WWW.IJRPR.COM

**International Journal of Research Publication and Reviews**

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

# Malicious URL Identification Using Deep Learning

***Sathya Keerthi S[1], Dr K Chitra[2]***

[1] Student Department of Computer Science Sri Krishna Aditya College of Arts and Science

[2] Department of Computer Science Dean of Research Sri Krishna Aditya College of Arts and Science

ABSTRACT :

Cyber-attacks are an inevitable threat in cybersecurity, making users into sharing their information under false pretenses regardless of its security (Public or Private). This project aims to develop a framework to differentiate between malicious links and legitimate ones. By analyzing URL structures, domain names, and hosting environments, researchers will identify specific patterns and features that indicate hacking attempts. Advanced deep learning algorithms will then classify links based on these patterns. This project uses a wide dataset of malicious and legitimate URLs to improve feature extraction techniques and classification models, aiming to enhance detection accuracy. Overall, the objective is to strengthen defenses against cyber threats, prevent successful identification of cyber-attacks, and protect user data. Ultimately, the findings aim to empower users and organizations with the knowledge and tools to identify and combat such attacks effectively and efficiently.

**Keywords**—Deep learning, Links, Algorithm, Neural Networks

## Introduction :

Take for instance, a multinational corporation operating in a high-stakes industry, such as finance or healthcare, where data security is paramount. One of the company's employees receives an email appearing to be from a trusted partner. The email contains a URL urging the recipient to verify sensitive account information. While the URL seems legitimate at a glance, it is a cleverly crafted phishing link designed to steal login credentials. Traditional security systems might fail to identify the URL as malicious due to its similarity to the authentic domain and its absence from existing blacklists.

However, the company's email security system is equipped with the proposed CNN-based malicious URL detection model. Upon receiving the email, the system automatically analyses the URL, extracting and evaluating its structural and semantic features. Unlike conventional methods, the CNN-based model identifies subtle irregularities, such as unexpected character patterns and abnormal subdomain structures, flagging the URL as potentially malicious.

The system then immediately quarantines the email, preventing it from reaching the employee's inbox. A notification is sent to the cybersecurity team, which investigates and confirms the phishing attempt. Meanwhile, the URL is added to a shared threat intelligence database, helping other organizations avoid falling victim to the same attack. This real-time application highlights the transformative potential of CNN-based models in proactively identifying and mitigating cybersecurity threats, safeguarding sensitive information, and enhancing trust in digital interactions.

This comprehensive and proactive approach underscores the necessity of adopting advanced deep learning techniques to tackle the ever-evolving challenges of cybersecurity in a digital-first world.

In the ever-expanding digital landscape, the rise of malicious URLs becomes a critical threat to cybersecurity, affecting individuals, organizations, and even governments. Cyber criminals constantly develop new methodologies to exploit vulnerabilities, making traditional detection mechanisms, such as blacklists and rule-based algorithms, increasingly low in usage for users. These conventional approaches often fail to keep up with the increasing volume and dynamic nature of emerging threats The increasing prevalence of malicious URLs has driven the development of diverse detection methodologies, each varying in efficacy. This comparative analysis explores three key approaches: blacklists, machine learning models, and a proposed deep learning methodology.

## Literature Survey :

**Studies of comparison(Existing Methodologies)**

In this comparative study, we are going to make 2 of the existing methodologies and a proposed methodologies where it leverages the robustness of secured space for users where the existing methodologies are based on:

Blacklisting (Stanford CS229)

Machine learning model (IJCSNS 2022)

So these methods were considered to be a bench marker among other models and take a quick explanation about these methods.

**Methodology 1 (Blacklists)**

The Blacklisting is one of the earliest methodology for URL detection as it contains the oldest way of detecting any kind of source from a group of data and it does not utilizes any of the algorithms of today's level of technology.

In simpler words all that a blacklisting approach needs a set of prior malicious URL set and the given URL from the user. And this approach can be done without any extra efforts.

**Methodology 2 (Machine Learning)**

In the detection process of URLs, Machine Learning often plays a vital role as there are distinct models available for identifying such links.

As explained about a more explained version of these works at [1] where we utilized it to make a comparative analysis among 4 Machine Learning algorithms in detail as follows.

