*DOI: 10.15849/IJASCA.250730.15*

*Int. J. Advance Soft Compu. Appl, Vol. 17, No. 2, July 2025*
*Print ISSN: 2710-1274, Online ISSN: 2074-8523*
*Copyright © Al-Zaytoonah University of Jordan (ZUJ)*

# Web Phishing Detection and Awareness Utilizing Hybrid Machine Learning Algorithms

**Omar Alheyasat[1,2]**

[1]Electrical Engineering Department, Faculty of Engineering, Al-Balqa Applied University, Al-Salt, Jordan
e-mail: omarah@bau.edu.jo
[2]Software Engineering Department, Faculty of Science and Information Technology,
Al Zaytoonah University of Jordan, Amman, Jordan
e-mail: o.heyasat@zuj.edu.jo

**Abstract**

***The growing risks of web phishing attacks necessitate advanced detection mechanisms to safeguard users from phishing sites. This paper proposes a Layered Web Phishing Detection (LWPD) system that combines hybrid machine learning algorithms with real-time user awareness mechanisms. The system involves two stages: (1) a hybrid ML model integrating unsupervised clustering (K-means) and supervised classification (Random Forest, XGBoost) to detect phishing URLs with 99.7% accuracy after feature engineering; and (2) a browser extension that alerts users in real time regarding suspicious links. Experiments with a balanced dataset (11,430 URLs, 87 features) demonstrate that feature reduction to 34 significant attributes enhances model performance without compromising high precision. The method put forward in this paper outperforms conventional ML methods by reducing feature dimensionality by 60% at the expense of just a 2% loss in accuracy. This study bridges the gap between theoretical detection and actual user awareness in the real world by presenting a viable solution for cybersecurity application.***

**Keywords**: *Phishing detection, hybrid machine learning, feature engineering, browser extension, cybersecurity.*

## 1 Introduction

Phishing attacks represent one of the most devious and damaging fronts of modern fraud. Phishing is defined as a method to fraud users to obtain sensitive information utilizing fake sites, accounts or services. They are easy to implement since they do not require deep technical knowledge, which makes it difficult to defend against them. Phishing is a form of online deception that takes advantage of people's trust, who often fall prey as attackers

***Received 4 March 2025; Accepted 10 June 2025***

impersonate someone reputable. Also, messages which feel genuine often send us to places subconsciously marked as trustworthy [1].

According to Egress Cybersecurity [27], Phishing schemes are rapidly evolving and are getting more sophisticated with time; which makes it difficult for existing detection systems and blacklists to detect them. Similarly, the 2024 Industry Cybersecurity Journal [28] shows that phishing is a significant contributor to data breaches, especially where sensitive data is involved. This necessitates the creation of more reliable tools. In response, recent literature has increasingly explored the adoption of machine learning and deep learning tools to improve the effectiveness of phishing detection systems. Machine learning models have shown improved performance in identifying underlying patterns and trends that traditional models would normally miss, particularly in cryptographic data [2, 3]. Analyzing features through cryptography is promising for detection of hidden patterns related to such attacks. Although some similar efforts have been made, there remains significant room for improvement [4].

The financial impact of phishing is significant. It has been reported in 2021 that billions of dollars have been lost duo to web phishing attacks. In addition, the report showed that more than 80% of the phishing attacks attempts to obtain users credentials, such as usernames and passwords. [5]. These credentials are used to access different services, such as emails and banking accounts. A wide range of methods and techniques have been proposed to address phishing [6-8]. However, since phishing depends on social engineering, it is easy to modify its techniques, which makes the issue difficult to control.

In this study, phishing attacks are divided into two main levels; detection and awareness. For the detection component, a new Layered Web Phishing Detection (LWPD) hybrid machine learning algorithm is proposed to tackle the phishing. This approach makes use of a combination of supervised and unsupervised learning models. For the awareness component, a new browser extension (BE) is implemented based on the (LWPD) algorithm to aware the user about any link before loading it. This approach is designed to improve users' understanding of phishing techniques and improve real-world detection. The suggested approach is both feasible and straightforward to apply.

The remainder of this paper is organized as follows; section 2 introduces the main related work that has been conducted in the area of web phishing detection. Section 3 overviews

the proposed method. In section 4, the experiment will be explained. The results will be discussed in section 5. Finally, a summary of the paper is provided in section 6.

