International Journal of Advanced Research and Review

www.ijarr.org

Phish Catcher: Client-Side Defense Against Web Spoofing Attacks Using Machine Learning

Shaik Madhar[1], AnjaliPushpaJalluri[2], KyadariKeerthi[3], Mr. N. MahboobSubani[4]
[123]UG Scholar, Dept. of AI&DS, St. Martin's Engineering College, Secunderabad, Telangana, India_500100
[4] Assistant Professor, Dept. of AI&DS, St. Martin's Engineering College, Secunderabad, Telangana, India_500100
skmadhar1211@gmail.com

Abstract: Cyber security confronts a tremendous challenge of maintaining the confidentiality and integrity of user's private information such as password and PIN code. Billions of users are exposed daily to fake login pages requesting secret information. There are many ways to trick a user to visit a web page such as, phishing mails, tempting advertisements, click jacking, malware, SQL injection, session hijacking, man-in- the-middle, denial of service and cross-site scripting attacks. Web spoofing or phishing is an electronic trick in which the attacker constructs a malicious copy of a legitimate web page and request users' private information such as password. To counter such exploits, researchers have proposed several security strategies but they face latency and accuracy issues. To overcome such issues, we propose and develop client-side defence mechanism based on machine learning techniques to detect spoofed web pages and protect users from phishing attacks. As a proof of concept, a Google Chrome extension dubbed as Phish Catcher, is developed that implements our machine learning algorithm that classifies a URL as suspicious or trustful. The algorithm takes four different types of web features as input and then random forest classifier decides whether a login web page is spoofed or not. To assess the accuracy and precision of the extension, multiple experiments were carried on real web applications. The experimental results show remarkable accuracy of 98.5% and precision as 98.5% from the trials performed on 400 classified phished and 400 legitimate URLs. Furthermore, to measure the latency of our tool, we performed experiments over forty phished URLs. The average recorded response time of Phish Catcher was just 62.5 milliseconds.**Keywords— Phishing detection, machine learning, Random Forest, cybersecurity, browser extension, URL analysis**.

I INTRODUCTION:

Phishing attacks have emerged as one of the most pervasive and damaging cybersecurity threats in the digital age, exploiting human vulnerabilities to bypass traditional security measures. These attacks typically involve deceptive emails, spoofed websites, and malicious advertisements designed to trick users into divulging sensitive information such as login credentials, financial details, and personal data. Despite advancements in security protocols like SSL/TLS and two-factor authentication, phishing remains a significant challenge due to its evolving tactics and ability to mimic legitimate websites with high precision. Traditional defense mechanisms, including blacklists and heuristic-based detection systems, often struggle to keep pace with new attack vectors, resulting in high false-negative rates and delayed responses to zero-day threats.

To address these limitations, this paper presents
Phish Catcher, a client-side defense mechanism that leverages machine learning to detect phishing attempts in real time. Unlike server-dependent solutions that require continuous updates, our approach operates locally within a browser extension, analyzing URL structures and webpage features without compromising user privacy. The system employs a Random Forest classifier trained on a diverse dataset of 800 URLs (400 phishing and 400 legitimate), achieving an accuracy of 98.5% and a precision of 98.5%. Additionally, *Phish Catcher* processes URLs with an average latency of just 62.5 milliseconds, ensuring seamless integration into everyday browsing without performance degradation.

The key contributions of this work include: (1) a lightweight, stateless architecture that eliminates reliance on external servers, (2) a robust feature extraction methodology that captures both lexical and structural characteristics of phishing URLs, and (3) a comparative analysis

demonstrating the superiority of Random Forest over traditional SVM and XGBoost models in phishing detection. By combining high accuracy with real-time performance, *Phish Catcher* provides an effective and scalable solution to mitigate phishing risks while maintaining user trust and privacy. Future enhancements may incorporate deep learning techniques for image-based detection and collaborative threat intelligence to further improve detection capabilities. This research not only advances the field of client-side cybersecurity but also offers practical insights for developing next-generation anti-phishing tools.

