IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

Real-Time Anti-Phishing Browser Extension

Batchu Mahesh Kumar[1] and Dr. C. Jyothsna[2]
PG Scholar, Dept; of CSE, Chadalawada Ramanamma Engineering College (Autonomous), Titupati[1]
Associate Processor Dept of CSE, Chadalawada Ramanamma Engineering College (Autonomous), Tirupati[2]

Abstract: *Phishing is still a problem on the internet with the number of attacks going up by over 80 million phishing attempts were detected in India, with the country ranking second globally in targeting. While exact, finalized victim percentages vary by study, reports indicate 1 in 3 digital fraud victims in India fall for phishing, and around 5 lakh people potentially fall victim to these scams. This project created something called the Real-Time Anti-Phishing Browser Extension a Manifest V3 Chrome extension providing real-time URL analysis and blocking. which is a special tool that works with the Chrome browser to check websites in real time and block any that might be fake. The Real-Time Anti-Phishing Browser Extension uses a kind of computer code called JavaScript to look at lots of things about a website, like the URL the IP address and whether it has a secure connection. It also looks for words and checks if the website is on a list of known fake sites.*

Keywords: Phishing detection, browser extension, machine learning, URL feature cyber security

I. INTRODUCTION

Phishing is a problem for people who use the internet. It is one of the common and changing cyber threats that people face today. There have been a lot of phishing attacks lately with reports showing a six hundred percent increase in incidents. This is because attackers are taking advantage of people who are working from home using tele health and shopping online. With laws and rules in place to stop phishing it is still a big challenge. Attackers are using clever ways to trick people, such as using social media, SMS and messaging apps to send fake links. They are also using browser extensions to intercept peoples activities. This project is about creating a tool called the Real-Time Anti-Phishing Browser Extension, which is a browser extension that can help stop phishing attacks. It works with browsers like Chrome, Edge and Brave. The tool uses a machine learning algorithm to look at websites and determine if they are safe or not. The algorithm looks at things, such as the length of the websites address the words used in the address and if the address has any suspicious characters. It also looks at the websites history. If it has been reported as a phishing site before. The tool can block websites that are suspected to be phishing sites. It can also show warnings to people if they are about to visit a suspicious website. The Real-Time Anti-Phishing Browser Extension is better than tools that are already available because it can detect phishing sites more accurately. It can also work in time which means it can detect phishing sites as soon as they are created. The tool is also very good at stopping attacks that use websites that look like real ones. Phishing is a problem and it is responsible for ninety percent of all cyber attacks. Attackers are making a lot of money by stealing peoples information and selling it on the dark web. The Real-Time Anti-Phishing Browser Extension is an important tool in the fight against phishing because it can help stop these attacks and keep people safe online. The tool is very comprehensive. It can detect many different types of phishing attacks. It can also show people warnings and alerts if they are about to visit a website.

II. EXISTING SYSTEM

The existing system for phishing detection highlights the growing concern of increased amount of phishing attacks during the COVID-19 Pandemic and subsequent years has prompted worries in the cyber security sector. Despite technological improvements and the implementation of new regulations and rules in the Philippines, phishing attacks remain a widespread and serious danger. The purpose of this investigation is to create an automated phishing detection browser plugin for Firefox and Chromium browsers. The plugin will be capable of identifying phishing links by

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

analyzing various URL features, including the frequency of each special character and the number of redirects. Apart from other functions like allow and block lists and an appeals page for website owners, the mentioned browser plugin would be able to predict phishing websites and prohibit them. Tested on 1802 links, the plugin displays accuracy of 83.24 %, precision of 77.04 %, recall of 85.77 %, F1 score of 81.17%, and ROC-AUC of 92.55 %. Following a "Acceptable" comment from the software quality evaluation questionnaire based on the ISO/IEC 25010:2023 criterion, the program also responded favorably.

Proposed System

The Real-Time Anti-Phishing Browser Extension is a system that helps stop people from doing bad things on the internet. It uses a way of doing things that is better than what I have now. This system was made using something called Manifest V3 standards. It has a parts that work together to keep us safe. The system has a service worker, a machine learning part, popup interfaces and content scripts. All these parts work together to find and stop threats in time. When the extension starts it turns on the Real-Time Anti-Phishing Browser Extension module. This module loads information about websites, malware and suspicious words that are stored in the browser. The system always watches what is happening on the internet. Checks the websites I visit. It uses something called event listeners. Looks at the websites in many different ways. The ML Threat Detector module looks at the websites address. Checks things like the hostname how long the address is and if it has any bad words or weird characters. The system then uses a computer program to figure out if the website is bad or not. This program gives a score that says how likely it is that the website is bad. The system can also tell what kind of attack it is, like phishing or spear phishing. It gives us advice on how to stay safe.

