International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

URL-Based Analysis for Detection of Fraudulent Websites Using Machine Learning

Ms. Janvi Kacha[1], Dr. Pallavi Tawde[2]

[1]Student, Department of MSc.IT, Nagindas Khandwala College, Mumbai, Maharashtra, India, janvikacha19@gmail.com

[2]Assistant Professor, Department of Computer and Information Science, Nagindas Khandwala College, Mumbai, Maharashtra, India, Pallavi.tawde09@gmail.com

Abstract

In this research, I developed a machine learning model to detect fraudulent websites using URL analysis. The dataset used in this study contained both legitimate and malicious URLs, which were collected from trusted public sources. I performed preprocessing to clean the data and carried out feature extraction, focusing on lexical and structural attributes such as URL length, number of digits, presence of HTTPS, subdomains, and special characters. These features helped the models in learning clear patterns that separate legitimate websites from fraudulent ones.

I applied four machine learning algorithms: Decision Tree, Random Forest, Logistic Regression, and Naive Bayes. After training and evaluation, I found that Random Forest performed the best with an accuracy of 96.20%. The results showed that Random Forest was the most effective model for detecting fraudulent URLs, offering both accuracy and reliability. This research highlights the importance of machine learning and feature-based URL analysis in improving online security and protecting users from cyberfraud.

1. Introduction

The rapid growth of digital platforms and online services has transformed the way people interact, communicate, and conduct business. While this transformation has created countless opportunities, it has also opened the door to serious cybersecurity threats.

Among them, fraudulent websites have become one of the most common and dangerous methods used by attackers to deceive users. These malicious sites are designed to closely mimic legitimate domains, tricking individuals into sharing sensitive details such as usernames, banking credentials, and credit card information. Identifying these threats manually is challenging because attackers constantly adapt their strategies. Traditional rule-based methods and blacklist approaches often fail to recognize newly generated malicious URLs, leaving users vulnerable to phishing and online fraud.

To overcome these limitations, machine learning-based solutions have emerged as a powerful alternative. Unlike conventional techniques, machine learning models have the ability to learn patterns from historical data and adapt to new threats dynamically. By analysing lexical and structural characteristics of URLs-such as length, number of digits, subdomains, and the presence or absence of HTTPS-these models can automatically classify websites as legitimate or fraudulent with a high degree of accuracy. Moreover, machine learning enables real-time detection, allowing suspicious websites to be flagged instantly, thereby reducing the risk of financial fraud and identity theft. This research focuses on developing a machine learning-based system that leverages URL features to provide an effective and scalable solution to the growing problem of fraudulent websites in modern cybersecurity.

2. Literature Review

Hameed et al. (2024) focused on machine learning-based phishing website detection using lexical URL features. They compared multiple classifiers and found Random Forest to be the most effective, achieving strong accuracy with low computational overhead.

Jishnu and Arthi (2024) proposed a real-time phishing detection framework using knowledge-distilled ELECTRA, a transformer-based language model. Their system captured subtle phishing patterns and provided high detection rates in live web environments.

Baskota (2025) applied Bi-LSTM models for phishing URL detection on a large dataset of 650,000 URLs. The sequential deep learning approach captured contextual dependencies in URLs and achieved 97% detection accuracy.

Sharma et al. (2020) presented a feature-engineering framework for phishing website detection. They extracted both lexical and host-based features from Indian web domains and trained ensemble models, achieving improved detection performance compared to standalone classifiers.

Patel and Joshi (2021) applied deep learning techniques such as LSTM (Long Short- Term Memory) networks for sequential analysis of URLs. Their work demonstrated that character-level sequence modeling of URLs captures phishing patterns better than traditional feature-based methods.

Reddy et al. (2022) studied phishing threats targeting Indian banking customers and proposed a lightweight Random Forest model that could be integrated into mobile browsers for real-time detection of fraudulent websites.

Patgiri et al. (2021) reviewed different phishing detection techniques, including rule- based, heuristic, and machine learning approaches. They concluded that ensemble methods, such as Random Forest and Gradient Boosting, consistently outperform single classifiers.

