INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2026, VOL. 72, NO. 2, PP. 1–11
Manuscript received July 27, 2026; revised April 2026. doi: 10.24425/ijet.2026.157910

Towards improved Phishing website detection: heuristic-based approaches vs. Machine Learning

Dmytro Shutenko, Anwar Hasan, Serhii Buchyk, and Ruslana Ziubina

***Abstract*—Phishing is widely acknowledged as one of the most insidious types of social engineering attacks. Despite substantial efforts to combat this issue, it continues to evolve in sophistication, resulting in increasing financial losses. Historically, countering phishing involved a blend of human vigilance and software-based detection mechanisms, primarily relying on list-based strategies. However, with the advent of advanced data science, innovative phishing detection techniques utilizing Machine Learning models have emerged and garnered significant research attention. This study aims to comprehensively compare the effectiveness of traditional heuristic-based and modern Machine Learning classification models, while addressing challenges associated with their efficiency. Experimental results involving the Random Forest classifier, although requiring slightly more computational power, demonstrated a substantial increase in detection accuracy (57.2% higher) and a remarkable reduction in testing time (11.28 seconds faster vs 0.01 seconds) when compared to heuristics using the same input data.**

***Keywords*—phishing websites detection; website features; heuristics; machine learning; random forest**

I. Introduction

PHISHING represents a form of cyber-criminal activity wherein adversaries employ social engineering tactics to deceive individuals into revealing sensitive information and passwords under the guise of a trustworthy entity. The inception of phishing preceded its prevalence on the Internet, with earlier manifestations occurring through phone-based methods, known as vishing. The contemporary methods of phishing execution predominantly involve email correspondence or web resources and services that guide victims to counterfeit websites mirroring familiar interfaces, urging them to input their login credentials. 91% of all attacks on individuals used social engineering techniques in some shape or form often containing phishing attempts according to latest data [1] as of Q1 2023. Another prevalent approach to target organizations is the utilization of mail-outs, wherein targeted recipients are lured into responding to queries purportedly posed by their own IT department [2]. Mail-outs not only gather data but also serve as a conduit for disseminating malware, frequently concealed within attachments. In 42% and 22% of recorded cases malware distribution methods in attacks on organizations were reported to have happened via websites and email respectively [1]. Interestingly, in another 23% of cases malware distribution happened via social networks and messaging apps. Therefore, despite big corporations like Google having come up with robust means to detect and automatically block phishing emails, Internet users are now simply targeted differently which is supported by an increasing number of recorded attacks of such kind [3]. Consequently, this work centers on phishing websites as one of the biggest Internet threats today.

Emerging in the mid-1990s, these threats were initially tackled through blacklisting approaches. As software development evolved, adversaries employed ever-creative tactics, resulting in significant financial losses for enterprises, governments, and individuals. Security experts introduced heuristic-based techniques to detect and thwart these threats, but adversaries constantly altered their tactics to challenge detection algorithms. The advent of advanced data science introduced innovative phishing detection techniques utilizing Machine Learning (ML) models, garnering research attention since the late 2000s. By harnessing ML's flexibility and complexity, researchers achieved promising lab results, with some solutions integrated as standalone services [4] or modules within existing systems. Despite this, researchers continue to explore pre-Machine Learning phishing detection methods, which we deem inefficient.

This paper aims to conduct phishing websites detection accuracy comparison of a hybrid phishing website detection system, combining URL and heuristic approaches with a Machine Learning model employing the same features. The structure unfolds in the following sections: Section 2 offers an overview of traditional phishing detection techniques, with an emphasis on pre-Machine Learning era methods' deficiencies. Sections 3 and 4 elaborate on the hybrid detection system's inspiration, methodology, and Machine Learning Model respectively. Additionally Section 3 covers the issue of features selection and Section 4 describes the dataset used to train the classifier. Section 5 outlines test procedures and presents comparative metrics such as detection accuracy and testing time. Section 6 concludes with final remarks and future research suggestions.

II. Phishing Detection Techniques

Phishing attacks can be categorized into two primary types: human-based social engineering, involving direct physical

D. Shutenko and A. Hasan are with the Department of Electrical and Computer Engineering, University of Waterloo, Canada (e-mail: {dshutenko, ahasan}@uwaterloo.ca).

S. Buchyk is with Taras Shevchenko National University of Kyiv, Ukraine (e-mail: buchyk@knu.ua).

R. Ziubina is with the University of Bielsko-Biala, Poland (e-mail: rziubina@ubb.edu.pl).

© The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0, https://creativecommons.org/licenses/by/4.0/), which permits use, distribution, and reproduction in any medium, provided that the Article is properly cited.

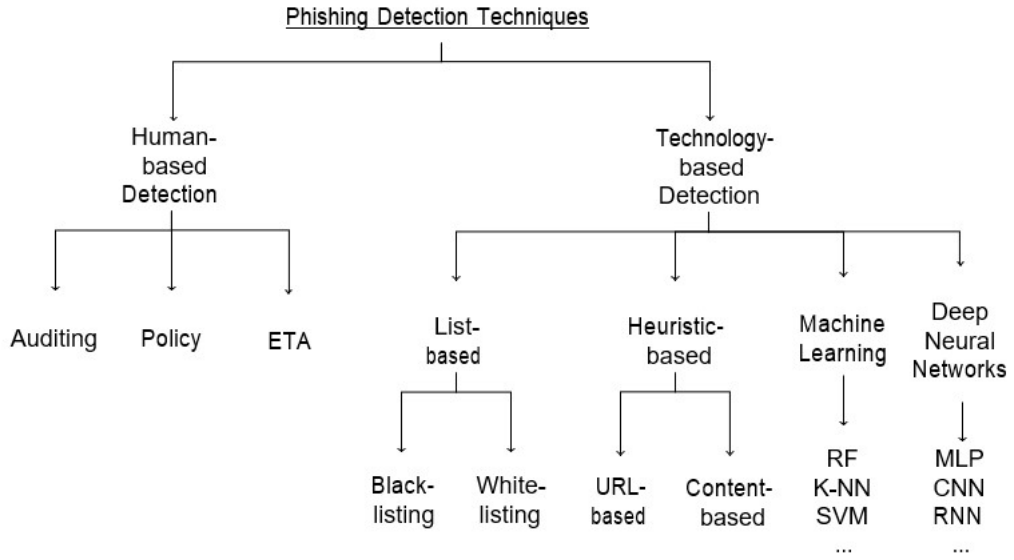

Fig. 1. Taxonomy of Phishing Attacks Detection Techniques.

interactions, often through phone calls, and technology-based social engineering, encompassing tactics like online social network impersonation, website phishing scams, and email phishing. Figure 1 presents the current taxonomy of phishing detection methods. Subsequent paragraphs of this section will focus on a comprehensive exploration of both categories of phishing detection methods. Furthermore, exploration will be conducted into the various approaches that information security specialists can employ to proactively mitigate the risks of sensitive data leakage.