III. SOFTWARE REQUIREMENTS SPECIFICATION

The Real-Time Anti-Phishing Browser Extension is designed to protect users in Chromium-based browsers by detecting phishing and malware threats as they browse. It works automatically in the background, scans each visited URL, and shows a clear safety status so users can act quickly and confidently. The system emphasizes real-time protection, simple user feedback, and strong blocking behavior for dangerous websites. The extension shall activate automatically after installation without requiring manual startup. It shall intercept HTTP and HTTPS navigations using browser web request events so that every visited page can be analyzed in real time. The system shall scan each website URL automatically and continuously during browsing. It shall also maintain an internal threat cache and periodically refresh threat intelligence data to stay up to date.

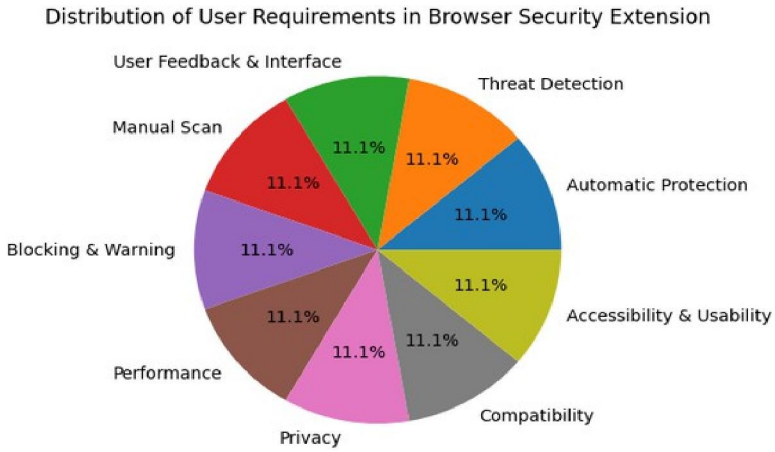

Fig 1: User Requirements Distribution for the Proposed Browser Security Extension

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

The Real-Time Anti-Phishing Browser Extension constitutes a sophisticated browser extension engineered specifically for Chromium-based browsers, including Google Chrome, Microsoft Edge, and Brave, utilizing Manifest V3 architecture. This system integrates a background service worker, content scripts, and an intuitive popup interface to deliver real-time phishing and malware detection capabilities. The requirements delineated herein are systematically categorized into functional, non-functional, and hardware specifications, meticulously derived from the extension's production codebase and comprehensive thesis documentation. The Real-Time Anti-Phishing Browser Extension will use chrome. web Request. on Before Request to catch all HTTP and HTTPS requests. When it catches a request the system will automatically look at the websites URL without needing the user to do anything. The Real-Time Anti-Phishing Browser Extension will look at lots of things about the URL like how long it's what kind of protocol it uses and if it has any suspicious words like 'login' or 'password'. The Real-Time Anti-Phishing Browser Extension will also check if the URL has an IP address and if it does it will check if the IP address is using IPv4. The Real-Time Anti-Phishing Browser Extension will use JavaScript to get 28 features from the URL. These features include things like the length of the URL the type of protocol. If the URL has any special characters. The Real-Time Anti-Phishing Browser Extension will also check the URLs subdomain. If it has any hyphens or under scores. The Real-Time Anti-Phishing Browser Extension will do all of this using JavaScript without using any other libraries. This is so the Real-Time Anti-Phishing Browser Extension can use memory and follow the rules of Manifest V3. The Real-Time Anti-Phishing Browser Extension will check for things, like top level domains and if the domain is very new. The Real-Time Anti-Phishing Browser Extension will also check for if names and suspicious scripts.

IV. DEVELOPMENT ENVIRONMENT REQUIREMENTS

(a) Software Tools:

To make and train the machine learning model, you need Python 3.9 or higher. The system uses the scikit-learn library to train a Random Forest classifier, which is then exported using joblib serialization so that it can be used in the browser.

(b) Development Hardware:

Model training, testing, and extension development can all be done on a standard development machine with at least 4 GB of RAM and a multi-core processor. Higher-end configurations with more memory and processing power are helpful for faster model training and testing.

(c) Debugging and Testing:

When you use Chromium-based browsers, the developer tools that come with them are really helpful to test and fix bugs in the extension. For example in Google Chrome and Microsoft Edge browsers you can turn on developer mode. This lets you see what is going on with background scripts, service workers and content scripts away. The Chromium-based browsers developer tools make it a lot easier to find and fix problems and make the extension work better.