## 2 Related Work

Phishing attacks propose threats to digital security on different levels, including users, organizations, and even governments. In recent years, there has been focus in the literature on prevention and detection of phishing. A strand of the literature has focused on applying traditional machine learning algorithms to classify phishing websites through a number of features. For instance, in [9], RandomForest, FilteredClassifier, and J-48 were found to be the best classifiers for phishing detection. The InfoGainAttributeEval method was applied for feature selection. Similarly, in [10], the Random Forest learning algorithm was used and showed a 99% detection accuracy. Furthermore, [11] identified common characteristics found in phishing websites, and also found that the Random Forest showed the best performance in detection with 94% accuracy. In [12], a large number of machine learning models were trialed separately for the detection of phishing, additionally, different combinations of machine learning models were applied for the same purpose.

A number of studies have focused on the awareness of users to phishing websites. In [13], a gamification approach was used to improve awareness. Furthermore, in [14], the psychological and behavioral features that affect the susceptibility of users to fall for a phishing attack has been discussed. Furthermore, in [16], Gradient-Boosted Decision Trees (GBDT) algorithm was applied to multidimensional features to predict user susceptibility to phishing. The results showed an accuracy of 89.04%. These studies highlight the importance of focusing on human factors alongside the technical defenses.

Another strand of the literature has explored the use of deep learning for phishing detection and awareness. For example, in [15], A combined classification framework utilizing CNNs and LSTM networks was introduced to identify phishing attacks that affects the images and text frames in the locator of the universal resource. The accuracy rate of the new algorithm (IPDS) was 93.28%, with an average detection time of 5 seconds. Additionally, in [17], deep learning models were applied for detection of web attacks. The Recurrent Neural Network (RNN) algorithm showed better performance over Artificial Neural Networks (ANNs), with 94% accuracy. Furthermore, [18] implemented a feature-based hybrid model using XGBoost for detection of phishing websites that focused on the real time detection of new phishing websites. The model achieved an accuracy of 99.17%. In the work of [25], a convolutional neural network was proposed to identify phishing websites through a deep learning-based framework. Meanwhile, in [26], strengths and

limitations of deep learning techniques were demonstrated as a survey that was used in phishing detection.

Besides websites, studies have explored phishing detection in other digital environments. For instance, [19] and [20] used machine learning techniques and neural networks to detect phishing in emails and websites, respectively. In [21], a detection methodology based on probabilistic semantic analysis was conducted. On the other hand, [22] explored SMS phishing detection using integrated pre-language transformer. Furthermore, in [23], Artificial intelligence techniques were utilized to identify malware on Android-based smartphones. In contrast, [24] utilized a semi-supervised majority voting mechanism to detect phishing attacks within the education domain.

Beyond detection algorithms, a number of studies focused on system-level mitigation strategies. For instance, in [29], a block chain technology was proposed for phishing detection and prevention by improving online transactions' security. Meanwhile, [30], demonstrated a zero-trust architecture and an evaluation for mitigation phishing attacks. In [31], the author proposed a stacked phishing detection model that utilized multi-machine learning models. This stacked structure has been utilized in different applications [32]. However, the authors did not mention how many algorithms should be stacked and what is the impact of supervised and unsupervised algorithms in the stack. In [33, 34], the authors proposed a detection mechanism utilizing SDN. The method utilizes a classical machine learning model implemented in the network controller.

Many detection methods were applied in the literature, showing strong results in both traditional machine learning and deep learning algorithms. Most studies were found to focus on single-model applications, with only a few combining unsupervised and supervised methods. Additionally, there is a clear scarcity in literature incorporating real-time awareness for users. In this study, a Layered Web Phishing Detection (LWPD) algorithm is proposed which addresses both detection and awareness. The (LPWD) algorithm integrates unsupervised learning in the first layer, and supervised learning in the second layer, and the output of the first layer is used as the input in the second layer. In addition, the new framework utilizes the output of the prediction model to alert the user about the link contents on real time.

