Fake Link Detector

Abstract

Fake Link Detector

The "Fake Link Detector" project is a pivotal solution in the ongoing battle against deceptive URLs, bolstering online security and safeguarding user privacy. With its user-friendly frontend interface, it empowers users of all technical backgrounds to shield themselves from harmful links, contributing to a safer digital environment for everyone.

On the backend, the project employs a sophisticated machine learning model, extensively trained on a diverse dataset. This model, in conjunction with a JavaScript-based API, enables real-time analysis of URLs, swiftly identifying suspicious links and alerting users to potential dangers. The project's real-time threat detection is a testament to its effectiveness in countering deceptive URLs and enhancing online security.

As with any innovative endeavour, there is room for further development. Future work may include refining the machine learning model and expanding the dataset to stay ahead of evolving threats and the tactics employed by malicious actors online.

In conclusion, the "Fake Link Detector" project is a vital tool for safeguarding online security and user privacy. It offers a proactive approach to combating the risks associated with deceptive URLs, ensuring that users can explore the digital world with greater confidence and security.

Contents

Chapter 1 : Introduction

1.1 Overview

The "Fake Link Detector" is a mini project that aims to enhance internet security by detecting fake or malicious links. With the increasing prevalence of cyber threats, users are constantly exposed to harmful links that can lead to phishing attacks, malware infections, and other cybercrimes. This project leverages JavaScript for the front-end and an API in the back-end to provide a tool that helps users identify potentially harmful links and protect themselves from online threats

1.2 Motivation

The motivation behind the development of the Fake Link Detector project is to address the growing concern of online security. Cyber threats have become more sophisticated, and individuals and organizations alike are at risk of falling victim to phishing scams and malware distribution. The project is motivated by the following factors:

Internet Security:

The internet is an essential part of our daily lives, but it is also rife with security risks. The project seeks to contribute to a safer online environment.

User Education:

Many users are unaware of the dangers posed by fake links. This project aims to educate and empower users to make informed decisions when clicking on links.

Protecting Privacy:

Malicious links can compromise user data, privacy, and financial information. The project strives to protect users from such threats.

1.3 Problem Statement and Objectives

Link Evaluation:

- Develop a system that can analyze and evaluate the authenticity of a given link. This includes identifying features that are indicative of deceptive links, such as:
  - Misleading domain names
  - Unusual subdomains
  - Suspicious URL parameters
  - Redirects to known malicious websites

- The system should be able to accurately classify links as legitimate, suspicious, or dangerous, even if the link is new or has been recently created.

User-Friendly Interface:

- Create a user-friendly front-end using HTML and CSS, making it accessible and easy to use. The interface should allow users to easily input URLs and receive feedback on the safety of the links.
- The interface should be visually appealing and intuitive, with clear and concise instructions. It should also be responsive, so that it can be used on a variety of devices, including smartphones and tablets.

Real-time Feedback:

- Provide real-time feedback to users on the safety of links they input, indicating whether the link is safe, suspicious, or potentially dangerous. This will help users to make informed decisions about whether to click on links, thereby reducing their risk of exposure to online threats.
- The feedback mechanism should be fast and efficient, so that users do not have to wait long for a response. It should also be accurate and reliable, so that users can be confident in the information they are receiving.

Back-end Integration:

- Implement a back-end system with an API that performs the actual link analysis and delivers results to the front-end. This will allow the front-end to be developed independently of the back-end, making it easier to maintain and update.
- The back-end system should be scalable and secure, so that it can handle a large number of requests without compromising performance or security.
- Ensure that the system can handle a large number of requests efficiently without compromising performance. This is important because the system will be used by a large number of users, and the number of requests is likely to increase over time.
- The system should also be able to handle unexpected spikes in traffic without crashing or becoming unresponsive.

By addressing all of these objectives, the Fake Link Detector project can develop a robust and effective system for detecting and preventing deceptive links. This system can help to protect users from a variety of online threats, including identity theft, data breaches, and financial losses.

Chapter 2 : Literature Survey

2.1 Survey of Existing Systems

Key findings from the literature survey:

Antivirus Software and Browser-Based Solutions: Many anti-virus software and web browsers incorporate link scanning features to identify and block malicious links. These systems often rely on databases of known malicious URLs and heuristic analysis of link characteristics.

