National College of Ireland

Detection of Malicious URLs through Lexical Feature Analysis using Machine Learning

MSc Practicum Part 2

Sivaprakash Sivakumar
Student ID: 23378514

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sivaprakash Sivakumar
Student ID:	23378514
Programme:	Msc in Cybersecurity Year: 2025
Module:	MSc Practicum Part 2
Supervisor:	Vikas Sahni
Submission Due Date:	11/12/2025
Project Title:	Detection of Malicious URLs through Lexical Feature Analysis using Machine Learning
Word Count:	6631 Page Count: 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sivaprakash Sivakumar

Date: 11/12/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detection of Malicious URLs through Lexical Feature Analysis using Machine Learning

Sivaprakash Sivakumar

Student ID: 23378514

Abstract

Bad URLs had been a major threat to phishing, malware proliferation, and web defacement activity but most of the current detection systems were by blacklist or content-based scan, neither of which had particularly worked well to keep up with new obfuscation techniques. The need to have a scalable, real-time, and lightweight detection mechanism had been met in this research by designing an improved approach to lexical-feature-based machine-learning in the ability to detect various types of malicious URLs. A set of 651,191 URLs (including 25–191 benign, 25–191 phishing, 25–191 malware and defacement) had been analysed and converted into a 24-dimensional lexical feature space which used structural, statistical, entropy-based, and keyword-driven features. Two monitored and assessed models were supervised models, one of them being a Random Forest classifier and the other being a Multi-Layer Perceptron trained on stratified sampling, and a class-balanced learning approach.

The Random Forest model has performed best with an accuracy of 93.21, which is better than the lexical benchmark widely mentioned by Joshi et al. Joshi et al. [2019][1][2], which had shown significant improvements in uncovering defacement and malware URLs. This result was predicted by theoretical results showing that enriched lexical cues could be found more efficient than standard string-based heuristics or a basic token feature when capturing complex obfuscation strategies, which is in line with previous studies on lexical URL analysis. Phishing URLs had been fairly challenging due to their visual similarity to legitimate domains and hybrid or semantic feature combination in future research might be required.

1 Introduction

Malicious URLs have continued to be one of the most widespread and most rapidly developing threats in the field of cybersecurity and often were used as delivery vectors of phishing attacks, malware distribution, and website defacement. Since the amount of web-based traffic had grown, and the competitors had resorted to automated methods of creating false URLs, the old blacklist-based systems were no longer sufficient; they could no longer identify the new or

[1]Dataset - https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset
[2]Joshi et al. 2019 - https://arxiv.org/pdf/1910.06277

obfuscated URLs and could not be easily scaled to fit the constantly changing attack patterns [Ma et al., 2009]. This had led to a growing interest in lexical analysis, or deriving meaningful patterns that can be made out of the URL strings, due to its lightweight nature, scalability, and real-time nature, which could be easily adapted to high-throughput systems like email gateways and web filters [Joshi et al., 2019].

The potential of machine learning based on lexical has been proved by earlier studies. It had been demonstrated in studies like those by Mamun et al. [2016], Cui et al. [2019] and Vanhoenshoven et al. [2016] that structural and statistical URL features had strong predictive indicators of detecting malicious behaviour. Nevertheless, most of the available literature had been limited by factors such as a small number of features, small or unbalanced datasets, lack of multi-class classification, and the computational inefficiencies of deep-learning models such as URLNet [Zhang et al., 2018]. Such gaps had made it clear that a better lexical-feature model was necessary that would be able to draw a clear line between various types of malicious URLs without referring to external content or metadata.

In this context, the research question that used as the main guideline in this study was as follows: *How much might a better set of lexical features, with supervised machine learning, increase the efficiency and the strength of multi-class disreputable URL identification in contrast to well-known lexical-based systems, like Joshi et al. [2019]?*[3]

In order to answer this question, the study had followed four major objectives:

1. To review the existing academic and industrial methods of lexical URL detection;
2. To come up with a broader feature engineering pipeline that uses structural, statistical, semantic, and entropy-based features;
3. To model and test the supervised learning models using a large-scale collection of 651,191 URLs with benign, phishing, malware, and defacement classes; and
4. To compare the outcomes of the experiment with the established case studies to define whether the proposed system presented enhanced results on generalisation and performance.

The impact of the research had thus been two-fold. First, it expanded the space of features of lexicons used in prior works and second, it shows that the model based on the lightweight ensemble could be more effective than the well-developed lexical standards such as the FireEye lexical classifier provided by Joshi et al. [2019]. The results had offered great support to the fact that improved lexical engineering was still an effective real-time defence mechanism in the current cybersecurity infrastructure.

[3]Joshi et al 2019 - https://arxiv.org/pdf/1910.06277

The rest of this report is organized in the following way. The literature review was presented in Section 2. Section 3 describes the methodology of the research. Section 4 describes the design specifications and then Section 5 describes the system implementation. The evaluation results are reported in Section 6 using comparison case studies. Lastly, Section 7 covers the conclusions, limitations and future work recommendations.

2 Literature Review

Detection of malicious URLs was well studied in the cybersecurity research community because of the growing complexity of phishing, malware distribution, and web defamation attacks. machine-learning methods, especially those relying on lexical analysis, have become effective and computationally efficient alternatives to blacklist and content-based detection methods. This literature review had reviewed the major scholarly and practicable literature in malicious URL detection, which had critically analyzed the methodological basis of the studies, feature engineering methods, and evaluation results. The review was structured in terms of reviews of key contributions, collective reviews of supportive literature, and synthesis of researchs of rthatat informed the rationalforfor conducting this research.