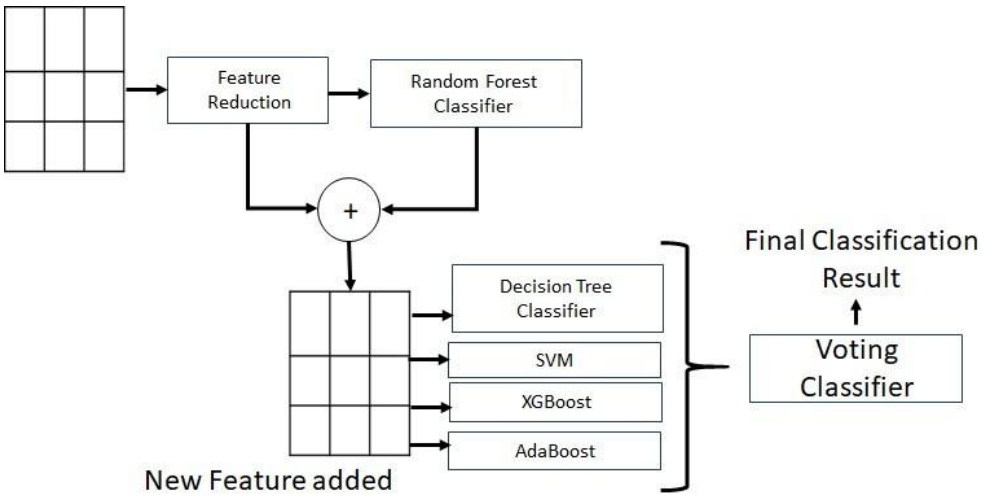
# 3 The Proposed Framework

This framework consists of two parts; detection algorithm and awareness method. In the detection part, a Layered Phishing Web Detection (LPWD) hybrid machine learning algorithm is proposed. This algorithm consists of two layers. In the first layer, the input features that extracted from URL link, page content, and external services are fed into an

unsupervised machine learning algorithm to cluster the URLs into normal or phished links (0, 1). In the second layer, the clustered output from the first layer is added to the features as a new feature, and that output is fed into a supervised machine learning algorithm. The proposed (LWPD) algorithm is shown in Fig.1.

Fig.1.The proposed (LWPD) algorithm

In the second part of this work (the awareness method) is proposed. The idea of this method




is to implement a browser extension (BE). This (BE) will be used to aware the user about the safety of any link before loading it. This stage will be done based on the output of the (LWPD) algorithm in order to give an alert message to the user if the URL is a phished link or not. The awareness method is shown in the right side of Fig.2. The proposed method of detection and awareness is shown in Fig.2.

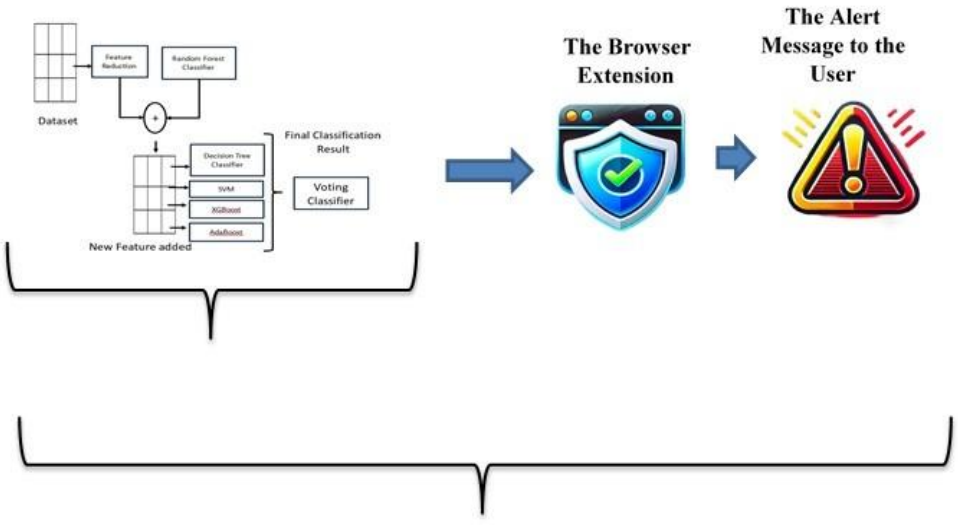



Fig. 2. The Proposed Platform

The proposed framework in this work was implemented and tested using two levels. The first level, the detection method was to test as inputs all the 87 extracted features of the data set. The list of all features and their types is shown in the experiment section. The second level of the framework is the awareness method, and it consist of the (BE) and the alert message to the user, the second level will be discussed in the next section.