Phishing Detection Tools:

Several tools are designed specifically to detect phishing attempts, which often involve fake links. These tools focus on analyzing the structure and content of web pages to identify deceptive elements.

Machine Learning Approaches:

Some researchers have explored machine learning techniques to detect malicious links. These methods use features extracted from URLs, such as domain reputation, path, and query parameters, to predict the likelihood of a link being malicious.

Community-Driven Solutions:

Various online communities and organizations maintain databases of known malicious links. They rely on user reports to identify and track deceptive URLs.

2.2 Limitations of Existing Systems

While existing systems offer valuable insights and solutions for identifying fake or malicious links, they come with certain limitations, prompting the need for the "Fake Link Detector" project.

These limitations include:

Resource Intensiveness:

Some link analysis methods, especially machine learning approaches, can be resource-intensive and may not be suitable for real-time analysis.

False Positives and Negatives:

Existing systems may produce false positives (legitimate links incorrectly identified as malicious) and false negatives (malicious links missed by the system), leading to user frustration and reduced trust in the tool.

Centralized Approaches:

Some systems rely on centralized databases and algorithms, making them susceptible to a single point of failure or manipulation by attackers.

Privacy Concerns:

Link analysis systems that require user data for analysis may raise privacy concerns among users who are hesitant to share their browsing behavior.

Existing Systems:

I. **Google Safe Browsing**

   Features:

   - Real-time Warnings: Google Safe Browsing provides real-time warnings to users when they attempt to access potentially malicious websites.

- Google Integration: Integrated into web browsers like Google Chrome and Firefox.

Limitations:

- Database Dependency: Relies on a database of known threats, which may not cover newly created malicious links.
- Privacy Concerns: Some users may have privacy concerns about Google's involvement in their web browsing.

II. PhishTank

Features:

- Community-Driven: A community-driven project where users submit and verify phishing URLs.
- Open Data: PhishTank provides an open dataset of known phishing URLs for various purposes.

Limitations:

- Dependent on User Input: Accuracy depends on user participation and reporting of malicious links.
- Quality Control: Not all submissions may be accurate, potentially leading to false positives and negatives.

Chapter 3 : Proposed System

3.1 Problem Statement

In today's digital landscape, deceptive hyperlinks have become a rampant problem, jeopardizing online security and trust. Our mission is to develop a robust platform that combines advanced algorithms with user reporting to detect and report potentially fraudulent links in real-time. This innovative approach will proactively safeguard users from scams, phishing schemes, and misinformation, promoting a safer online experience. With the proliferation of scams and fake news, our platform aims to be a critical defense against these threats. By fostering trust and security, we aim to create a more reliable and trustworthy digital ecosystem where users can navigate the internet with confidence.

3.2 Proposed methodology

URL Analysis:

Theory: This technique involves parsing and analyzing the URL structure. Customization can be done by employing regular expressions to identify suspicious patterns, such as extra subdomains, uncommon TLDs (top-level domains), or variations of popular domain names.
Customization: Customize the regex patterns to suit specific threats or deception tactics commonly seen in your target audience or region.

Domain Reputation:

Theory: Assess the reputation of the linked domain using services like Who is XML API. Customization can involve using additional reputation services or creating an in-house reputation system that considers the historical behavior of websites.
Customization: Customize reputation scoring to prioritize certain indicators over others, aligning with your site's specific goals and risk factors.

HTTPS Verification:

Theory: Verifying if the link uses HTTPS. Customization can involve inspecting SSL certificates, checking for mixed content, and assessing the validity of the HTTPS implementation.
Customization: Customize HTTPS verification to focus on aspects that are most relevant to your users' security concerns.

Real-time Scanning:

Theory: Continuously monitoring links for changes in behavior. Customization can involve defining thresholds for what constitutes a significant change and how often links are scanned.
Customization: Customize the scanning frequency and change detection criteria based on the dynamic nature of the threats you aim to counter.

3.3 System Design

Web Application:

The fake link detector website is primarily a web application, allowing users to input and verify links

.

API Integrations:

Utilize APIs for services like Who Is XML API for domain reputation assessment.

