International Journal of Recent Research and Review, Special Issues- 2025
ISSN 2277 – 8322

MALICIOUS URLs DETECTION

Mr. Pawan Sen[1], Anuradha Maurya[2], Vinay Kumar Tanwar[3]

[1]Head of Department Arya College of Engineering[2,3]Research Scholar

[1,2,3]Department of CS/IT

[1,2,3]Arya College of Engineering, Jaipur

[2]anumaurya2690@gmail.com, [3]tanwarvinay63 @gmail.com

Abstract- The proliferation of malicious URLs poses a significant cyber threat, enabling a wide range of attacks, including phishing, malware distribution, and identity theft. Traditional detection methods often rely on blacklists or manually engineered features, which struggle to keep pace with the dynamic nature of malicious URL generation. In this study, we explore a Natural Language Processing (NLP)-based approach to malicious URL detection, treating URLs as sequences of tokens and leveraging textual patterns to distinguish between benign and harmful links. We employ tokenization techniques suited to URL structures and apply machine learning models such as TF-IDF with logistic regression, as well as deep learning models like LSTM and transformers, to classify URLs. Our experiments, conducted on benchmark datasets, demonstrate that NLP-based models can effectively learn semantic and syntactic cues indicative of malicious intent, achieving high detection accuracy while maintaining low false positive rates. This research highlights the potential of NLP methodologies in enhancing automated cybersecurity systems and provides a scalable framework for real-time malicious URL detection.

Keywords- Malicious URL Detection, Natural Language Processing (NLP), URL Classification, Phishing Detection, Machine Learning, Deep Learning, Cybersecurity.

1. INTRODUCTION

The rapid expansion of the internet has led to a corresponding rise in cyber threats, with malicious URLs emerging as a primary vector for attacks such as phishing, malware dissemination, and data breaches. These URLs often mimic legitimate web addresses, making them difficult to detect through conventional means. Traditional approaches like blacklisting and heuristic-based systems are increasingly insufficient, as attackers continuously generate new, obfuscated links to bypass static defences.

In response to these challenges, the application of Natural Language Processing (NLP) techniques to URL analysis has gained momentum. By treating URLs as textual data, NLP allows systems to identify patterns, structures, and linguistic cues that may indicate malicious behavior. This approach enables the detection of previously unseen or zero-day threats by analyzing the semantic and syntactic features embedded within the URL strings.

This research aims to explore and evaluate NLP-based models for detecting malicious URLs, leveraging both traditional machine learning methods and deep learning architectures. Through comprehensive experimentation on publicly available datasets, we demonstrate how NLP techniques can enhance detection accuracy, reduce false positives, and offer a scalable solution for real-time cybersecurity applications.

2. TRADITIONAL DETECTION METHODS

Traditional approaches to malicious URL detection primarily rely on blacklisting and heuristic-based systems.While these methods can effectively block known threats, they often fail to identify newly generated or obfuscated malicious URLs.The dynamic nature of cyber threats necessitates more adaptive and intelligent detection mechanisms.

1. Machine Learning Approaches

Machine learning (ML) techniques have been extensively explored for malicious URL detection.These methods involve extracting features from URLs and training classifiers to distinguish between benign and malicious links.Commonly used classifiers include Support Vector Machines (SVM), Random Forests, and Naïve Bayes.However, the effectiveness of these models heavily depends on the quality of feature engineering and may not generalize well to novel threats.

2. Deep Learning and NLP-Based Methods

Recent advancements have seen the integration of Natural Language Processing (NLP) techniques and deep learning models for URL analysis.By treating URLs as sequences of characters or tokens, models can learn complex patterns indicative of malicious intent.For instance, transformer-based models like BERT and its variants have shown promise in capturing contextual information within URLs, enhancing detection accuracy.

3. URL ATTACK METHODS

Attackers use phishing URLs to attract users to open a fake website, where access to the user's computer is attempted in order to steal the user's private information, such as credit card numbers. Non-expert users can be easily fooled into clicking through to a phishing website by making barely noticeable misspellings in the URL, suchas changingwww.facebook.com towww.facebo0k.com, which makes user data more vulnerable.

Attacks occur when spammers create web pages in an attempt to fool the browser engine into perceiving they are legitimate whenthey are not. By illegally improving their rank, spammers want to deceive and attract more users to their spam websites. Spammers send spam emailsthat contain spam URLs to harm and infect the systems of their victims using spyware and adware.