Human-based detection methods involve direct human intervention in identifying and preventing phishing attacks. These methods rely on human judgment to determine whether encountered activities are indicative of social engineering attacks. Initially, it was the primary means of detecting such attacks, given the lack of automated systems at the time. As a result, it has undergone substantial research. Nowadays, three distinct approaches fall under human-based mitigation: policy, auditing, and the ETA framework — comprising education, training, and awareness. The significance of employing human decision-making in mitigation strategies is given a lot of emphasis in [3]. Another set of approaches to detect and prevent phishing attacks involves utilizing technology-based solutions. These methods gained prominence as phishing incidents proliferated, prompting the demand for email filters and background website analyzers. This initiated an ongoing competition between adversaries and security experts, each refining their techniques to outsmart the other. Over the past three decades, technology-based methods have undergone comprehensive exploration in countering social engineering attacks. In this work, an overview of the following phishing technology based detection methods will be presented:

1) List Based detection;
2) Heuristics detection;
3) Machine Learning models;
4) Deep Neural Networks.

First and foremost, there are such list based solutions as blacklists and whitelists. A blacklist or whitelist operates solely based on the URLs contained within each list, thus missing out on URLs not present in the list. This approach often leads to a trade-off between false positives (whitelist) and false negatives (blacklist) as mentioned in [5]. Consistently updating these lists is crucial, but in practice, these methods have shown limited effectiveness due to the challenge of staying up to date with the continuous influx of new phishing websites being generated.

Secondly, various heuristic techniques prove effective in detecting phishing patterns by analyzing characteristic attributes and patterns associated with deceptive websites. These methodologies rely on predefined rules to assess the likelihood of a website's maliciousness, contributing to a flexible and efficient phishing detection system.

In [7], researchers proposed an anti-phishing strategy centered on identifying irregularities within web pages. This method extracts anomalies from diverse sources, such as URLs, page titles, cookies, login forms, DNS data, and SSL certificates. By comparing these anomalies to an extensive dataset of known malicious patterns, the offered approach has the potential to uncover zero-day phishing attacks which are phishing websites not yet identified and therefore non-existent in the databases mentioned earlier. While some argue that this doesn't significantly differentiate from blacklists, as blacklists necessitate precise matches to identify phishing sites, it's worth noting that heuristic methods have a better chance of identifying new malicious payloads [8]. However, this adaptability also increases the risk of false positives, disrupting normal operations and causing system overhead. Major email clients and web browsers have already integrated heuristic-based detectors to identify phishing attacks among others. Moreover, numerous antivirus solutions incorporate heuristic-based phishing detection. In an unchanging threat landscape,

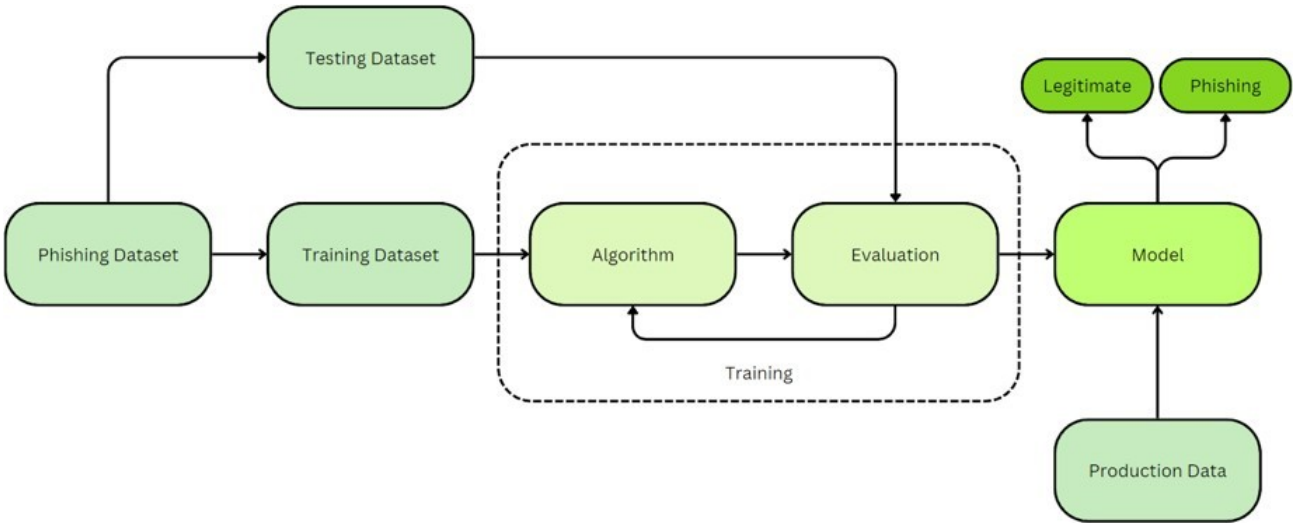

Fig. 2. Utilizing Machine Learning for phishing attack detection [6].

these methods could eventually establish an exhaustive set of signatures for identifying potential threats.

While heuristic detection methods perform adequately in identifying reported phishing patterns, some researchers argue that they only marginally surpass basic blacklists, as indicated in [9]. These methods may also introduce significant system overhead since they necessitate the creation of a signature for each entry, followed by a comparison against numerous known signatures within a given database. It can be argued that such solutions could be effective if a portion of data could be sent to the cloud for analysis and subsequent comparison with known malicious signatures. The outcome could then be relayed to the host, notifying the user of potential threats or even automatically filtering out such content before it becomes visible. However, practical factors frequently hinder the utilization of cloud-based methods for some types of data due to security regulations. This situation presents a dilemma where a choice must be made between two suboptimal alternatives.

Moving forward, more contemporary and highly promising methodologies have emerged, gaining effectiveness through rigorous research - Machine Learning-driven approaches. These techniques leverage algorithms that learn from extensive datasets containing information about known phishing websites, emails, and even text messages, enabling them to identify suspicious entities with remarkable accuracy. A recent study [10] treats phishing detection as a Machine Learning classification challenge. Here, the decision- making process determines whether a given website is legitimate or a phishing platform. Essentially, ML models undertake the task of analyzing dynamic phishing patterns, discerning the optimal combinations of characteristics for identifying malicious behavior, and filtering out outdated data. This aligns with the notion proposed in [11], advocating AI algorithms as the foundation for constructing viable models to combat the evolving nature of phishing threats. In essence, numerous Machine Learning-based solutions capitalize on systematically acquired knowledge regarding key traits that have proven effective in identifying elements commonly associated with phishing websites, as outlined in a study [12].

