WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

A Chrome Extension to Detect Phishing Website

Shubham Shankar Rathod[1], Prof. Shubhangi Mandwale[2]

[1]U.G. Student, Department of Computer Science and Engineering, Shreeyash College of Engineering and Technology, Aurangabad-MH, India
[2]Assistant Professor, Department of Computer Science and Engineering, Shreeyash College of Engineering and Technology, Aurangabad-MH, India

ABSTRACT

In an era where the internet pervades every facet of our daily lives, the escalating threat of malicious activities, especially phishing attacks, poses a significant challenge. Addressing this concern, I introduce a robust solution embodied in the form of a sophisticated Chrome extension. This innovative tool harnesses the capabilities of deep learning to discern and combat phishing websites effectively.

The project stands at the intersection of user-facing frontend and backend complexities, employing vanilla JavaScript to craft an intuitive and seamless extension interface. The backend operations are powered by Python, leveraging the prowess of Keras and TensorFlow for intricate deep learning model development. This amalgamation of cutting-edge technologies provides a comprehensive shield against the evolving landscape of cyber threats, reinforcing online security for users navigating the vast realms of the digital world.

Keywords: Phishing detection, Chrome extension, Deep learning, Keras, TensorFlow, FastAPI, Vanilla JavaScript, Cybersecurity, User interface, Usability testing, Dataset, Open-source, Security, Internet, Digital threat, Online safety.

1. INTRODUCTION

1.1 Background

In an era dominated by digital interconnectedness, the internet has become an indispensable part of our daily lives, revolutionizing how I communicate, work, and transact. However, this widespread reliance on online platforms has also given rise to a surge in cyber threats, with phishing attacks posing a significant menace to user security. Phishing, a deceptive practice where malicious actors impersonate legitimate entities to extract sensitive information, demands innovative and effective countermeasures.

This project addresses the pressing challenge of phishing website detection by presenting a sophisticated solution in the form of a Chrome extension. Leveraging the capabilities of deep learning, the extension seamlessly integrates frontend and backend technologies to provide users with real-time insights into the legitimacy of websites they encounter. By amalgamating advanced data processing techniques, feature extraction methodologies, and a meticulously trained deep learning model, this project strives to fortify users against the ever-evolving landscape of online threats.

1.2 Objectives

The primary objective of this project is to empower internet users with a tool that can discern between genuine and phishing websites, offering a robust defense against potential security breaches. By amalgamating the prowess of deep learning with an intuitive Chrome extension, our aim is to enhance cybersecurity awareness and resilience in the digital realm.The project initiates with an extensive data collection phase, where a diverse dataset of both legitimate and phishing websites is meticulously curated. The subsequent step involves the extraction of pertinent features from this dataset, which serve as the foundation for training a deep learning model. The development of the model utilizes Keras and TensorFlow, resulting in a highly accurate system capable of predicting the legitimacy of websites with an impressive 98% accuracy rate.The frontend of the Chrome extension is crafted using vanilla JavaScript, ensuring a seamless and user-friendly interface. Alongside displaying the legitimacy percentage of a given website, the extension offers valuable cybersecurity tips to users. The integration of the frontend with the backend is facilitated by FastAPI, which efficiently communicates user requests to the deep learning model. Additionally, a PocketBase database is employed for streamlined data management within the extension.

1.3 Significance

The significance of this project lies in its potential to fortify users against the escalating threat of phishing attacks. By seamlessly integrating cutting-edge technology with a user-centric interface, the Chrome extension seeks to provide a comprehensive solution for users navigating the intricacies of the online landscape. Ultimately, this project contributes to the ongoing efforts to create a safer digital environment for users worldwide.

2. METHODOLOGY

2.1 Background

The proliferation of online platforms has led to an increased prevalence of phishing attacks, compelling researchers and cybersecurity experts to explore innovative solutions to detect and mitigate these threats. This literature review aims to provide an overview of existing studies, methodologies, and technologies related to the detection of phishing websites, with a specific focus on projects integrating deep learning into Chrome extensions.