# 4 Experimentation

This experiment consists of two main parts. In the first part, a hybrid machine learning method is constructed and trained to detect phished websites. The second part involves developing a website add-in that automatically detects links in any opened web pages, allowing users to determine whether a link is phished or not and providing alerts for detected phishing sites.

## 4.1 The Dataset

The experiments in this work were conducted based on a publicly available dataset from [31] specifically designed for machine learning-based phishing detection system benchmarking. The dataset comprises 11,430 URLs distributed equally in terms of the phishing and legitimate scenarios for a balanced dataset to facilitate the training and testing of robust models. Each URL is represented by 87 features that are classified into three classes as shown in the following Table 1.

Table 1: Features description used in dataset

| **Category Of the features** | **No. of Features Extracted** | **Description of the features** |
|---|---|---|
| **features of the syntax and structure of the URLs** | 56 | re all explicitly extracted from the URL composition, such as its length, special character use, and the morphology of the URL itself. |
| **features of the content of their corresponding web pages** | 24 | ted from the actual content of the webpage corresponding to the URL, capturing elements such as HTML tags, embedded links, and text patterns. |
| **features acquired when making a query to external services** | 7 | ıed by querying external resources and services, such as domain reputation databases and WHOIS records, to provide additional context about the URL |

Distribution of the dataset is one of the advantages of the dataset, balance representation of the dataset with fair evaluation by machine learning models. It includes the same ratio of 50% phishing URLs and 50% legitimate URLs, minimizing bias and improving robustness of research findings on phishing detection. The next table shows the features names and type as shown in Table 2:

Table 2. Names and types of the features

| Attribute_Name | Data_Type | Attribute_Name_Modified | Data_Type |
|---|---|---|---|
| web_address_id | Integer | max_word_count_raw | Float |
| url_size | Integer | max_token_in_host | Float |
| hostname_length | Integer | max_token_raw | Float |
| ip_address | Integer | avg_token_count | Float |
| dot_count | Integer | avg_token_path | Float |
| hyphen_count | Integer | phishing_signals | Integer |
| question_mark_count | Integer | domain_appears_in_brand | Integer |
| and_operator_count | Integer | brand_in_subdomain | Integer |
| or_operator_count | Integer | suspicious_domain_extension | Integer |
| equal_sign_count | Integer | anomaly_report | Integer |
| tilde_count | Integer | hyperlink_count | Integer |
| percent_sign_count | Integer | internal_link_ratio | Float |
| slash_count | Integer | external_link_ratio | Float |
| colon_count | Integer | internal_css_count | Integer |
| comma_count | Integer | multi_link_ratio | Float |
| semicolon_count | Integer | external_redirect_ratio | Float |
| dollar_sign_count | Integer | internal_redirect_ratio | Float |
| space_count | Integer | internal_error_ratio | Float |
| token_count | Integer | has_login_form | Integer |
| comma_token_count | Integer | external_favicon_detected | Integer |
| http_token_count | Integer | email_submission_possible | Integer |
| https_keyword_ratio | Integer | internal_media_ratio | Float |
| digit_ratio_in_url | Integer | suspicious_form_handler | Integer |
| digit_ratio_in_host | Integer | embedded_iframes_present | Integer |
| punycode_usage | Integer | popup_frequency | Integer |
| extension_in_path | Integer | safe_anchor_ratio | Float |
| non_standard_subdomain | Integer | mouseover_script_usage | Integer |
| subdomain_count | Integer | right_click_disabled | Integer |
| randomized_domain_name | Integer | empty_title_flag | Integer |
| shortening_service_in_path | Integer | multiple_tlds_in_domain | Integer |
| redirect_count | Integer | copyright_info_in_domain | Integer |
| no_word_length | Integer | indexed_by_google | Integer |
| repeated_characters | Integer | domain_age_in_days | Integer |
| max_word_in_host | Integer | site_traffic_level | Integer |
| min_word_in_path | Integer | dns_data_available | Integer |

Before training the model, and for the purpose of data preprocessing, the data set was heavily preprocessed and sorted to ensure integrity of data. Missing values were handled and all inconsistencies within the feature space were addressed in order to preserve the integrity of the data set. The composition of the data set (50% phishing URLs and 50% legitimate URLs) was maintained while preprocessing the data to avoid bringing bias in when training as well as evaluating the models. In this study, the dataset initially consisted of 82 features. To refine the feature set, a correlation-based filtering approach was employed. Initially, features with a correlation coefficient greater than 0.5 with the target variable were selected; however, only one feature met this criterion. The threshold was then reduced to 0.3, yielding six features. Further lowering the threshold to 0.1 resulted in a total of 21 selected features.