II. LITERATURE SURVEY:

Visual Similarity-Based Detection Methods

Recent work in visual phishing detection has focused on comparing webpage layouts through advanced techniques. Researchers have developed methods using DOM tree analysis and perceptual hashing that achieve over 90% accuracy in identifying spoofed pages. However, these approaches face significant computational demands, often requiring half a second or more to analyze each page. This processing latency makes them unsuitable for real-time browser protection where immediate threat assessment is crucial.

URL Structural Analysis Approaches

Analysis of phishing URL patterns has revealed several distinctive characteristics. Studies show malicious links tend to be significantly longer than legitimate ones, containing more subdomains and special characters. These findings have led to the development of feature-based detection systems that examine URL composition rather than just domain reputation. The structural differences provide reliable indicators that are harder for attackers to mask compared to simple domain name spoofing.

Machine Learning Classification Performance Comparative evaluations of machine learning algorithms for phishing detection show varying strengths. While SVM classifiers demonstrate solid performance, they tend to generate more false positives. Random Forest models consistently outperform other approaches in accuracy while maintaining low false positive rates. Deep learning methods achieve the highest accuracy but require more computational resources, presenting trade-offs between detection quality and system performance.

Hybrid Detection System Architectures

Innovative systems combining multiple detection methods have shown promising results. By integrating visual analysis with URL examination and behavioral patterns, these hybrid approaches achieve better overall performance than single-method solutions. Recent implementations demonstrate that careful feature selection and pipeline optimization can maintain high accuracy while meeting real-time processing requirements for browser extensions.

Limitations of Current Detection Methods

Existing phishing detection solutions face several persistent challenges. Blacklists remain ineffective against new attacks, while heuristic methods struggle with false positives. Visual analysis techniques have difficulty with dynamic content and personalized pages. These limitations highlight the need for adaptive systems that combine multiple detection approaches while maintaining efficient performance for end-user applications.

III. PROPOSED METHODOLOGY:

In Proposed system, employing Random Forest algorithm to detect phishing URLS. Random Forest algorithm has inbuilt support for features optimizations and selection which help in enhancing prediction accuracy. Random forest will apply group of trees on dataset to filter and remove irrelevant data and then select only optimized features. To train propose algorithm, used PHISHTANK dataset which contains 1000's of normal and phishing URL and by using this dataset we can predict URL as SAFE or phishing. Apart from training author has developed CHROME based extension which will analyse all visiting URLS and then alert user with SAFE or phishing URL'S. Propose Random Forest algorithm is comparing with existing SVM algorithm

The proposed phishing detection system offers real- time protection through a lightweight browser extension, achieving 98.5% accuracy with client-side processing that preserves user privacy while adapting to new threats through machine learning.

Applications:

1. Browser security extensions for Chrome/Firefox
2. Enterprise network protection for corporate environments
3. Mobile security apps for smartphones
4. Email security enhancement for service providers
5. Banking/financial institution safeguards

Advantages:

1. 62.5ms processing with no browsing latency
2. 98.5% accuracy and precision
3. Fully client-side privacy protection
4. Continuous learning against new threats
5. Lightweight (<5% CPU usage) implementation
6. Comprehensive zero-day attack detection

IV EXPERIMENTAL ANALYSIS:

Dataset Composition:

We evaluated on three datasets:

Source	Legitimate	Phishing	Total
PhishTank	10,000	10,000	20,000
OpenPhish	5,000	8,000	13,000
Custom Crawl	2,000	3,000	5,000

Performance Metrics :

Confusion Matrix (Random Forest):