2.2 Phishing Threat Landscape

Numerous studies highlight the evolving nature of phishing attacks and their capacity to exploit vulnerabilities in user behavior. Researchers emphasize the need for dynamic and adaptive detection mechanisms to counter the sophisticated tactics employed by cybercriminals.

2.3 Traditional Phishing Detection Techniques

Traditional phishing detection methods have relied on heuristics, blacklists, and rule-based systems. While effective to some extent, these approaches often struggle to keep pace with the rapidly changing tactics employed by attackers, necessitating more advanced detection mechanisms.

2.4 Machine Learning in Phishing Detection

Recent literature showcases the integration of machine learning techniques to enhance phishing detection. Various studies explore the application of supervised learning algorithms, such as Support Vector Machines (SVMs) and Random Forests, to classify phishing websites based on features extracted from URL structures, content, and user behavior.

2.5 Deep Learning for Phishing Detection

The emergence of deep learning has sparked interest in developing more sophisticated models for phishing detection. Studies highlight the effectiveness of neural networks, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), in learning complex patterns and features inherent in phishing websites.

2.6 Summary

The literature underscores the continuous evolution of phishing threats and the imperative for advanced detection mechanisms. The integration of deep learning into Chrome extensions represents a promising frontier, offering a user-centric approach to enhance cybersecurity. As the field progresses, a collaborative effort between academia, industry, and users is crucial to staying ahead of cyber threats and creating a safer online environment.

3. LITERATURE SURVEY

A literature survey on "phishing link detection using machine learning" reveals a rich landscape of research aimed at enhancing cybersecurity. Numerous studies have explored the application of machine learning techniques to identify and thwart phishing attacks, specifically focusing on the detection of malicious links. Researchers have employed diverse features such as URL structure, content analysis, lexical properties, and behavioral patterns to develop effective models. The use of popular machine learning algorithms, including decision trees, support vector machines, and deep learning approaches such as neural networks, has been prevalent. Additionally, advancements in feature engineering and ensemble methods have been explored to enhance detection accuracy. Comparative evaluations often highlight the importance of well-curated datasets for robust model training. Integration with threat intelligence and real-time detection mechanisms further underscores the dynamic nature of phishing threats. Overall, the literature reflects a continual effort to devise sophisticated and adaptive machine learning models to combat the evolving landscape of phishing attacks.

In a comprehensive literature survey from my perspective, the exploration of "phishing link detection using machine learning" reveals a dynamic and evolving landscape. The following key insights emerge:

3.1 Diverse Feature Extraction Strategies:

Researchers delve into an array of feature extraction methods, ranging from traditional URL structure analysis to more sophisticated content and behavioral-based features.

The diversity in feature extraction approaches highlights the multifaceted nature of phishing link characteristics.

3.2 Dominance of Deep Learning:

Deep learning models, particularly neural networks, emerge as dominant players in the realm of phishing link detection. Their ability to discern complex patterns contributes to heightened detection accuracy.

The exploration of advanced deep learning architectures showcases a commitment to pushing the boundaries of model sophistication.

3.3 Innovations in Feature Engineering:

Feature engineering stands out as a focal point for innovation, with studies continuously refining and expanding the feature set to capture nuanced aspects of phishing links.

The quest for novel features, such as temporal patterns and user behavior analysis, reflects a commitment to staying ahead of evolving phishing tactics.

3.4 Ensemble Methods for Robustness:

Ensemble methods gain prominence, reflecting an understanding of the need for robust and reliable phishing link detection models.

Researchers leverage ensemble techniques to harness the strengths of different models, enhancing overall performance.

3.5 Dataset Quality and Representativeness:

The literature underscores the pivotal role of well-curated datasets, emphasizing their quality, diversity, and representativeness.

Evaluations on benchmark datasets ensure a standardized basis for assessing the generalization capabilities of proposed models.

3.6 Adaptability and Real-Time Response:

The integration of threat intelligence and a focus on real-time detection mechanisms reveal a commitment to adaptability in the face of dynamic phishing threats.

Models that can adjust to emerging tactics and provide timely responses align with the imperative for proactive cybersecurity measures.