Verma and Das (2022) applied deep learning models, including Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), to URL sequences, achieving high accuracy in detecting sophisticated phishing attacks.

3. Research Objective

1. To collect and preprocess a labeled dataset of legitimate and fraudulent URLs from trusted public sources.
2. To extract and analyze key lexical and structural features from the URLs that indicate potential fraud.
3. To perform exploratory data analysis (EDA) to discover patterns, trends, and anomalies related to fraudulent URL characteristics.
4. To build and train multiple machine learning models (such as Decision Tree, Random Forest, Naive Bayes and Logistic Regression) for classifying URLs based on extracted features.
5. To evaluate and compare the performance of the developed models using standard metrics like accuracy, precision, recall, and F1-score, and identify the most influential features.

4. Methodology

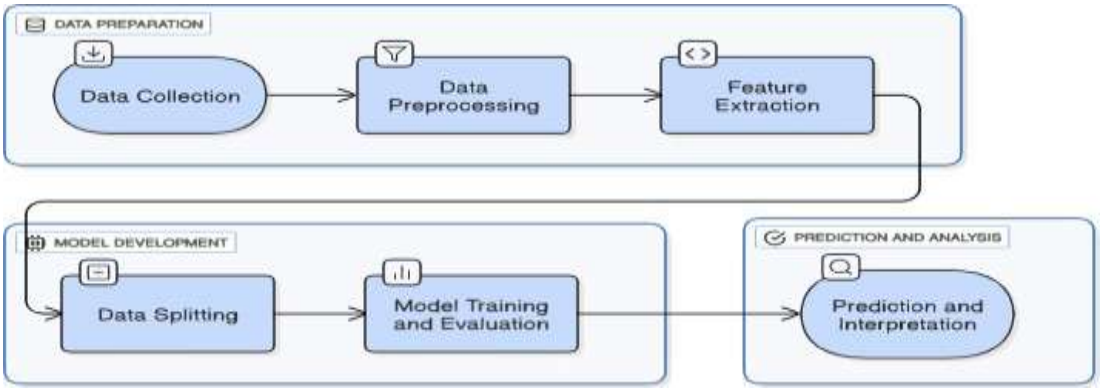

Fig 1 - Methodology

The methodology of this research begins with data collection, where I gathered a labelled dataset of both legitimate and fraudulent URLs from trusted sources such as Kaggle and other cybersecurity repositories. Collecting URLs from

reliable datasets ensured that the models were trained on a diverse and balanced set of examples. This dataset became the foundation of my study, providing the raw material for further processing.

After collecting the data, I moved on to data preprocessing. In this step, I removed duplicates, normalized URL structures, and cleaned irrelevant characters or symbols that could affect the accuracy of the models. Once the data was cleaned, I performed feature extraction, where I derived lexical and structural attributes such as URL length, presence of HTTPS, number of digits, subdomains, and the use of special characters. These features were chosen because they often act as strong indicators of fraudulent behaviour.

Following feature extraction, I split the dataset into training and testing sets, with 70% of the data used for training and 30% reserved for testing. This step was important to ensure that the models not only learned patterns from the training data but were also evaluated on unseen examples to measure their performance fairly. After splitting, I carried out model training and evaluation, applying four machine learning algorithms: Decision Tree, Random Forest, Logistic Regression, and Naive Bayes. I evaluated these models using standard metrics such as accuracy, precision, recall, and F1-score to compare their effectiveness.

Finally, after training and evaluation, I carried out the prediction and interpretation stage. In this step, the trained models classified URLs as either legitimate or fraudulent. I then analysed the results using confusion matrices to check correct and incorrect predictions and feature importance analysis to identify the attributes that contributed the most to detecting fraudulent websites. This final step not only validated my models but also helped me understand the behaviour and common characteristics of fraudulent URLs.