To introduce an additional informative feature, The dataset was used to train a Random Forest classification model. The trained model was then used to predict the target variable for the entire dataset, achieving an accuracy of 91%. The predicted output was subsequently added as a new feature, increasing the total number of features to 22.
With the enhanced dataset, multiple supervised machine learning models were trained to improve predictive performance. To further optimize the final classification accuracy, an ensemble learning approach was adopted using a voting classifier. Hard voting was applied to combine predictions from four models: Random Forest Classifier, XGBClassifier, DecisionTreeClassifier, and Support Vector Machine (SVM). The models were trained and assessed using a 10-fold cross-validation approach.
To evaluate the effectiveness of the final model, accuracy, precision, recall, and F1-score were employed as assessment criteria. These results were compared against classical machine learning models to highlight improvements achieved through the proposed methodology.

The performance metrics are defined as follows:

Accuracy refers to the percentage of total instances that were correctly identified by the model. Equ.1 shows accuracy calculation

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

Precision measures how many of the instances labeled as positive by the model were actually correct. Equ.2 shows precision calculation

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

Recall: Frequently referred to as sensitivity or the true positive rate, quantifies the fraction of real positive cases that the model successfully detected.

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

The F1-score: represents the harmonic mean of precision and recall, offering a balanced measure of both.

$$F1 - Score = 2 * \frac{Precision*Recall}{Precision+Recall} \qquad (4)$$

Table 3. Variables names and their description

| Variable | Description |
|---|---|
| TP | Correctly predicted positive instances. |
| TN | Correctly predicted negative instances. |
| FP | Incorrectly predicted positive instances. |
| FN | Incorrectly predicted negative instances. |

Python and JavaScript have been used to create a browser add-in that reads the HTML of the retrieved webpage, extracts all links in the `<a>` anchor elements, and uses the hybrid model to check for phishing detection. If any of these links are identified as phished, an alert is shown to the user. To evaluate the add-in, a new HTML page was created containing 20 normal links and 3 phished links.

# 5 Results and Discussions

The assessment of the proposed method commenced with the use of a downloaded dataset containing 87 features to train and test various classical supervised and unsupervised machine learning models. Table 1 shows the results of 6 classical binary classifiers supervised machine learning algorithms and one unsupervised model. We can observe from the table that unsupervised K-means model reported that lowest accuracy with 56% and XGBoost and Random frost reported over than 96%. This shows how supervised models overcome the K-means unsupervised model.

Table 4. The results of 7 supervised classical binary classifiers and one supervised model.

| Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| SVM | 0.953919 | 0.954441 | 0.953261 | 0.953821 |
| Kmeans | 0.563246 | 0.869105 | 0.328568 | 0.408296 |
| Random Forest | 0.962669 | 0.964498 | 0.966113 | 0.962117 |
| XGBoost | 0.965702 | 0.965880 | 0.965412 | 0.965618 |
| AdaBoost | 0.935137 | 0.941550 | 0.927788 | 0.934551 |
| Gradient Boosting | 0.956136 | 0.956079 | 0.956064 | 0.956176 |
| Decision Tree | 0.930823 | 0.935353 | 0.931061 | 0.932045 |

In the second scenario, the correlation between the features and the output binary classification has been calculated. The heat map in figure 3 shows this correlation. To select less number of features, a filter has been leveraged to select the features with higher correlation value. A value of 0.5 has been applied. However, only one feature has a correlation value over 0.5, which is the “google index”. The filter has been reduced to 0.3. 5 features are reported; “IP”, “ratio digits URL”, “phish hints”, domain in title and “google index”. These features were used to train the same machine learning models, with their accuracy reported in Table 5.