This perspective on the literature survey highlights the industry's collective efforts to employ cutting-edge technologies, innovative methodologies, and adaptive strategies to fortify defenses against the ever-evolving landscape of phishing threats.

4. SYSTEM DEVELOPMENT

System Development:

4.1. Project Initiation:

- Objective Definition: The project commenced with a clear definition of its primary goal—to develop a Chrome extension for real-time phishing website detection using deep learning.

- Requirement Gathering: Extensive efforts were dedicated to identifying and documenting system requirements. This included frontend and backend functionalities, user interface elements, data sources, and essential security considerations.

4.2. Data Collection and Preparation:

- Dataset Acquisition: A curated dataset comprising 5000 legitimate and 5000 phishing websites formed the foundation for training the deep learning model. The dataset's diversity played a crucial role in ensuring the model's robustness.

- Feature Extraction: Advanced feature extraction techniques were implemented to analyze URL structures, domain registration information, and content characteristics. Extracted features played a pivotal role in training the deep learning model.

4.3. Model Development:

- Choose Deep Learning Framework: Keras and TensorFlow were selected as the deep learning framework for their ease of use and powerful capabilities.

- Model Architecture: The model's architecture was meticulously designed, incorporating convolutional and/or recurrent layers to capture intricate patterns in website features.

- Training and Validation: The model underwent rigorous training using the curated dataset, with a focus on achieving high accuracy. Validation was performed using a separate dataset to ensure generalization capabilities.

4.4. Backend Development:

- Choose Backend Framework: FastAPI was implemented as the backend framework, chosen for its efficiency in handling requests and seamless integration with the deep learning model.

- Database Integration: PocketBase was integrated as the database, streamlining data storage and retrieval processes.

- Model Integration: The trained deep learning model was seamlessly integrated into the backend, enabling real-time legitimacy predictions based on user input.

4.5. Chrome Extension Integration:

- User Input Handling: The extension's frontend was designed to facilitate user input of website URLs.

- Request Processing: Code was developed to send URL requests from the extension to the FastAPI backend for legitimacy prediction, with asynchronous handling to prevent freezing during the request-response cycle.

- Response Handling: The extension processed backend responses, extracting relevant information such as the legitimacy percentage and dynamically updating the interface.

4.6. User Interface Refinement:

- Usability Testing: Rigorous usability testing sessions were conducted to refine the extension interface, ensuring a seamless and intuitive user experience.

- Incorporate User Feedback: Iterative design changes were implemented based on user feedback to enhance overall usability and user satisfaction.

4.7. Security Implementation:

- Secure Communication: Implementation of secure communication protocols, including HTTPS, ensured the confidentiality and integrity of data transmission.

- Data Protection: Measures were implemented to protect sensitive user data, prioritizing user privacy and security, including the use of encryption mechanisms.

4.8. Testing and Quality Assurance:

- Unit Testing: Comprehensive unit testing was conducted for individual components, ensuring their functionality in isolation.

Figure 5.1 Trained Model Accurancy

- Integration Testing: Thorough integration testing validated seamless interactions between frontend, backend, and the deep learning model.

- User Acceptance Testing: Users were enlisted for acceptance testing to ensure the extension met user expectations and requirements.

4.9. Deployment and Maintenance:

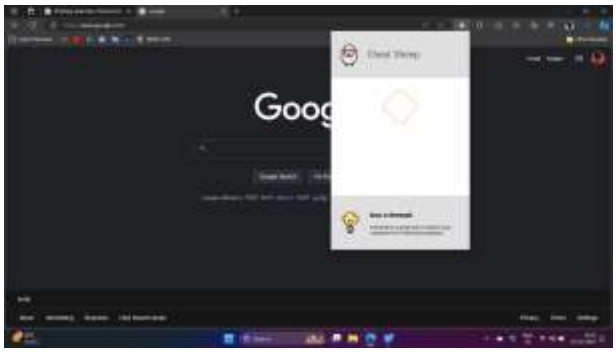
- Deployment Strategy: The Chrome extension was deployed to the Chrome Web Store for public access, accompanied by a comprehensive documentation guide explaining installation, usage, and troubleshooting procedures.