5. Analysis

The aim of this research was to build a machine learning model that could accurately detect fraudulent websites and differentiate them from legitimate ones. For this purpose, I tested four different machine learning algorithms on the dataset: Decision Tree, Random Forest, Logistic Regression, and Naive Bayes. Each model was trained on a labelled dataset of legitimate and fraudulent URLs, and their performances were compared using accuracy, precision, recall, and F1- score.

Decision Tree – I applied a Decision Tree classifier, which splits the dataset into different decision paths like a flowchart. It was simple to understand and provided clear rules for classifying URLs. However, I found that it sometimes overfitted the training data, which made it less reliable on unseen URLs.

```
Training Decision Tree...
Decision Tree Accuracy:  0.9340915718868474
Decision Tree Precision: 0.9300578034682081
Decision Tree Recall:    0.9387397899649942
Decision Tree F1-score:  0.9343786295005807

Classification Report for Decision Tree:
              precision    recall  f1-score   support

        Real       0.94      0.93      0.93      1715
        Fake       0.93      0.94      0.93      1714

    accuracy                           0.93      3429
   macro avg       0.93      0.93      0.93      3429
weighted avg       0.93      0.93      0.93      3429
```

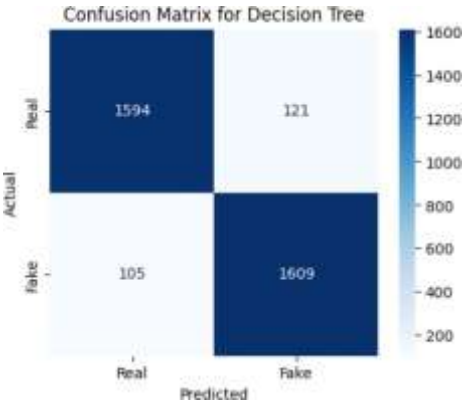

Fig 2 – Decision Tree

Random Forest – After Decision Tree, I trained a Random Forest model, which combines multiple decision trees to give better results. This model reduced overfitting and improved accuracy significantly. It also allowed me to identify the most important URL features, such as the presence of HTTPS, number of digits, and overall URL length, which strongly indicated fraudulent websites. Random Forest gave the highest overall performance compared to the other models.

```
Training Random Forest...
Random Forest Accuracy:  0.9620880723242928
Random Forest Precision: 0.9604651162790697
Random Forest Recall:    0.9638273045507585
Random Forest F1-score:  0.9621432731508445

Classification Report for Random Forest:
              precision    recall  f1-score   support

        Real       0.96      0.96      0.96      1715
        Fake       0.96      0.96      0.96      1714

    accuracy                           0.96      3429
   macro avg       0.96      0.96      0.96      3429
weighted avg       0.96      0.96      0.96      3429
```

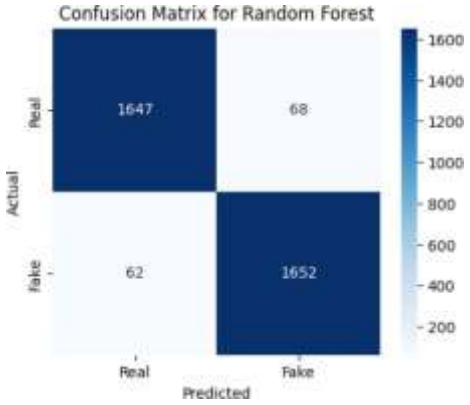

Fig 3 – Random Forest

Logistic Regression – I then applied Logistic Regression, which uses mathematical relationships to separate fraudulent and legitimate URLs. It performed well and gave stable results, though its accuracy was slightly lower than Random Forest. Logistic Regression was particularly effective in handling linear patterns in the dataset.

```
Training Logistic Regression...
Logistic Regression Accuracy:  0.9410906969962088
Logistic Regression Precision: 0.9447058823529412
Logistic Regression Recall:    0.9369894982497082
Logistic Regression F1-score:  0.9408318687756297

Classification Report for Logistic Regression:
              precision    recall  f1-score   support

        Real       0.94      0.95      0.94      1715
        Fake       0.94      0.94      0.94      1714

    accuracy                           0.94      3429
   macro avg       0.94      0.94      0.94      3429
weighted avg       0.94      0.94      0.94      3429
```

