International Journal of

DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X ww.ijdim.com Original Research Paper

Pish Catcher: Machine Learning-Based Client-Side Defence Mechanism for Web Spoofing Attack Detection

Mr B L Narayana[1], Chimalamarri Vamsi Krishna[2], Buradagunta Raj Karun[3], Katuri Tulasi Ram[4], Aluka Prabhakar Raju[5], Dasuri Venkata Nikhil Kumar[6]

[1]Assistant Professor, St. Ann's College of Engineering & Technology, Chirala, Andhra Pradesh-523157

[2-6] UG Student, St. Ann's College of Engineering & Technology, Chirala, Andhra Pradesh- 523157

ABSTRACT: Web spoofing attacks are becoming increasingly advanced, making it difficult for users to differentiate between real and fraudulent websites designed to steal sensitive information. As phishing techniques evolve, strong client-side protection has become essential. This work aims to develop an automatic client-side mechanism that detects suspicious or spoofed web pages before users unknowingly share confidential data. Traditional manual checks such as verifying URLs, examining website appearance, or relying on browser indicators are often ineffective because attackers create visually identical replicas and misleading domain names, making user awareness alone unreliable. To build an accurate machine learning model, datasets from sources like PhishTank and legitimate website repositories are collected and processed for algorithms such as XGBoost. The proposed system functions as a client-side detection tool that extracts URL-based and content-based features in real time and uses models like Random Forest, XGBoost, Decision Tree, Logistic Regression, and SVM to classify pages as legitimate or spoofed, offering immediate protection without depending on server-side operations.

Key words: *Web Spoofing, Phishing Detection, Machine Learning, XGBoost, Client-Side Protection.*

1. INTRODUCTION

The rapid expansion of the internet has transformed nearly every aspect of modern life, from e-commerce and online banking to e-health, education, and social networking. Billions of users worldwide rely on web applications and online services that require authentication through login pages. While these platforms provide convenience and accessibility, they also expose users to significant cybersecurity risks. Among these, phishing and web spoofing attacks have emerged as some of the most pervasive and damaging threats. In such attacks, adversaries construct fraudulent websites that mimic legitimate ones, tricking unsuspecting users into divulging sensitive information such as usernames, passwords, and financial credentials.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Phishing attacks have become more advanced, targeting sensitive data through techniques like spoofed websites, email manipulation, and malware injection, making traditional security methods less effective. Existing anti-phishing solutions are mainly server-side or client-side, where server-side methods are less practical and client-side approaches offer better user-level protection. However, blacklist and heuristic-based client-side systems struggle with zero-day attacks and often face limitations in real-time performance and scalability.

To address these limitations, researchers have increasingly turned to machine learning (ML) techniques for phishing detection. ML models can learn complex patterns from large datasets of legitimate and phishing URLs, enabling them to identify previously unseen attacks. Algorithms such as Naïve Bayes, Support Vector Machines (SVM), Logistic Regression, and Random Forest have been applied with promising results. However, challenges remain in balancing accuracy, precision, and response time, particularly in real-world deployments where latency is critical.

This work introduces **PhishCatcher**, a client-side defense mechanism implemented as a Google Chrome extension. PhishCatcher leverages machine learning to classify login web pages as legitimate or spoofed in real time. The system extracts multiple categories of web features—including URL structure, domain properties, and content attributes—and applies a Random Forest classifier to determine authenticity. Unlike stateful tools that rely on stored screenshots or lists, PhishCatcher operates in a stateless manner, minimizing latency and storage overhead. Experimental evaluation on real web applications demonstrates remarkable performance, achieving 98.5% accuracy and precision, with an average response time of just 62.5 milliseconds. These results highlight the feasibility of deploying lightweight, client-side ML solutions for phishing defense.

2. LITERATURE SURVEY

W. Khan et al. [1] proposed Spoof Catch, a client-side protection tool designed to combat phishing attacks. Their research focused on developing a tool that operates on the client side to detect and prevent phishing attempts by analyzing and filtering out potentially harmful content. This approach aimed to enhance user security by providing an additional layer of protection against phishing threats. B. Schneier [2] discussed the limitations of two-factor authentication (2FA) in his article. He argued that while 2FA provides an extra layer of

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

security, it is often insufficient in protecting against sophisticated attacks. Schneier's analysis highlighted the need for more robust security measures beyond 2FA to effectively combat modern cybersecurity threats. S. Garera et al. [3] introduced a framework for detecting and measuring phishing attacks. Their work provided a structured approach to identifying phishing threats by using various detection techniques and metrics. This framework aimed to improve the accuracy and reliability of phishing detection systems.

N. Chou et al. [9] presented client-side defenses against web-based identity theft. Their research focused on developing methods to protect users from identity theft by analyzing and securing web interactions, thereby reducing the risk of unauthorized access to personal information. B. Hämmerli and R. Sommer [10] edited proceedings from the 4th International Conference DIMVA 2007, which covered various aspects of intrusion detection and malware vulnerability assessment. The conference proceedings included research on methods and techniques for detecting and mitigating security threats in diverse computing environments.

3 SYSTEM ARCHITECTURE

The system architecture begins with a preprocessed phishing URL dataset, followed by Canopy feature selection to reduce irrelevant attributes. Cross-validation splits data into 70% training and 30% testing sets. Grid search optimizes hyper parameters, and an ensemble model combining LR, SVC, and DT is trained to generate accurate phishing predictions.

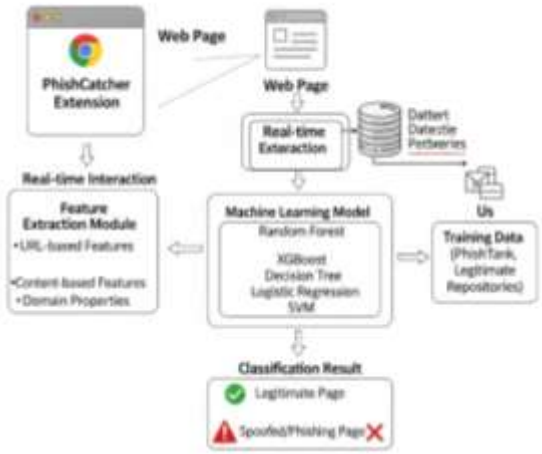

Fig1: **System Architecture**

3.1 METHODOLOGY

This section outlines the step-by-step process used to develop the PISHCATCHER system for detecting phishing and spoofed URLs using machine learning. The methodology includes data acquisition, preprocessing, feature engineering, model training, evaluation, and deployment.

3.2 Dataset Acquisition

The dataset used for training and evaluation is sourced from repositories such as PhishTank and legitimate website databases. It contains labeled URLs (phishing = 1, legitimate = 0) along with

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

domain-level and content-based features. These include ranking metrics, domain resolution indicators, cardinality ratios, and Jaccard similarity coefficients between various URL components.

3.3 Data Pre processing

Preprocessing was performed to ensure the dataset was clean and suitable for machine learning tasks. Missing values were handled either by deletion or imputation. Categorical variables were encoded into numerical form, and feature scaling was applied to normalize values across attributes. The normalization was performed using Min-Max scaling:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

This ensured that all features contributed equally to the learning process. Finally, the dataset was split into training and testing subsets using an 80:20 ratio.

3.4 Feature Engineering

Feature extraction was critical to improving model performance. A custom function was designed to compute URL characteristics such as domain length, path length, and frequency of special characters (dots, slashes, hyphens). Additionally, similarity measures were calculated using the Jaccard coefficient:

$$J(A,B) = \frac{|\ A \cap B\ |}{|\ A \cup B\ |}$$

This helped quantify structural similarities between URL components, which are often exploited in spoofing attacks.

3.5 Model Training

Three machine learning models were trained: Support Vector Machine (SVM), Random Forest, and XGBoost.

- **SVM** constructed hyperplanes to separate phishing and legitimate classes.
- **Random Forest** used an ensemble of decision trees to improve robustness.
- **XGBoost** applied gradient boosting to minimize classification errors iteratively. Its objective function is defined as:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where $l$is the loss function and $\Omega(f_k)$is the regularization term for tree f_k.

3.6 Model Evaluation

The models were evaluated using accuracy, precision, recall, and F1-score. Confusion matrices were also generated to visualize classification outcomes. The F1-score was calculated as:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

This provided a balanced measure of performance, especially in cases of class imbalance.

3.7 Client-Side Deployment

Finally, the model was integrated into a Google Chrome extension named PISHCATCHER. Operating at the client-side, it extracts features from visited URLs in real time and classifies them instantly. This ensures proactive protection against spoofing attacks without relying on server-side infrastructure.

4. DESIGN AND CONSTRUCTION

The design and construction of the PISHCATCHER system follow a systematic pipeline to ensure robust phishing detection and real-time client-side protection. The process begins with the acquisition of a labeled dataset containing both legitimate and phishing URLs. These URLs are enriched with additional attributes such as ranking, domain resolution, cardinality ratios, and similarity coefficients, which provide a strong foundation for classification.

The raw dataset is then preprocessed to make it suitable for machine learning tasks. Missing values are handled through imputation, categorical variables are encoded numerically, and feature scaling is applied to normalize values across attributes. The dataset is split into training and testing subsets to evaluate generalization performance.

Feature engineering plays a crucial role in enhancing accuracy. A custom function extracts structural and semantic attributes from URLs, including domain length, path length, frequency of special characters, and similarity measures. These engineered features capture hidden patterns that differentiate phishing URLs from legitimate ones, strengthening the predictive capability of the models.

Three machine learning models are constructed and trained: Support Vector Machine, Random Forest, and XGBoost. Each model is trained on the processed dataset to classify URLs as phishing or legitimate. Random Forest leverages ensemble learning, SVM constructs hyperplanes for separation, and XGBoost applies gradient boosting for iterative error minimization.

The models are evaluated using metrics such as accuracy, precision, recall, and F1-score. Confusion matrices are generated to visualize classification outcomes, providing insights into strengths and weaknesses. Comparative analysis shows that XGBoost consistently achieves superior performance, making it the most suitable model for deployment.

Finally, the trained XGBoost model is integrated into a Google Chrome extension named PISHCATCHER. Operating at the client-side,

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

the extension extracts features from visited URLs in real time and classifies them instantly. This design ensures proactive protection against spoofing attacks without relying on server-side infrastructure, offering users immediate defense with minimal latency.

5 RESULTS AND DISCUSSION

The phishing detection system was evaluated using a dataset of over 11,000 URLs, comprising both legitimate and phishing websites. Multiple machine learning models—Decision Tree, Naïve Bayes, and SVM—were implemented as baselines, while the proposed XGBoost model was tested for enhanced performance. Comparative analysis was conducted using accuracy, precision, recall, F1-score, and confusion matrix evaluation.

	url	label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	1
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	1
2	serviciosbys.com/paypal.cgi.bin.get-into.herf...	1
3	mail.printakid.com/www.online.americanexpress...	1
4	thewhiskeydregs.com/wp-content/themes/widescre...	1
...	...	...
96002	xbox360.ign.com/objects/850/850402.html	0
96003	games.teamxbox.com/xbox-360/1860/Dead-Space/	0
96004	www.gamespot.com/xbox360/action/deadspace/	0
96005	en.wikipedia.org/wiki/Dead_Space_(video_game)	0
96006	www.angelfire.com/goth/devilmaycrytonite/	0

96007 rows × 2 columns

Fig 2: Dataset Phishing Detection

Figure 2 presents a phishing detection dataset containing 96,007 entries with two columns: "url" and "label." Each row represents a website URL, and the label indicates whether it is phishing (1) or legitimate (0). The top rows show suspicious URLs labeled as phishing, while the bottom rows contain safe URLs. This dataset is ideal for training machine learning models to classify malicious websites based on URL patterns. It supports binary classification and can be used to evaluate algorithms like Decision Tree, SVM, and XGBoost. Proper preprocessing and feature extraction from this dataset are crucial for building an effective cybersecurity solution.

Fig 3: client side spoofing detection

The figure client side spoof detection open any website in Chrome. Then click on the Extensions (puzzle icon) in the top-right corner of the browser and select your installed extension (for example, Phish Catcher). Once you click it, the extension will activate and a popup window will appear on the right side of the screen showing the website security analysis results.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

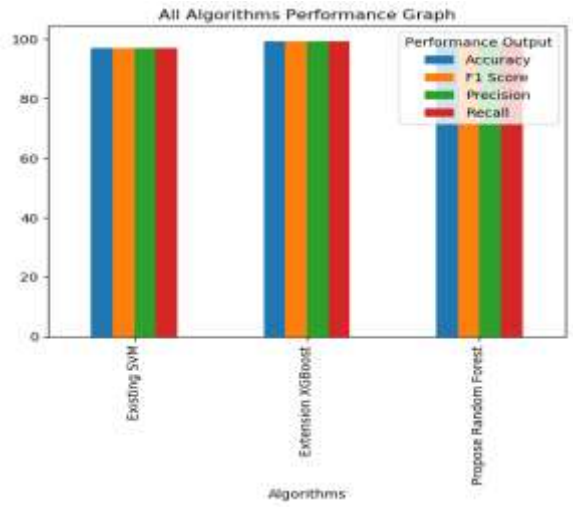

Fig 4: Performance Comparison Graph of SVM, RFR, XGBoost Classifiers

Figure 4 The performance comparison graph highlights the effectiveness of three classifiers SVM, Random Forest, and XG Boost across four key metrics: accuracy, precision, recall, and F1-score. XG Boost consistently leads with near-perfect scores, followed by Random Forest with strong results. SVM performs well but shows slightly lower precision and recall, confirming XGBoost's superiority.

6. CONCLUSION

The reserach focused on developing a robust machine learning model to effectively distinguish between legitimate and phishing URLs. Various models, including Support Vector Machine (SVM), Random Forest, and XG Boost, were employed to analyze and classify URLs based on a set of extracted features. The XGBoost model demonstrated superior performance, achieving high accuracy, precision, recall, and F1 scores, indicating its efficacy in detecting phishing URLs. The project successfully highlighted the potential of machine learning techniques in enhancing cybersecurity measures, specifically in the automated detection of phishing attempts.

Future Scope

The feature set contains thirty features, though, the addition of more automated features might be a great idea to improve the overall performance. Some other discriminative classifiers such as SVM can also be implemented for the prediction of fake or real URL by training larger data-sets. Evaluation metrics may also be evolved by using different tools for a better performance analysis.

REFERENCES

[1] W. Khan, A. Ahmad, A. Qamar, M. Kamran, and M. Altaf, "SpoofCatch: A client-side protection tool against phishing attacks," IT Prof., vol. 23, no. 2, pp. 65-74, Mar. 2021.

[2] B. Schneier, "Two-factor authentication: Too little too late," Commun. ACM, vol. 48, no. 4, pp. 136, Apr. 2005.

[3] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

measurement of phishing attacks," Proc. ACM Workshop Recurring malcode, pp. 1-8, Nov. 2007.

[4] R. Oppliger and S. Gajek, "Effective protection against phishing and web spoofing," Proc. IFIP Int. Conf. Commun. Multimedia Secur., pp. 32-41, 2005.

[5] T. Pietraszek and C. V. Berghe, "Defending against injection attacks through context-sensitive string evaluation," Proc. Int. Workshop Recent Adv. Intrusion Detection, pp. 124-145, 2005.

[6] M. Johns, B. Braun, M. Schrank, and J. Posegga, "Reliable protection against session fixation attacks," Proc. ACM Symp. Appl. Comput., pp. 1531-1537, 2011.

[7] M. Bugliesi, S. Calzavara, R. Focardi, and W. Khan, "Automatic and robust client-side protection for cookie-based sessions," Proc. Int. Symp. Eng. Secure Softw. Syst., pp. 161-178, 2014.

[8] A. Herzberg and A. Gbara, "Protecting (even naive) web users from spoofing and phishing attacks," 2004.

[9] N. Chou, R. Ledesma, Y. Teraguchi, and J. Mitchell, "Client-side defense against web-based identity theft," Proc. NDSS, 2004.

[10] B. Hämmerli and R. Sommer, "Detection of Intrusions and Malware and Vulnerability Assessment: 4th International Conference DIMVA 2007 Lucerne Switzerland July12-132007 Proceedings," vol. 4579, 2007.