Phishing Detection in Web Domains: new intelligent tool leveraging the effectiveness of emerging Generative models

Carmine Ambrosino*
c.ambrosino11@studenti.unisa.it
University of Salerno
Fisciano, Italy

Maurizio Atzori*
atzori@unica.it
University of Cagliari
Cagliari, Italy

Stefano Cirillo*
scirillo@unisa.it
University of Salerno
Fisciano, Italy

Domenico Desiato*
domenico.desiato@uniba.it
University of Bari
Bari, Italy

Simona Ettari*
s.ettari@studenti.unisa.it
University of Salerno
Fisciano, Italy

Giuseppe Polese*
gpolese@unisa.it
University of Salerno
Fisciano, Italy

Giandomenico Solimando*
gsolimando@unisa.it
University of Salerno
Fisciano, Italy

Abstract

The rapid growth of online services has heightened concerns about user protection from cyber threats, particularly phishing, which poses significant risks to cyber-social security. To this end, we propose a novel tool for phishing detection called U-Proof. Our tool uses both state-of-the-art LLMs and traditional ML models to detect phishing websites. In particular, we evaluate the phishing detection capabilities of different LLMs and compare them with several ML models to analyze the impact of different model architectures on the identification of phishing websites. For a comprehensive experimental evaluation, we use a combination of public and custom datasets. These include active phishing websites from September 2024, as well as URLs from banks and postal services. Furthermore, the tool includes explanations to enhance user awareness of phishing tactics, supporting broader educational efforts to reduce risks.

CCS Concepts

• **Security and privacy** → **Web application security**; Human and societal aspects of security and privacy; • **Computing methodologies** → *Natural language processing*.

Keywords

Phishing Detection, Machine Learning, Large Language Models

ACM Reference Format:
Carmine Ambrosino, Maurizio Atzori, Stefano Cirillo, Domenico Desiato, Simona Ettari, Giuseppe Polese, and Giandomenico Solimando. 2026. Phishing Detection in Web Domains: new intelligent tool leveraging the effectiveness of emerging Generative models. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining (WSDM '26), February 22–26, 2026, Boise, ID, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3773966.3779407

*These authors contributed equally.

WSDM '26, Boise, ID, USA

ACM ISBN 979-8-4007-2292-9/2026/02
https://doi.org/10.1145/3773966.3779407

1 Introduction

Motivation. Traditional phishing detection methods, mainly based on rule-based systems or conventional machine learning models, have only partially mitigated phishing attacks [7]. These approaches typically focus on known patterns, such as suspicious URLs, domain anomalies, or email characteristics [11], but struggle to keep pace with evolving phishing techniques. Modern attacks increasingly exploit data from social networks and non-anonymized sensitive information to craft targeted campaigns [4, 5]. The dynamic and adaptive nature of phishing attacks requires more flexible and intelligent solutions to known threats and emerging sophisticated phishing strategies. In response to this, attention has shifted toward technologies relying on generative AIs, particularly Large Language Models (LLMs). The latter showed their ability to understand human-like text, offering new possibilities for detecting phishing attacks by analyzing content at a semantic level, which could provide deeper insights into the context of potential attacks.

Contributions. Our study introduces U-Proof, a novel tool that combines the strengths of LLMs with traditional machine learning (ML) models to identify phishing attacks. In particular, by integrating these two approaches, U-Proof aims to detect phishing websites with improved speed, accuracy, and adaptability. LLMs, known for their contextual understanding and ability to process complex language patterns, can complement traditional ML models that, on the other hand, work on structural features, providing a more comprehensive solution for phishing detection. Through this combined approach, we explore the potential of LLMs not only as core components capable of improving the identification of phishing sites through semantic and contextual analysis. To guide the design of the U-Proof, we try to answer the following two RQs with an extensive evaluation on real phishing domains: *RQ1: Can we trust LLMs to detect phishing sites? RQ2: How do traditional ML models compare to LLMs for phishing detection?*

Related Work. Several studies have evaluated ML-based phishing detection from URL content. Random Forest often outperforms more complex models, achieving accuracies around 97% [12], while ensemble methods such as XGBoost and LightGBM reach about 96% on large-scale datasets [9]. Frameworks combining URL, HTML, and derived features, as well as optimized stacking and voting ensembles, consistently exceed 97% accuracy and can reach up to

99.79% [6, 10]. More advanced DL approaches, including CNNs and RNN/LSTMs, report up to 94% accuracy for phishing detection [3], while models applied to email phishing, including BERT-based architectures, achieve up to 99.61% [1]. Overall, ML and DL methods are effective for security [2] and phishing detection, but require large, continuously updated datasets and may suffer from limited generalization under biased or scarce data.

2 Identifying Malicious Domains Combining LLMs with Traditional Predictive Models

The problem of identifying phishing domains can thus be seen as a classification task, where the goal is to predict whether a given domain is malicious or not. More formally, let us consider a set of $U = \{v_1, v_2, \ldots, v_n\}$, where each v_i represents domains presented on the web. The problem of identifying a malicious domain requires associating with each URL v a type $h \in \{0, 1\}$, where 1 represents the value for a malicious domain and 0 otherwise. Starting from this, we can define the set $\Gamma = \{(v, h) \mid v \in U, h \in \{0, 1\}\}$ containing all the domains to which an h value is associated. The set Γ contains all the legitimism and phishing domains, i.e., $\Gamma = \Gamma_L \cup \Gamma_P$ with $\Gamma_L = \{(v, h) \mid v \in \Gamma, h = 0\}$ and $\Gamma_P = \{(v, h) \mid v \in \Gamma, h = 1\}$.

Starting from this, we aim to define an intelligent tool specifically tailored to discern malicious domains within online platforms, based on a response obtained by a machine learning classifier Ψ, and two generative models denoted as Λ_1 and Λ_2, which are combined through an ensemble decision mechanism.

Given a domain $v \in U$, the ML classifier Ψ produces a binary prediction $\Psi(v) \in \{0, 1\}$ based on URL-level and page-level features, while each generative model Λ_i, independently analyzes the same input and returns a classification outcome $\Lambda_i(v) \in \{0, 1\}$, derived from its output. Since each LLM typically returns a textual response, it is forced via the prompt to return a structured answer that contains the decision about the URL, the motivation, and a probability score associated with the decision. The structured answer is then analyzed to extract the decision of each LLM. The final decision function $\Phi(v)$ is defined according to a majority voting strategy over the three models, such that $\Phi(v) = 1$ iff $\Psi(v)+\Lambda_1(v)+\Lambda_2(v) \geq 2, 0$ otherwise. Accordingly, a URL is classified as phishing if at least two of the three models agree on the malicious label, and legitimate otherwise.

3 Dataset Overview

The *PhiUSIIL Phishing URL* Dataset [10] provides critical insights into distinguishing phishing URLs from legitimate ones through its comprehensive features, such as URL length and domain characteristics. It contains 134,847 phishing URLs and 99,722 legitimate URLs, each with 56 associated categorical and numeric features.

Figure 1 shows the distributions of the URL and the domain length for the URL in the dataset. As we can see, phishing URLs exhibit a greater length compared to legitimate URLs. This observation suggests that attackers use longer URLs to obscure malicious content to mimic legitimate domains more effectively.

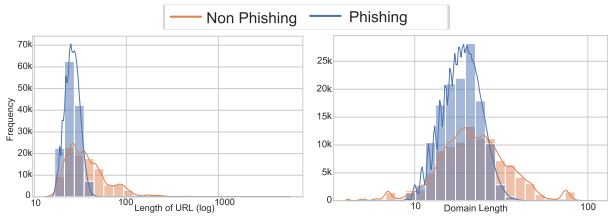

Figure 1: Characteristics of phishing and legitimate URLs.

This characteristic might be exploited as a distinguishing feature for detecting phishing URLs, since the increased length often results from additional subdomains, directories, or parameters that make the address appear more complex or authentic. In addition to URL structure, phishing domains themselves also tend to be longer, as attackers frequently add random characters or imitate legitimate brand names to obfuscate the malicious nature of the site further.

4 Can we trust LLMs in detecting phishing sites?

To address RQ1, we evaluate the ability of several LLMs to identify phishing websites using the prompt described in Section 6.3. Table 2 reports the classification results obtained by performing three executions for each URL. To ensure a fair assessment, we use local models without Internet access (Mistral 7B and Llama 3.1 7B) and models released before the *PhiUSIIL Phishing URL* dataset (Llama 2 7B and Llama 2 13B), reducing the chance they were trained on it.

As we can see, the best classification results are obtained by GPT-4o and Copilot, which reach a score of 1 across all metrics considered, i.e., accuracy, precision, recall, and F1-score. On the other hand, Google Gemini is the model that obtains the worst classification results, with a score of 0.80 across all utility metrics considered, i.e., accuracy, precision, recall, and F1-score. Yet, Google Gemini is followed, in terms of classification results, by Mistral 7*B* and Claude 3 Sonnet that slightly deviate from Google Gemini by reaching a score of 0.85 in accuracy, 1 in precision, 0.70 in recall, and 0.82 in F1-score, respectively. Moreover, Table 2 highlights that GPT 4*o* mini and Claude 3 Haiku are the second-best classifiers that slightly deviate from the best ones by reaching a score of 0.95 in accuracy, 1 in precision, 0.90 in recall, and 0.95 in F1-score, respectively. Llama 3.1 8*B* obtains a score of 0.90 across all utility metrics considered, i.e., accuracy, precision, recall, and F1-score, respectively, by highlighting that it generally offers promising results in terms of phishing classification. Finally, both Llama 2 7B and Llama 2 13B achieve an accuracy of 0.85, while differing in the remaining metrics: Llama 2 7B achieved a score of 0.78 for precision, 0.80 for the recall, and 0.75 for the F1-score, whereas Llama 2 13B achieves 0.84 in precision, 0.84 in recall, and 0.79 in F1-score.

5 How do traditional ML models compare to LLMs for phishing detection?

In the previous section, we assessed various LLMs in identifying phishing websites. Now, we evaluate different machine learning models on the same task to compare their classification effectiveness. Table 1 reports the classification results for different ML models by performing three executions for each URL. Additional details on the training and testing procedures, including the dataset split strategy, are provided in the project repository[1]. As we can see, KNN and Decision Tree excelled with a score of 0.98 across accuracy, precision, recall, and F1-score, while Bernoulli Naïve Bayes (BNB) performed the worst at 0.88, and Logistic Regression followed with a score of 0.92. The SVM model, with a score of 0.97, was the second-best, while both Random Forest and AdaBoost scored 0.95. To enhance performance, we utilized the Stacking Classifier, which

[1] https://github.com/carmine-ambrosino/u-proof.git

Model	A	P_0	R_0	$F1_0$	P_1	R_1	$F1_1$
Logistic Regression	± 0.92	± 0.90	± 0.92	± 0.91	± 0.94	± 0.93	± 0.93
KNN	± 0.98	± 1.00	± 0.97	± 0.98	± 0.97	± 1.00	± 0.98
Decision Tree	± 0.95	± 0.91	± 0.97	± 0.94	± 0.98	± 0.93	± 0.95
Bernoulli Naive Bayes	± 0.88	± 1.00	± 0.72	± 0.84	± 0.83	± 1.00	± 0.91
SVM	± 0.95	± 0.93	± 0.96	± 0.94	± 0.97	± 0.94	± 0.96
Random Forest	± 0.96	± 0.97	± 0.97	± 0.97	± 0.98	± 0.97	± 0.97
AdaBoost	± 0.95	± 0.91	± 0.97	± 0.94	± 0.98	± 0.93	± 0.95
Stacking Classifier	± 0.95	± 0.93	± 0.96	± 0.94	± 0.97	± 0.94	± 0.96

Table 1: Results of phishing URL detection using ML models.

also achieved a score of 0.98 across all metrics. Comparing LLMs and ML approaches, KNN and Decision Tree models deviate slightly from the best LLMs (GPT 4*o* and Copilot) by 0.02. BNB outperformed Google Gemini by 0.08, showing that weaker ML models can sometimes beat LLMs. Overall, ML models usually achieve slightly higher accuracy than most LLMs, but not the top ones.

6 The U-Proof tool

U-Proof is a phishing-domain detection tool for both individual users and organizations. As shown in Figure 2, it exposes an *nginx* container as a reverse proxy to external clients, forwarding requests to a *Flask* container running the application logic, which combines a stacking-based ML model with two open-source LLMs, Llama 3.1 8B and Mistral 7B. The source code and screenshots of U-Proof are available in the associated GitHub repository[1].

6.1 Functionalities

The U-Proof tool provides several key functionalities aimed at supporting both individual users and organizations in defending against phishing threats.

On-the-fly Detection through LLMs and ML. U-Proof analyzes URLs and domain-level features in real time, classifying them as legitimate or phishing using a hybrid architecture that combines machine learning and large language models. ML classifiers use structural features such as URL length and domain entropy, while LLMs provide contextual semantic analysis based on linguistic patterns and expert-like reasoning. A majority-voting strategy aggregates predictions from multiple models and assigns the final label to the class with the most votes, improving robustness, reducing false positives, and enhancing decision stability.

LLM-based Explanations for Phishing URL Detection. The main functionality of U-Proof is the ability to provide explanations alongside predictions. Rather than returning a binary output, the tool generates detailed justifications of its decisions, highlighting suspicious indicators such as URL length and abnormal subdomain usage. These explanations, produced by the LLMs, reproduce the reasoning process of a cybersecurity expert, thereby fostering user trust and supporting transparency in the detection process. The explanatory component also plays an educational role, since it can be employed to raise awareness about phishing strategies and to strengthen organizational training activities.

Model	A	P_0	R_0	$F1_0$	P_1	R_1	$F1_1$
GPT 4o CLOSE	± 1.00	± 1.00	± 1.00	± 1.00	± 1.00	± 1.00	± 1.00
GPT 4o mini CLOSE	± 0.95	± 1.00	± 0.90	± 0.90	± 0.95	± 1.00	± 0.97
Claude 3 Sonnet CLOSE	± 0.85	± 1.00	± 0.70	± 0.82	± 0.85	± 1.00	± 0.92
Claude 3 Haiku CLOSE	± 0.95	± 1.00	± 0.90	± 0.90	± 0.95	± 1.00	± 0.97
Copilot CLOSE	± 1.00	± 1.00	± 1.00	± 1.00	± 1.00	± 1.00	± 1.00
Gemini CLOSE	± 0.80	± 0.80	± 0.80	± 0.80	± 0.80	± 0.80	± 0.80
Llama 3.1 8B OPEN	± 0.90	± 0.90	± 0.90	± 0.90	± 0.90	± 0.90	± 0.90
Mistral 7B OPEN	± 0.85	± 1.00	± 0.70	± 0.82	± 0.85	± 1.00	± 0.92
Llama 2 7B[1] OPEN	± 0.85	± 0.76	± 0.82	± 0.79	± 0.82	± 0.72	± 0.77
Llama 2 13B[1] OPEN	± 0.85	± 0.80	± 0.85	± 0.82	± 0.86	± 0.76	± 0.81

Class 0 denotes legitimate URLs; Class 1 denotes all others. True Positives (TP): phishing URLs correctly detected; True Negatives (TN): legitimate URLs correctly detected; False Positives (FP): misclassified legitimate URLs; False Negatives (FN): misclassified phishing URLs.[1] Released before the dataset [10]. CLOSE Closed-source LLM, OPEN Open-source LLM.

Table 2: Results of phishing URL detection using LLMs.

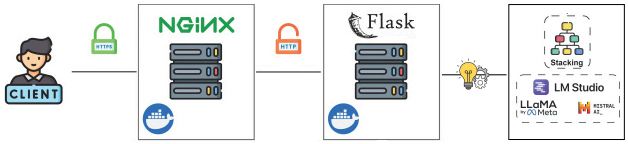

Figure 2: Overview of the modules underlying U-Proof.

6.2 Classification with ML Models

The ML-based phishing detector uses supervised learning trained on the labeled dataset in Section 3. We evaluate seven ML classifiers under identical experimental conditions, including the same data splits, preprocessing pipeline, scoring metric, and hyperparameter tuning strategy based on grid search, to ensure fair comparison. We also train a stacking ensemble, fitting a Logistic Regression meta-learner on out-of-fold predictions from the best base models and evaluating it on a held-out test set.

The dataset is split according to an 80/20 hold-out strategy, and all experiments use a fixed random seed (42) for reproducibility. Hyperparameter optimization is performed exclusively on the training set using 2-fold cross-validation with shuffling enabled. The task is formulated as a binary classification problem, where label 1 denotes phishing URLs and 0 legitimate ones. All models are trained on a shared set of phishing-related features, which are also used at inference when analyzing unknown URLs. URL features are obtained by direct parsing; page-level features come from controlled HTTP requests and shallow HTML inspection, without loading external resources or executing scripts. If a page cannot be accessed or parsed, the model handles missing features consistently. Full implementation and experimental details are in the official repository.

6.3 Interacting with LLMs

Using LLMs for specialized tasks requires carefully designed prompt engineering templates to ensure reliable behavior. We define an ad hoc prompt that simulates a cybersecurity expert assessing whether a URL is a phishing site. All LLM interactions are run locally in LMStudio, using the released models without extra training or fine-tuning. For reproducibility, the official model cards of the deployed LLMs are linked in the tool repository[1]. Each LLM is evaluated on a balanced subset of 200 randomly selected URLs from the dataset in Section 3 (100 phishing, 100 legitimate), ensuring a fair and controlled evaluation. The template is developed using the Manual Template Engineering approach, which uses human intuition to design effective prompts [8]. To this end, we have defined a prompting function $f_{\text{URL-evaluations}}(x)$ that aims to complete the sentence x to achieve the prompt sentence $x' = f_{\text{URL-evaluations}}(x)$:

Template of $f_{\text{URL-evaluations}}(v)$

> You are a cybersecurity expert. Analyze the following URL to determine whether it might be a phishing site or a legitimate one: [T]. Evaluate whether to visit the site based on the provided URL and identify any suspicious or legitimate signals. Clearly explain your decision in detail by using the signals you considered. If you cannot access the site, justify. If the site does not exist, clearly state that it is unreachable and assess whether the URL still suggests a phishing attempt. In the role of a cybersecurity expert, using the website URL provided, evaluate whether the site could be a phishing or legitimate site. [T]

where [ϒ] is the slot containing the URL and [T] is the slot related to the response of the LLM containing the textual response.

7 Discussion, Limitations, and Trade-offs

The proposed hybrid architecture, which combines traditional binary ML classifiers with LLMs through majority voting, improves robustness and interpretability but introduces challenges that may affect practical deployment. Although experimental results show that conventional ML models can achieve performance comparable to LLM-based approaches, LLMs offer broader evaluative capabilities by leveraging reasoning, contextual understanding, and external knowledge, thus improving resilience to novel and evolving phishing strategies. However, this comes with trade-offs in scalability and cost. Traditional ML models are lightweight and fit for large-scale, high-throughput scenarios, while LLM inference has higher latency and cost, especially with proprietary APIs. These limitations are further amplified by the majority voting, which requires multiple model invocations per URL.

From a security perspective, the hybrid design increases resistance to evasion by combining heterogeneous detection mechanisms, but also introduces new risks, such as prompt injection, reliance on external services, and privacy or reproducibility concerns due to evolving third-party models. Overall, while LLM integration enhances generalization and enables human-readable explanations, it increases latency, reduces scalability, and raises operational costs. Practical deployments may therefore benefit from adaptive strategies, such as invoking LLMs only for ambiguous cases, to better balance robustness, efficiency, and resource constraints.

8 Demonstration Plan

Our demonstration highlights how U-Proof helps users identify potentially dangerous websites that threaten data privacy and personal information. It blocks access to phishing domains, safeguarding sensitive data for both individuals and organizations. We utilize a combination of public and custom datasets for thorough evaluation, showcasing U-Proof's phishing detection capabilities in two scenarios: utilizing LLMs and ML.

Scenario 1. Phishing detection by using U-Proof with ML and LLMs. Let us suppose that a security analyst in an organization is responsible for monitoring web traffic to prevent employees from accessing malicious websites. The analyst can use the U-Proof to automatically process a batch of URLs extracted from emails and browsing logs. U-Proof evaluates each URL through its hybrid architecture, where machine learning models contribute with structural feature analysis, while LLMs provide contextual semantic reasoning and explanations. By combining the two approaches in a majority-voting strategy, U-Proof delivers both a reliable classification and a human-readable justification. This allows the analyst to identify phishing URLs with high accuracy and efficiently provide a security report for employees. The combination of ML and LLM models within U-Proof proves particularly effective for large-scale monitoring and organizational defense against phishing threats.

Scenario 2. Explanation of phishing detection using U-Proof. Let us suppose that a user browses the internet for their daily activities, such as looking for news websites, sending emails, reading a book, and so on. Such a user can use U-Proof to examine URLs visited, and thanks to a new ad-hoc prompt template defined to mimic a cybersecurity expert in evaluating a URL, they can easily identify phishing sites. The prompt template is developed using the Manual Template Engineering approach, which leverages human intuition to design effective prompts. In this scenario, the user can exploit LLMs' capabilities within the U-Proof to receive an in-depth explanation that enhances awareness of phishing tactics.

9 Conclusion and Future Works

We introduced U-Proof, a phishing detection tool that integrates LLMs' contextual reasoning with traditional ML pattern recognition. Experiments on a public phishing dataset show that LLMs are particularly effective at capturing content- and context-based phishing cues, while ML models reliably detect known patterns. Their combination in U-Proof improves detection accuracy and reduces false positives, especially for phishing sites that closely mimic legitimate ones.

In the future, we aim to enhance the U-Proof tool with advanced techniques for phishing detection, such as anomaly detection and reinforcement learning. Additionally, we plan to investigate the ethical and privacy implications of utilizing LLMs in security, particularly in relation to personal data and regulatory compliance.

Acknowledgments

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

References

[1] Samer Atawneh and Hamzah Aljehani. 2023. Phishing email detection model using deep learning. *Electronics* (2023).
[2] Maurizio Atzori, Eleonora Calò, Loredana Caruccio, Stefano Cirillo, Giuseppe Polese, and Giandomenico Solimando. 2024. Evaluating password strength based on information spread on social networks: A combined approach relying on data reconstruction and generative models. *Online Soc. Networks Media* 42 (2024).
[3] Cagatay Catal, Görkem Giray, Bedir Tekinerdogan, Sandeep Kumar, and Suyash Shukla. 2022. Applications of deep learning for phishing detection: a systematic literature review. *Knowledge and Information Systems* (2022).
[4] Francesca Cerruto, Stefano Cirillo, Domenico Desiato, Simone Michele Gambardella, and Giuseppe Polese. 2022. Social network data analysis to highlight privacy threats in sharing data. *Journal of Big Data* 9, 1 (2022), 19.
[5] Stefano Cirillo, Domenico Desiato, Michele Scalera, and Giandomenico Solimando. 2023. A Visual Privacy Tool to Help Users in Preserving Social Network Data. In *Proc. of the 9th International Symposium on End-User Development (IS-EUD 2023), Cagliari, Italy, June 6-8, 2023*, Vol. 3408. CEUR-WS.org.
[6] Gangu Dharmaraju, Tatapudi Nirosh Kumar, P PattabhiRama Mohan, Raja Rao Pbv, and A Lakshmanarao. 2024. Phishing Website Detection through Ensemble Machine Learning Techniques. In *2024 2nd International Conference on Computer, Communication and Control (IC4)*. IEEE.
[7] Richa Goenka, Meenu Chawla, and Namita Tiwari. 2024. A comprehensive survey of phishing: Mediums, intended targets, attack and defence techniques and a novel taxonomy. *International Journal of Information Security* 23, 2 (2024).
[8] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023).
[9] J Pathmanaban, Prabahar Godwin James, P Ashok, B Ragesh, S Aakash, and N Kaushik. 2024. Phishing Website Detection Using Machine Learning. In *2024 2nd International Conference on Networking and Communications (ICNWC)*. IEEE.
[10] Arvind Prasad and Shalini Chandra. 2024. PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. *Computers & Security* (2024).
[11] Orvila Sarker, Asangi Jayatilaka, and et al. Haggag. 2024. A Multi-vocal Literature Review on challenges and critical success factors of phishing education, training and awareness. *Journal of Systems and Software* 208 (2024).
[12] Md Milon Uddin, Kazi Arfatul Islam, Muntasir Mamun, Vivek Kumar Tiwari, and Jounsup Park. 2022. A Comparative Analysis of Machine Learning-Based Website Phishing Detection Using URL Information. In *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. IEEE.