## Methodologies Included (Existing System)

The existing system for malicious URL detection relies on a combination of traditional and machine learning-based approaches, each with its strengths and limitations. Consider a scenario where an employee receives an email containing a link to a malicious website known for hosting phishing campaigns. The URL, xyz.com, has been actively used in prior attacks and is already reported to cybersecurity databases. In this case, blacklisting proves highly effective. The system checks the URL against a database of known threats and immediately identifies it as malicious, preventing the user from accessing the site. While simple and efficient, blacklisting works well in such scenarios where the malicious URL is already documented but fails against newly generated or unused threats. Now, imagine a scenario where an attacker sends a phishing email with a URL that is cleverly obfuscated, such as secure-mysite.com, designed to resemble a legitimate domain. This URL is not present in any blacklist, but machine learning models trained on features like domain age, special character usage, and the similarity of subdomains to trusted websites can analyze it. The model identifies patterns suggesting malicious intent, such as an unusually recent domain registration or excessive subdomain nesting, classifying the URL as suspicious. Machine learning models excel in such scenarios by leveraging feature-based analysis, though they may struggle with complex patterns not covered in their training data.

***Blacklisting Approach***

One of the wisest and oldest approach to identifying malicious URL is blacklisting approach where a given URL gets compared with other URLs.

Also, these URLs are very complicative in nature to identify and it needs a lot of manual and paperwork to make such methodology work with the flow. And also, it is suitable for the scenarios that were risen at that time. Especially, this stood out among other model so much so that in the case of every research paper this methodology has been mentioned as it was one of the successful methodology papers such as [1], [2], [3] and [4].

This method revolves around the concept of maintaining a "blacklist," which is essentially a database of URLs and domains that have been identified as harmful. These blacklists are curated by cybersecurity organizations, internet service providers (ISPs), and specialized security software vendors. They serve as a repository of reported and verified malicious links, which users and systems can consult to avoid interacting with potentially harmful web resources.

Blacklists are typically updated regularly to keep up with the dynamic nature of cyber threats. The process of curating a blacklist involves a combination of automated threat detection, user-reported incidents, and intelligence gathered from global cybersecurity networks. Once a URL is flagged as malicious due to its association with phishing, malware distribution, or other cyberattacks it is added to the blacklist. This ensures that subsequent attempts to access the same URL are blocked by systems integrated with the blacklist.

**WORKING OF BLACKLISTS:**

1. ***URL Matching****:*
   Whenever a user attempts to access a website, the system compares the requested URL against entries in the blacklist. If a match is found, the URL is immediately flagged as malicious, and access is denied.
2. ***Threat Detection Sources****:*
   Blacklist databases are enriched using multiple sources. Automated systems continuously scan the web for phishing sites and malware-hosting domains. Cybersecurity experts verify and report suspicious URLs, while user feedback also plays a role in identifying potential threats.
3. ***Real-Time Blocking****:*
   Systems utilizing blacklists, such as web browsers, email security filters, and firewalls, are designed for real-time operation. When a flagged URL is detected, the system proactively blocks it, preventing users from accessing the malicious resource.
4. ***Regular Updates****:*
   To remain effective, blacklists must be updated frequently. Cybercriminals create new malicious URLs daily, and outdated blacklists can leave systems vulnerable to attack.
   Blacklists are integrated into various cybersecurity systems, including browser-based safe browsing tools, network firewalls, and email filtering solutions. These systems leverage the blacklist to protect users from malicious content during their online activities.

**DISADVANTAGES:**

Even though this blacklist method is convincible for URL detection, it has some notable demerits such as.

i. **Ineffectiveness Against New Threats**:
   Blacklists only contain known malicious URLs, making them ineffective against new or previously unidentified threats.

ii. **High Maintenance Overhead**:
Continuous updating of the blacklist database is resource-intensive, requiring regular monitoring and reporting of new threats.

iii. **Susceptibility to Evasion Techniques**:
Attackers can bypass blacklists using obfuscation methods such as URL shortening, typo squatting (e.g., gogle.com instead of google.com), or dynamically generated domains.

iv. **False Negatives**:
Malicious URLs not present in the blacklist will be treated as safe, creating a security gap.

v. **Limited Scope**:
Blacklists cannot adapt to the evolving tactics of cybercriminals, as they rely solely on static data without contextual analysis or pattern recognition.