Some attacks direct users to a malicious website that typically installs malware on the user's device that can be exploited for file corruption, keystroke logging, and even identity theft. Malware is a type of malicious software that can steal someone's personal information and damage a computer. One example of malware isthe drive-by download, defined as the unintentional download of malware caused by a user being tricked into visiting a malicious website. More examples include ransomware, keyloggers, trojanhorses, spyware, scareware, computer worms, and viruses.

Some attack redirects the user to a malicious website that has been altered by hackers in one or more aspects, such as itsvisual appearanceor someof thesite's contents. Hacktivists strive to take down a website for several reasons . This form of action occurs when the attackers discover the vulnerabilities of the website and utilize those vulnerabilities to compromise the website and modify the content on the web page without the owner's authorization, which is technically known as penetrating a website [11]. The classification of malicious URL attacks by ML techniques can be binary, such as either malicious or benign. Conversely, multi-classification is not restricted to any number of classes except that it has more than two, such as benign, phishing, suspicious, malware, spam, and others.

4. TECHNIQUES TO DETECT MALICIOUS URL

The detection of malicious URLs is a crucial task in cybersecurity, aiming to prevent phishing attacks, malware distribution, and other online threats. Over the years, various techniques have been developed, ranging from traditional rule-based systems to advanced machine learning and deep learning approaches. This section outlines the primary techniques used in detecting malicious URLs:

4.1 Blacklist-Based Detection

Blacklist-based techniques rely on maintaining a database of known malicious URLs. When a user accesses a URL, it is checked against this list to determine its legitimacy. While efficient for detecting previously identified threats, these methods fail against new or obfuscated URLs and suffer from poor scalability and high maintenance requirements.

4.2 Heuristic-Based Detection

Heuristic approaches apply manually crafted rules to detect suspicious patterns in URLs, such as excessive use of special characters, long domain names, or suspicious keywords (e.g., “login”, “verify”). These techniques can identify novel threats but are limited by the scope and generality of the defined rules, often resulting in high false positives.

4.3 Machine Learning-Based Detection

Machine learning models use features extracted from URLs—such as length, number of dots, presence of IP address, and lexical tokens—to train classifiers like Support Vector Machines (SVM), Decision Trees, and Random Forests. These models can generalize better than blacklists and heuristics but often require feature engineering and may not handle complex URL patterns effectively.

4.4 Deep Learning-Based Detection

Deep learning models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Bidirectional GRUs (BiGRU), eliminate the need for manual feature engineering by learning patterns directly from raw URL sequences. These models are more robust to zero-day threats and obfuscation techniques, particularly when combined with word or character embeddings.

4.5 Natural Language Processing (NLP)-Driven Detection

Treating URLs as a form of structured text, NLP techniques enable semantic and syntactic analysis to uncover hidden patterns indicative of malicious intent. Approaches like tokenization, TF-IDF, and word embeddings (Word2Vec, FastText) convert URLs into numerical formats suitable for machine learning. Advanced models like BERT, RoBERTa, and transformer-based classifiers further enhance detection by capturing context and relationships within URL components.

4.6 Hybrid Approaches

Hybrid systems combine multiple detection strategies e.g., blacklists, heuristic rules, and machine learning to maximize detection accuracy and reduce false positives. These approaches often use ensemble learning or multi-stage pipelines to leverage the strengths of each method.

5. DATA PREPROCESSING

Data preprocessing is a critical step in building effective models for malicious URL detection. Given the structured and sometimes obfuscated nature of URLs, preprocessing transforms raw input into a clean and consistent format suitable for feature extraction and modeling. The following are key preprocessing steps applied in this research:

5.1 URL Normalization

Normalization ensures that URLs follow a standard structure by:Converting all characters to lowercase, removing redundant or trailing slashes, replacing encoded characters (e.g., %20) with their standard representations, and eliminating session IDs or query parameters that do not contribute to classification.

5.2 Tokenization

Tokenization involves splitting URLs into meaningful substrings or tokens. This is typically done using delimiters such as /, ., -, =, and ?. For example, the URL:

https://login.bank.example.com/secure-login?session=abc123

might be tokenized into:

["https", "login", "bank", "example", "com", "secure", "login", "session", "abc123"]

Tokenization helps in capturing structural and semantic patterns often used by malicious actors.

5.3 Noise Removal

URLs often contain dynamic values or tracking parameters (e.g., UTM codes, random hashes) that may introduce noise. Removing or masking such elements prevents the model from overfitting to irrelevant patterns.

5.4 Label Encoding

URLs in the dataset are labeled as malicious or benign. These categorical labels are encoded into binary format (e.g., 1 for malicious, 0 for benign) to facilitate classification tasks.