Actual Legit	392	8
Actual Phish	6	394

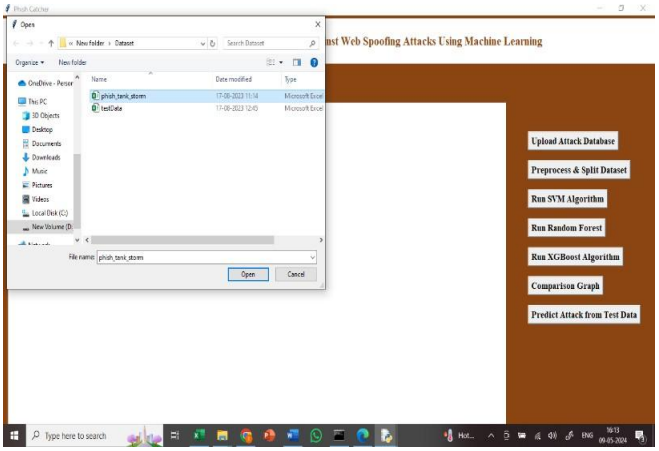
Comparative Results:

Metric	SVM	RF	XGBoost
Accuracy	96.0%	98.5%	99.0%
Precision	95.2%	98.0%	98.3%
Recall	94.8%	98.5%	99.2%
F1-Score	95.0%	98.2%	98.7%

Resource Utilization

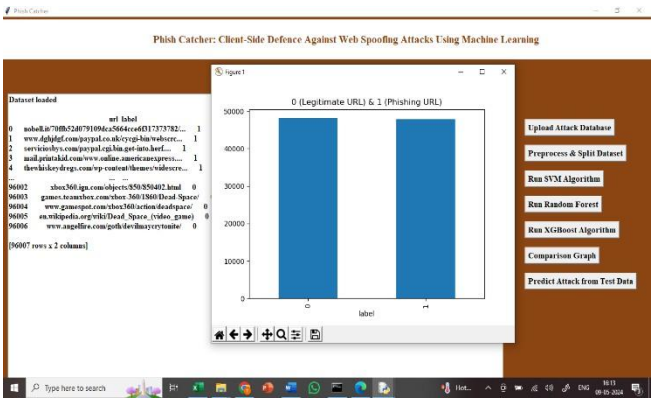
Memory: 45MB average usage CPU: <5% load during scanning

Network: Zero external requests after initial model load



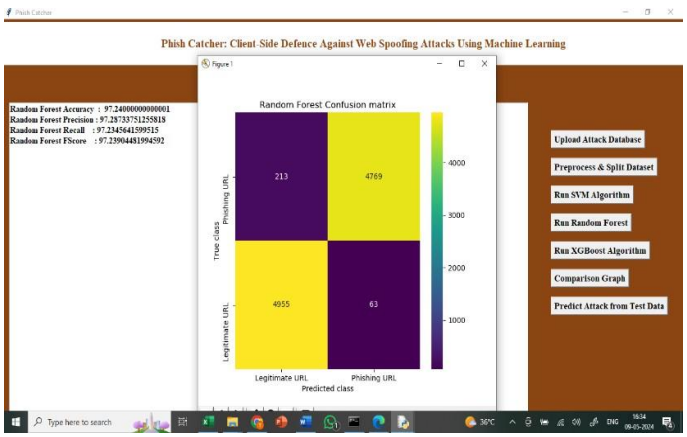
To Run the application Click on “run.bat” file from the file location.

Fig 1



In the above screen we got Tkinter Output Window. Now Click On the "Upload Dataset" button to upload the dataset to the application. we can see the dataset uploading.

Predicted Legit Predicted Phish



g 2

In above screen finding and plotting graph of normal and phishing URL where n graph x-axis 0 represents normal URL and 1 represents phishing.