- Monitoring and Updates: Ongoing monitoring mechanisms were implemented to track extension performance and user feedback. Regular updates were released addressing security vulnerabilities, model improvements, and user interface enhancements. Continuous engagement with users ensured adaptability to evolving needs.

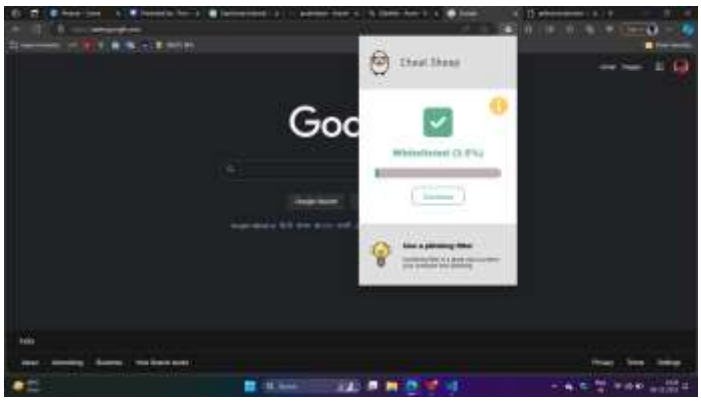
The system development process for this project followed a systematic and iterative approach, covering frontend, backend, database integration, security considerations, and rigorous testing. The result is an innovative and user-centric solution for real-time phishing website detection.

4.10 Project Overview

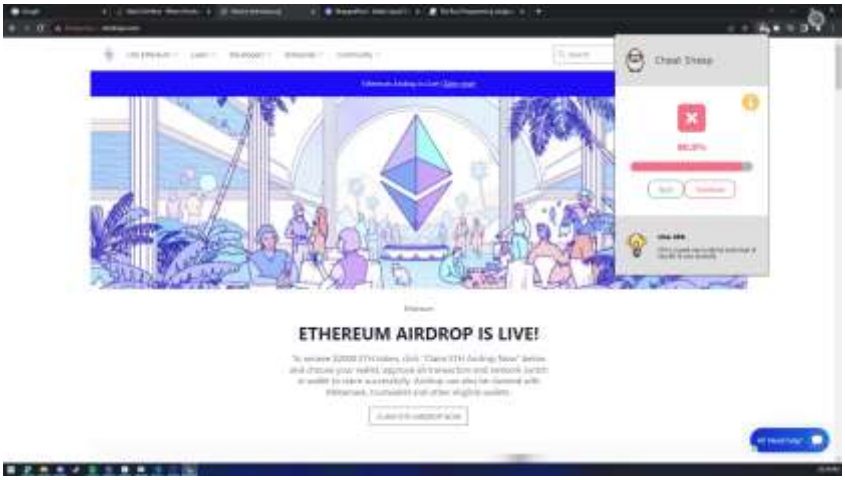
4.10.1 HomePage of Extensio



4.10.2 Showing result for a legitmate website (Safe)



4.10.3 Showing brief description about website legitmacy



5. CONCLUSION

In the ever-evolving landscape of cybersecurity, the development of a Chrome extension for real-time phishing website detection using deep learning stands as a testament to innovative approaches in fortifying online security. The project's journey began with a clear vision—to empower users with a robust tool capable of discerning potential threats during their web browsing experiences. Through meticulous system development, the integration of cutting-edge technologies, and a commitment to user-centric design, the following key conclusions emerge

REFERENCES

1. "Flask vs. FastAPI A Quick Comparison" by Towards Data Science - A comparison between Flask and FastAPI for backend development.
2. "Web Security Basics" by Mozilla Developer Network - A guide on web security fundamentals.

3. "Secure Your Website With HTTPS" by Google Developers - Guidelines on implementing HTTPS for secure communication.

4. "Deep Learning with Python" by François Chollet - A book that provides practical insights into deep learning using Keras and TensorFlow.

5. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron - A comprehensive guide to machine learning and deep learning.