5.5 Feature Representation

After preprocessing, URLs are transformed into a format compatible with machine learning or deep learning models. Common representation techniques include:

Bag of Words (BoW) or TF-IDF: For classical ML models

Character/Word Embeddings: For deep learning, where each token is mapped to a vector space

Sequence Padding: Ensuring uniform input length across samples for sequence-based models like LSTM or Transformer

6. FEATURE EXTRACTION USING NLP TECHNIQUES

Feature extraction plays a vital role in transforming preprocessed URLs into a numerical form that machine learning and deep learning models can interpret. By treating URLs as a special form of structured text, Natural Language Processing (NLP) techniques allow the extraction of both syntactic and semantic patterns that are critical for identifying malicious behaviors. This section outlines the key NLP-based methods employed for feature extraction.

6.1 Character-Level Embeddings

Character-level modeling treats each URL as a sequence of individual characters. This is particularly effective in detecting:

- Obfuscated or misspelled words (e.g., g00gle.com instead of google.com)
- Malicious use of symbols (e.g., %, @, //)

Each character is assigned an embedding vector, and sequences are input into neural networks such as CNNs or RNNs. This method is robust to morphological variations and captures low-level patterns that are often missed by word-level approaches.

6.2 Token-Level (Word-Level) Embeddings

URLs are split into tokens using delimiters like /, ., -, and ?. Each token can be mapped to an embedding vector using techniques like:

One-hot encoding (basic, sparse representation)

Word2Vec, GloVe, or FastText (dense, pretrained embeddings)

Custom-trained embeddings on URL corpora

Token-level embeddings help models understand word-like components in URLs, such as login, secure, or update, which often signify malicious intent.

6.3 Contextual Embeddings

Contextual embeddings are generated using transformer-based language models like BERT, RoBERTa, or DistilBERT, which consider the surrounding context of each token in the URL. This is crucial for distinguishing between benign and suspicious usage of common tokens. For instance:

login.example.com (likely safe)

example.com/login-verification (potentially malicious)

These models allow the detection system to understand not just the presence of a token, but how it is used within the entire URL structure.

6.4 Statistical NLP Features

In addition to embeddings, the following statistical features are often computed:

URL length: Malicious URLs tend to be unusually long.

Number of subdomains: Excessive subdomains may indicate phishing.

Frequency of suspicious keywords: e.g., “verify”, “account”, “secure”.

Character entropy: High entropy may indicate encoded or obfuscated content.

These handcrafted features are often concatenated with learned embeddings to enrich the model’s understanding of the URL.

6.5 Sequential Modeling and Attention

Once embeddings are obtained, sequential models such as LSTMs, GRUs, or BiLSTMs can process the sequences to capture order-dependent patterns. To enhance interpretability and performancemechanisms are applied to assign greater weight to critical tokens (e.g., login, paypal, update-password).

7. CONCLUSION

Through the application of Natural Language Processing techniques—specifically character-level and token-level embeddings, sequential deep learning models and attention mechanisms—this study demonstrates that URLs can be effectively treated as structured text sequences. This allows the model to

learn nuanced patterns, contextual relationships, and subtle indicators of malicious intent without heavy reliance on manual feature engineering or third-party blacklist databases.

In addition, the model's interpretability, offered by the attention mechanism, provides insights into which parts of a URL contribute most significantly to classification decisions, making it more transparent and trustworthy for cybersecurity analysts.

Future Work- While the current model is effective, several avenues for future enhancement exist: Real-time Deployment: Optimizing the model for real-time detection in large-scale systems. Adversarial Robustness: Integrating techniques to defend against adversarial attacks designed to bypass detection. Multimodal Inputs: Combining URL data with WHOIS, DNS, and page content analysis for a more holistic threat assessment. Cross-lingual Capability: Enhancing the model to detect threats in URLs that use multiple languages or non-Latin scripts.

REFERENCES

[1] J. Ma, L. K. Saul, S. Savage and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009.

[2] T. Le, H. Tran, T. Nguyen and N. Nguyen, "PhishTank URL detection using deep learning," in International Conference on Awareness Science and Technology (iCAST), Fukuoka, Japan, 2018.

[3] D. Sahoo, C. H. Liu and S. C. Hoi, "Malicious URL detection using machine learning: A survey," 2017.

[4] R. B. Basnet, A. H. Sung and Q. Liu, "Rule-based phishing attack detection," in International Conference on Security and Management (SAM), 2012.

[5] S. Marchal, K. Saari, N. Singh and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets," in IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016.

[6] R. Vinayakumar, K. P. Soman and P. Poornachandran, "Detecting malicious URLs using deep learning techniques," in International Conference on Advances in Computing, Communications and Informatics (ICACCI), Trivandrum, India, 2019.