***Machine Learning (IJCSNS 2022)***

This machine learning model works based on different algorithms as mentioned in [1] also it cross compared 4 distinct algorithms and makes a suitable existing methodology reference for this study.

***Data Collection***

This On identifying specific URLs, they were classified into 4 different categories based on their level of threats and their structure, they are:

**Malware:**

These URLs direct users to websites designed to distribute malicious software, such as viruses, worms, ransomware, or spyware. The primary goal is to compromise the user's system, steal data, or disrupt operations.

**Phishing**

These URLs direct users to websites designed to distribute malicious software, such as viruses, worms, ransomware, or spyware. The primary goal is to compromise the user's system, steal data, or disrupt operations.

**Benign**

These URLs direct users to websites designed to distribute malicious software, The primary goal is to compromise the user's system, steal data, or disrupt operations.

**Defacement**

These URLs direct users to websites designed to distribute malicious software, such as viruses, worms, ransomware, or spyware.
These data are then collected and processed in such a way that the URL gets divided into 7 distinct parts such as,
Scheme, Subdomain, Second-Level Domain, Top-Level Domain, Subdirectory, Page and Query.

| ***Dataset filed*** | ***Explanation*** |
|---|---|
| Domain_token_count | No. of token in a URL |
| executable | URL is pointing to executable file or not |
| NumberofDotsinURL | Count the number of dots (.) in URL |
| Arguments_LongestWordLength | Count the character in the largest word in URL |
| NumberRate_Domain | Occurrence of domain name in dataset |
| Entropy_Domain | Domain entropy |
| class | Classification of URL |

**Fig 1 URL dataset fields in IJCSNS 2022**

To make this model work in a flow they have utilized the Weka 3.8.6 for the experiment where for every distinct algorithm, the colllected, processed data form the dataset ISCX URL 2016 where the model certainly focuses on 3 categories such as:

Accuracy
Precision
Recall

The result of the following experiment gets displayed in the following table as follows,

| Classifier | Accuracy | Precision | Recall |
|---|---|---|---|
| Random Forest | 0.949 | 0.974 | 0.93 |
| Decision Tree | 0.915 | 0.961 | 0.92 |
| SVM | 0.927 | 0.935 | 0.956 |
| K-NN | 0.933 | 0.969 | 0.949 |

**Fig 2 URL detection analysis of IJCSNS 2022**

**NOTE:** This experiment also carried out its experiment without the preprocessing for a better comparison foe this experiment

**DRAWBACKS:**

I. The feature extraction mechanism is not comprehensive, which limits the performance of the classifiers.
II. The systems only utilize traditional machine learning algorithms.
III. The discussion on performance metrics is incomplete, particularly in terms of false positive and false negative rates.
IV. The machine learning models presented classify URLs as malicious or benign based on static criteria and lack dynamic adaptability to detect obfuscated or polymorphic URLs in real-time.
V. The trained models may struggle to generalize effectively to new datasets containing unseen patterns or URLs, limiting their robustness against evolving cyber threats.

Even though this model does have such drawbacks to be noticed this made a good framework and a distict benchmarker model for this doamin of research as whenever any other research work that will take place in near future will definitely contains this machine learning methodology for use.

***Proposed Methodology (Deep Learning)***

In order to make a new methodology to identify malicious URLs form a set of links as to whether they belong to the existing categories of URLs dataset, the method does stands out in several possible ways as to how it could potentially be the next level of detection mechanism where it have to become a standout methodology as the benchmarking detection model.
To make this happen the detection method follows a distinct and different approach on detecting URLs as it follows the algorithm in deep learning which is CNN algorithm where it follows a neural network on a convolutional level.

**CNN in detection:**
CNN-Convolutional Neural Network
A **Convolutional Neural Network (CNN)** is a type of deep learning algorithm specifically designed for processing structured data, such as images, sequences, or text. CNNs are particularly effective in tasks that involve spatial or hierarchical data, making them suitable for image recognition, natural language processing, and even malicious URL detection.
*Need for CNN model:*

- **Convolutional Layers**: These layers apply kernels to input data to extract features such as patterns, edges, or textures.
- **Pooling Layers**: These layers reduce the spatial dimensions of the data.
- **Fully Connected Layers**: These layers integrate the extracted features and make predictions by mapping the learned features to output classes.
- **Non-Linearity**: Activation functions, such as Rectified Linear Unit, introduce non-linearities to help the network learn complex patterns.

