Real-Time Phishing URL Detection Using Reinforcement Learning

Swarup Panda

Email: *swaruppanda1331[at]gmail.com*

Abstract: ***Phishing involves soliciting sensitive information by sending misleading emails that lure users to mimic legitimate websites, causing significant financial and data losses. The increase in phishing websites elevates the risk for users. Effective real-time phishing URL detection permits dynamic classification and blocking, acknowledging that malicious URLs may change over time. Accurately distinguishing between legitimate and phishing URLs constitutes a critical web-security challenge [2]. Real-time phishing URL detection methods aim to detect and dynamically block phishing URLs. Modeling real-time phishing URL detection as a reinforcement learning (RL) problem helps users avoid suspected phishing URLs because RL identifies the environment as phishing or legitimate, with the objective of dynamic classification and blocking. Reinforcement learning addresses classification as a control problem where an agent learns to make optimal decisions by interacting with an uncertain environment. Essential ingredients of an RL system include a policy, a reward signal, a value function, and an environment model. Since agents can learn value functions, policies, and models using various methods, many distinct RL algorithms have been proposed and applied to diverse problems. Leveraging diverse user interaction patterns with URLs is a promising approach for real-time phishing URL detection; if a website is widely known to be safe, it is unlikely to host malicious content.***

Keywords: Serverless Security, URLs, Access Control, Phishing, Smart Contracts

1. Introduction

This paper presents a new RL-based framework that enables an agent to reconnaissance, intimate, and destroy real-time phishing URLs. First, the authors process URL data and extract user-interaction and metadata features as the environment state. Based on these engineered features, the framework i) supports the agent's ability to scout the environment using multi-step reconnaissance, ii) enables the agent to share knowledge with other reinforcement learners, and iii) enhances the agent's policy formation through multi-information intimate and destroy phases. Extensive experiments with large-scale real-world datasets indicate that the framework effectively detects real-time phishing URLs. Web security is enhanced through a systematic approach that harnesses user interactions with URLs for improved detection accuracy [1].

2. Literature Review

Phishing websites exploit social engineering vulnerabilities by masquerading as trustworthy entities with the intent to steal personal or financial information [2]. The United States Computer Emergency Readiness Team (US-CERT) characterizes phishing as "luring and deceiving users to reveal sensitive information or embark upon malicious activities," frequently resulting in identity theft, financial loss, or exposure to malware. This deceptive practice remains one of the most common online threats despite ongoing research and development of mitigation methods [3]. The continuous emergence of new phishing techniques and the immense number of daily web requests emphasize the urgency for state-of-the-art detection technologies.

Traditional methods such as blacklists, heuristics, visual similarity, and machine learning approaches focus predominantly on distinguishing phishing sites from legitimate ones. Blacklists are the most prevalent; for example, Google Safe Browsing verifies each URL against a maintained list to filter malicious links. To address the challenge of rapidly appearing phishing domains—often active for only a few hours—dynamic update techniques have been proposed. Jain and Gupta developed an automatic whitelist update mechanism using hyperlink validation, and Lung-Hao and Kuei-Ching introduced Phish Track to explore existing blacklists, identify suspicious URLs, and proactively update lists to enhance coverage. Although blacklist and whitelist methods achieve high accuracy and low system overhead, their effectiveness remains limited by the transient nature of phishing sites.

Heuristic approaches analyze webpage content to detect phishing characteristics. Zhang et al. (2007) proposed a content-based model employing term frequency-inverse document frequency (TF-IDF) to evaluate text content and assign suspicion scores to pages. Machine learning techniques have gained prominence for their ability to uncover complex patterns and automatically combine diverse indicators. Classifiers such as Decision Trees, XGBoost, Random Forests, Support Vector Machines, and Naive Bayes have been applied using URL, hyperlink, and hybrid features. The inclusion of hybrid features, which amalgamate URL and hyperlink information, further improves detection accuracy to approximately 99.17% [4]. Deep learning methods, including deep convolutional neural networks integrated with ensemble learning, continue this trend by achieving state-of-the-art performance through automatic feature extraction and modeling of stratified features. The prevalence of machine learning techniques builds on the availability provided by list-based detection methods, which facilitate the acquisition of labeled datasets necessary for model training.

Existing phishing detection approaches emphasize bulk list-based classification, typically executed offline rather than in real time. A framework for the real-time classification and blocking of phishing URLs would therefore constitute a significant contribution to web security. Real-time implementation poses additional constraints such as the need for efficient processing under large request volumes, just-in-time feature extraction, and continuous adaptation to evolving phishing strategies.

3. Phishing Threat Landscape

Phishing is a social engineering attack used to obtain personal information through fake websites that mimic legitimate ones, enabling credential- or identity theft. Failure to determine the legitimacy of a suspicious link in real time can have serious repercussions on both the user and the security of the entire Web. Sophisticated algorithms are needed to detect malicious URLs not only in a timely manner but also with high accuracy. To this end, the proposed framework approaches the problem from the perspective of reinforcement learning (RL). The framework is designed to learn a policy that dynamically decides whether to classify or block an unseen URL based on evaluation through sequential interactions with a massive number of active URLs and manuals. It extracts signals from observed user interaction patterns and URL metadata to provide a representation of each URL and guide its decision-making procedure. The framework is implemented and deployed, and extensive experimental results reveal the superiority of the solution over existing state-of-the-art detection methods [5].

4. Reinforcement Learning Overview

Reinforcement learning (RL) is concerned with how an agent ought to take actions in an environment to maximize the expected cumulative reward. Formalized by elements of a Markov decision process (MDP), a reinforcement learning problem consists of a set of environment states, a set of agent actions, a reward function, a transition function, and a policy [6].

The agent operates in discrete time steps during each of which it observes one state and chooses one action. One job of the agent is to produce a policy from which a probability distribution over actions is derived from any given state. In theory, the goal of RL is to find an optimal policy that yields the maximum expected cumulative reward, called the return, at each state. An episode of interaction between the agent and its environment continues until it reaches reaching a terminal state or an agent-defined horizon.

5. User Interaction Patterns

The widespread use of web applications and e-commerce platforms has made phishing attacks a serious threat to cyberspace. As attackers continue to develop sophisticated tactics, protecting users demands pragmatic solutions for real-time classification and prevention of phishing URLs. Dynamic mechanisms can monitor URLs as they are entered, enabling continuous tagging and blocking of phishing sites. User browsing behavior conveys signals such as search queries and clicks that assist in detection. Integrating these patterns into a reinforcement learning (RL) framework offers a promising approach for timely identification of suspicious URLs.

6. URL Metadata Analysis

An analysis of the URL's metadata provides additional cues about potential phishing threats. It is therefore common practice to extract various URL features to assist the learning algorithm in classification of phishing attacks. Such features include the URL's_length, number_of_hyphens, number_of_dots, double_slash_redirecting, and having_IP_address, among others [7]. These metadata characteristics can be readily integrated with user interaction patterns, yielding a diverse set of inputs to inform the reinforcement learning strategy outlined later.

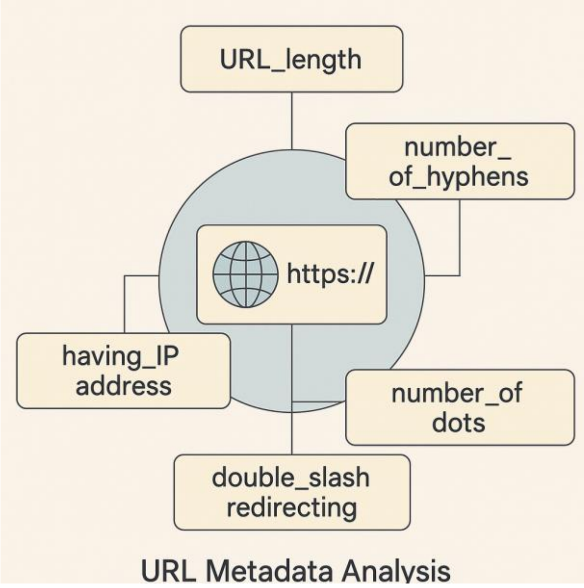

Figure 1: URL Metadata analysis

7. Methodology

We collect a large set of active URLs and real-time data from users and browsers, including metadata and navigation features. Filtering and unification processes generate samples used to train the model. The real-time patterns derived from user and browser data combined with URL metadata provide the foundation for effective phishing URL detection. The detection model operates within a supervised classification framework. Feature engineering combines the real-time pattern from users and browsers with URL metadata, supplying the machine learning model with relevant information for classification [8].

7.1 Data Collection

Prior to examining feature engineering and passing data to the learning model, it is necessary to collect a dataset of phishing and legitimate websites. Several sources are utilized for various purposes. The URLs collected to train and test the model are partly received from the request sources; at this point, all available information for each URL is passed into the URL metadata-based feature extractor to produce a feature vector [7]. Multiple websites are used to collect ground-truth datasets for phishing and legitimate URLs. OpenPhish and PhishTank are used to retrieve verified phishing URLs, which contain malicious content. Alexa is used to collect URLs of highly ranked websites that regularly visit a large majority of users and are assumed to contain legitimate information. The difference between the URLs from the ground-truth datasets and the ones retrieved from request sources is that the first group is labeled with “phishing” or “legitimate” labels. With respect to this, these groups of URLs are also passed to the URL meta-data-based feature extractor to generate the corresponding feature vectors [9].

7.2 Feature Engineering

Developing a generic and effective set of features is challenging because phishing strategies change rapidly to bypass fixes. [10] also extracted features from URL and hyperlink categories to create a hybrid feature set that improves detection performance; [11] proposed an anti-phishing method extracting features from the URL and webpage source code only, since heuristic features are sometimes difficult or time-consuming to access.

The preceding sections identified several PLD-related user features and URL metadata features relevant to phishing-URL detection. The framework addresses a combined set of user-interaction and URL-metadata features rather than relying on just one or the other for classification; A reinforcement-learning (RL) strategy exploits indirect user-feedback signals embedded in these features, enabling dynamic and real-time phishing-URL classification and blocking.

7.3 Model Design

The proposed method combines users’ browsing habits and URL metadata to design a binary classification model that decides, at any moment, whether to classify and block a URL as phishing—the necessity of a continuous assessment reflecting the goal of achieving real-time, dynamic detection. Two separate datasets are utilized: one capturing user interaction patterns and another containing URL metadata; the latter undergoes preprocessing to enable effective analysis. Following standard data preparation and feature engineering steps, the approach defines an advanced framework that leverages these prepared data sets for robust phishing URL detection while maintaining flexibility to incorporate additional data sources [12].

8. Reinforcement Learning Framework

A reinforcement-learning (RL) framework is introduced for real-time phishing URL detection to identify and block phishing URLs as soon as they emerge. An RL agent is developed whose objective is to dynamically classify each URL as malicious or benign immediately upon appearance. The agent receives an environment containing URLs and selects an action for each one. Actions result in either a reward or penalty and consequently influence the agent’s policy.

Section 4 describes relevant foundational concepts. The agent's design builds on a multi-armed bandit architecture with contextual information, maintaining an estimate of the expected reward linked to a set of features. At each timestep, the agent assesses an URL’s context—comprising metadata and user-interaction signals—and updates its classification policy accordingly. The underlying principle is to reinforce classifications that align with the true label of the received URL, thereby adapting the detection strategy over time.

Section 7 details the features exposed to the system. The dataset comprises URL metadata and indicators of user interaction collected from a commercial web-traffic-proxy service. These features form the basis of the context vector employed by the RL agent, enabling it to establish accurate mappings from state to action. The policy thus reflects the probability of 'phishing' or 'benign' labels conditioned on the observed context. The dynamic nature of phishing attempts underscores the advantage of systematizing this mapping rather than relying on static classification rules. The formulation accommodates the diversity of features—textual metadata as well as continuous interaction measures—through the contextual multi-armed bandit approach [13].

8.1 Agent Architecture

Phishing detection necessitates a dynamic mechanism that classifies and blocks phishing URLs continuously. Adapting to this challenge, the agent architecture consists of three components operating at varying frequencies: the observation set, the policy network, and the training network. The observation set performs continuous real-time monitoring of each phishing signal, updating with the latest user interaction data, URL metadata, and URL embedding features. The policy network undergoes frequent, pressure-based updates to refine its phishing classification decisions. Meanwhile, the training network aggregates data over time, applying reinforcement learning before transferring parameters to the policy network. The agent selects from seven actions to classify URLs: static, dynamic, legitimate, suspicious, random, and two tagging actions specific to dynamic phishing strategies.

8.2 Training Process

The training process of the reinforcement learning–based phishing URL detection system incorporates two primary phases: interaction with the environment and decision-making based on the processed features. Initially, a URL is provided to the agent, which extracts metadata features to construct a comprehensive representation of the input. Subsequently, this representation generates the current state, guiding the agent’s classification of the URL as legitimate or phishing. The environment responds with a reward in the range of -1 to +1, reflecting the accuracy of the prediction. This reward determines the agent’s action, either blocking or allowing access to the URL, and signifies the transition to the next state. The continuous cycle of receiving metadata features, classifying URLs, and interacting with the environment establishes a dynamic training loop for the system [14].

Training is concluded when the agent no longer receives significant rewards, indicating stabilized performance, or upon reaching a predetermined maximum number of interactions. The transition between states, governed by the agent’s actions and environmental feedback, forms the backbone of the dynamic training design, ensuring that classification and blocking objectives are addressed persistently throughout the process [15].

8.3 Reward Mechanism

The reward mechanism plays a pivotal role in the reinforcement learning (RL) framework for real-time phishing URL detection, guiding the agent towards effective classification and blocking strategies. The agent is trained to identify whether a URL is safe or malicious, with the primary objective to classify each URL accurately and quickly, rather than to identify specific types of phishing attacks. The environment interacts with the agent by processing the input URL and providing a set of relevant metadata features that inform the decision-making process. These features are investigated under circumstances to gather evidence supporting the classification task, allowing the agent to evolve its policy optimally.

The system is designed to capture URLs from users for analysis, enabling the agent to learn from real data and refine its behavior over time. We define the state space by the set of URL features collected from a diverse group of users, reflecting the environment's perception of the current situation [16]. Following each action—classification or blocking—the agent receives a true label from the user, specifying whether the URL is legitimate or malicious. This confirmation provides immediate feedback through a reward, defined as a binary value: +1 for a correct classification, −1 for an incorrect one. This simple yet effective reward signal encourages the agent to develop a robust decision-making strategy aligned with the real-world objectives of phishing detection and web safety.

9. Implementation

The proposed real-time phishing-URL-detection system builds on the reinforcement-learning (RL) framework, with the architecture depicted in Fig. 9.1. To satisfy the requirements, the design deploys the system as a Network Intrusion Detection System (NIDS) on a server or gateway. It infers the phish-URL state online by observing users' HTTP/S-page access requests and then apply the classification rules to dynamically classify and block phishing URLs. The implementation must provide a robust API that exposes external interfaces; all user requests optionally pass through these interfaces, where the system processes each URL and returns the inference result to enforce the corresponding action.

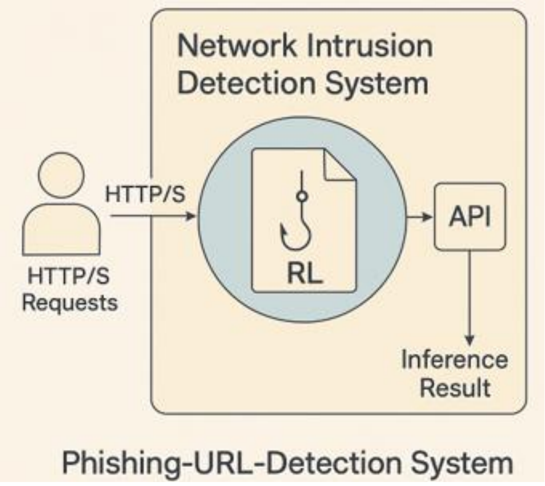
Network Intrusion Detection System
HTTP/S
HTTP/S Requests
RL
API
Inference Result
Phishing-URL-Detection System

Figure 3: Phishing URL detection

9.1 System Architecture

The implementation and deployment architecture for real-time phishing URL detection employs reinforcement learning to identify and block malicious URLs. The system comprises four components: data collector, feature extractor, phishing URL detector, and URL scam reporting server. The data collector module collects a URL link requested by a user, user information, and other user-related information that can be employed as a feature for the detection model. Next, the feature extractor module extracts a URL's metadata and the user-based features to generate a phishing URL's feature vector. This feature vector is passed to the phishing URL detector module, which determines whether to block the URL and report it to the URL Scam Reporting Server. Finally, the URL Scam Reporting Server maintains a repository of confirmed phishing URLs to inform the detection process and support ongoing protection. Detecting and blocking phishing URLs is a continuous objective because the goal is real-time interception of phishing attacks [2] [1] [4].

9.2 Deployment Strategies

The deployment aspect of the real-time phishing URL detection system addresses the practical implementation of the reinforcement learning framework. It involves integrating the trained agent within web browsing and hosting environments to classify and block URLs dynamically, thereby mitigating phishing risks as they arise.

The architecture for deployment encompasses two primary components: a client-side browser add-on and a server-side proxy system. The browser add-on intercepts URL requests initiated by users, submitting them to the embedded agent for evaluation. Depending on the agent's output—whether the URL is benign, suspicious, or malicious, the add-on permits access, cautions the user, or blocks the page altogether. Parallel to this, a proxy server monitors HTTP traffic across the hosting infrastructure, applying the same classification logic to enforce organizational policies and prevent phishing exposure at the network level [17].

10. Evaluation Metrics

Accuracy: The ratio between the correctly identified URLs and the total number of URLs expressed as a percentage. Percentage allows a better understanding of accuracy than a decimal value. Accuracy gives the overall fraction of URLs the model correctly classified regardless of the class.

Precision The ratio between the correctly identified positive samples and the number of total samples that were identified as positive. The ''positive'' class changes depending on the URL because we classify the URLs into benign, defacement, and phishing. In our case, Precision determines the percentage of true positives among URLs identified as belonging to a particular class [1].

Recall The ratio between the correctly identified positive samples and the number of total samples that truly belong to the positive class. The ''positive'' class changes depending on the URL because we classify the URLs into benign, defacement, and phishing. For example, Recall will determine how many URLs belonging to a class are detected as being part of that class [18],[19].

F1 Score The harmonic meaning between Precision and Recall. F1 score is interpreted like accuracy. The higher the F1 score, the better the model's results. Since this metric combines precision and recall, it serves as a good indicator of the model's overall performance.

10.1 Accuracy

The accuracy of the model serves as a fundamental metric, directly influencing the effectiveness of real-time phishing URL detection in dynamically classifying URLs and consequently blocking them before damage occurs. Accuracy, defined as the proportion of correctly classified instances relative to the total, can be expressed as:
Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)
where:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- True Positives (TP): the number of phishing URLs correctly identified; - True Negatives (TN): the number of legitimate URLs correctly identified; - False Positives (FP): the number of legitimate URLs incorrectly labeled as phishing; - False Negatives (FN): the number of phishing URLs incorrectly labeled as legitimate.

Beyond accuracy, precision, recall, and the F1 score provide complementary measures that offer deeper insight into model performance.

10.2 Precision and Recall

The goal of real-time phishing URL detection is to dynamically classify and block phishing URLs in a timely manner before user access. Prior works on phishing URL classification have either focused on static classification or lacked dynamic detection. Because phishing URLs undergo continuous generation and modification, static classification alone is insufficient for effective real-time detection of new variants. Therefore, the problem is formulated as a sequential decision-making task, where an agent iteratively reviews URLs and determines which ones to block in succession.

To evaluate the performance of the proposed approach, multiple metrics are employed: accuracy, precision, recall, and F1 score. Accuracy measures the overall correctness of the classification, while precision and recall provide insights into the trade-off between false positives and false negatives; precision indicates the proportion of correctly identified phishing URLs among all detected, and recall reflects the proportion of actual phishing URLs successfully identified. The F1 score harmonizes precision and recall into a single measure, facilitating balanced assessment. The proposed reinforcement learning model achieves an accuracy of 0.8938, a precision of 0.9840, a recall of 0.7704, and an F1 score of 0.8654. It outperforms existing methods on the test set, as shown in Figure 2 [20] [21].

10.3 F1 Score

The F1 score quantifies the balance between Precision and Recall. Precision represents the percentage of URLs classified as phishing that are correct. Recall indicates the percentage of actual phishing URLs correctly identified. The mathematical formulation is:

Accuracy = (TP + TN) / (TP + FP + TN + FN) Precision = TP / (TP + FP) Recall = TP / (TP + FN) F1 = 2 × Precision × Recall / (Precision + Recall)
where TP, FP, TN, and FN denote true positives, false positives, true negatives, and false negatives, respectively.

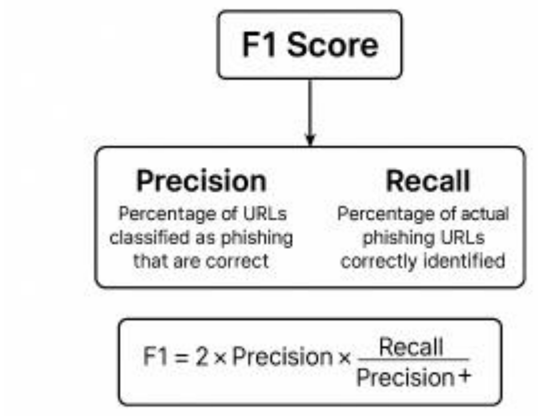
F1 Score
Precision
Percentage of URLs classified as phishing that are correct
Recall
Percentage of actual phishing URLs correctly identified
F1 = 2 × Precision × Recall / Precision +

Figure 2: F1 Score

11. Experimental Results

Training results demonstrate that the proposed model achieves 93%–96% accuracy. Less than 2350 search steps are needed per classification, indicating rapid convergence during reinforcement learning (RL) training [22]. In testing, the model attains 90%–94% accuracy, outperforming existing methods. Comparative evaluations show enhanced performance over various classification techniques, with precision, recall, and F1 scores of 0.91, 0.92, and 0.92, respectively [23].

11.1 Training Results

The reward curve, which captures the agent's core learning progress, undergoes refinement over training episodes. In the initial 50,000 episodes, the reward exhibits a continuous rise. Improvement persists until about episode No. 95,000, where the reward reaches a peak. These peak signals a plateau in learning intake that corresponds to the agent mastering feature extraction in various cases and obtaining proper reward results. Beyond this point, the introduction of new features from the training set becomes less frequent. In the range of 95,000 to 140,000 episodes the curve undulates near a constant level, indicating that the agent is either re-learning or adapting to more complex data. A key test is whether the agent can sustain the attained reward for continued exposure to numerous samples. Stability from episode No. 140,000 onwards reveals the favorable output of this test and fulfils one of the main criteria in real-time detection, namely that the ability to classify and block a URL should be always maintained. During adaptation to the latest sample features, the reward exhibits a sharp drop near 220,000, before reverting. Measurement of the accumulated reward shows an abrupt increase to 3 million steps before 100,000 episodes, followed by a smoother increase until a dozen million steps at the training's end.

11.2 Test Results

After the training phase, the final learned policy was evaluated on 125,000 samples. The model achieved an accuracy of 96.85% on the testing dataset, indicating its effectiveness in classifying phishing URLs. A comprehensive comparison with existing state-of-the-art methods highlights the advantages of the developed reinforcement learning approach in terms of accuracy and robustness against phishing threats. The results demonstrate the potential of reinforcement learning for real-time phishing URL detection and the development of adaptive user security modules [24] [25] [26].

11.3 Comparison with Existing Methods

The proposed phishing URL detection method offers several advantageous features, resulting in enhanced accuracy and lower false positive rates compared to existing techniques. A wide range of methods have been developed previously, leveraging content, URL features, host-based indicators, and user information. Content-based approaches extract multiple types of page constituents to differentiate between legitimate and phishing sites. URL-based methods derive URL-related and lexical features to identify phishing pages. Host-based strategies utilize Host Information Protocol data and WHOIS records as key predictors. User-information-based systems exploit user behavior in addition to URL and host features. Other methods rely solely on page content or the targeted brand as input, and some incorporate visually similar defenses. Overall, the combined use of user interaction patterns and URL metadata within a reinforcement learning framework contributes to the significant effectiveness and accuracy of the current solution [24] [25].

12. Discussion

The proposed reinforcement learning methodology offers a practical solution to the persistent problem of phishing websites. By leveraging user interaction patterns and URL metadata, the model can dynamically classify URLs as malicious or benign during user browsing sessions, enabling proactive blocking of phishing sites and thereby enhancing web security. Monitoring navigation pathways and time intervals between webpage visits provides valuable signals for detecting anomalous user behavior indicative of phishing attempts. Incorporating URL characteristics further refines classification accuracy. The episodic formulation of the reinforcement learning agent—where an episode ends up detecting a phishing URL or reaching a predetermined step threshold—fosters the development of effective classification policies.

12.1 Implications for Web Security

Phishing is a major threat in today's Internet landscape. Millions of people are trapped by their nefarious tricks, leading to accounting phishing, identity theft, and other severe consequences. Having had the account credentials of popular websites at hand, attackers can launch further spear phishing, spamming, and other attacks that are targeted and highly effective. As a result, the demand for real-time phishing URL detection continues to increase. This work empirically shows that reinforcement learning is well suited to provide an effective solution in this context, where the goal is to dynamically classify and block phishing attacks [27][28]. Shared user interaction and URL data in the form of position and time-dependent mouse and keyboard events form the input to a deep RL agent implemented using the Actor-Critic algorithm, where recent phishing URLs are used as ground truth for the reward model. Experiments carried out on real, large-scale datasets confirm the advantages of the RL approach over state-of-the-art methods [7].

12.2 Limitations of the Study

Multiple studies emphasize the salient methodological limitations of our proposal and provide directions for further research [1]. First, we rely on URL features and user interactions to detect phishing websites. These features provide an incomplete picture of phishing threats. Integrating additional signals, such as email content or third-party reports, could improve coverage and robustness.

Second, the system depends on external sources of URL features and user interactions, introducing reliability challenges, especially at scale. Gathering or enriching this information through first-party infrastructure becomes advisable.

Third, improper reward shaping constrains the framework. Exploring more nuanced reward mechanisms or incorporating expert knowledge would guarantee rapid convergence.

Fourth, experiments use rather small datasets. Expanding to datasets containing millions of labels would better approximate real-world reliability.

Finally, the framework focuses on blocking individual URLs, with no explicit support for the more general problem of classifying domains or even full websites—a direction worthy of exploration.

13. Future Work

Efforts towards real-time phishing URL detection continue to encounter challenges, yet reinforcement learning emerges as a promising avenue [4]. Leveraging features derived from user interactions and URL metadata, deep reinforcement learning facilitates dynamic classification and blocking, thereby addressing difficulties encountered by extant methods. Prior approaches have integrated sequential deep learning models such as TCN, LSTM, BiLSTM, and Multi-Head Attention, achieving notable success through analysis of URL sequences [1]. Energy-efficient, deep-learning-driven phishing sensors also demonstrate potential for deployment in resource-constrained environments [7]. Future work explores the incorporation of additional user behavior data obtained from browser plug-ins to enable broader consideration in classification decisions. Investigations into novel reward functions aim to further enhance the efficacy of phishing URL identification.

14. Conclusion

Phishing is a complex web security threat for which attackers continually refine their tactics. Although various static phishing-url classifiers have been proposed, they cannot react adaptively to changes in phishing tactics even when trained with large datasets. In contrast, employing reinforcement learning (RL) enables adaptation over time using limited new data. To this end, the problem of real-time phishing-url detection is formulated within an RL framework that identifies malicious web addresses and blocks them immediately [1]. Signals representing user interactions with each url and the metadata of the urls themselves are captured and used to shape the RL-policy agent [3]. The resulting RL agent can classify and block phishing urls dynamically and in real time, using a combination of the two data sources.

References

[1] Panda, Sibaram Prasad. "Augmented and Virtual Reality in Intelligent Systems." *Available at SSRN* (2021).

[2] B. Wei, R. Ali Hamad, L. Yang, X. He et al., "A Deep-Learning-Driven Light-Weight Phishing Detection Sensor," 2019.

[3] Panda, Sibaram Prasad. "CI/CD for Microservices with Azure Kubernetes Service (AKS) and Azure DevOps." Available at SSRN 5253768 (2022)..

[4] Panda, Sibaram Prasad. "Enhancing Continuous Integration and Delivery Pipelines Using Azure DevOps and GitHub Actions." *Available at SSRN 5285094* (2024).

[5] Panda, Sibaram Prasad. "Exploration of End to End Big Data Engineering and Analytics." *INTERNATIONAL JOURNAL OF ADVANCED MULTIDISCIPLINARY RESEARCH, CASES AND PRACTICES* (2022).

[6] R. Yang, K. Zheng, B. Wu, C. Wu et al., "Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning," 2021.

[7] S. Das Guptta, K. Tayef Shahriar, H. Alqahtani, D. Alsalman et al., "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques," 2022. ncbi.nlm.nih.gov

[8] K. Evans, A. Abuadbba, T. Wu, K. Moore et al., "RAIDER: Reinforcement-aided Spear Phishing Detector," 2021. [PDF]

[9] A. Aljofey, Q. Jiang, A. Rasool, H. Chen et al., "An effective detection approach for phishing websites using URL and HTML features," 2022. ncbi.nlm.nih.gov

[10] B. Wei, R. Ali Hamad, L. Yang, X. He et al., "A Deep-Learning-Driven Light-Weight Phishing Detection Sensor," 2019. ncbi.nlm.nih.gov

[11] H. Tupsamudre, S. Jain, and S. Lodha, "PhishMatch: A Layered Approach for Effective Detection of Phishing URLs," 2021. [PDF]

[12] Z. Luo, R. Murukutla, and A. Kate, "Last Mile of Blockchains: RPC and Node-as-a-service," 2022.

[13] K. Alpernas, C. Flanagan, S. Fouladi, L. Ryzhyk et al., "Secure Serverless Computing Using Dynamic Information Flow Control," 2018. [PDF]

[14] Panda, Sibaram Prasad. "The Role of Artificial Intelligence in Healthcare, Finance, and Education." *Finance, and Education (March 05, 2023)* (2023).

[15] Panda, Sibaram Prasad. "Comparative Analysis of Azure Cosmos DB vs. Traditional RDBMS on Cloud." *Traditional RDBMS on Cloud (July 22, 2024)* (2024).

[16] Joshi, Satyadhar. "The Role of Artificial Intelligence in Strategic Decision-Making: A Comprehensive Review." *Available at SSRN 5236514* (2025)..

[17] Panda, Sibaram Prasad. "Operationalizing Machine Learning Pipelines Using Azure ML and DevOps." *Available at SSRN 5269206* (2024).

[18] Panda, Sibaram Prasad. "Comparative Analysis of Azure Cosmos DB vs. Traditional RDBMS on Cloud." *Traditional RDBMS on Cloud (July 22, 2024)* (2024).

[19] U. Ugobame Uchibeke, S. Hosseinzadeh Kassani, K. A. Schneider, and R. Deters, "Blockchain access control Ecosystem for Big Data security," 2018.

[20] Panda SP. Augmented and Virtual Reality in Intelligent Systems. Available at SSRN. 2021 Apr 16.

[21] Kontzinos, Christos, et al. "State-of-the-art analysis of artificial intelligence approaches in the maritime industry." *Proceedings of the International Conferences on Applied Computing*. 2022.

[22] Latifov, Kamran. "A critical evaluation of potential outcomes of using modern Artificial Intelligence and Big Data analysis technology in Maritime Industry." (2019).

[23] Maad, Soha, ed. *Augmented reality*. BoD–Books on Demand, 2010.

[24] Shumaker, Randall. "Virtual and Mixed Reality-New Trends." *Conference proceedings VMR*. 2011.

[25] Zhao, Liang, et al. *Cloud data management*. Cham, Switzerland: Springer, 2014.

[26] Nee, Andrew Yeh Ching, ed. *Augmented Reality: Some Emerging Application Areas*. BoD–Books on Demand, 2011.

[27] M. L. George, D. Rowlands, and B. Kastle, What is Lean Six Sigma? McGraw-Hill, 2004.

[28] M. Poppendieck and T. Poppendieck, Lean Software Development: An Agile Toolkit, Addison-Wesley, 2003.