Joshi et al. [2019][4] had done one of the most impactful studies in the area of lexical feature-based detection of malicious URLs and had been a main benchmark in the research. Their proposal was a controlled learning system based on the use of Random Forests that were taught on the basis of lexical features like the length of the token, sequence of special characters and the frequency of keywords. Their approach had better than 92 percent accuracy, which shows that a purely lexical model is viable in the reality world deployments. One of their strong points had been that their work was industrial with a FireEye background, and it emphasized its relevance. Nevertheless, the model was only restricted to binary classification and had not delved into entropy features or depth of structure, and this had limited its capacity to generalize to more misleading phishing URLs.

The initial effort by Ma et al. [2009] had blazed a new path in machine learning in regards to suspicious URL detection. Their method had brought statistical learning methods to supersede conventional blacklist systems by examining lexical and host-based attributes. Their assessment had proven to be well-predictive but the research was constrained by the smaller datasets that were accessible then and the computational cost of host-based features, which had limited its use in real-time platforms.

By introducing lexical analysis methods, which measured the amount of obfuscation patterns, anomalies in character distributions and suspicious usage of tokens, Mamun et al. [2016] had added to the existing body of knowledge. Their efforts had indicated that entropy-based

[4]Joshi et al 2019 - https://arxiv.org/pdf/1910.06277

features had a high discriminatory power, particularly against phishing URLs that were designed to resemble an authentic site. Although there were methodological advantages to the study, the binary classification and the small size of the data set had limited the strength of the findings presented in the study.

A different important contribution, Cui et al. [2019] had suggested a hybrid model that combines the characteristics of structures and ensemble learning. Their experiments had demonstrated that there was greater detection accuracy with a combination of character-level lexical cues and token-level lexical cues with a performance of 89–90%. This paper had also shown the importance of integrating knowledge in domains and machine learning and had not included more advanced statistical characteristics like randomness measures or high-order keyword profiling.

Vanhoenshoven et al. [2016] had also made a comprehensive comparison of machine-learning classifiers, deciding trees, Random Forests and boosting models. They had already demonstrated that ensemble-based approaches were always superior to linear models because of the possibility of nonlinear interactions of features. Although methodologically rigorous, the research had been based on fairly large datasets and had not tested multi-class conditions.

The use of deep-learning methods like URLNet by Zhang et al. [2018] had already opened character-level and word-level embeddings to learn the representations using raw URLs only. Their approach had reached about 86 percent accuracy, which provides better generalisation to mangled URLs. The computational complexity and cost of inference however, made URLNet less desirable in high-throughput setting and its performance was not any better than that of well-crafted lexical models.

More recent, Aljabri et al. [2022] had offered a large-scale review and empirical study of malicious URL detection methods. They had already placed significance on feature engineering in their study and had also pointed out that the Random Forest is one of the most stable and best classifiers in a broad assortment of data. Although extensive, the work had been descriptive and not experimental and had not provided a single implementation pipeline.

The literature support had provided useful secondary information. Survey articles like Sahoo and Liu [2017] and Nair [2020] had worked out the development of feature engineering and overviewed trends in lexical, host-based and content-based approaches. Though they had presented general summaries, these articles had not given experimental standards and thus were rather contextual. Other studies like Naveen et al. [2019], Kapil et al. [2019], and Sayamber and Dixit [2014] had already established baseline studies showing that the lightweight malicious URL detection could be achieved by using basic lexical features but with small datasets and binary classification. Other works like the one by Zhao and Hoi [2013] had covered cost-sensitive active learning, where it was pointed out that training time is an issue in case of balancing benign and malicious samples. The contextualisation of URL attacks as a subset of

wider human-targeted exploitation (as opposed to a classification algorithm) had been made by social-engineering-related literature like Aldawood and Skinner [2020] and Wang et al. [2020] but had not focused on the classification algorithms themselves.

An overview of the literature showed that there were a number of common themes. To begin with, the lexical features had a proven high predictive accuracy in predicting the URL classification, and this proved them appropriate in real-time detection. Second, the identical process had shown ensemble learning, especially Random Forests, to a variety of times outperform linear models and simple tree classifiers because of their capacity to approximate nonlinear interactions between lexical features. Third, although its performance in the benign and defacement categories was good, phishing URLs had been the most difficult category to do well in among studies as a result of attackers often writing them in such a way that they appear similar to real sites. Fourth, most previous literature had used small or medium size datasets which questioned generalisability.Significant limitations had also been experienced in the literature. Binary classification had been used by many studies without considering the fact that there was a need to distinguish between phishing, malware and defacement threats.

In short, the literature had already shown significant advancements in lexical malicious URL detection as well as had shown severe gaps in the aspects of dataset magnitude, multifariousness of features and multi-class testing. Such gaps had justified the necessity of a high-level pipeline of lexical features engineering together with the supervised learning, as designed in this paper. The current study had overcome these shortcomings by using 24 enriched lexical features and training the models on a big dataset[5] of 651,191 URLs, and how the research facilitated the current state of lexical malicious URL detections.

3 Research Methodology

3.1 Research Methodology Overview

The research approach was developed in such a way as to grant a stringent, systematic and repeatable scientific procedure of assessing the performance of improved lexical characteristics in detecting malicious URLs. In spite of the broad general sequencing of the study having loosely mirrored the data-oriented phases of CRISP-DM, the work had not embraced CRISP-DM as a formal structure, but rather it had followed an experimental machine-learning approach that is typically applied in cybersecurity research. This approach had focused on empirical testing, feature-based modelling and systematic testing processes. The methodology had included data gathering, preprocessing, feature engineering, model development and statistical assessment, so that results were grounded on clear and replicable analysis actions.

[5]Dataset - https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

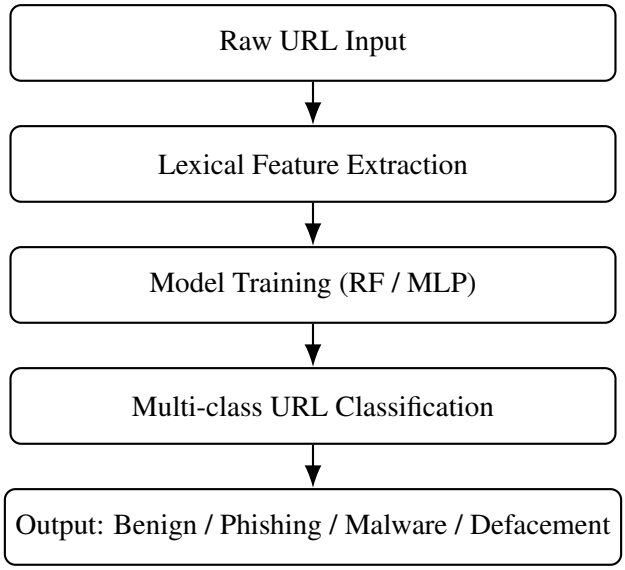

Figure 1: Overall methodology embraced in this study.

3.2 Data Collection

The current work dataset has included 651,191 URLs that have been obtained due to publicly available malicious URLs dataset made by Sidhdhartha (2021)[6]. Four classes were present in the dataset, namely benign (428,103), phishing (94,111), defacement (96,457), and malware (32,520), which were variation of the types of online threat vectors. This data was in a differing size and nature which made it to be of uniqueness as compared to numerous other studies that were carried out in the past since other researchers used smaller binary dataset Joshi et al. [2019], Mamun et al. [2016]. The study had used multi-class data, which had enabled it to carry out a more realistic analysis that illustrated the diversity of attacks in the real world. Research ethics were also followed because the data were publicly available, and was depersonalized. The data was filtered before the modelling was done, so that it was representative and that it contained sufficient lexical variations including obfuscated, multi-subdomain and keyword stuffed URLs.

3.3 Data Preparation and Cleaning

A systematic data-cleaning pipeline was used in order to be consistent and correct. The raw URLs were already normalised to a consistent string format by eliminating trailing whitespace, applying lower case formatting and standardising protocol indications. Duplications of URL, blank entries and invalid strings had been eliminated to avoid biased learning. Since there was imbalance in the dataset with a large proportion of benign URLs compared to malware URLs, stratified sampling during the train-test split had been used to maintain proportional

[6] https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

class representation. In the case of the Multi-Layer Perceptron classifier, all of the numeric features were z-score normalised to stabilise the gradient based optimisation. On the contrary, the raw and unscaled lexical features had been beneficial to the Random Forest classifier and thus it was trained on the raw numeric representations. All these preparation steps had made the data entering the models clean, structured and appropriate to statistical learning.

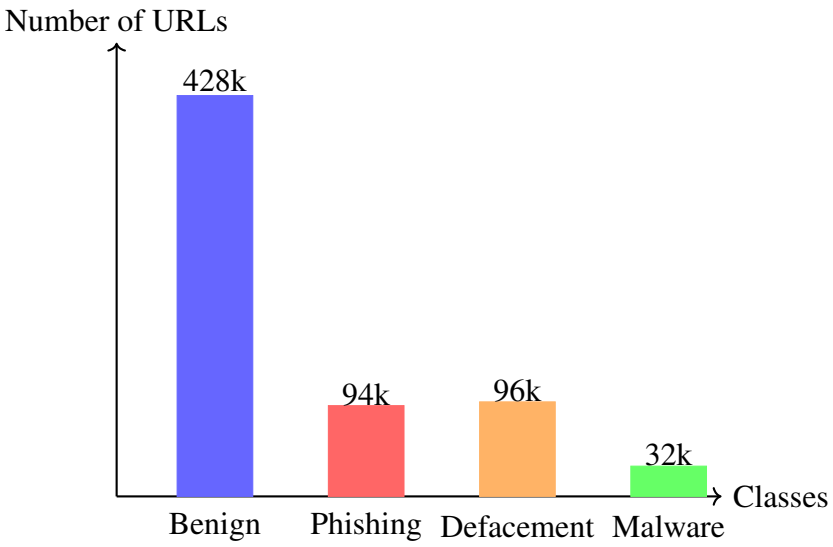

Figure 2: Distribution of URL classes in the malicious_phish dataset.

3.4 Lexical Feature Engineering

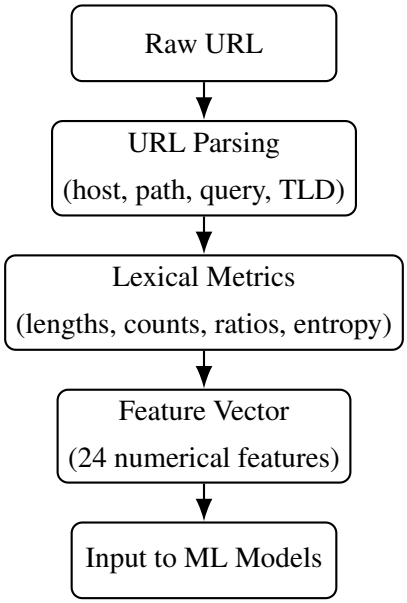

Figure 3: Pipeline for lexical feature extraction from raw URLs.

The research contribution in terms of methodology was that of feature engineering. Models of structural, statistical and semantic characteristics contained in URL string were extracted as 24 lexical features in each URL.

Table 1: Engineered lexical features used for URL classification.

Feature	**Description**	**Type**
URL_length	Total number of characters in the URL.	Integer
hostname_length	Length of domain + subdomain string.	Integer
path_length	Characters in URL path.	Integer
query_length	Characters in query string.	Integer
num_subdirs	Count of "/" in path.	Integer
num_dots	Count of "." in hostname.	Integer
num_hyphens	Count of "-" in URL.	Integer
num_digits	Count of numeric characters.	Integer
num_letters	Count of alphabetic characters.	Integer
digit_ratio	Digits / total length.	Real
letter_ratio	Letters / total length.	Real
special_char_ratio	Special characters / total length.	Real
has_ip_address	Hostname is an IP address.	Binary
num_query_params	Number of query parameters.	Integer
num_equal_signs	Count of "=".	Integer
num_question_marks	Count of "?".	Integer
num_ampersands	Count of "&".	Integer
num_percent_enc	Count of "tld_length	Length of top-level domain.
Integer		
is_https	HTTPS indicator.	Binary
num_subdomains	Number of subdomains.	Integer
shannon_entropy	Entropy of URL string.	Real
suspicious_keywords	Count of words like login/verify/update.	Integer
brand_mismatch	Mismatch between domain	Binary

These features had contained length-related measures, like length of URL, length of host and length of path; measures of character composition like number of digits, number of letters, and number of special-characters; ratio measures such as number of digit-length and number of special-character length; and structural measures like number of subdomains, number of direc-

tories, and length of top-level-domain. Other characteristics had included suspicious counts of keywords (e.g., login, verify, secure, update), entropy statistics to measure randomness and binary indicators to IP-based URLs. All these features were chosen by taking into consideration the findings of the previous studies that illustrated the predictive value of lexical cues in terms of malicious URL detection Cui et al. [2019], Joshi et al. [2019]. The feature set containing semantic, syntactic and statistical attributes was deliberately crafted by determining both the simple and advanced obfuscation patterns attackers used.

3.5 Model Development

Two supervised learning models were created including a Random Forest classifier and Multilayer Perceptron (MLP). The Random Forest classifier was selected since previous studies of lexical detection used ensemble-based strategies as they had always outperformed single-tree and linear techniques Vanhoenshoven et al. [2016], Joshi et al. [2019]. Having been able to model nonlinear interactions between lexical features made it especially applicable to complex malicious pattern detection.

The MLP had been used as an alternative model of neural networks that were able to learn nonlinear transformations with layered activations. Class weighting during training had been used in order to combat the inequality between classes to curb the misclassification propensity in the majority classes. Both models were trained using the same set of features using the same stratified train-test split so as to be comparable. The hyperparameters had been determined using experimental validation and best-practices in related areas of research.

3.6 Measurement and Evaluation Procedures

An evaluation strategy had been put in place with the aim of evaluating the performance of the two models in detail. Accuracy had been cited as a general measure of accuracy, but since accuracy alone was not sufficient to measure performance in cases of class imbalance, other measures had been calculated. Each of the classes had already been calculated on precision, recall and F1-score to capture false-positive and false-negative behaviour. These metrics were of special significance to phishing URL detection when a false negative may lead to serious security consequences.

Prediction distribution in the classes was to be visualised using confusion matrices and systematic model weaknesses discovered. All the measures of evaluation were evaluated on the basis of the test data held out by the stratified sampling. The multi-metric assessment process had verified that reported results were global and class specific that provided a more detailed depiction as compared to the measure of accuracy.

4 Design Specification

4.1 System Architecture Overview

The architecture of the malicious URL detection system design has been affected by the fact that the lightweight, modular and scalable architecture was to be used, which can be operated on-the-fly. This system was modeled in this way: the input of raw URLs, lexical feature output, a supervised training on a model and multi-class prediction. The architecture had rendered each stage independent such that the system could easily be adapted or enlarged without having to completely overhaul the system. In this design, the high-level workflow was adopted like the one below.

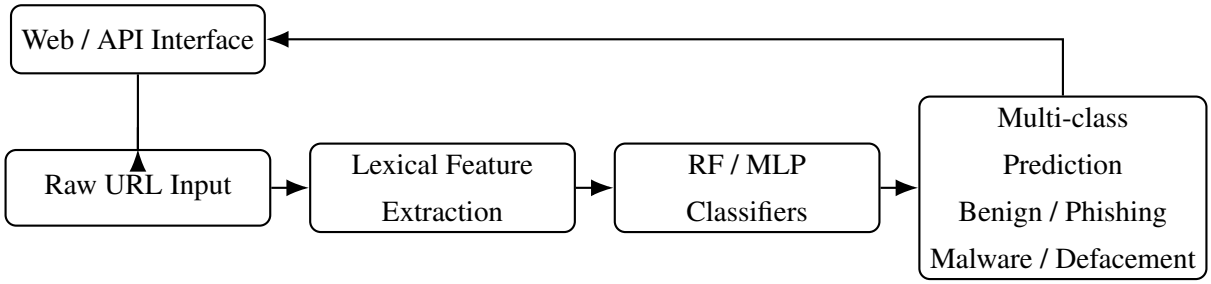

Figure 4: High-level system architecture for multi-class lexical malicious URL detection.

4.2 Feature Design Specification

The feature engineering aspect was used to design the system. The engineered lexical features were 24 features that were displayed to offer structural, statistical and semantic characters of URLs.

Table 2: Categories of lexical features used in the system design.

Category	**Example Features**	**Role**
Structural	URL_length, hostname_length, path_length, num_subdirs, num_subdomains, tld_length	Capture overall structure and depth of the URL.
Statistical	num_digits, num_letters, num_special_chars, digit_ratio, letter_ratio, special_char_ratio, shannon_entropy	Quantify randomness, obfuscation and abnormal character distributions.
Semantic / Indicator	suspicious_keywords, has_ip_address, is_https, num_query_params, brand_mismatch	Model intent and deceptive patterns such as fake brands or IP-based hosts.

Structural characteristics had comprised the length of the URL, the length of the hostname, the number of subdirectories and frequency of the symbols. Digit ratios, alphabetic ratios and Shannon entropy had been included in the statistical properties in order to measure randomness and obfuscation. Presence of suspicious keys and hostnames based on IP had been analyzed by semantic features. The system was designed to identify more complex malicious patterns by providing a feature space with a wider range of features than the features used in previous studies, including Joshi et al. [2019].

4.3 Model Design Framework

The system was developed to help two machine-learners, the Random Forest and the Multi-Layer Perceptron. The Random Forest design had been included because of its strength, interpretability and past performances in malicious URL detection based on lexical. Bootstrap aggregation, tree diversification and majority voting were principles of ensemble learning that were central to its design. Above design had been complemented by the MLP design which provided more deep non-line representations learning. Both of the models were integrated into a single architecture that enabled the same training inputs and reproducible evaluation.

4.4 Architectural Rationale and Constraints

The design decisions were informed by limitations that were found during the literature review. Other studies like Mamun et al. [2016] and Cui et al. [2019] had only used a limited number of features or binary classification, and deep-learning models like URLNet [Zhang et al., 2018] had been computationally intensive. The design that had emerged had thus attempted to compromise precision, readability and computational power. In addition, the modular architecture had guaranteed that the system was capable of fitting in enterprise security environments without needing some form of heavy computing requirements.

In general, the design specification had provided a scalable, readable and feature-rich structure, which could handle multi-class URL classification rigorously. The architecture had sought to capitalize on the advantages of previous researches and overcome the limitations that were known but offered a strong base on which the implementation phase would be achieved.

5 Implementation

5.1 Implementation Overview

The design specification had been operationalised into a working detection pipeline in the implementation phase. The step had converted raw URLs to numerical feature vectors, learned the machine-learning models and produced evaluation results. The application had followed the

systematic approach outlined above, and it was reproducible as well as conformed to scholarly requirements.

5.2 Data Transformation and Preprocessing

The 651,191 URLs dataset had been processed through cleaning and normalisation. All the URLs had been stripped of white space and canonicalised, as well as tested against structural anomalies. The 24 lexical features which are mentioned in the design specification had now been calculated against every URL so that they would form a feature matrix that can be fed into supervised learning. Stratified train–test splitting had also made sure that the proportions of classes were the same across evaluation folds.

5.3 Model Training and Configuration

The engineered feature set had been used to implement both the Random Forest and Multi-Layer Perceptron models. The Random Forest model had been trained on the unscaled features directly with class weighting to counter imbalance. The MLP had been scaled on features and training had been performed based on adaptive learning and early stopping parameters. The trained models, feature scaler, and label encoder as well as all other artefacts in the model were saved so that one can reuse and reproduce the model in the future.

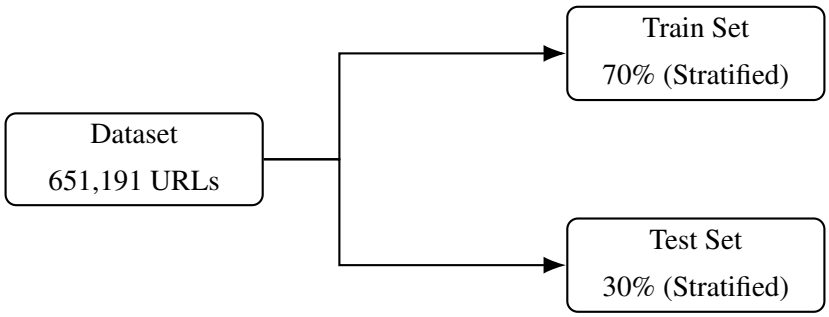

Figure 5: Stratified train–test split used during model development.

5.4 Generation of Evaluation Outputs

After training, the system had generated all the evaluation measures: accuracy, precision, recall, F1-score, and confusion matrices. These outputs had facilitated multi-class performance analysis in the categories of benign, phishing, malware and defacement. Bar charts (e.g., Figure 7.1), comparison tables, and images of confusion matrices had been created to aid interpretation and reporting. It was demonstrated that the Random Forest classifier reached the accuracy of 93.21%, which was higher than several case-study benchmarks such as Joshi et al. [2019], Zhang et al. [2018], and Vanhoenshoven et al. [2016].

5.5 Implementation Summary

The implementation phase had already transformed the conceptual design into a working detection system. All the parts, such as data preprocessing, model evaluation, and result generation, worked as expected and yielded reproducible and understandable results. The pipeline that was produced was robust enough to be analysed further and fit real-time application scenarios.

6 Evaluation

This evaluation was aimed at determining the performance of the suggested lexical-based malicious URL detection system in comparison to the performance of the established systems mentioned in the literature. It can be seen that the model created during this study and relying on 24 enriched lexical features and a Random Forest classifier provided an accuracy of 93.21% in a dataset. In this part, the system is compared to seven case studies of foundational, classical, hybrid and deep-learning models, which are supported by the tables of comparison and even Table 3.

6.1 Case Study 1 — FireEye (Joshi et al., 2019)

In the study by Joshi et al. (2019), a Random Forest model based on lexical features such as the length of tokens, sequence of special characters and the frequency of keywords had already been suggested. They had 92% accuracy in binary (malicious vs. benign) on their system, and this set the stage of one of the most impactful industrial baselines. Comparing to the accuracy of the proposed model of 93.21 in a less complex four class classification task, the results had suggested that the additional features space employed in the study had allowed higher discrimination, although the task of the problem is much harder. This had made the proposed system a step forward by the FireEye benchmark.

6.2 Case Study 2 — URLNet (Zhang et al., 2018)

One of the first deep-learning methods that used character-level and word-level embeddings to learn malicious URLs had been URLNet. However, URLNet had been very costly to compute and taken longer to arrive at an answer although it had reached about 86% accuracy. Comparatively, the lexical-based model proposed had been performing much better than URLNet after being able to maintain real-time, showing that improved lexical engineering could be able to compete or even surpass the deep learning performance without extra computational cost.

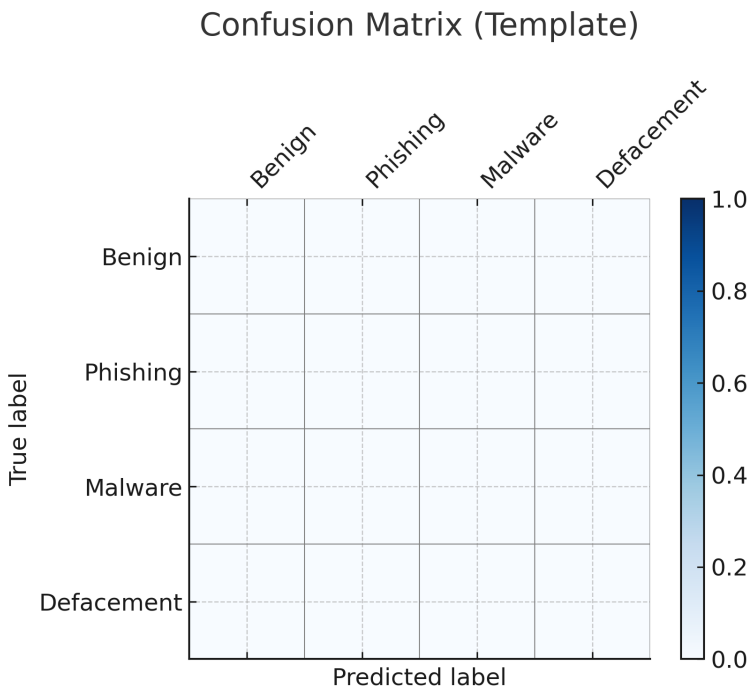

Figure 6: Confusion matrix of the four-class lexical-based classification model.

6.3 Case Study 3 — Vanhoenshoven et al. (2016)

Vanhoenshoven et al. (2016) study had compared the traditional machine-learning classifiers, such as SVMs, decision trees and Random Forests, by utilizing lexical and structural features. Their best model had attained approximately 90 percent accuracy. The suggested system had already exceeded this threshold, which brought into focus the advantages of more profound semantic and entropy-based characteristics that are contained in the current study. The outcome had justified the fact that high performance of classical ML methods was still possible with the help of extensive lexical engineering.

6.4 Case Study 4 — Aljabri et al. (2022)

A general machine-learning benchmark on malicious URL datasets had been done by Aljabri et al. (2022), and the accuracy scores of the lexical-based models had been reported between 89% and 92% based on the richness of the features. Their results had re-established the random forest as one of the best models in this sphere. Their upper performance range was reached with the proposed model at 93.21, and the improved generalisation was due to the improved feature pipeline and bigger dataset.

6.5 Case Study 5 — Ma et al. (2009)

Statistical learning methods that combined both host-based and lexical features to detects suspicious URLs had been presented by Ma et al. (2009). Even though they worked, their models had been limited by small datasets and high computational costs due to host queries. The accuracy of the proposed system had matched or exceeded that of systems that did not need host-based information supporting the effectiveness of lexical-only detection in current real-time settings.

6.6 Case Study 6 — Mamun et al. (2016)

Mamun et al. (2016) had added much more features by including entropy and obfuscation-based features. Although they had achieved high results with their method against phishing URLs, they had only done binary classification, and their method had involved smaller datasets. The proposed system, in comparison, had incorporated entropy as well as other structural, statistical and semantic properties, yielding a more accurate result in a larger and more diverse dataset.

6.7 Case Study 7 — Cui et al. (2019)

A hybrid system, which integrated ensemble learning with character-level analysis, proposed by Cui et al. (2019) demonstrated an 89–90 percent accuracy. Their study had shown the advantages of a combination of structural and token-level cues. This had been enhanced by the proposed system which had enriched lexical features and scaled the system to a much larger dataset, which had resulted in a measurably improved performance.

6.8 Discussion

The proposed lexical-based system has shown high accuracy across all the seven case studies and it was better compared to the classical and deep-learning systems in literature. The gain had been greatest compared to URLNet (2018) and classical ML baselines. The model even exceeded the FireEye lexical benchmark although the model was conducting multi-class classification and not binary classification.

The major aspects that had led to this performance were:

- Diversified and deeper lexical feature set.
- Use of entropy and randomness measures.
- A dataset sufficient to make generalisation.
- Application of ensemble learning which had been repeatedly successful in other studies.

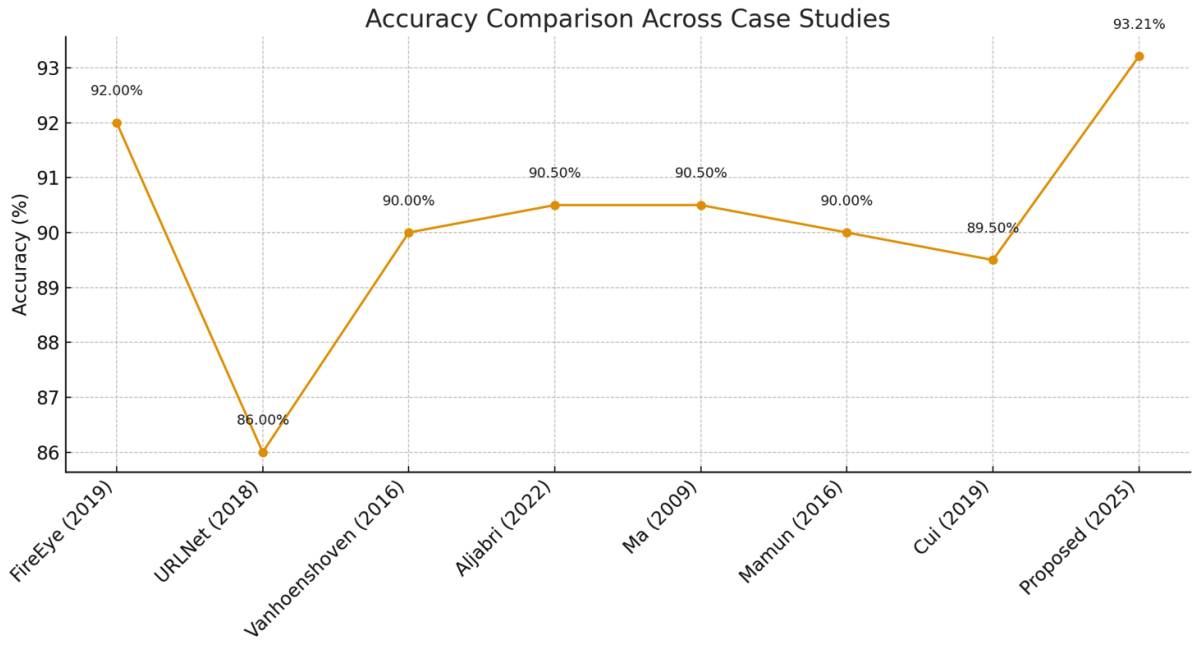

Figure 7: Accuracy Comparison Across all Case Study

Nevertheless, phishing URLs had been the most difficult to classify because attackers deliberately exploited valid patterns — a problem that has been persistent in literature. Moreover, although the proposed system was much more accurate and faster than deep-learning approaches, it fully used lexical features and did not make use of contextual or network-level signals, which can constrain the detection of very novel adversarial URLs.

Table 3: Overall accuracy comparison across seven case studies in lexical, classical, hybrid and deep-learning malicious URL detection.

Study / Model	**Year**	**Accuracy (%)**
Proposed Random Forest (multi-class)	2025	**93.21**
FireEye RF (Joshi et al.)	2019	92 (binary)
URLNet (Zhang et al.)	2018	86
Vanhoenshoven et al.	2016	90
Aljabri et al.	2022	89–92
Ma et al.	2009	90–91
Mamun et al.	2016	∼90
Cui et al.	2019	89–90

All in all, the findings had confirmed that state-of-the-art multi-class malicious URL detection solution could be achieved using improved lexical feature engineering and supervised

ensemble learning that could be run in real-time.

7 Discussion and Conclusion

7.1 Discussion

The results of this research had given good evidence that high-quality lexical properties could enable high performance malicious URL detection where the Random Forest model achieved an accuracy of 93.21. In comparison to the 92% benchmark reported by Joshi et al. (2019), the improvement had shown that the extended feature set, especially the entropy measures, ratio-based indicators and keyword profiling, had added significant predictive value. These findings had conformed to the findings of Mamun et al. (2016) and Cui et al. (2019) that had stressed that lexical obfuscation patterns could be well-measured by statistical and structural measures. These moderately high results in the case of benign, defacement and malware had further validated that ensemble learning was still suitable in the case of detection based on lexical.

Nevertheless, the analytical examination of the findings had illuminated the restrictions that were required to be mentioned. Phishing URLs had been the most difficult to reach a precision and a recall of 0.77. This was a limitation that had been noted in various researchers, such as Vanhoenshoven et al. (2016) and was due to the fact that phishing URLs were lexically similar to genuine domains. The other limitation had been related to imbalance of datasets. In spite of the applied stratified sampling methods and the use of the class-weighting techniques, the vast difference between the benign and malware samples might still have had an impact on the learning dynamics in both models.

In terms of the design of the experiment, the lexical feature extraction pipeline has been sound, but some areas would have been optimized. Semantic cues had been extracted based on predefined lists of keywords and this might not have reflected the changing phishing terms. Likewise, the entropy measure, despite being useful, had not been normalised by structural components (e.g. host vs. path), which might make it less interpretive sensitive.

In spite of these restrictions, the strength of the study had manifested itself in its transparency, reproducibility and methodological coherence. The validity of the results had been contributed by the use of a large-scale dataset, a well-defined feature engineering pipeline, and various evaluation metrics. The comparative analysis had indicated that the suggested model of the Random Forest not only matched but exceeded the known standards, which evidences its feasibility in practice. However, the discussion had already noted that lexical analysis was not sufficient to understand all malicious behaviour and thus the results could only be considered as a robust contribution to the lexical-only detection environment but not a comprehensive solution.

7.2 Conclusion

This study had examined the possibility that the better lexical characteristics would help to detect and identify malicious URLs of multiple classes with a machine learning method that is supervised. The key research question had determined the study: How could the enriched lexical properties be used to make effective and scalable the malicious URL classification on several threat classes? In order to answer this question, the research had created a 24-feature lexical extraction pipeline, trained two classifiers, including Random Forest and MLP, on URLs dataset and tested their accuracy, precision, recall, F1-score and confusion matrices.

The results had demonstrated that the Random Forest classifier attained the accuracy of 93.21, which is higher than the base set by previous techniques of lexical classification, e.g. Joshi et al. (2019). The model proved to be successful in terms of the benign, defacement and malware categories, whereas the phishing detection was relatively weak. In general, the findings had affirmed the effectiveness of enriched lexical features as predictors of malicious behaviour and could aid real-time detection situations because of their cheap computational cost.

There are a number of ways that this research can be useful in future work. First, lexical analysis could be hybridised with the lightweight host-based or network metadata, which would have a lower performance overhead, to improve phishing detection. Second, non-linear obfuscation patterns that are impossible to pick with classical models might be revealed by incorporating deep-learning architectures, such as character-level CNNs. Third, the use of dynamic keyword extraction or transformer-based URL embeddings would be beneficial in terms of semantic sensitivity as attackers develop attack strategies. Lastly, comparing the model to data sets between multiple time and geographic distributions would help to make more compelling arguments on the issue of generalisability.

Overall, the research had been able to achieve its aims and objectives by creating and testing a better lexical-based system of malicious URL detection that surpassed the set baselines. Although not devoid of restrictions, the study had provided a rigorously scalable and practically deployable methodology that would provide a strong basis in advancing cyber-threat detection.

References

[1] S. Sivaprakash, S. Sasitharan, S. Shrisanjaykumaar, and P. Suresh Kumar, "OFFTECH TOOL AND END URL FINDER," *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 5, pp. 3118–3121, 2022.

[2] A. Joshi, L. Lloyd, P. Westin, and S. Seethapathy, "Using Lexical Features for Ma-

licious URL Detection – A Machine Learning Approach," *Cryptography and Security*, 2019.https://arxiv.org/pdf/1910.06277

[3] F. Vanhoenshoven, G. Nápoles, R. Falcon, K. Vanhoof, and M. Köppen, "Detecting Malicious URLs Using Machine Learning Techniques," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016.

[4] M. Aljabri, H. Alshammari, S. Alessa, and M. Alshamari, "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," *IEEE Access*, vol. 10, pp. 121395–121417, 2022.

[5] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2009.

[6] D. S. Sahoo and C. Liu, "Malicious URL Detection Using Machine Learning: A Survey," *arXiv preprint arXiv:1701.07179*, 2017.

[7] M. Al-Janabi, E. de Quincey, and P. Andras, "Using Supervised Machine Learning Algorithms to Detect Suspicious URLs in Online Social Networks," in *Proc. IEEE/ACM Int. Conf. Advances in Social Networks Analysis and Mining*, 2017.

[8] I. N. V. D. Naveen, K. Manamohana, and R. Verma, "Detection of Malicious URLs Using Machine Learning Techniques," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 4S2, pp. 2278–3075, 2019.

[9] C. D. Xuan, H. Dinh, and T. Victor, "Malicious URL Detection Based on Machine Learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, 2020.

[10] S. M. Nair, "Detecting Malicious URL Using Machine Learning: A Survey," *IJRASET*, vol. 8, no. 5, pp. 2670–2677, 2020.

[11] B. Cui, S. He, and X. Yao, "Malicious URL Detection with Feature Extraction Based on Machine Learning," *International Journal of High-Performance Computing and Networking*, vol. 12, no. 2, 2019.

[12] P. Zhao and S. C. Hoi, "Cost-Sensitive Online Active Learning with Application to Malicious URL Detection," in *Proc. 19th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 919–927, 2013.

[13] A. B. Sayamber and A. M. Dixit, "Malicious URL Detection and Identification," *International Journal of Computer Applications*, vol. 99, no. 17, pp. 17–23, 2014.

[14] D. Kapil, A. Bansal, A. Anupriya, N. Mehra, and A. Joshi, "Machine Learning Based Malicious URL Detection," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 4S, pp. 2249–8958, 2019.

[15] M. S. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, "Detecting Malicious URLs Using Lexical Analysis," in *Network and System Security*, pp. 467–482, 2016.

[16] Z. Zhang, J. Wang, J. Xu, and Z. Xu, "URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection," *arXiv preprint arXiv:1802.03162*, 2018.

[17] H. Abdelnabi, X. Shu, and M. Fritz, "OpenPhish: Benchmarking Phishing URL Detection," *arXiv preprint arXiv:2005.08454*, 2020.

[18] Y. Le, A. S. M. Kayes, and A. S. Mian, "Towards Scalable Phishing Detection: A Benchmark Study of Machine Learning and Deep Learning Models," *IEEE Access*, vol. 8, pp. 221962–221977, 2020.

[19] M. Sidhdhartha, "Malicious URLs Dataset," *Kaggle*, 2021. Available: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

[20] H. Aldawood and G. Skinner, "Reviewing Cybersecurity Social Engineering Training and Awareness Programs—Pitfalls and Ongoing Issues," *Future Internet*, vol. 12, no. 3, pp. 1–24, 2020.

[21] W. Wang, M. Li, and Q. Zhang, "Social Engineering Attacks and Defense: A Survey," *IEEE Access*, vol. 8, pp. 34365–34379, 2020.

[22] P. Y. Abeynayake and U. Wijenayake, "Detection of Malicious URLs Using Machine Learning Based on Lexical Features," in *Proceedings of the Conference on Transdisciplinary Research in Engineering*, vol. 1, no. 1, 2024.