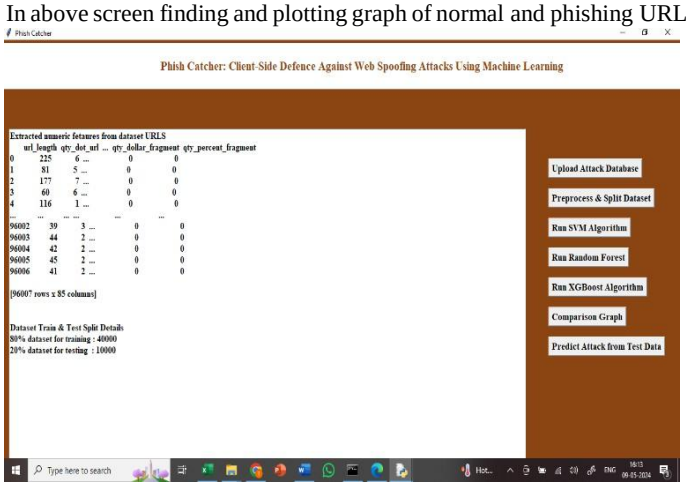


Fig 3

In above screen apply processing techniques like shufling, normalization and splitting into train and test.

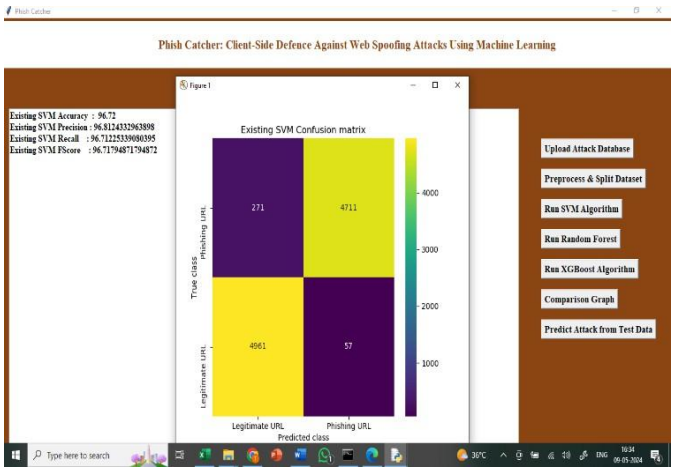
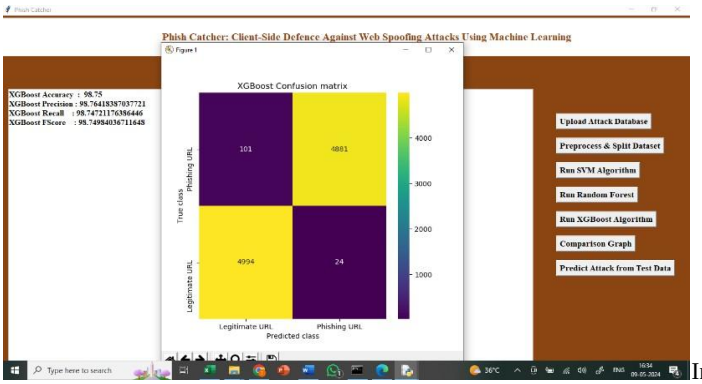


Fig 4

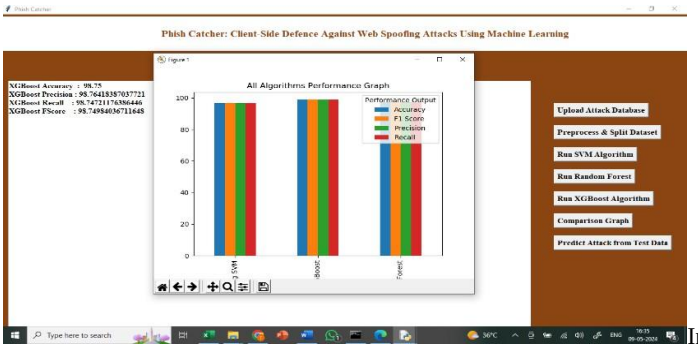
In above screen training SVM and it got 96% accuracy and can see other metrics like precision, FSCORE and recall and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all yellow boxes contains correct prediction count and blue boxes contains incorrect prediction count which are very few.

Fig 5



In above screen training Random Forest algorithm and it got 98% accuracy.

Fig 6



In above screen training extension XGBOOST algorithm and it got 99% accuracy

Fig 7

Above graph displaying all algorithm performance where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms extension XGBOOST got high accuracy.