**PROCEDURES INVOLVED:**
In this model, there are certain steps that needed to be followed during the construction and execution of the mechanism such as,

**STEP 1** Break down each URL into smaller components or tokens, such as individual characters or substrings. For example, a URL like https://example.com/login might be tokenized into segments like https, example, com, and login.

**STEP 2** This step makes sure that the CNN can focus on analysing specific parts of the URL, such as subdomains or directory paths, which might indicate phishing attempts.

**STEP 3** Convert the tokens into numerical representations that the CNN can process.

**STEP 4** Methods like one-hot encoding or embedding layers can be used to transform tokens into vectors that capture their relationships and significance.

**STEP 5** This step translates text format data into a format suitable for computational processing while preserving semantic meaning.

**STEP 6** Treat the tokenized URL as a sequence of characters or words, representing it in a structured format.

**STEP 7** Apply convolutional filters to this sequence to detect features, such as recurring patterns or anomalous structures, indicative of malicious intent.

**STEP 8** These filters allow the CNN to automatically extract meaningful patterns without manual feature engineering.

**STEP 9** Train the CNN to recognize characteristics typical of phishing URLs, such as obfuscated strings (e.g., g00gle.com instead of google.com), suspicious subdomains, or excessive URL length.

**STEP 10** By analysing these features, the CNN can learn to identify subtle differences between legitimate and phishing URLs effectively.

**STEP 11** Use a labelled dataset comprising examples of both phishing and legitimate URLs.

**STEP 12** Each URL in the dataset is described with its classification, enabling the CNN to learn patterns that distinguish between malicious and benign URLs.

**STEP 13** Split the dataset into training, validation, and testing subsets.

**STEP 14** Use the validation set to fine-tune the model and prevent overfitting.

**STEP 15** Evaluate the trained CNN on the testing set to assess its generalization ability to unseen URLs.

**STEP 16** After successful training, integrate the CNN into a phishing detection system.

**STEP 17** The system analyses incoming URLs, extracting their features and classifying them as phishing or legitimate based on learned patterns.

**STEP 18** This real-time classification ensures proactive protection against phishing threats, enhancing security measures for users and systems.

These procedures form a comprehensive framework for leveraging CNNs in phishing URL detection, ensuring high accuracy, adaptability, and real-time applicability.

**ADVANTAGES:**

- Improved accuracy in detecting malicious URLs through advanced machine learning techniques.
- More reliable detection reduces the chances of incorrectly flagged URLs.
- Examining a broader range of URL attributes and metadata for a more thorough threat assessment.
- Automatically learns complex calculations from raw data.
- Enhanced detection accuracy and robustness

## A Cross Comparitive Study :

This study makes over a 3 kind of methodologies in which two of them were already in-use and also got old enough to be replaced. Also, the proposed method is one of the best methods which could potentially be replaced by the existing systems. As previously mentioned, the methods were explained in such a way that the proposed system's method includes the step-by-step procedural instruction as to how the whole thing works.
This study makes over a 3 kind of methodologies in which two of them were already in-use and also got old enough to be replaced. Also, the proposed method is one of the best methods which could potentially be replaced with the existing systems. As previously mentioned, the methods were explained in such a way that the proposed system's method includes the step-by-step procedural instruction as to how the whole thing works.

After some time span of using blacklisting approach it came out with certain drawbacks, similarly to cover up the drawbacks that were faced in blacklisting approach it does not made any significant difference except few changes that could potentially be useful in avoiding certain circumstances.

***Why machine learning stands tall?***

Even though there are several methodologies released to identify URLs such as **Heuristic Models** which utilized a set of predefined rules based on observed characteristics of malicious URLs, **Reputation-Based Systems** those assessed the reputation of a domain or IP address based on historical data, user feedback, and reported incidents, **Pattern Matching** which identified malicious URLs based on specific patterns or signatures observed in known threats, **Statistical Models** that leverages statistical properties of URLs, such as the frequency of specific characters, domain age, or URL length distribution and **Rule-Based Systems** which combined elements of heuristic and pattern-matching methods to create more comprehensive detection frameworks, these can only managed to overcome the drawbacks of what their previous models faced.

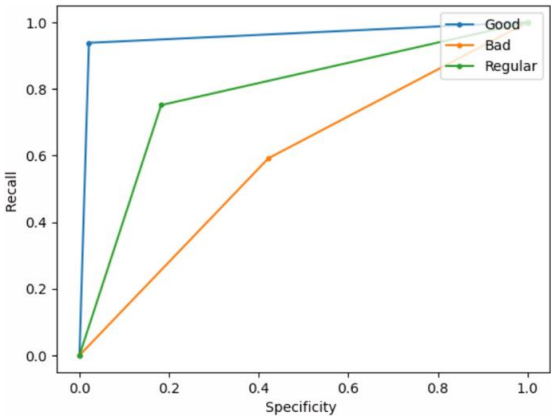


**Fig 3 ROC curve of result in [3]**

But in the case of machine learning model of [1], not only it follows the traditional way and fills out the gaps of drawbacks, but also it provides a newer way of approaching towards this issue and got away with it. By making further improvements and a whole different approach not only in a single algorithm but also in three different algorithms and some more of them yet to come in this field of machine learning detection of malicious links.

This makes a stand out methodology in this field and made a bench-marker software product on machine learning because of the sole reason on making a model which thinking out of the box so making something new out of the occurring issue that exists over time.

## Rsults And Conclusion :

The proposed system focuses on following the path of the machine learning method's ideology in which making a system that comprises and satisfies the following set of rules,
Rectifying the problem statements.
Reducing the drawbacks faced even better if fulfilled every drawbacks.
Making it outsmart every other existing system by utilising new methods that were not been available in prior methodologies.
On applying these metrics in the proposed system that consists of deep learning algorithm in which it identifies malicious links by following the above metrics in every possible ways. So lets look at the cross comparative analysis on the existing system and proposed methodology.

| METHODOLOGY | FEATURE EXTRACTION | ALGORITHMS USED | IMPROVEMENTS | LIMITATIONS |
|---|---|---|---|---|
| Comparison method | List of URL'S based on experience | Hashing, String matching, Binary search | First ever methodology, Easy to use | Out-Dated model, Needs to be attacked to be updated |
| Using Machine learning techniques (IJCSNS-2022) | Random Forest, YRL's features | SVM, DT (DECISION TREE), K-NN | Satisfied defects of previous model, Improved calculations, Better than prior models of ML | Feature extraction is non comprehensive, Used old ML algorithms for detection |
| Proposed System | Tokenization, Sequence padding, Vocabulary Indexing. | CNN (Convolutional Neural Network) | Improved accuracy. Reliable detection, Enhanced Robustness, | Solely relies on only one deep learning algorithm, Follows similar pattern of prior models |

**Fig 4 Cross Comparitive analysis on proposed and existing methodologies on finding malicious links**

These were the primary and needed drawbacks to be resolved alongside getting into satisfying the metrics those mentioned in the above conclusive section of this research so as a result this has all the potential tools needed to become the next bench-marking system on finding malicious links.

***Acknowledgment***

I would like to express my sincere gratitude to Sri Krishna Adithya College of Arts and Science and the Department of Computer Science for providing me with the opportunity to undertake and complete this research paper. Their unwavering support, guidance, and encouragement throughout this project have been invaluable.

REFERENCES :


1. Aljahdalic, A. O., Banafee, S., & Aljohani, T. (2024). URL filtering using machine learning algorithms. Information Security Journal A Global Perspective, 33(3), 193–203. https://doi.org/10.1080/19393555.2023.2193350
2. Chong, C., Liu, D., & Lee, W. (n.d.). Malicious URL Detection. Stanford.edu. Retrieved January 6, 2025, from https://cs229.stanford.edu/proj2012/ChongLiu-MaliciousURLDetection.pdf
3. Patgiri, R., Biswas, A., & Nayak, S. (2023). deepBF: Malicious URL detection using learned Bloom Filter and evolutionary deep learning. Computer Communications, 200, 30–41. https://doi.org/10.1016/j.comcom.2022.12.027
4. Reyes-Dorta, N., Caballero-Gil, P., & Rosa-Remedios, C. (2024). Detection of malicious URLs using machine learning. Wireless Networks, 30(9), 7543–7560. https://doi.org/10.1007/s11276-024-03700-w
5. Saqib. (2022). URL filtering by using machine learning. International Journal of Computer Science and Network Security, 22(8), 275–279. https://doi.org/10.22937/ijcsns.2022.22.8.34