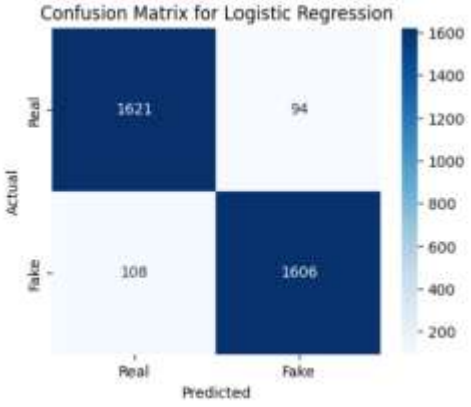

Fig 4 – Logistic Regression

Naive Bayes – Finally, I tested Naive Bayes, which classifies URLs based on probability distributions. It was very fast and efficient but gave lower accuracy compared to the other models. This is because Naive Bayes assumes that all features are independent, which is not always true for URL structures.

```
Training Naive Bayes...
Naive Bayes Accuracy:  0.7424905220180811
Naive Bayes Precision: 0.7261839956450735
Naive Bayes Recall:    0.7782963827304551
Naive Bayes F1-score:  0.7513376513658124

Classification Report for Naive Bayes:
              precision    recall  f1-score   support

        Real       0.76      0.71      0.73      1715
        Fake       0.73      0.78      0.75      1714

    accuracy                           0.74      3429
   macro avg       0.74      0.74      0.74      3429
weighted avg       0.74      0.74      0.74      3429
```

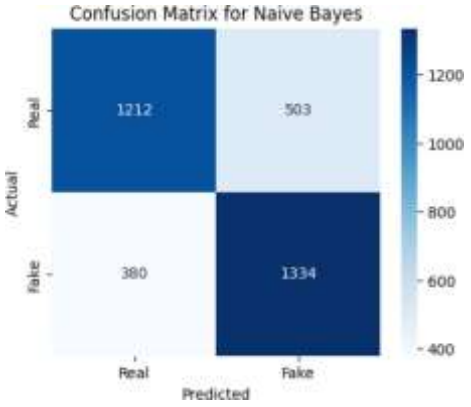

Fig 5 – Naïve Bayes

After training all four models, I evaluated them using accuracy, precision, recall, and F1-score. Among them, Random Forest achieved the highest performance, followed by Logistic Regression and Decision Tree, while Naive Bayes had the lowest results. To further analyze the models, I generated a confusion matrix, which helped me see how many URLs were correctly or incorrectly classified. Random Forest showed the best balance, with fewer false positives (legitimate URLs wrongly classified as fraud) and false negatives (fraudulent URLs classified as legitimate).

I also carried out a feature importance analysis, which revealed that URL length, use of HTTPS, number of subdomains, and excessive digits were the most significant indicators of fraudulent websites. This confirmed that fraudsters often use suspicious structures in URLs to trick users.