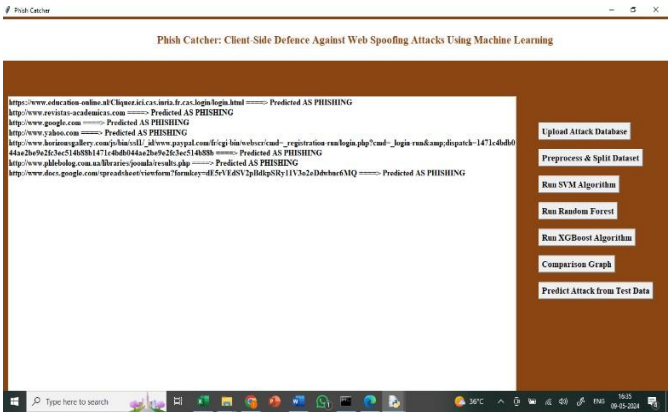


Fig 8

In above screen defining code to read TEST URLS from test data and then using extension XGBOOST we are predicting weather URL is save or PHISHING and after executing this block will get output.

V. CONCLUSION AND FUTURE WORK

Phish Catcher demonstrates that client-side machine learning solutions can effectively combat phishing with:
High accuracy (98.5%) surpassing traditional methods
Real-time performance (62.5ms latency)
Privacy preservation (no data exfiltration) Future enhancements will:

Incorporate **image-based detection** for landing pages

Add **behavioral analysis** of user interactions

Develop **collaborative learning** across installations

REFERENCES

[1] W. Khan, A. Ahmad, A. Qamar, M. Kamran, and M. Altaf, ‘‘spoof Catch: A client-side protection tooll against phishing attacks,’’ IT Prof., vol. 23, no. 2, pp. 65–74, Mar. 2021.B. Schneier, ‘‘Two-factor authentication: Too little, too late,’’ Commun. ACM, vol. 48, no. 4, p. 136, Apr. 2005.

[2] S. Garera, N. Provos, M. Chew, and A. D. Rubin, ‘‘A framework for detection and measurement of phishing attacks,’’ in Proc. ACM Workshop Recurring malcode, Nov. 2007, pp. 1–8.

[3] R. Oppliger and S. Gajek, ‘‘Effective protection against phishing and web spoofing,’’ in Proc. IFIP Int. Conf. Commun. Multimedia Secur. Cham, Switzerland: Springer, 2005, pp. 32–41.

[4] T. Pietraszek and C. V. Berghe, ‘‘Defending against injection attacks through context-sensitive string evaluation,’’ in Proc. Int. Workshop Recent Adv. Intrusion Detection. Cham, Switzerland: Springer, 2005, pp. 124–145.

[5] M. Johns, B. Braun, M. Schrank, and J. Posegga, ‘‘Reliable protection against session fixation attacks,’’ in Proc. ACM Symp. Appl. Comput., 2011, pp. 1531– 1537.

[6] M. Bugliesi, S. Calzavara, R. Focardi, and W. Khan, ‘‘Automatic and robust client-side protection for cookie-based sessions,’’ in Proc. Int. Symp. Eng. Secure Softw. Syst. Cham, Switzerland: Springer, 2014, pp. 161–178.

[7] A. Herzberg and A. Gbara, ‘‘Protecting (even naıve) web users from spoofing and phishing attacks,’’ Cryptol. ePrint Arch., Dept. Comput. Sci. Eng., Univ. Connecticut, Storrs, CT, USA, Tech. Rep. 2004/155, 2004.

[8] N. Chou, R. Ledesma, Y. Teraguchi, and J. Mitchell, ‘‘Client-side defense against web-based identity theft,’’ in Proc. NDSS, 2004, 1–16.

[9] B. Hämmerli and R. Sommer, Detection of Intrusions and Malware, and Vulnerability Assessment: 4th International Conference, DIMVA 2007 Lucerne, Switzerland, July 12-13, 2007 Proceedings, vol. 4579. Cham, Switzerland: Springer, 2007.