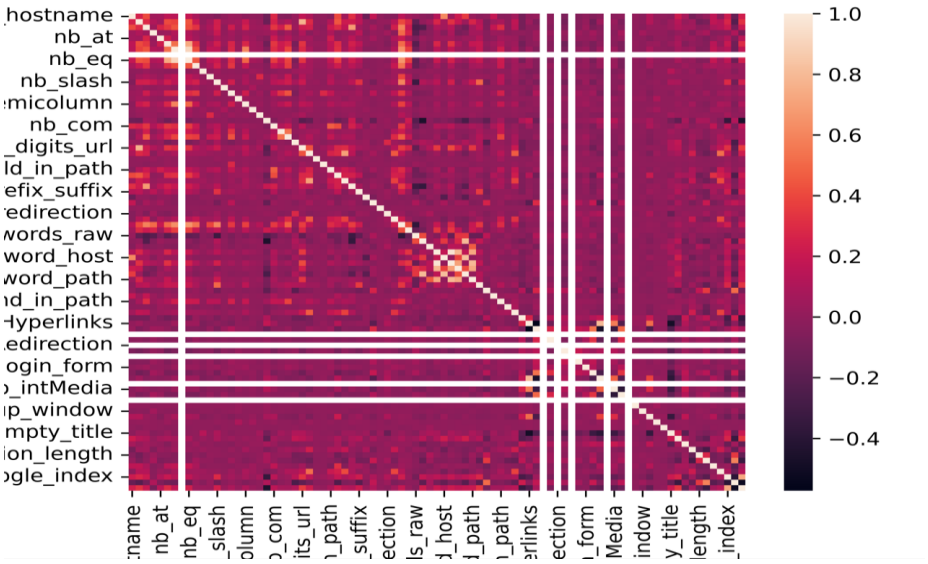



Fig 3. The heat map and the correlation between the input features and the resulting output binary classification.

The table clearly indicates that the accuracy of the models has been reduced for the supervised and for the unsupervised models. A reduction of over 8% has been reported for all classical models. To enhance these models, the filter value has been reduced to 0.1.

Table 5. The accuracy of the different models

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| SVM | 0.884739 | 0.884060 | 0.885257 | 0.884587 |
| Kmeans | 0.431738 | 0.524284 | 0.640592 | 0.516653 |
| Random Forest | 0.870040 | 0.870742 | 0.868662 | 0.869592 |
| XGBoost | 0.883690 | 0.877396 | 0.891800 | 0.884429 |
| AdaBoost | 0.866308 | 0.843627 | 0.899514 | 0.870391 |
| Gradient Boosting | 0.887423 | 0.883618 | 0.892034 | 0.887726 |
| Decision Tree | 0.867708 | 0.875130 | 0.856511 | 0.865989 |

34 features have been selected as shown in table 3.

Table 6. The names of the 34 selected features and their accuracy.

| **Feature Name** | **Accuracy** |
|---|---|
| Length url | 0.248580 |
| Length hostname | 0.238322 |
| ip | 0.321698 |
| Nb dots | 0.207029 |
| Nb at | 0.142915 |
| Nb qm | 0.294319 |
| Nb and | 0.170546 |
| Nb eq | 0.233386 |
| Nb slash | 0.242270 |
| Nb semicolumn | 0.103554 |
| Nb com | 0.156284 |

| | |
|---|---|
| https token | 0.114669 |
| Ratio digits_url | 0.356395 |
| Ratio digits_host | 0.224335 |
| Tld in subdomain | 0.208884 |
| Abnormal subdomain | 0.128160 |
| Nb subdomains | 0.112891 |
| Prefix suffix | 0.214681 |
| Shortening service | 0.106120 |
| Length words raw | 0.192010 |
| Shortest word host | 0.223084 |
| Longest words raw | 0.200147 |
| Longest word host | 0.124516 |
| Longest word path | 0.212709 |
| Avg words raw | 0.167564 |
| Avg word host | 0.193502 |
| Avg word path | 0.197256 |
| Phish hints | 0.335393 |
| Suspicious tld | 0.110090 |
| Statistical report | 0.143944 |
| Empty title | 0.207043 |
| Domain in title | 0.342807 |
| Dns record | 0.122119 |
| Google index | 0.731171 |

Table 7 demonstrates how accurate the same models are when applied to these features. The table reveals the accuracy levels achieved by the models has enhanced with more than 5% and that the reported accuracy is less than 2% of the same models with all the features. We have selected these features to train the hybrid model. Table 8 shows the accuracy of the hybrid model