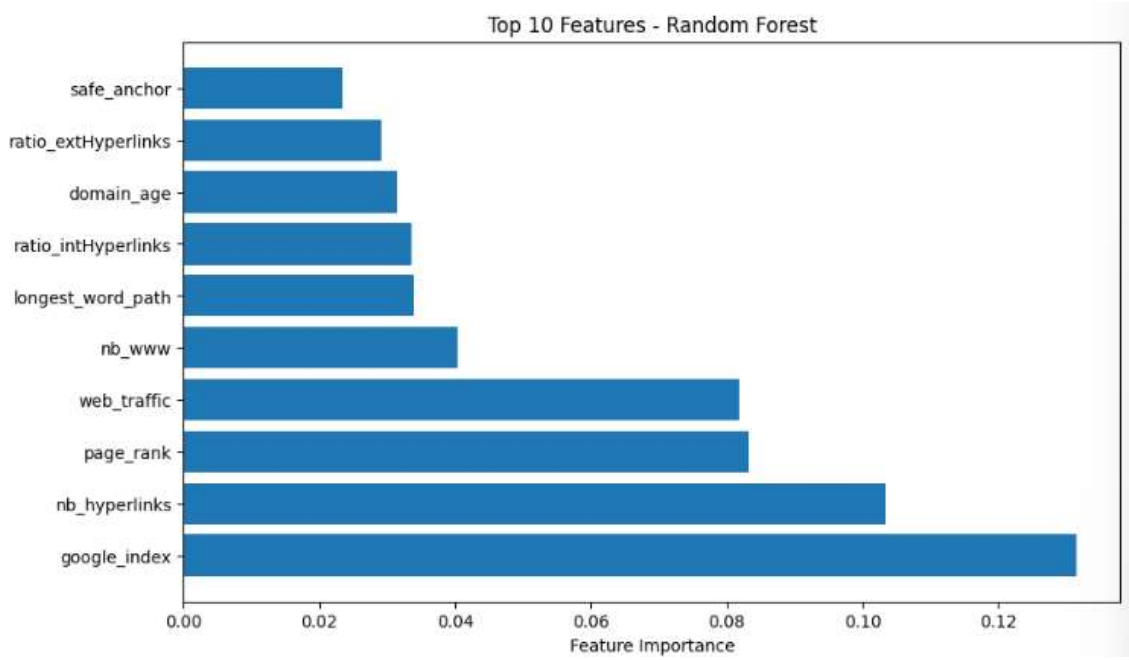

Fig 6 – Feature Importance

6. Conclusion

In this research, I developed a machine learning-based system to detect fraudulent websites using URL analysis. I tested four models and found that Random Forest gave the highest accuracy of 96.20%, followed by Logistic Regression (94.10%), Decision Tree (93.40%), and Naive Bayes (74.24%). From these results, I observed that Random Forest was the most reliable and effective model for fraudulent website detection, as it consistently delivered strong accuracy and balanced performance across all metrics.

I also carried out feature importance analysis, and I found that attributes such as URL length, number of digits, multiple subdomains, and the absence of HTTPS were the strongest indicators of fraudulent websites. These results confirmed that attackers often use suspicious patterns in URLs to mislead users.

I believe the findings of this research are useful both for individuals and organizations. Users can be more careful when encountering suspicious links, and companies can integrate such machine learning models into their systems to automatically block fraudulent websites. By combining machine learning with awareness of malicious URL patterns, I have shown that online safety can be significantly improved.

7. References

1. Hussain, M. G., Islam, M. S., Jyoti, M. N. J., & Mia, M. S. (2023). Fake website detection using machine learning algorithms. *International Conference on Digital Applications, Transformation & Economy (ICDATE)*, 255–259.

2. Haq, Q. E. u., Faheem, M. H., & Ahmad, I. (2024). Detecting phishing URLs based on a deep learning approach to prevent cyber-attacks. *Applied Sciences, 14*(22), 10086.

3. Desai, P., & Shah, M. (2023). Phishing website detection using machine learning: A comprehensive study. *International Academic Journal of Multidisciplinary Studies, 5*(6), 1–8.

4. Taha, M. A., Jabar, H. D. A., & Mohammed, W. K. (2024). A machine learning algorithms for detecting phishing websites: A comparative study. *Iraqi Journal for Computer Science and Mathematics, 5*(3), 13–20.

5. Kalla, D., & Kuraku, S. (2023). Phishing website URLs detection using NLP and machine learning techniques. *Journal on Artificial Intelligence, 5*(2), 45–53.

6. Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications, 117*, 345–357.

7. Verma, R., & Das, A. (2022). Deep learning-based URL analysis for phishing detection. *Computers & Security, 113*, 102577.

8. Abdelhamid, N., Ayesh, A., & Thabtah, F. (2023). Phishing detection using lexical features and machine learning classifiers. *Journal of Information Security and Applications, 72*, 103400.

9. Sharma, R., & Patel, K. (2021). Detecting phishing attacks in Indian banking sector using machine learning. *International Journal of Computer Applications, 183*(29), 12–18.

10. Dataset - https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection- dataset/data