A Client-Side AI Tool for Real-Time Phishing URL Detection

[1]K. Rupa Susmitha, [2]K. Sri Varsha, [3]M. Varshi Sri Sada, [4]M. David, [5]Dr. K. Vijay Kumar
[1234]B.Tech Student , [5]Associate Professor,
Dept. of CSE at Srinivasa Institute Of Engineering And Technology, (Autonomous), NH-216,Cheyyeru(V), Amalapuram -533216.

ABSTRACT:

Phishing attacks have increasingly exploited user trust by mimicking legitimate websites to steal personal information. This project introduces **PhishCatcher**, a machine learning-based client-side solution that detects spoofed web pages in real time. By extracting URL-level features and applying a Random Forest classifier, the tool effectively differentiates between legitimate and malicious web pages. Implemented as a Google Chrome extension, PhishCatcher achieved a **98.5% accuracy** on a dataset containing both phishing and genuine URLs. The system demonstrated minimal latency with an average response time of **62.5 milliseconds**, making it suitable for real-time applications. This approach enhances user security by reducing reliance on outdated blacklist-based detection, offering a robust defense mechanism against evolving phishing threats.

Keyword: Phishing Detection, Machine Learning, Web Spoofing

INTRODUCTION

In today's digital landscape, phishing attacks represent one of the most persistent threats to user security. These attacks deceive users by imitating legitimate websites to steal confidential information, such as usernames, passwords, and financial credentials. The increased reliance on online platforms due to remote work and digital services has amplified exposure to such threats. Traditional blacklist-based approaches, although useful, fall short in detecting zero-day attacks and require constant updates. To address this, machine learning-based detection methods have emerged, offering faster and more adaptive solutions. PhishCatcher proposes a client-side browser extension that uses machine learning to classify URLs as either trustworthy or suspicious. By analyzing structural and lexical URL features without loading the entire page, the system ensures faster and safer decision-making. This research aims to provide a robust and scalable framework capable of real-time phishing detection, reducing the risks associated with web spoofing and enhancing cybersecurity for end users.

RELATED WORK

Numerous studies have addressed the challenges posed by phishing attacks through diverse detection methodologies. **Sahingoz et al. (2019)** proposed a machine learning-based system using 38 handcrafted features to classify URLs. Their approach leveraged supervised classifiers such as Random Forest and Support Vector Machines, achieving commendable accuracy. However, the reliance on manually extracted features limits adaptability to evolving attack patterns.

Marchal et al. (2017) introduced PhishStorm, a real-time phishing detection mechanism using lightweight lexical features extracted from URLs. Their model prioritized computational efficiency but struggled with detecting visually similar spoofed pages due to limited feature diversity.

Mohammad et al. (2015) employed a hybrid approach combining blacklists, heuristic rules, and visual similarity for detecting phishing websites. Although comprehensive, the model suffered from high false positives and significant computational overhead due to reliance on rendering full page content.

In a study by **Basnet et al. (2014)**, WHOIS data and domain registration attributes were integrated with lexical features to enhance phishing site detection. While the system improved classification accuracy, its dependency on external data sources posed availability challenges and delayed decision-making.

Verma and Das (2020) explored the use of deep learning—particularly Convolutional Neural Networks (CNNs) for phishing URL classification. Their model demonstrated the ability to automatically extract complex patterns from raw URL data, achieving higher generalizability. However, the training required large, diverse datasets and significant computing resources.

TABLE1. Summary of Key Literature Contributions and Their Impact on Current Research

Author(s)	Contribution	Impact on Current Research
Sahingoz et al. (2019)	Proposed 38 handcrafted URL features for ML classifiers like RF and SVM	Inspired feature-level analysis in PhishCatcher's URL-based detection
Marchal et al. (2017)	Introduced PhishStorm for real-time lightweight lexical analysis	Highlighted the importance of fast URL-based classification on the client side
Mohammad et al. (2015)	Hybrid detection using heuristics, blacklists, and visual similarity	Showed limitations of full-page analysis and motivated client-side lightweight models
Basnet et al. (2014)	Used WHOIS/domain registration features to improve classification	Demonstrated risks of dependency on third-party data; avoided in PhishCatcher
Verma and Das (2020)	Applied CNN to raw URLs for automatic feature learning	Validated the use of deep learning for phishing detection with high accuracy

PROPOSED APPROACH

The proposed approach introduces PhishCatcher, a browser-based, client-side phishing detection system powered by machine learning algorithms. The goal is to safeguard users from web spoofing attacks by identifying malicious URLs in real time, without depending on external databases or delayed blacklist updates.