Training the Machine Learning model for such a detection system requires a relevant dataset that has features that are related to both phishing and legitimate website classes. Previous research demonstrates that by employing robust ML techniques like Random Forest, k-Nearest Neighbor (K-NN), and Support Vector Machine (SVM), high detection accuracy can be achieved. Figure 2, adapted from [6], illustrates a typical scenario used to train an ML model to differentiate phishing websites from legitimate ones. The particular algorithm has little influence on the common approach taken for Machine Learning while the utilization of a dataset comprising entries from both classes is paramount.

In [13], findings indicate that the Random Forest model outperformed its counterparts on a relatively modest dataset. This model was the best performing model with a True Positive of 100%, a True Negative of 90.48%, and an accuracy of 98.35%. A prediction was made by averaging the result obtained from five individual Decision Trees. This helps to reduce the problem of overfitting, a problem peculiar to the Decision Tree Algorithm.

Machine Learning and Deep Neural Networks usually require large datasets and long periods of training to be effective. In [14], authors highlight the challenge of obtaining the requisite datasets, which are often scarce unless deliberate efforts are made to collect samples. Moreover, these datasets can become obsolete if not regularly updated with newly identified instances. The dynamic nature of phishing attacks renders older datasets inadequate, as adversaries continuously devise more sophisticated strategies in response to catching up defense mechanisms. Additionally, the process of gathering information itself is prone to inaccuracies, leading to a higher false positive rate and rendering the system cumbersome rather than efficient. Furthermore, the inherent unpredictability of Machine Learning models and the substantial time required for their training, testing, and deployment further compound the challenges.

The use of technology is often accompanied by added cost, complexity and overall system overhead. The systems that have been mentioned in this work would require a significant financial investment by an organization in most cases without a clear measure of cost-benefit once they will be deployed. Thus, spending large amounts of money on such systems, their training, deployment, management and maintenance can be irrelevant. The added complexity of the systems also means that there is potential for a business process to be interrupted in case those systems malfunction yielding false positives one after another. Thus the task is to prioritize research and development of systems that perform effectively in real-world scenarios, avoiding allocating resources to revise outdated techniques. In the following chapters, the aim is to determine whether continued research and enhancement of pre-ML era phishing detection methods hold merit, or if researchers' efforts should predominantly be channeled into the development of ML-based solutions.

To assess the value of pre-Machine Learning era phishing detection methods future research, a decision was made to merge the strengths of current solutions, resulting in the formation of a hybrid system. This section discusses the process of designing this system, its underlying structure, and website features chosen to achieve peak detection accuracy. Taking cues from the heuristic principles outlined in [15], a theory was formulated suggesting that integrating various phishing detection techniques could not only boost the efficiency of existing systems but also reduce the limitations linked to using each method used on its own. The system we're proposing consists of three main parts that examine the given website to find any signs of potential danger as illustrated in Figure 3. In the upcoming paragraphs, a deeper dive will be taken into each of these components, explaining their functionality and how they have been put into practice.

III. Heuristic Phishing Websites Detection System

To assess the value of pre-Machine Learning era phishing detection methods future research, a decision was made to merge the strengths of current solutions, resulting in the formation of a hybrid system. This section discusses the process of designing this system, its underlying structure, and website features chosen to achieve peak detection accuracy. Taking cues from the heuristic principles outlined in [15], a theory was formulated suggesting that integrating various phishing detection techniques could not only boost the efficiency of existing systems but also reduce the limitations linked to using each method used on its own. The system we're proposing consists of three main parts that examine the given website to find any signs of potential danger as illustrated in Figure 3. In the upcoming paragraphs, a deeper dive will be taken into each of these components, explaining their functionality and how they have been put into practice.

URL Analysis: Initially, the focus is on harnessing the power of blacklist-based filtering. This requires the first component to have an up-to-date collection of reported threats. To achieve this, the decision was made to utilize the Google Safe Browsing Lookup API (version 4) as an independent service. This choice offloads the resource-intensive tasks, like handling extensive infrastructure and high computation power, to the API of a trusted vendor. Meanwhile, it keeps our URL-based detection method nimble and flexible. The component is designed in such a way that if Google identifies a specific website as malicious, our system will swiftly flag the entered URL as phishing and terminate future tests to speed up the process of analysis. It is fair to state Google is using best practices which include a comprehensive examination of various URL attributes to confirm the fraudulent nature of a website before blacklisting it. Therefore, it is assumed that the Google Safe Browsing API, upon which the system relies, can compensate for a selected set of features that will later be compared to our Machine Learning-powered system. The logical structure of the system reveals that no additional components are employed if Google has already blacklisted the website and the system triggers a fraud notice immediately.

Content Analysis: To detect irregularities within fraudulent websites present in the web page's Document Object Model (DOM) and client-side scripts (JavaScript functions), the approach aims to streamline the process. Instead of generating complex signatures, website content will be retrieved, and specific data points will be extracted for analysis. This approach not only conserves computational resources but also eliminates the need for signature comparison with an existing database. We will leverage third-party packages to simplify and accelerate the parsing process. Following a methodology outlined in [16], the initial feature set is to be hand-crafted.

TABLE I
URL FEATURES

ID	Feature Name	ID	Feature Name	ID	Feature Name
1	URL length	11	'http' in path	21	Number of (@) characters
2	Hostname length	12	Number of subdomains	22	Number of (%) percent characters
3	Port	13	Number of 'www'	23	Number of (-) hyphen characters
4	Shortening service	14	Number of 'com'	24	Number of (;) semicolon characters
5	Character repeat	15	Number of redirections	25	Number of ($) dollar sign characters
6	Brand in path	16	Number of (=) equals characters	26	Number of (,) comma characters
7	Punycode	17	Number of (*) star characters	27	Number of (.) dot characters
8	IP	18	Number of (\) slash characters	28	Number of (&) and characters
9	Prefix, suffix	19	Number of (~) tilde characters	29	Number of (:) colon characters
10	Https token	20	Number of (——) or characters	30	Number of (_) underscore characters

Domain Analysis: This component will harness domain-specific information, along with insights into a website's ranking on major search engines, which are procured through third-party services such as Whois and Google. Relevant packages will be used to facilitate this type of data retrieval