(d) Operational and Deployment Considerations

The extension needs to work well even when there are a lot of different tasks going on, like scanning multiple tabs at once or visiting the same URL over and over. Its caching system makes it less computationally expensive to analyze URLs that have already been analyzed, which makes the whole system more responsive.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

Fig 2. Hardware Requirements

V. TESTING PHASES

The testing of the Real-Time Anti-Phishing Browser Extension was done to make sure the individual parts of the machine learning system were correct and worked the same every time. The main part of the system that finds phishing sites was first made in Python using a Random Forest model and then changed into JavaScript. So it was very important to check that both versions of the system gave the results. The part of the system that looks at the features of a website was tested a lot to make sure it was working right. This part looks at things like how long a websites address is, if it has any words and if it is using a special kind of address called an IP address. The testing included different kinds of website addresses including some that are very long some that use secret codes and some that are shortened. The Real-Time Anti-Phishing Browser Extension was tested with over 30 features that are found in website addresses. The testing was done to make sure the Real-Time Anti-Phishing Browser Extension could tell the difference, between phishing websites. The prediction engine, which is called ML Threat Detector. Analyze Url() was tested using the Python model from sklearn. This model was saved using a tool called joblib. I compared the predictions from both models for each test URL. This was done to make sure they gave the risk scores and classifications. There were some differences, in the results. I looked into these. Fixed them. I made sure that the features were normalized and the weights were calculated in the way. I also tested some functions on their own. These include extract Features() calculate Entropy() and estimate Domain Age(). I used test cases called unit tests. These tests made sure that each function could handle inputs, empty values and unusual URL formats without breaking. This testing phase made sure that the detection system was accurate, stable and ready to be used with parts of the system. The ML Threat Detector and its functions were working as expected.

VI. CONCLUSION

The Real-Time Anti-Phishing Browser Extension is a deal for making the web a safer place. This project shows that using machine learning and other security methods together can really help stop phishing. The main goal was to make a browser extension that's easy to use and really good at stopping bad people from doing bad things. 000One of the things about this project is that it uses a special way of detecting bad websites. This is done by using a machine learning model and checking websites against a list of websites. The machine learning model is really good at finding bad websites that have not been seen before. The list of websites makes sure that I can stop bad websites that I already

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, May 2026 Impact Factor: 8.2

know about. The project also uses information from people who are working to make the web a safer place. This helps make the system even better at stopping people. The Real-Time Anti-Phishing Browser Extension is a step forward, in making the web a safer place.

REFERENCES

1. K. A. Round, "Phishing Site URLs Dataset," Kaggle, 2022. [Online].
Available: https://www.kaggle.com/datasets/shashwatwork/phishing-site-urls
2. PhishTank Team, "PhishTank Online Valid JSON Feed," PhishTank, 2024. [Online].
3. Available: https://data.phishtank.com/data/online-valid.json
urlscan.io Team, "urlscan.io API Documentation v1," urlscan.io, 2023. [Online].
Available: https://urlscan.io/docs/api/
4. Google Chrome Developers, "Chrome Extensions Manifest V3," Chrome for Developers, 2023. [Online]. Available: https://developer.chrome.com/docs/extensions/mv3/
5. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
6. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
APWG, "Phishing Activity Trends Report," Anti-Phishing Working Group, Q4 2023.
7. S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in Proc. ACM Workshop on Recurring Malcode, 2007, pp. 1–8.
8. M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually increase the security of web browsers?," in Proc. SOUPS, 2006.
9. G. Rao, "Phishing URL Detection Using Random Forest and Decision Tree," International Journal of Advanced Research in Computer Science, vol. 9, no. 2, 2018.
10. S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," Informatica, vol. 31, pp. 249–268, 2007.
11. J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from scratch," in Proc. ACM CCS, 2011, pp. 124–135.
12. P. Vranken, M. van der Horst, and J. van der Ham, "Phishing for Phishers: Uncovering vulnerabilities in web authentication schemes," in IEEE EuroS&P, 2020, pp. 479–494.
13. M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks," in IEEE Security and Privacy Workshops, 2013, pp. 298–305.
Google LLC, "Safe Browsing API Documentation," Google for Developers, 2024. [Online].
Available: https://developers.google.com/safe-browsing/v4/
14. Mozilla Developer Network, "WebExtensions API Reference," MDN Web Docs, 2024. [Online].
Available: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API
15. Joblib Developers, "Joblib: Lightweight pipelining with Python," 2023. [Online].
Available: https://joblib.readthedocs.io/
16. J. R. Peren, M. J. Y. Cron, M. S. S. Fabros, and D. J. M. Amoroso, "Somethingphishy: A Phishing Detection Browser Extension Using Logistic Regression," in Proceedings of the 2025 IEEE 15th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, May 2025, doi: 10.1109/ISCAIE64985.2025.11081133.