Backend:

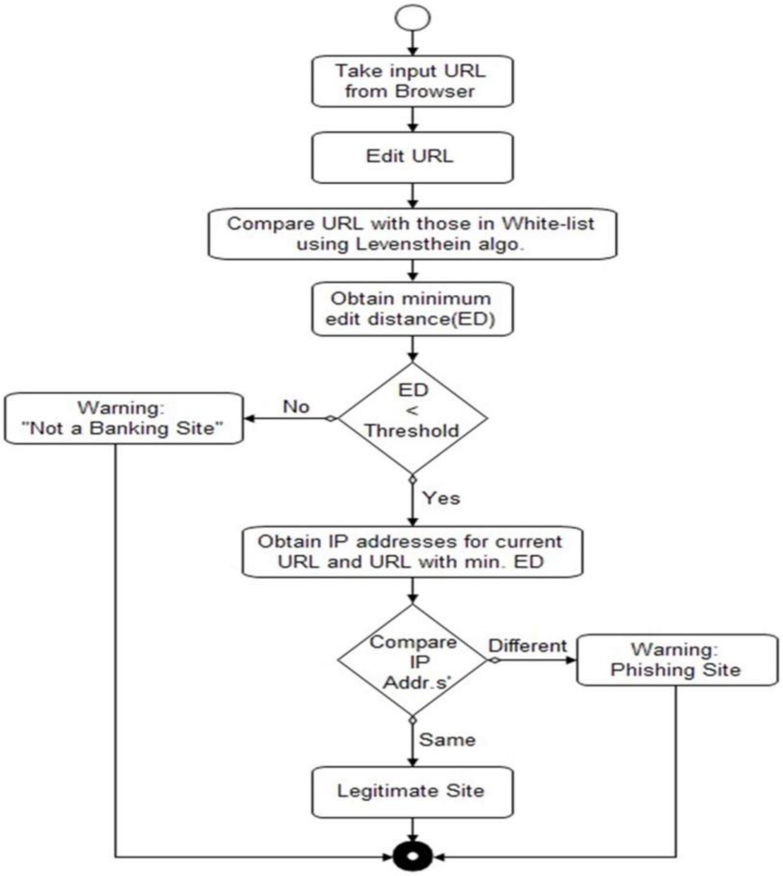
Server-side code can be implemented using JavaScript, Node.js, or another backend language

.

Frontend:

Use HTML, CSS for the user interface

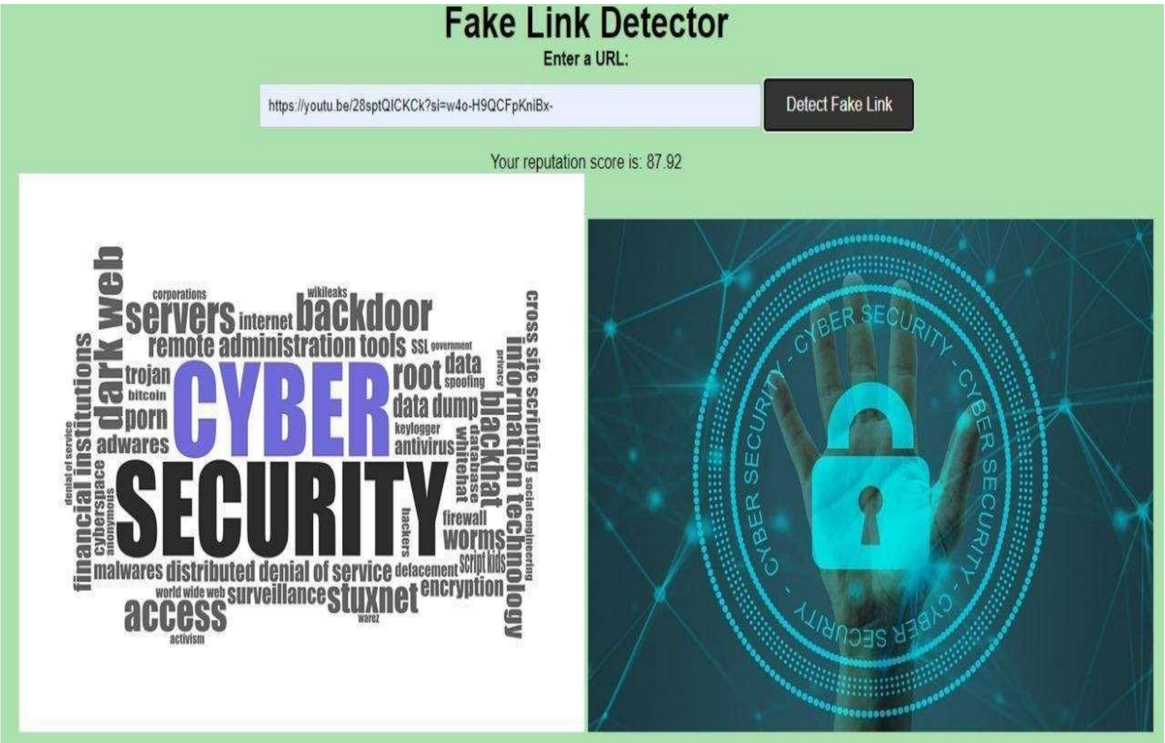
Flowchart:


Take input URL from Browser
Edit URL
Compare URL with those in White-list using Levensthein algo.
Obtain minimum edit distance(ED)
ED < Threshold
No
Warning: "Not a Banking Site"
Yes
Obtain IP addresses for current URL and URL with min. ED
Compare IP Addr.s'
Different
Warning: Phishing Site
Same
Legitimate Site

3.4 Details of Hardware and Software Requirements

- VS code
- Extensions For Java Script
- Extensions For CSS
- Extensions For HTML
- Extensions For LiveShare
- WHOISXMLAPI's Domain Reputation API

Chapter 4 : Results and Discussion

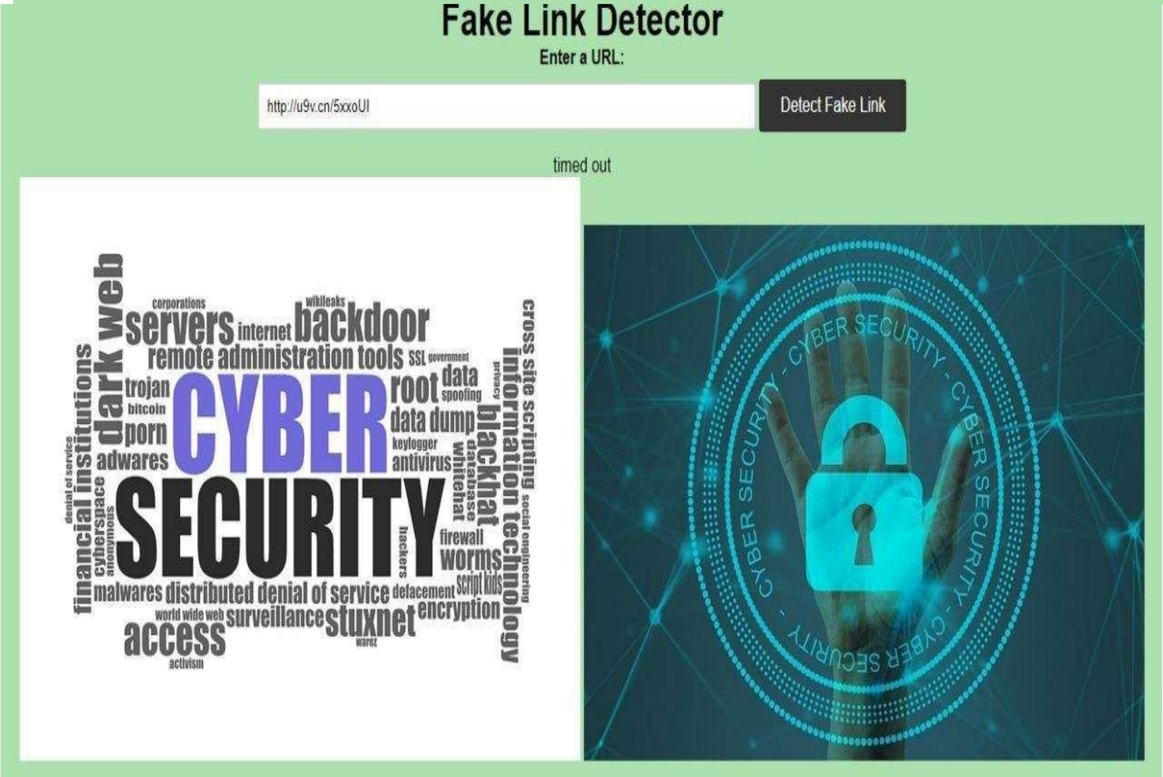


Fake Link Detector

Enter a URL:

https://youtu.be/28sptQICKCk?si=w4o-H9QCFpKniBx-

Your reputation score is: 87.92



Fake Link Detector

Enter a URL:

http://u9v.cn/5xxoUI

timed out

Chapter 5 : Conclusion and Future Scope

5.1 Conclusion

The "Fake Link Detector" mini project has already shown its potential in enhancing online security and privacy. To ensure its continued effectiveness in the ever-evolving landscape of online threats, several avenues for future development and expansion should be considered. Firstly, investing in advanced machine learning models utilizing cutting-edge natural language processing and deep learning techniques can improve accuracy and the detection of new deceptive URL types. Integrating real-time threat intelligence feeds can enhance the project's capabilities by providing users with timely and accurate threat alerts. Implementing a user feedback and reporting mechanism is crucial for user engagement and improving the machine learning model. Expanding the project into a mobile application can make it more accessible and convenient for users, while developing browser extensions for popular web browsers can offer real-time URL scanning during web browsing. These enhancements and expansions will further solidify the Fake Link Detector as a valuable tool in the ongoing fight against deceptive URLs, ensuring a safer online environment for users.

5.2 Future Scope

The Fake Link Detector mini project has already demonstrated its potential in enhancing online security and privacy. To ensure its continued effectiveness and relevance in the ever-evolving landscape of online threats, there are several avenues for future development and expansion:

Advanced Machine Learning Models: To stay ahead of increasingly sophisticated deceptive URL tactics, the project can invest in researching and implementing more advanced machine learning models. These models can utilize the latest techniques in natural language processing and deep learning to improve accuracy and the ability to detect new types of deceptive URLs.

User Feedback and Reporting Mechanism: Implementing a user feedback and reporting mechanism is crucial. This feature would allow users to report suspicious URLs that might not have been detected by the system. User reports can help improve the machine learning model and dataset by incorporating real-world examples of deceptive links.

In summary, the Fake Link Detector project has a bright future with numerous opportunities for expansion and improvement.

References

- P. Seth and M. Damle, "A Comprehensive Study of Classification of Phishing Attacks with its AI/I Detection," 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC), Bengaluru, India, 2022, pp. 370-375, doi: 10.1109/IIHC55949.2022.10060305.

- M. M. Elsheh and K. Swayeb, "Phishing Website Detection Using a Hybrid Approach Based on Support Vector Machine and Ant Colony Optimization," 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), Benghazi, Libya, 2023, pp. 402-406, doi: 10.1109/MI-STA57575.2023.10169464.

- A. Alswailem, B. Alabdullah, N. Alrumayh and A. Alsedrani, "Detecting Phishing Websites Using Machine Learning," 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2019, pp. 1-6, doi: 10.1109/CAIS.2019.8769571.

- A. Mandadi, S. Boppana, V. Ravella and R. Kavitha, "Phishing Website Detection Using Machine Learning," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-4, doi: 10.1109/I2CT54291.2022.9824801.

- M. Mohammed, K. K. Prasanth and S. V. Sai Subhash, "Phishing Detection Using Machine Learning Algorithms," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2022, pp. 921-924, doi: 10.1109/ICSSIT53264.2022.9716269.

- H. Yuan, X. Chen, Y. Li, Z. Yang and W. Liu, "Detecting Phishing Websites and Targets Based on URLs and Webpage Links," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 2018, pp. 3669-3674, doi: 10.1109/ICPR.2018.8546262.

- HTML & CSS: The Complete Reference, Fifth Edition by Thomas A. Powell

- HTML & CSS Design and Build Websites by Jon Duckett

- HTML &CSS Quick start guide by David DuRocher