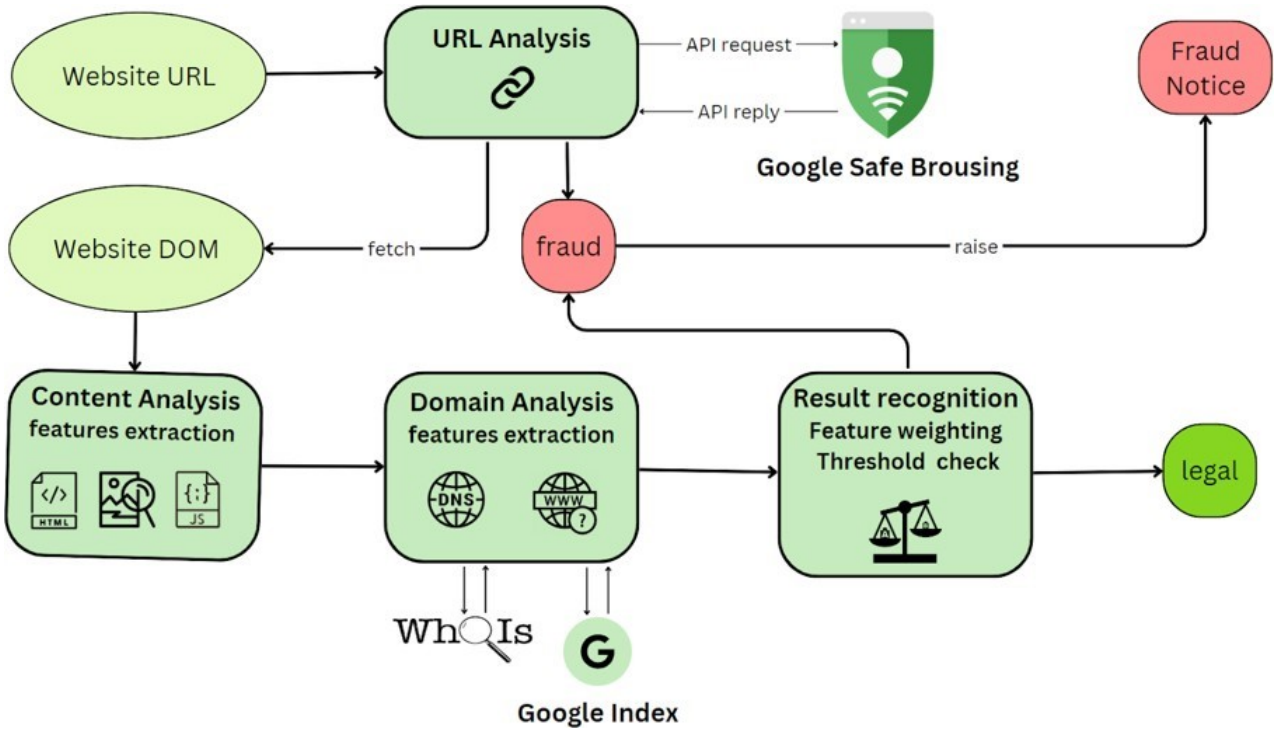

Fig. 3. Heuristic Phishing Websites Detection System.

TABLE II
CONTENT FEATURES

ID	Feature Name	ID	Feature Name	ID	Feature Name
31	Number of hyperlinks	35	Domain with copyright	39	Usage of iframe
32	Internal Hyperlinks Ratio	36	Login form	40	Empty title
33	External Hyperlinks Ratio	37	External favicon	41	Domain in title
34	Null Hyperlinks Ratio	38	Email submission	42	Pop Up Window

TABLE III
DOMAIN FEATURES

ID	Feature Name
43	Google Index
44	WHOIS registered domain
45	Domain registration length

and analysis process. This component plays an important role in our phishing detection system by shedding light on domain registry status and its history, ultimately enhancing our ability to identify potential red flags.

A subset of website features for analysis was previously alluded to, drawing from prior research of works [8], [11], [15]–[21]. This selection process aimed to identify the most effective features while discarding those that proved ineffective taking into account corresponding performance for every component of our system.

To eliminate redundancy among features targeting similar patterns with equal influence and to emphasize meaningful distinctions, features that did not significantly impact detection rates were removed. Consequently, an intermediary selection of 45 features was obtained, representing a reduction of approximately 35% from the total number of features extracted from the literature analyzed. This intermediary set of features includes 30 URL features (refer to Table 1), 12 Content features (refer to Table 2), and 3 Domain features (refer to Table 3).

The decision to reduce the number of features instead of increasing them, which was initially driven by the aim to capture a broader range of malicious patterns [8], can be achieved by using 2 ranking methods, in particular Feature Selection by Filtering Method (FSFM) and Feature Selection by Omitting Redundant Features (FSOR).

The FSFM process is implemented by using Information Gain (IG) which measures the extent to which the features are mixed up [19]. The FSOR process is implemented by using the Relief Ranking Filter to rank all extracted features, identifying the most desired ones over those that don't bring much input into the detection process.

Additionally, as it's pointed out in work [15], IG is employed in measuring the relevance of attribute K in class L. As the mutual information value between classes K and attribute L gets higher, the relevance between classes K and attribute L gets higher, as follows:

$$IG(L, K) = H(L) - H(L|K), \tag{1}$$

where

$$H(L) = -\sum_{c \in C} P(c) \log P(c) \tag{2}$$

is the entropy of the class L,

$$H(L|K) = -\sum_{k \in K} P(k) \sum_{c \in C} P(c|k) \log P(c|k) \tag{3}$$

is the conditional entropy of the class given attribute K.

Since the testing dataset is balanced with $P(positive) = P(negative) = 0.5$, the entropy of the class is

$$H(Y) = -\sum_{i} P(y_i) \log_2 P(y_i) = -\left(0.5 \log_2 0.5 + 0.5 \log_2 0.5\right) = 1.$$

Thus, the information gain obtained from attribute K can be formulated as

$$IG(L, K) = 1 - H(L|K). \tag{4}$$

The minimum value of $IG(L, K)$ happens only if $H(L|K) = 1$ which indicates that attribute K and class L have no relation to one another at all. In contrast, there is a tendency to select attribute K that usually appears in one class L as either positive or negative. In other words, a set of attributes that appears only once in one class is classified as the best set of features. This indicates that the maximum $IG(L|K)$ is attained when $P(K)$ is equivalent to $P(K|L)$ resulting in $P(L_1|K)$ and $H(L_1|K)$ being equivalent to 0.5. When $P(K) = P(K|L_2)$, then the value of $P(K|L_2)$ results in $P(L_1|K)$ and $H(L_1|K) = 0$. The value of $IG(L, K)$ is therefore varied from 0 to 0.4.

The Relief Ranking Filter, an essential component of feature selection process, serves the purpose of feature weighting and identifying most unique and redundant ones. Its core principle involves randomly selecting instances, computing their nearest neighbors, and adjusting a feature weighting vector to assign more weight to features that distinguish an instance from neighbors of different classes. Specifically, the Relief Ranking Filter aims to establish a suitable probability estimate, which can subsequently be assigned as the weight for each feature f as depicted as follows:

$$w_f = \frac{p_d \times x}{c_d} - \frac{p_s \times x}{c_s} \tag{5}$$

where w_f is the weight for every feature f, p_d is the probability of a different value of feature x across different classes c_d, and p_s represents the probability of different values of feature x within the same class c_s. This method is said to yield good performance in numerous domains as pointed out in [22].

By applying the concepts of IG and Relief Ranking Filter, the final selection of features used in the system was determined, resulting in approximately a 45% reduction from the total extracted features. This final feature set differs from the intermediary set obtained after the removal of redundant features. This final feature set differs from the intermediary set after the removal of redundant features. The selected features comprise 25 website attributes, denoted by ID Features 1, 2, 3, 6, 8, 10, 16, 18, 23, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45. Through the elimination of less effective features, a reduction in processing time and an improvement in system performance are achieved, particularly when operating on a lower-specification computer. This is essential as the system is intended to be hosted on-premise rather than in a cloud environment.

Table 4 presents different categories of website features, each assigned a weight corresponding to its perceived significance. Weights are evenly distributed among all features of a given category with an exception of the URL category where weight is consolidated due to the system's implementation details (all individual features get reduced to Google Safe Browsing API's return value). While some works, such as Yang et al. [17], argue that certain features carry a more substantial impact on the detection process, we believe that numeric representation can introduce researcher and data biases, which vary for each specific study. In our approach, the aim is to mitigate these biases by testing the efficiency of uniformly distributed weights against those provided by the Machine Learning model. This comparison will be detailed in the upcoming section.

TABLE IV
FEATURES WEIGHT DISTRIBUTION

Website Attribute Category	Feature IDs	Weight
URL	1, 2, 3, 6, 8, 10, 16, 18, 23, 28, 30	0.45
DOM	31, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42	0.40
Domain	43, 44, 45	0.15

The hypothesis explored in previous research [8] suggested that enhancing the system's effectiveness could be achieved by introducing new features and optimizing the distribution of attribute values. Yet, having stress tested this hypothesis in practice, it became clear that the quantity of features isn't the key factor, but rather the unique impact each feature has on detection accuracy.

The challenge of precisely defining the threshold for identifying suspicious patterns remains an open question for future research. For this study we define it to be 0.45 of the phishing index P as follows:

$$P = \sum_{f=1}^{n} k_f \cdot w_f \tag{6}$$

where k_f represents the coefficient of each feature contributing to phishing, w_f is its respective weight, and n is the number of features. The coefficient k_f is defined as 0 if a feature doesn't contribute to phishing and 1 if it does.

To validate the system proposed in this study, a web-based application was developed for conducting tests. To ensure optimal speed and swift performance, we integrated URL-related checks on the client side of the application. Meanwhile, checks related to website content and domain were executed on the server side. This approach helped us overcome Cross-Origin Resource Sharing policy restrictions and offload resource-intensive tasks from the client, thus optimizing overall performance of the system. Consequently,

we created an open-source test application hosted on GitHub [23], utilizing Flask, JavaScript for the client side, and Python for the server side. User interface of offered system can be seen on Figure 4. A number of third-party Python libraries were employed including BeautifulSoup, requests, tldextract, validators, urllib, whois, selenium and others.

Once the solution is refined, a plugin can be developed to automate the detection process, seamlessly working in the background to enhance user convenience. In Section 5 of this paper, we evaluate the system's performance using real-world data.

IV. Phishing Websites Detection Machine Learning Model

This section is dedicated to the development of a Machine Learning model designed for the detection of phishing websites. Importantly, this model will operate using the same input data as the Heuristic system introduced in Section 3. As detailed earlier, a total of 25 features were selected for analysis using the FSFM and FSOR ranking methods. However, this is just one piece of the puzzle when aiming to construct an ML model capable of detecting zero-day phishing websites. The other crucial elements are the classifier itself and the training dataset. Among the various approaches proposed in the literature, data mining-based methods have proven effective in identifying phishing attacks, as highlighted in [20]. For instance, researchers in study [9] employed various data mining techniques to categorize web pages as either legitimate or phishing. They utilized multiple classifiers to develop an efficient phishing detection system, with the Random Forest classifier demonstrating superior performance in detecting phishing attempts. It's important to note that these techniques rely on ML libraries written in Python and, as such, cannot be executed in most web browsers in real-time. However, the primary objective of this research is to compare traditional heuristic systems with Machine Learning-based ones in terms of their real-time phishing detection accuracy. To achieve this, we adopted a strategy where predictions are made on the server side and subsequently relayed to the client's browser, possibly through an extension or other architecturally suitable means, while adhering to best traffic control practices. Algorithm 1 provides the pseudocode for the Random Forest (RF) Machine Learning classifier chosen for this study.

The study makes use of Kaggle as a convenient platform for addressing data science problems, much like the one presented herein. Additionally, we employ Python scikit-learn and pandas libraries for the design and training of our Machine Learning model. Other Python libraries numpy and matplotlib are used to cover the needs associated with linear algebra and plotting respectively. The initial configuration of the proposed RF model is illustrated in Figure 6. While the heuristic system outlined in Section 3 partly relies on the research team's subjectivity when defining attribute weights, thresholds, and the overall program logic (which entails decision-making for each attribute), Random Forest in contrast, utilizes numerous decision trees. It derives predictions by averaging the outputs of each individual tree component.

TABLE V
Algorithm 1: Random Forest (RF) pseudocode [24]

Algorithm: Random Forest (RF)
Require: Training data D
Ensure: RF classification model with c classifiers

for $i = 1$ to c **do**
  Sample D with replacement to obtain D_i
  Create root node N_i with D_i
  call `BuildTree`(N_i)
end for

function `BuildTree`(N):
  if N contains instances of only one class **then**
    return
  else
    Randomly select $x\%$ of the features in N
    Select feature F with highest Information Gain
    Create f child nodes $N_1, \ldots, N_f$ based on values $F_1, \ldots, F_f$ of F
    for $i = 1$ to f **do**
      Set D_i to instances in N where feature $F = F_i$
      call `BuildTree`(N_i)
    end for
  end if

TABLE VI
Dataset details

Category	Training split	Testing split
Input data	11400	12
Duplicate data	0	0
Extracted features	25	25
Phishing URLs ratio	50%	50%
Legitimate URLs ratio	50%	50%
Date of creation	May 2020	October 2023

Figure 6, shows a snippet of a publicly available training dataset mined in [21], which was vital for training the Random Forest classifier. The dataset comprises 2 parts for the total of 11412 URLs with 25 extracted features discussed earlier. The dataset is balanced, it contains exactly 50% phishing and 50% legitimate URLs. Training dataset was originally designed to be used as a benchmark for Machine Learning based phishing detection systems.

Testing split was sourced from PhishTank [22] and most recent instances witnessed. PhishTank is an antiphishing website where users can submit, verify, track or share suspicious and phishing websites. It maintains a phishing archive consisting of valid, unknown, online or offline phishing sites. Dataset details are given in Table 6.

It is acknowledged that utilizing such a small testing split comes with inherent risks and inevitably introduces some deviation from the actual accuracy of the proposed model. Nevertheless, our primary objective in this study is not to source the most up-to-date dataset, but rather to identify a more effective phishing detection method. Therefore, we have chosen to acknowledge this limitation for the time being. Future research, however, should be carried out with a significantly

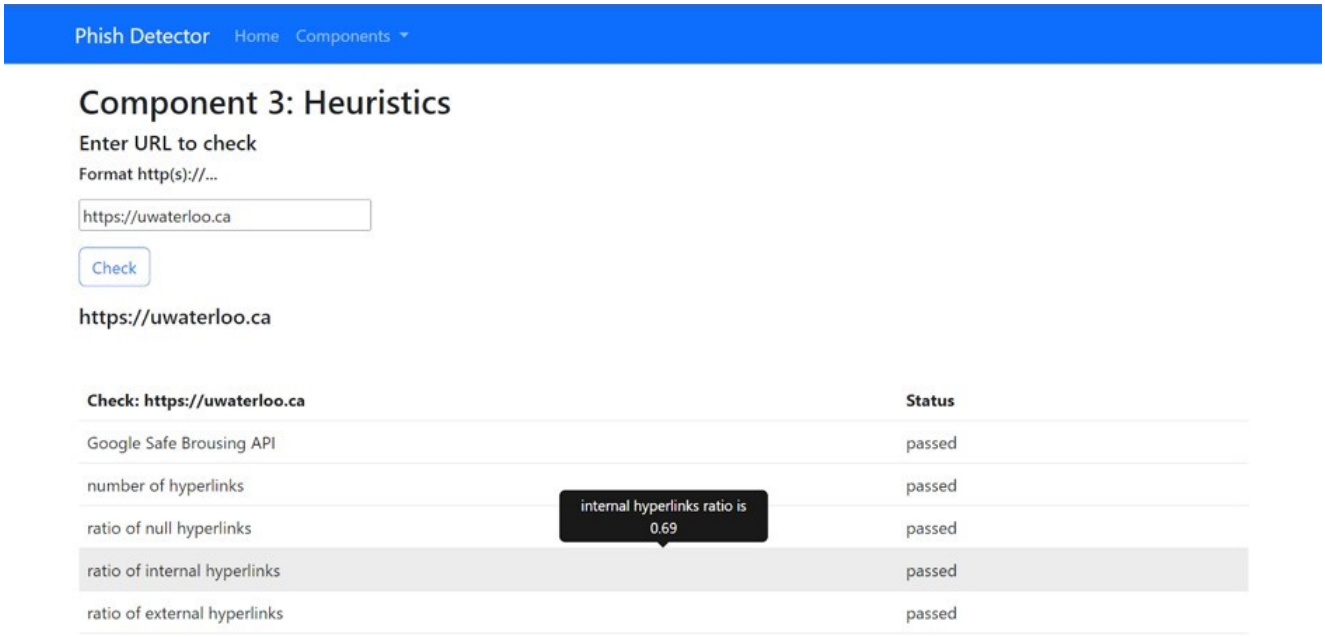

Fig. 4. Heuristic Phishing Websites Detection System User Interface.

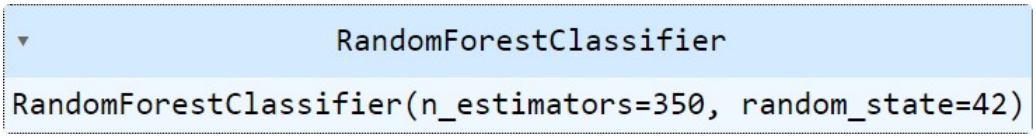

Fig. 5. Model Configuration parameters.

larger and more recent dataset. Figure 7 features a dataset heatmap which is a two-dimensional representation that shows the relationship between columns of data. This heatmap was generated using matplotlib Python library's pyplot plotting interface, helping us showcase the data's correlations better.

After training the RF classifier, the evaluation resulted in an accuracy score of 1.00000, which is equivalent to 100%. The confusion matrix displayed the following results:

$$[[57000] \quad [05700]] \tag{7}$$

This outcome indicates that the training was successful, with the model effectively identifying 50% of phishing and 50% of legitimate URLs. In Section 5 of this paper, we evaluate the proposed model's performance using the newest data from the testing split.

V. Detection Accuracy Comparison

VI. Detection Accuracy Comparison

In this Section we attempt to quantitatively evaluate the performance of both Heuristic system and Random Forest model developed in Section 3 and Section 4 respectively, by using several classic Machine Learning evaluation metrics discussed in [**?**]. The specific metrics and descriptions are given in Table 7. In this study, accuracy is utilized to evaluate the performances of proposed systems as follows:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \times 100\% \tag{8}$$

Another performance measure for a RF model is F defined as follows:

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall'} \tag{9}$$

where

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN} \tag{10}$$

Finally, the last performance metrics used for both solutions is average time taken to conduct a single entry test calculated as follows

$$t = t_n/n \tag{11}$$

where t_n is time taken to conduct n tests.

The Random Forest's performance in terms of F-measure and accuracy has shown that this ensemble Machine Learning technique was quite effective and reliable in detecting phishing websites and achieved better accuracy and took substantially less in terms of testing time than the heuristic-based system. Implementation details, resources utilized, challenges faced and results achieved after testing are summarized in Table 8.

As can be inferred from the Table 8, this study proves that Machine Learning methods excel in the detection of phishing web pages when compared to heuristic approaches. Both were evaluated using the same input data and selected features. This

Fig. 6. Dataset snippet.

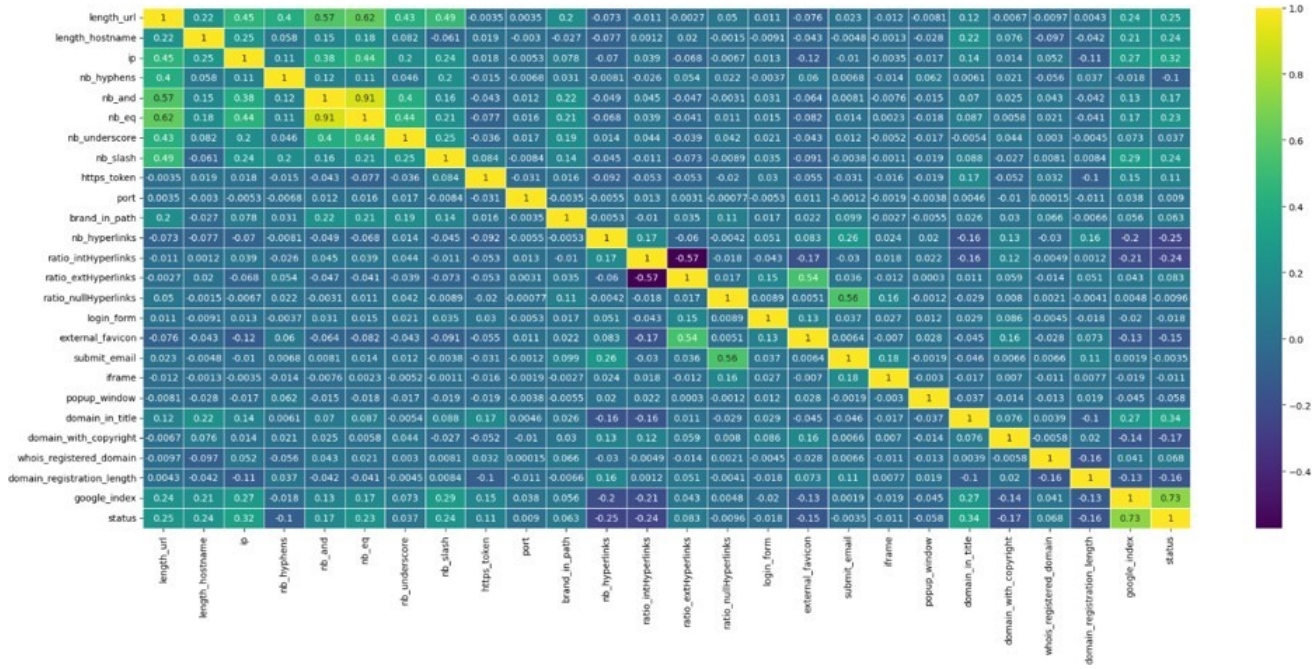

Fig. 7. Dataset columns correlation heatmap.

TABLE VII
PERFORMANCE EVALUATION METRICS

Metric	Description
TP	Number of websites correctly classified as Phishing in test set
TN	Number of websites correctly classified as Legitimate in test set
FP	Number of websites incorrectly classified as Phishing in test set
FN	Number of websites incorrectly classified as Legitimate in test set

study proves the superiority of Machine Learning models over traditional heuristics-based methods in the realm of phishing website detection. The tests involving the Random Forest classifier, although demanding slightly more computational power, showcased a substantial increase in detection accuracy (57.2% higher) and a remarkable reduction in testing time (11.27 seconds faster). Therefore it is evident that further research into phishing websites detection has to be primarily directed at identifying most efficient classifiers rather than attempting to perfect traditional heuristic-based solutions.

Regardless of the path we choose to advance the development of a web plugin aimed at analyzing web traffic to detect and prevent phishing, it's imperative to consider potential privacy concerns. Therefore, ensuring that any solution we

TABLE VIII
KEY FINDINGS

	Heuristics (URL + DOM + Domain)	**Machine Learning (Random Forest)**
Implementation	Flask Web App	Python scripts
Runtime environment	Web browser + virtual server	Kaggle platform
Minimal hardware requirements		
Storage	1 GB HDD/SSD	3 GB HDD/SSD
RAM	2 GB	8 GB
CPU	Pentium 4	Intel Core i5
Power supply	500 W	600 W
GPU	—	2 GB
Dataset		
Training split	—	11,412 URLs (mined May 2020)
Testing split	12 URLs (mined October 2023)	12 URLs (mined October 2023)
Selected Features	25 (see Table 4 for details)	25 (see Table 4 for details)
Performance metrics		
True Positives	1	6
True Negatives	6	0
False Positives	0	1
False Negatives	5	5
Precision	—	0.86
Recall	—	0.55
F-measure	—	0.67
Accuracy	58.33%	91.67%
Average test time	11.28 seconds	0.01 seconds
Challenges	Feature weighting Threshold identification	Dataset collection (web scraping)

create anonymizes user traffic is of paramount importance to avoid raising privacy-related concerns.

Phishing poses a significant and growing threat as we increasingly rely on technology, with adversaries using deceptive tactics to steal private data from users. This has serious implications, resulting in data breaches and financial losses for companies and institutions. Traditional approaches, based on human judgment and security policies, are not sufficient to combat this evolving threat. In response, automated technology-based solutions are needed to enhance our defense mechanisms.

In this study, exploration of heuristic approaches revealed that simply increasing the quantity of analyzed features doesn't guarantee improved detection accuracy. Instead, the unique impact of each feature on detection accuracy is crucial. Information Gain and Relief Ranking Filter concepts were employed to identify relevant features, resulting in the development of a web-based application that allowed manual testing across three categories: URL, DOM, and domain, totaling 25 website features.

Furthermore, adoption of a Machine Learning RF classifier model, utilizing a publicly available dataset, was made to intelligently detect phishing websites. To ensure compatibility, the number of website features in the dataset was adjusted to match the heuristic method, and the most up-to-date testing data, totaling 11,412 entries, was added. Our comprehensive study compared both methods head-to-head, demonstrating that the RF classifier outperforms heuristic approaches in terms of accuracy (heuristics: 58.33%, ML: 91.67%) and testing speed, although it has slightly higher computational requirements.

This research reinforces the notion that Machine Learning models significantly outperform traditional heuristic-based methods in the context of phishing website detection. This work highlights the need for continued research focused on identifying the most efficient classifiers and perfecting feature selection techniques. Efforts will be directed towards enhancing the results, ultimately leading to the development of next-generation solutions. Future endeavors will involve optimizing feature selection techniques outlined in Section 3, training models for maximum efficiency, minimal testing time, and resource usage. Additionally, focus will be on developing a web plugin for URL analysis, ensuring user traffic remains anonymized.

VII. CONCLUSION

Phishing poses a significant and growing threat as we increasingly rely on technology, with adversaries using deceptive tactics to steal private data from users. This has serious implications, resulting in data breaches and financial losses for companies and institutions. Traditional approaches, based on human judgment and security policies, are not sufficient to combat this evolving threat. In response, automated technology-based solutions are needed to enhance our defense mechanisms.

In this study, exploration of heuristic approaches revealed that simply increasing the quantity of analyzed features doesn't guarantee improved detection accuracy. Instead, the unique impact of each feature on detection accuracy is crucial. Information Gain and Relief Ranking Filter concepts were employed to identify relevant features, resulting in the development of a web-based application that allowed manual testing across three categories: URL, DOM, and domain, totaling 25 website features.

Furthermore, adoption of a Machine Learning RF classifier model, utilizing a publicly available dataset, was made to intelligently detect phishing websites. To ensure compatibility, the number of website features in the dataset was adjusted to match the heuristic method, and the most up-to-date testing data, totaling 11,412 entries, was added. Our comprehensive study compared both methods head-to-head, demonstrating that the RF classifier outperforms heuristic approaches in terms of accuracy (heuristics: 58.33%, ML: 91.67%) and testing speed, although it has slightly higher computational requirements.

This research reinforces the notion that Machine Learning models significantly outperform traditional heuristic-based methods in the context of phishing website detection. This work highlights the need for continued research focused on identifying the most efficient classifiers and perfecting feature selection techniques. Efforts will be directed towards enhancing the results, ultimately leading to the development of next-generation solutions. Future endeavors will involve optimizing feature selection techniques outlined in Section 3, training models for maximum efficiency, minimal testing time, and resource usage. Additionally, focus will be on developing a web plugin for URL analysis, ensuring user traffic remains anonymized.

ACKNOWLEDGMENT

The research was conducted at the University of Waterloo and endorsed by the Ministry of Education and Science of Ukraine. This project received funding and the MITACS Globalink Research Award. The views presented in this work are those of the authors and do not necessarily represent the perspective of the organization concerning their findings. The authors extend heartfelt gratitude to individuals who have played an instrumental role in making this work a reality, including Professor Serhiy Yarusevych, Dean Mary Wells, Kittiphon Phalakarn, Andrii Fesenko, Kristjan Krips, Tharun Abraham Aju, Ivanna Kvasna, and Andrea McKinney.

REFERENCES

[1] Positive Technologies, "Cybersecurity threatscape q1 2023 report," Positive Technologies, Technical Report, 2023, accessed: 2025-07-27. [Online]. Available: https://www.ptsecurity.com/ww-en/analytics/cybersecurity-threatscape-2023-q1/

[2] K. D. Mitnick, "Are you the weak link," *Harvard Business Review*, vol. 81, pp. 18–20, 2003.

[3] Federal Bureau of Investigation, "Internet crime report 2022," Internet Crime Complaint Center, Technical Report, 2022, accessed: 2025-07-27. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf

[4] Fortinet, "Fortinet secure web gateway," 2023, accessed: 2025-07-27. [Online]. Available: https://www.fortinet.com/products/secure-web-gateway/fortiproxy

[5] R. S. Rao and S. T. Ali, "Phishshield: A desktop application to detect phishing webpages through heuristic approach," *Procedia Computer Science*, vol. 54, pp. 147–156, 2015.

[6] A. Basit *et al.*, "A comprehensive survey of ai-enabled phishing attacks detection techniques," *Telecommunication Systems*, vol. 76, pp. 139–154, 2021.

[7] J. Poderys, M. Artuso, C. M. O. Lensbøl, H. L. Christiansen, and J. Soler, "Caching at the mobile edge: A practical implementation," *IEEE Access*, vol. 6, pp. 8630–8637, 2018.

[8] S. Buchyk, D. Shutenko, and S. Toliupa, "Phishing attacks detection," in *CEUR Workshop Proceedings*, vol. 3384, 2022, pp. 193–201, accessed: 2025-07-27. [Online]. Available: https://ceur-ws.org/Vol-3384/Short_7.pdf

[9] A. Subasi and E. Kremic, "Comparison of adaboost with multiboosting for phishing website detection," *Procedia Computer Science*, vol. 168, pp. 272–278, 2020.

[10] B. Wei *et al.*, "A deep-learning-driven light-weight phishing detection sensor," *Sensors*, vol. 19, p. 4258, 2019.

[11] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, "Ai meta-learners and extra trees algorithm for the detection of phishing websites," *IEEE Access*, vol. 8, pp. 142 532–142 542, 2020.

[12] M. Bhagwat, P. Patil, and T. Vishawanath, "A methodical overview on detection, identification and proactive prevention of phishing websites," in *Proc. 3rd Int. Conf. on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. IEEE, 2021, pp. 1505–1508.

[13] T. O. Ojewumi *et al.*, "Performance evaluation of machine learning tools for detection of phishing attacks on web pages," *Scientific African*, vol. 16, p. e01165, 2022.

[14] A. U. Zulkurnain, A. Hamidy, A. B. Husain, and H. Chizari, "Social engineering attack mitigation," *International Journal of Mathematics and Computational Science*, vol. 1, pp. 188–198, 2015.

[15] S. Shabudin, N. S. Sani, K. A. Z. Ariffin, and M. Aliff, "Feature selection for phishing website classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, 2020.

[16] C. Opara, Y. Chen, and B. Wei, "Look before you leap: Detecting phishing web pages by exploiting raw url and html characteristics," *Expert Systems with Applications*, vol. 236, p. 121183, 2023.

[17] R. Yang *et al.*, "Phishing website detection based on deep convolutional neural network and random forest ensemble learning," *Sensors*, vol. 21, p. 8281, 2021.

[18] E. Zhu *et al.*, "Ofs-nn: An effective phishing websites detection model based on optimal feature selection and neural network," *IEEE Access*, vol. 7, pp. 73 271–73 284, 2019.

[19] A. I. Pratiwi and Adiwijaya, "On the feature selection and classification based on information gain for document sentiment analysis," *Applied Computational Intelligence and Soft Computing*, vol. 2018, pp. 1–5, 2018.

[20] A. B. Altamimi *et al.*, "Phishcatcher: Client-side defense against web spoofing attacks using machine learning," *IEEE Access*, 2023.

[21] A. Hannousse and S. Yahiouche, "Web page phishing detection," https://data.mendeley.com/datasets/c2gw7fy2j4/3, 2021, accessed: 2025-07-27.

[22] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine Learning*, vol. 53, pp. 23–69, 2003.

[23] D. Shutenko, "Hybrid system for phishing website detection," https://github.com/dimashutenko/Hybrid-system-for-phishing-website-detection, 2023, accessed: 2025-07-27.

[24] H. Guo *et al.*, "Forecasting mining capital cost for open-pit mining projects based on artificial neural network approach," *Resources Policy*, vol. 74, p. 101474, 2021.