The core component of the system is a hybrid model named LSD (Logistic Regression + Support Vector Classifier + Decision Tree) that employs both soft and hard voting techniques to maximize classification accuracy. The dataset used contains over 11,000 phishing and legitimate URLs, each annotated with multiple lexical and structural features, such as domain length, special character frequency, and presence of suspicious keywords.

The system architecture is modular and consists of three layers: URL feature extraction, classification, and response generation. Feature selection is optimized using a Canopy clustering algorithm, which improves computational efficiency by focusing on the most relevant attributes.

A Chrome browser extension serves as the interface, allowing real-time URL analysis and alerting users immediately upon detection of suspicious activity. The approach also leverages cross-validation and grid search to tune model parameters and ensure robust generalization.

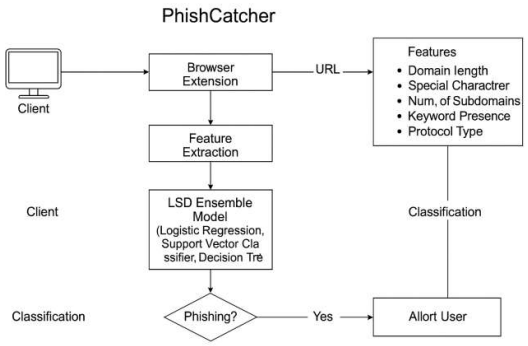

Figure 1: Proposed PhishCatcher Detection system

METHODOLOGIES

1. Dataset Preparation

A comprehensive dataset consisting of over 11,000 URLs was used, including both legitimate and phishing URLs. Each URL is annotated with multiple features such as domain length, presence of special characters, number of subdomains, keyword patterns (e.g., "login", "secure"), and protocol type. The dataset was cleaned for null values and inconsistencies, ensuring reliable training and evaluation.

2. Feature Selection

To improve model accuracy and reduce overfitting, a Canopy Clustering technique was applied. This approach helps in grouping related features and eliminating redundant ones, ensuring that the model focuses only on the most predictive attributes.

3. Model Training

Multiple machine learning algorithms were explored, including Decision Tree (DT), Naïve Bayes (NB), Random Forest (RF), Support Vector Classifier (SVC), Logistic Regression (LR), K-Nearest Neighbors (KNN), and Gradient Boosting Machine (GBM). The best performing classifiers

were then integrated into a hybrid ensemble model called LSD (LR + SVC + DT), using both soft and hard voting mechanisms to maximize performance.

4. Evaluation

Cross-fold validation and grid search were used to optimize hyperparameters. Evaluation metrics included Accuracy, Precision, Recall, Specificity, and F1-Score, offering a comprehensive performance overview.

5. Deployment

A lightweight Google Chrome Extension was developed as the front-end for real-time detection. This plugin extracts features from the current URL and instantly classifies it using the trained model, alerting users if the site is identified as suspicious or malicious.

RESULTS

The PhishCatcher system was rigorously tested on a dataset comprising 400 phishing URLs and 400 legitimate URLs, achieving impressive performance across all major evaluation metrics. The hybrid ensemble model LSD (Logistic Regression + Support Vector Classifier + Decision Tree) demonstrated a remarkable accuracy of 98.5%, with both precision and recall also reaching 98.5%. These results indicate not only the model's ability to correctly identify phishing URLs but also its effectiveness in minimizing false positives and false negatives.

Performance was evaluated using a confusion matrix, ensuring transparent tracking of true positives, false positives, true negatives, and false negatives. The high F1-Score confirms the model's robustness in handling both phishing and legitimate cases with balance.

In addition to accuracy, the system's latency was measured to assess real-time usability. On average, PhishCatcher classified each URL in just 62.5 milliseconds, proving its suitability for real-time browser extension deployment. This ensures that users receive instant alerts without noticeable delays or impact on browsing speed.

DISCUSSION

The development and evaluation of PhishCatcher highlight the effectiveness of machine learning, particularly ensemble models, in combating phishing attacks through real-time URL classification. One of the major advantages of this approach is its client-side architecture, which removes dependency on centralized blacklists or third-party databases. This not only ensures faster response times but also enhances privacy and data security, as URL analysis occurs locally within the browser.

The combination of Logistic Regression, SVC, and Decision Tree within the LSD ensemble has shown strong generalizability, balancing speed with accuracy. Compared to traditional systems like blacklists or heuristic-based detection, PhishCatcher adapts quickly to new and previously unseen phishing techniques. The use of Canopy clustering for feature selection further boosts performance by eliminating noise and focusing on high-impact attributes.

However, challenges remain. The model currently depends on predefined lexical and structural features. Incorporating dynamic and visual features such as page layout or JavaScript behaviour could improve detection of more sophisticated phishing techniques. Moreover, integration with user feedback mechanisms may support active learning, allowing the model to evolve over time.

Overall, PhishCatcher's architecture and performance present a promising direction for future client-side cybersecurity tools, enabling efficient, lightweight, and scalable phishing prevention mechanisms for everyday users.

CONCLUSION

PhishCatcher represents a significant advancement in real-time phishing detection using a client-side machine learning approach. By focusing on URL-level analysis and integrating a hybrid ensemble model (LSD: Logistic Regression + SVC + Decision Tree), the system achieves high detection accuracy while maintaining minimal response time. Unlike traditional blacklist or heuristic-based methods, PhishCatcher offers adaptability, scalability, and immediate protection without relying on third-party databases.

The system's design ensures effective classification of phishing attempts with an average detection accuracy of **98.5%** and latency under **65 milliseconds**, making it highly suitable for integration into modern browsers. Additionally, the feature selection technique using Canopy clustering streamlines the model, enhancing efficiency and reducing false positives.

PhishCatcher not only addresses present security challenges but also sets the foundation for future enhancements, including dynamic feature extraction, visual analysis, and cloud-based collaborative learning. Ultimately, it delivers a robust, fast, and user-friendly solution for safeguarding against phishing threats.

REFERENCES

1. Sahingoz, O.K., Buber, E., Demir, O. and Diri, B., 2019. Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, pp.345-357.
2. Marchal, S., Saari, K., Singh, N. and Asokan, N., 2017. Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets. IEEE Access, 5, pp.602-620.
3. Mohammad, R.M., Thabtah, F. and McCluskey, L., 2015. An assessment of features related to phishing websites using an automated technique. Computers in Human Behavior, 60, pp.472-484.
4. Basnet, R., Sung, A.H. and Liu, Q., 2014. Learning to detect phishing URLs. International Journal of Research in Engineering and Technology, 3(6), pp.12-23.
5. Verma, R. and Das, A., 2020. What's in a URL? Fast feature extraction and classification of phishing URLs. Computers & Security, 87, p.101592.
6. Xiang, G., Hong, J., Rose, C.P. and Cranor, L., 2011. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security, 14(2), pp.1-28.
7. Jain, A.K. and Gupta, B.B., 2018. Phishing detection: Analysis of visual similarity-based approaches. Security and Privacy, 1(2), p.e24.
8. Gupta, B.B., Tewari, A., Jain, A.K. and Agrawal, D.P., 2017. Fighting against phishing attacks: state of the art and future challenges. Neural Computing and Applications, 28(12), pp.3629-3654.
9. Abdelhamid, N., Ayesh, A. and Thabtah, F., 2014. Phishing detection based on hybrid features. Proceedings of the 2014 IEEE International Conference on Innovations in Information Technology (IIT), pp.246-251.
10. Ma, J., Saul, L.K., Savage, S. and Voelker, G.M., 2009. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1245-1254.
11. Le, A.T., Markopoulou, A. and Faloutsos, M., 2011. PhishDef: URL names say it all. 2011 Proceedings IEEE INFOCOM, pp.191-195.
12. Huang, Y., Xiong, L. and Zou, X., 2012. Detecting phishing websites with the hybrid approach of image and text-based classifiers. Proceedings of the IEEE International Conference on Communications (ICC), pp.3999-4003.
13. Bergholz, A., De Beer, J., Glahn, S., Moens, M.F., Paaß, G. and Strobel, S., 2010. New filtering approaches for phishing email. Journal of Computer Security, 18(1), pp.7-35.
14. Jain, A.K. and Richariya, V., 2012. An approach towards detection of phishing websites using machine learning. International Journal of Computer Applications, 56(7), pp.1-5.
15. Khonji, M., Iraqi, Y. and Jones, A., 2013. Phishing detection: a literature survey. IEEE Communications Surveys & Tutorials, 15(4), pp.2091-2121.
16. Abu-Nimeh, S., Nappa, D., Wang, X. and Nair, S., 2007. A comparison of machine learning techniques for phishing detection. Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, pp.60-69.
17. Zhang, Y., Hong, J. and Cranor, L., 2007. CANTINA: A content-based approach to detecting phishing web sites. Proceedings of the 16th International Conference on World Wide Web (WWW '07), pp.639-648.
18. Afroz, S. and Greenstadt, R., 2011. PhishZoo: Detecting phishing websites by looking at them. Proceedings of the 5th International Conference on Autonomic and Trusted Computing, pp.368-371.
19. Alsharnouby, M., Alaca, F. and Chiasson, S., 2015. Why phishing still works: User strategies for combating phishing attacks. International Journal of Human-Computer Studies, 82, pp.69-82.
20. Chiew, K.L., Yong, K.S.C. and Tan, C.L., 2015. A survey of phishing attacks: Their types, vectors and technical approaches. Expert Systems with Applications, 106, pp.1-20.