Table 7. The accuracy of the used models

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| **SVM** | 0.920904 | 0.917685 | 0.924517 | 0.921059 |
| **Kmeans** | 0.572227 | 0.873886 | 0.345625 | 0.424398 |
| **Random Forest** | 0.931287 | 0.932076 | 0.932696 | 0.934350 |
| **XGBoost** | 0.935837 | 0.940363 | 0.930593 | 0.935408 |
| **AdaBoost** | 0.891974 | 0.879784 | 0.907690 | 0.893486 |
| **Gradient Boosting** | 0.921371 | 0.920111 | 0.922413 | 0.921342 |
| **Decision Tree** | 0.890106 | 0.890462 | 0.888526 | 0.889886 |

We can observe form the table that an accuracy of 99.7% has been reported with less than 50% of the features. In addition, a 91% accuracy has been reported with only 5 features. Which means a reduction of 5% of the classical models with only 8% of the full feature set. We utilized this model in the implementation process of our awareness add-ins extension of the browser since it's simpler and viable.

Table 8. Accuracy after adding the filters

| Filter | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| >0.1 | 0.997201 | 0.998131 | 0.995793 | 0.997076 |
| >0.3 | 0.912505 | 0.912624 | 0.912832 | 0.912394 |

Finally, the model has been implemented in JavaScript as a browser extension named "ad-ins." The extension scans all <a> tags on the page to identify external links (i.e., links that start with "http" or "https"). If a link is detected as a phishing attempt, an alert is displayed. Figure 4 shows a test webpage created with 20 embedded links. Figure 5 illustrates how the extension generates an alert for a detected phishing link. Additionally, the alert includes the estimated probability that the link is malicious.


Fig 4. Test Webpage


Figure 5. The Browser Extension Alert for one of the links in the page

# 6 Conclusion

In this paper, a web phishing detection and awareness platform was introduced, consisting of two key modules. The first module is a hybrid multi-level machine learning algorithm that integrates unsupervised and supervised learning in order to detect web phishing. The

second module is based on the output of the first module, where a browser extension was developed for improving users' awareness of the phished links by giving an alert message when a malicious link is encountered in real-time.

The findings from the experiments strongly highlight the importance of feature engineering in the design of reliable web phishing detection systems. The inclusion of an augmented feature, which was derived from the original feature set, greatly improved the model's capability to distinguish phishing URLs from genuine ones. The enhancement was consistent across different machine learning approaches, which indicated the generalizability and robustness of the proposed approach.

The two-stage phishing detection algorithm presented in this study provides a strong foundation for the improvement of the accuracy of machine learning-based phishing detectors. The integration of real-time user awareness enhances its practical applicability. Overall, this research supports the advancement of robust, accurate, and data-centric systems for identifying phishing websites, providing meaningful guidance for both cybersecurity professionals and academic investigators.

**ACKNOWLEDGEMENTS**
This research was carried out on sabbatical leave at Al-Zaytoonah University of Jordan in the 2024/2025 academic year. The author would like to thank both Al-Balqa Applied University and Al-Zaytoonah University of Jordan for their continued support.

# References

[1] Alqahtani, S., Alzahrani, A., Algarny, S., et al. (2024). Phishing detection using deep learning: A systematic review. Electronics, 13(19), 3823.
[2] Aljofey, A., Alruwaili, M., Alsaedi, O., et al. (2024). A survey on phishing detection using machine learning: Recent advances and future directions. Artificial Intelligence Review.
[3] Liu, H., Huang, J., & Wang, Y. (2024). AntiPhishStack: An LSTM-based stacked generalization model for phishing URL detection. arXiv preprint, arXiv:2401.08947.
[4] Rathore, S., Jang, J., & Park, J. (2024). Evaluating robustness of phishing webpage detection models against adversarial attacks. arXiv preprint, arXiv:2407.20361.
[5] Egress Cybersecurity. (2021). Phishing Statistics.
[6] Alazaidah, R., Al-Shaikh, A., Al-Mousa, M. R., et al. (2024). Website phishing detection using machine learning techniques. Journal of Statistics Applications & Probability, 13(1).
[7] Zara, U., Ayyub, K., Khan, H. U., et al. (2024). Phishing website detection using deep learning models. IEEE Access, 12, 167072–167087.
[8] Helali, R. G. M. (2024). Phishing detection using hybrid machine learning techniques. Zhongguo Kuangye Daxue Xuebao, 29(2), 45–52.
[9] Rizky, R., & Hakim, Z. (2019). Analysis and design of VoIP server using Asterisk in Statistics and Statistical Informatics Communication of Banten Province using Ppdioo Method. Journal of Physics: Conference Series, 1179(1), 012160.
[10] Hanson, W. A., & Kalyanam, K. (2020). Internet Marketing and E-commerce (Student ed.). Thomson/South-Western.

