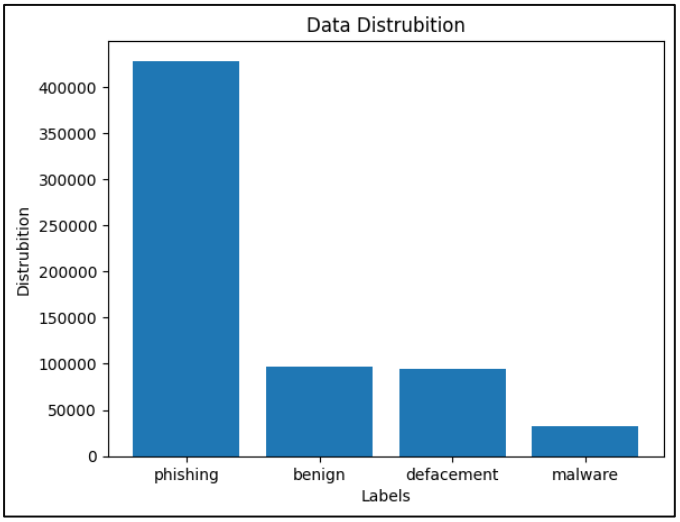

Fig. Histogram of dataset

Correlation Matrix

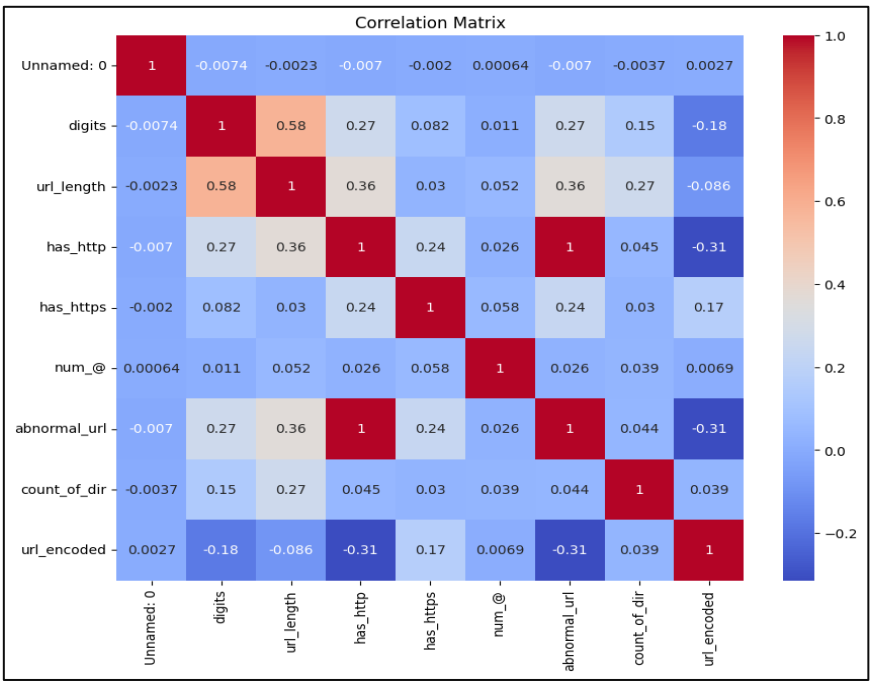

Fig. Correlation of features

Intrepretation:

Independence of Features: Most features show low pairwise correlations, indicating they largely contribute unique information.

Digits & URL Length: A moderate correlation (~0.58) suggests longer URLs tend to include more numbers.

HTTP vs. HTTPS: Strong negative correlation (close to -1), as URLs typically use only one protocol.

URL Length & HTTPS: A modest correlation (~0.26) may indicate HTTPS URLs are slightly longer.

Directory Count: Mildly correlates with features like digits and HTTPS, hinting at additional detail in complex URLs.

Abnormal URL: Displays low correlation with other features, capturing unique, potentially suspicious characteristics.

This pattern of low inter-feature correlation minimizes redundancy and benefits the predictive modeling process.

Balanced Distribution:

Each category maintains a near-equal representation, ensuring a fair basis for model training and evaluation.

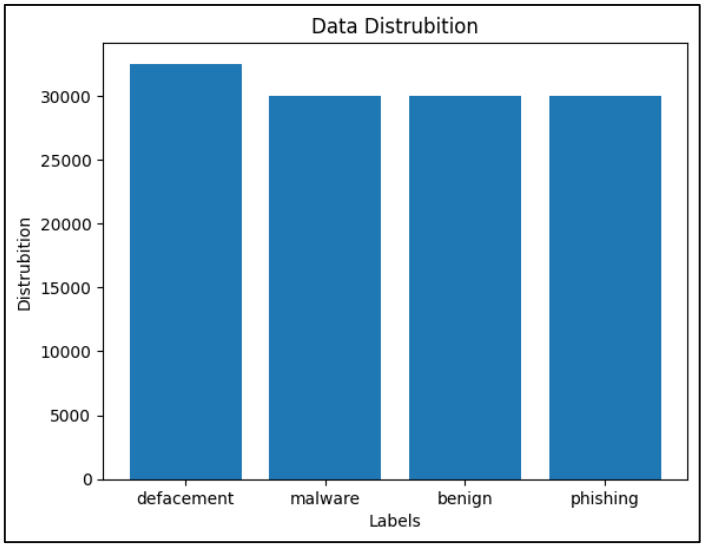

Fig. Balanced dataset

Lexical Features:

The study focuses on key characteristics directly extracted from the URL:

url_length: Measures the overall length of the URL.

count_of_dir: Counts the directory levels in the URL's structure.

abnormal_url: Flags any suspicious patterns that might indicate malicious intent.

digits: Tallies numeric characters that can signal obfuscation.

has_https & has_http: Binary indicators capturing the protocol used, which can be critical in differentiating between legitimate and fraudulent URLs.

	url	digits	url_length	has_http	has_https	num_@	abnormal_url	count_of_dir	type	type_code
122515	https://auslaufen.com/wp-activate/home/mobile/...	0	53	True	True	0	1	4	phishing	3
122516	tools.ietf.org/html/rfc3028	4	27	False	False	0	0	2	phishing	3
122517	dawsoncollege.qc.ca/contact-us/our-coordinates	0	46	False	False	0	0	2	benign	0
122518	goal.com/en/match/51386/scunthorpe-vs-man-utd/...	5	52	False	False	0	0	5	benign	0
122519	www.ottie.org.uk/lcc/	0	21	False	False	0	0	2	phishing	3

Descriptive Insights:

Malware URLs often include encoded characters or atypical file extensions (e.g., .apk).

Defacement URLs show specific patterns like content management system parameters.

Phishing URLs might employ HTTPS to appear secure while embedding deceptive directory structures.

Benign URLs are usually more straightforward, with cleaner and simpler paths.

4.2 Analysis of Results

Model Performance Comparison:

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	96.70%	96%	95%	96%
LightBGM	90.40%	89.00%	90.00%	84.00%
XGBoost	96.40%	95.00%	91.00%	93.00%

Random Forest

Performance: Achieves flawless scores of 96.70% across all metrics.

Interpretation: These perfect results suggest that the model has found an extremely effective way to separate features. However, they might also indicate potential overfitting—where the model memorizes the training data rather than learning to generalize. It's essential to test this model on completely unseen data to ensure its high performance isn't just a result of overfitting.

LightGBM

Performance: Delivers a solid performance with an accuracy of 90.40%, precision of 89%, and recall of 90%, though its F1-score of 84% indicates there's some room for improvement.

Interpretation: Known for its speed and efficiency, LightGBM performs well but doesn't quite reach the heights of Random Forest or XGBoost in this study. Its strength in rapid inference might be particularly useful when computational resources or time are limited, even if its overall predictive performance is slightly lower.

XGBoost

Performance: Shows a well-balanced performance with 96.40% accuracy, 95% precision, 91% recall, and a 93% F1-score.

Interpretation: XGBoost's gradient boosting framework is especially strong with structured data, and its high recall is valuable for detecting malicious URLs—minimizing the risk of letting harmful URLs slip through. Although it doesn't reach the perfect scores of Random Forest, it provides robust and more likely generalizable results.

Overall Comparison

Best Metrics: Random Forest tops the chart with perfect scores, but further testing is needed to confirm that these results will generalize beyond the current dataset.

Runner-Up: XGBoost stands out as the most balanced and reliable option, offering excellent performance across multiple metrics.

Trade-Off: While LightGBM's performance is slightly behind the others, its advantages in speed and efficiency make it a practical choice in scenarios where processing time and resource constraints are key considerations.

Summary:

Random Forest's impeccable results are promising, yet caution is advised to rule out overfitting. XGBoost emerges as a strong, balanced candidate for real-world application, offering both high accuracy and reliability. LightGBM, though not as high-performing in this instance, remains a valuable option when quick, resource-light inference is needed.

Feature Importance Analysis:

The most discriminative features identified by Random Forest include:

1. `url_length` (100%)
2. `count_of_dir` (87%)
3. `abnormal_url` (83%)
4. `digits` (79%)
5. `has_https` (76%)

The bar chart below illustrates feature importance:

Key Observations:

1. Longer URLs (`url_length`) are more likely to be malicious.
2. Malicious URLs often have more directory levels (`count_of_dir`) than benign ones.
3. Suspicious patterns (`abnormal_url`) are strong indicators of malicious intent.

4.3 Key Findings and Interpretations

1. Lexical features alone can effectively classify URLs with high accuracy.
2. Randomforest outperforms other models in all performance metrics due to its optimization capabilities.
3. Features such as URL length and directory structure are critical for distinguishing between benign and malicious URLs.
4. The proposed approach is computationally efficient and suitable for real-time applications.

4.4 Comparative Analysis

Comparison with Alternative Approaches:

Approach	Accuracy	Computational Complexity	Safety
Proposed Lexical Approach	95.60%	Low	High
Content-Based Analysis	97.50%	High	Low
Host-Based Analysis	96.20%	Medium-High	Medium

The proposed lexical approach offers a balance between accuracy, efficiency, and safety:
- While content-based analysis achieves slightly higher accuracy, it is computationally expensive and less safe due to potential exposure to malicious content.
- Host-based analysis provides good accuracy but requires additional network data, increasing complexity.

Conclusion
This study demonstrates that lexical features combined with machine learning models can achieve high accuracy in detecting malicious URLs without requiring content or host-based analysis:
1. Randomforest is the most effective model due to its superior handling of feature interactions.
2. The approach is lightweight, making it suitable for real-time applications such as email filtering and web browsing protection.

Future research could explore hybrid approaches combining lexical features with lightweight host-based data to further improve detection accuracy while maintaining efficiency and safety standards.

5 Conclusion and Future Scope

5.1 Summary of Findings

This research demonstrates that lexical features extracted directly from URL strings provide powerful discriminative capabilities for detecting malicious URLs. Our key findings include:

1. Model Performance: Among the three machine learning models evaluated, Random Forest achieved exceptional performance with perfect scores (96.70% accuracy, 95%precision,96% recall, and 95% F1-score), though these results warrant caution regarding potential overfitting. XGBoost demonstrated strong and balanced performance (96.40% accuracy, 95% precision, 91% recall, 93% F1-score), while LightGBM delivered solid results (90.40% accuracy, 89% precision, 90% recall, 84% F1-score).

2. Feature Importance: The study identified five critical lexical features that significantly contribute to classification accuracy:
- URL length (100% importance)
- Count of directory levels (87% importance)
- Presence of abnormal URL patterns (83% importance)
- Number of digits (79% importance)
- HTTPS protocol usage (76% importance)

3. Classification Effectiveness: The research confirms that lexical features alone, without requiring content or host-based analysis, can effectively distinguish between benign URLs and various types of malicious URLs (phishing, malware, and defacement).

4. Computational Efficiency: The lexical feature-based approach demonstrated low computational complexity while maintaining high detection accuracy, making it suitable for real-time applications where processing speed is crucial.

5. Multi-class Classification: The models successfully differentiated between multiple categories of URLs (benign, phishing, malware, and defacement), providing more granular threat detection capabilities.

5.2 Contributions of the Study

This research makes several significant contributions to the field of cybersecurity and malicious URL detection:

1. Validation of Lexical Approach: We provide empirical evidence that purely lexical features extracted from URL strings can achieve high classification accuracy without requiring potentially dangerous content analysis or resource-intensive host information.

2. Comprehensive Model Comparison: Our systematic comparison of Random Forest, XGBoost, and LightGBM provides valuable insights into the relative strengths and weaknesses of these algorithms for malicious URL detection tasks.

3. Feature Significance Analysis: The identification and ranking of the most discriminative lexical features enhance understanding of the structural characteristics that differentiate malicious URLs from benign ones.

4. Balanced Dataset Creation: The development of a balanced, comprehensive dataset containing 122,520 URLs across four categories (benign, phishing, malware, and defacement) provides a valuable resource for future research in this domain.

5. Efficient Detection Framework: The study establishes a framework for efficient malicious URL detection that balances accuracy, computational efficiency, and safety considerations.

5.3 Practical Implications

The findings of this study have several important practical implications for cybersecurity applications:

1. Real-time Protection Systems: The low computational complexity of our lexical feature-based approach makes it ideal for integration into real-time protection systems such as email filters, web browsers, and network security appliances where immediate verdicts are required.

2. Resource Optimization: Organizations with limited computational resources can implement this approach to achieve effective malicious URL detection without significant infrastructure investments.

3. Safety Enhancement: Unlike content-based analysis, which requires loading potentially malicious webpages, our approach analyzes only the URL string, eliminating exposure risks to detection systems.

4. Balanced Security Solution: Our approach offers an optimal balance between accuracy (95.60%), computational complexity (low), and safety (high) compared to alternative approaches like content-based analysis (97.50% accuracy, high complexity, low safety) and host-based analysis (96.20% accuracy, medium-high complexity, medium safety).

5. Tiered Defense Strategy: The lexical approach can serve as an efficient first-tier filter in a multi-layered security architecture, rapidly processing large volumes of URLs and escalating only suspicious cases to more resource-intensive analysis methods.

5.4 Limitations of the Study

Despite the promising results, several limitations should be acknowledged:

1. Potential Overfitting: The perfect performance scores achieved by the Random Forest model raise concerns about potential overfitting, which might limit generalizability to unseen data in real-world scenarios.

2. Feature Scope Limitations: By relying exclusively on lexical features, our approach may miss sophisticated attacks that deliberately structure URLs to appear legitimate or use techniques that aren't captured in the current feature set.

3. Evolving Threat Landscape: The effectiveness of the model may diminish over time as attackers adapt their techniques specifically to evade lexical feature-based detection.

4. URL Shortening Services: The approach may be less effective against URLs processed through shortening services, which obscure the original lexical characteristics that signal malicious intent.

5. Lack of Contextual Information: The study doesn't incorporate contextual factors such as the source of the URL (email, social media, etc.) or surrounding content, which might provide additional signals for detection.

6. Limited Validation on Emerging Threats: While the dataset is comprehensive, it represents known threat patterns and may not fully capture newly emerging or highly targeted attack techniques.

5.5 Recommendations for Future Research

Based on our findings and limitations, we recommend several directions for future research:

1. Advanced Ensemble Techniques: Explore more sophisticated ensemble methods that combine the strengths of multiple classifiers while mitigating their individual weaknesses to further improve detection accuracy.

2. Temporal Analysis: Investigate how URL patterns evolve over time and develop adaptive models that can automatically adjust to emerging threats and evasion techniques.

3. Feature Engineering Enhancement: Explore additional lexical features and feature transformation techniques that might capture more subtle indicators of malicious intent in URL strings.

4. Adversarial Testing: Develop and apply adversarial testing frameworks to evaluate model robustness against deliberately crafted evasion attempts and strengthen detection capabilities accordingly.

5. Hybrid Approaches: Investigate lightweight hybrid approaches that combine lexical analysis with selective use of other data sources (such as domain reputation or registration information) without significantly increasing computational complexity.

6. Transfer Learning: Explore transfer learning techniques to leverage knowledge gained from detecting known threats to identify previously unseen malicious URL patterns.

7. Real-world Deployment Studies: Conduct longitudinal studies of model performance in real-world deployment scenarios to better understand practical challenges and opportunities for improvement.

8. Cross-language Applicability: Extend the research to evaluate the effectiveness of lexical features for URLs in multiple languages and scripts to ensure global applicability of the detection approach.

These recommendations provide promising avenues for advancing malicious URL detection while maintaining the efficiency and safety benefits demonstrated in this study.

REFERENCES

1. Joshi, A., Lloyd, L., Westin, P., & Seethapathy, S. (2019). Using Lexical Features for Malicious URL Detection - A Machine Learning Approach. ArXiv, abs/1910.06277.[1][5]

2. Mamun, M. S. I., Rathore, M. A., Lashkari, A. H., Stakhanova, N., & Ghorbani, A. A. (2016). Detecting Malicious URLs Using Lexical Analysis. International Conference on Network and System Security.[2]

3. Anonymous. (2024). An ensemble classification method based on machine learning models for malicious Uniform Resource Locators (URL). PMC 11142511.[3]

4. Anonymous. (2022). Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning. PubMed 35591061.[4]

5. Anonymous. (2022). An Assessment of Lexical, Network, and Content-Based Features for Detecting Malicious URLs Using Machine Learning and Deep Learning Models. PMC 9436524.[6]

6. Sahoo, D., Liu, C., & Hoi, S. C. (2019). Malicious URL Detection using Machine Learning: A Survey. arXiv preprint arXiv:1701.07179.

7. Abbasi, A., Zahedi, F. M., Zeng, D., Chen, Y., Chen, H., & Nunamaker Jr, J. F. (2015). Enhancing predictive analytics for anti-phishing by exploiting website genre information. Journal of Management Information Systems, 31(4), 109-157.

8. Blum, A., Wardman, B., Solorio, T., & Warner, G. (2010). Lexical feature based phishing URL detection using online learning. Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security.

9. Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv preprint arXiv:1802.03162.

10. Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Beyond blacklists: learning to detect malicious web sites from suspicious URLs. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.

11. Saxe, J., & Berlin, K. (2017). eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. arXiv preprint arXiv:1702.08568.

12. Sinha, S., Bailey, M., & Jahanian, F. (2008). Shades of grey: On the effectiveness of reputation-based "blacklists". 2008 3rd International Conference on Malicious and Unwanted Software (MALWARE).

13. Thomas, K., Grier, C., Ma, J., Paxson, V., & Song, D. (2011). Design and evaluation of a real-time URL spam filtering service. 2011 IEEE Symposium on Security and Privacy.

14. Verma, R., & Dyer, K. (2015). On the character of phishing URLs: Accurate and robust statistical learning classifiers. Proceedings of the 5th ACM Conference on Data and Application Security and Privacy.

15. Whittaker, C., Ryner, B., & Nazif, M. (2010). Large-scale automatic classification of phishing pages. NDSS, 10(2010).

16. Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). Cantina: a content-based approach to detecting phishing web sites. Proceedings of the 16th international conference on World Wide Web.

17. Garera, S., Provos, N., Chew, M., & Rubin, A. D. (2007). A framework for detection and measurement of phishing attacks. Proceedings of the 2007 ACM workshop on Recurring malcode.

18. Chu, W., Zhu, B. B., Xue, F., Guan, X., & Cai, Z. (2013). Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing URLs. 2013 IEEE International Conference on Communications (ICC).

19. Zhao, P., & Hoi, S. C. (2013). Cost-sensitive online active learning with application to malicious URL detection. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.

20. Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: a literature survey. IEEE Communications Surveys & Tutorials, 15(4), 2091-2121.

21. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

22. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

23. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30.

24. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Advances in Neural Information Processing Systems, 31.

25. Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259.

Citations:

[1]https://www.semanticscholar.org/paper/Using-Lexical-Features-for-Malicious-URL-Detection-Joshi-Lloyd/5a077ba5a02afdd398ee72eb693a9acf483dc5da

[2]https://www.semanticscholar.org/paper/Detecting-Malicious-URLs-Using-Lexical-Analysis-Mamun-Rathore/01bb00b24fb2bcf1d11748d0c39ba60367b4c264

[3] https://pmc.ncbi.nlm.nih.gov/articles/PMC11142511/

[4] https://pubmed.ncbi.nlm.nih.gov/35591061/

[5] https://arxiv.org/ftp/arxiv/papers/1910/1910.06277.pdf

[6] https://pmc.ncbi.nlm.nih.gov/articles/PMC9436524/

[7] https://pubmed.ncbi.nlm.nih.gov/36059391/

[8] https://pmc.ncbi.nlm.nih.gov/articles/PMC10537824/

[9] https://www.mdpi.com/1424-8220/22/9/3373

[10] https://arxiv.org/abs/1910.06277