[11] Sellars, A. (2018). Twenty years of web scraping and the Computer Fraud and Abuse Act. Boston University Journal of Science & Technology Law, 24, 372–398.
[12] Adebowale, M. A., et al. (2020). Intelligent phishing detection scheme using deep learning algorithms. Journal of Enterprise Information Management.
[13] Kumar, S., & Singh, A. (2023). Enhancing user awareness of phishing attacks: A gamification approach. Computers & Security, 124, 102987.
[14] Vishwanath, A., & Zhang, Y. (2023). The role of human factors in phishing susceptibility: A review and research agenda. Journal of Cybersecurity, 9(1).
[15] Karim, A., Shahroz, M., Mustofa, K., et al. (2023). Phishing detection system through hybrid machine learning based on URL. IEEE Access, 11, 36805–36822.
[16] Yang, R., Zheng, K., Wu, B., et al. (2022). Predicting user susceptibility to phishing based on multidimensional features. Computational Intelligence and Neuroscience, 2022.
[17] Christy Eunaicy, J. J., & Suguna, S. (2022). Web attack detection using deep learning models. Materials Today: Proceedings.
[18] Das Guptta, S., Shahriar, K. T., et al. (2022). Modeling hybrid feature-based phishing website detection using machine learning techniques. Annals of Data Science.
[19] Unnithan, N. A., Harikrishnan, N. B., et al. (2018). Detecting phishing emails using machine learning techniques. In Proceedings of the 1st Anti-Phishing Shared Task Pilot 4th ACM IWSPA Co-Located 8th ACM Conference on Data and Application Security and Privacy (CODASPY).
[20] Rangapur, A., & Jubilson, A. (2021). Precise URL phishing detection using neural networks. arXiv preprint, arXiv:2110.13424.
[21] Ramanathan, V., & Wechsler, H. (2012). phishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training. EURASIP Journal on Information Security, 2012(1), 1–22.
[22] Ulfath, R. E., Alqahtani, H., et al. (2022). Hybrid CNN-GRU framework with integrated pre-trained language transformer for SMS phishing detection.
[23] Alkahtani, H., & Aldhyani, T. H. H. (2022). Artificial intelligence algorithms for malware detection in Android-operated mobile devices. Sensors, 22(6), 2268.
[24] Yin, X., & Zheng, X. (2022). A semisupervised majority weighted vote antiphishing attacks IDS for the education industry. Computational Intelligence and Neuroscience, 2022.
[25] Yin, X., & Zheng, X. (2022). A semisupervised majority weighted vote antiphishing attacks IDS for the education industry. Computational Intelligence and Neuroscience, 2022.
[26] Alkahtani, H., & Aldhyani, T. H. H. (2022). Artificial intelligence algorithms for malware detection in Android-operated mobile devices. Sensors, 22(6), 2268.
[27] Egress Cybersecurity. (2021). Phishing Statistics.
[28] Industry Cybersecurity Journal. (2024). Phishing and Data Breaches Report.
[29] (2023). Blockchain-based phishing detection and prevention methods.
[30] (2024). Zero-trust architecture evaluation for phishing attack mitigation.

[31] Masoud, M., Jaradat, Y., & Alsakarnah, R. (2022). A non-content multilayers hybrid machine learning web phishing detection model. International Review on Modelling and Simulations (IREMOS), 15(2), 108–115.

[32] Jannoud, I., Masoud, M. Z., Jaradat, Y., Manaserah, A., & Zaidan, D. (2024). A multi-layered hybrid machine learning algorithm (MLHA) for type II diabetes classification. Procedia Computer Science, 237, 445–452.

[33] Masoud, M., Jaradat, Y., & Jannoud, I. (2017). On detecting Wi-Fi unauthorized access utilizing software defined network (SDN) and machine learning algorithms. International Review on Computers and Software.

[34] Masoud, M., Jaradat, Y., & Ahmad, A. Q. (2016). On tackling social engineering web phishing attacks utilizing software defined networks (SDN) approach. In 2016 2nd International Conference on Open Source Software Computing (OSSCOM) (pp. 1–6). IEEE.

[35] Kaggle. (n.d.). Web Page Phishing Detection Dataset. Retrieved from https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset