I.J. Wireless and Microwave Technologies, **2024, 6, 48-67**
Published Online on December 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2024.06.04

Enhanced Phishing URLs Detection using Feature Selection and Machine Learning Approaches

Dharmaraj R. Patil*
Department of Computer Engineering, R.C.Patel Institute of Technology, Shirpur, India
E-mail: dharmaraj.patil@rcpit.ac.in
ORCID iD: https://orcid.org/0000-0001-7634-2769
*Corresponding Author

Rajnikant B. Wagh
Department of Computer Engineering, R.C.Patel Institute of Technology, Shirpur, India
E-mail: rajnikantw@gmail.com
ORCID iD: https://orcid.org/0000-0003-2997-6034

Vipul D. Punjabi
Department of Computer Engineering, R.C.Patel Institute of Technology, Shirpur, India
E-mail: vipul.punjabi@rcpit.ac.in
ORCID iD: https://orcid.org/0000-0003-4221-7657

Shailendra M. Pardeshi
Department of Computer Engineering, R.C.Patel Institute of Technology, Shirpur, India
E-mail: shailendra.pardeshi@rcpit.ac.in
ORCID iD: https://orcid.org/0000-0001-6297-6074

Received: 10 June, 2024; Revised: 15 August, 2024; Accepted: 17 October, 2024; Published: 08 December, 2024

Abstract: Phishing threats continue to compromise online security by using deceptive URLs to lure users and extract sensitive information. This paper presents a method for detecting phishing URLs that employs optimal feature selection techniques to improve detection system accuracy and efficiency. The proposed approach aims to enhance performance by identifying the most relevant features from a comprehensive set and applying various machine learning algorithms, including Decision Trees, XGBoost, Random Forest, Extra Trees, Logistic Regression, AdaBoost, and K-Nearest Neighbors. Key features are selected from an extensive feature set using techniques such as information gain, information gain ratio, and chi-square (χ^2). Evaluation results indicate promising outcomes, with the potential to surpass existing methods. The Extra Trees classifier, combined with the chi-square feature selection method, achieved an accuracy, precision, recall, and F-measure of 98.23% using a subset of 28 features out of a total of 48. Integrating optimal feature selection not only reduces computational demands but also enhances the effectiveness of phishing URL detection systems.

Index Terms: Phishing Website Detection, Web Security, Feature Selection, Machine Learning, Cyber Security.

1. Introduction

The rapid expansion of internet usage has transformed how individuals and organizations manage daily activities, driving increased convenience and connectivity in communication and commerce [1, 2, 3, 4]. However, this surge in online interactions has also led to a significant rise in cybersecurity threats, with phishing attacks becoming a widespread and sophisticated hazard. Phishing, a deceptive tactic in which attackers impersonate trusted entities to trick users into revealing sensitive information, poses serious risks to the confidentiality and integrity of both personal and organizational data [5, 6, 7, 8]. While the digital era has offered unparalleled advantages, it has also been marked by an alarming increase in cyber threats, with phishing attacks standing out as a persistent and evolving danger [9].

Traditional phishing detection methods often struggle to keep pace with the constantly changing tactics used by cybercriminals, highlighting the need for more advanced and adaptive detection mechanisms [11]. As phishing techniques become more sophisticated, effective countermeasures are essential. In this context, feature selection—an

integral component of machine learning—plays a vital role by optimizing datasets to identify and prioritize the features most indicative of phishing characteristics [12]. According to the APWG Phishing Activity Report for the Second Quarter of 2023, notable patterns in phishing attacks emerged during this period, underscoring the continued evolution of these threats [13].

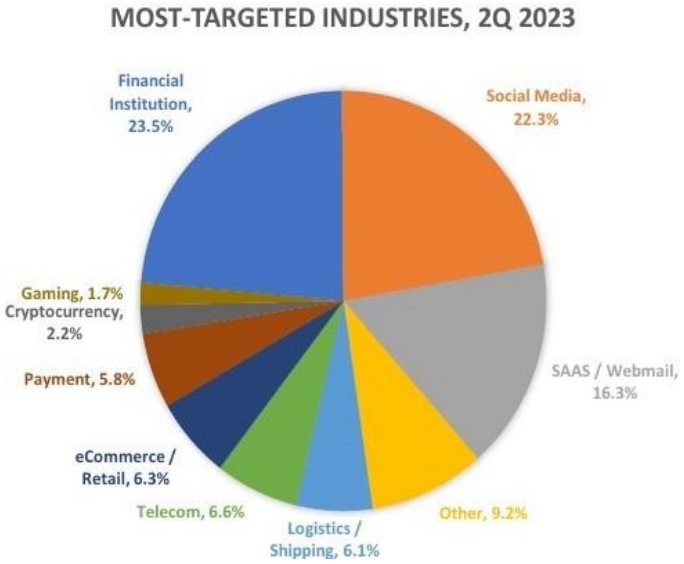

Fig. 1. Most-Targeted Industries, 2Q 2023. Source: https://apwg.org/trendsreports/ [13].

1. In Q2 2023, the APWG documented a total of 1,286,208 phishing attacks, marking the third-highest quarterly total in the organization's recorded history.
2. A noteworthy observation pertains to Business Email Compromise attacks, revealing a substantial increase in the average wire transfer amount requested. Specifically, the average amount in Q2 2023 surged to $293,359, showing a notable uptick of 57% compared to the previous quarter's average of $187,053.
3. The financial sector maintained its unfortunate status as the most-targeted industry, accounting for 23.5% of all phishing attacks. Additionally, attacks against online payment services constituted another 5.8% of the overall attack landscape.
4. A concerning trend highlighted in the report is the escalating volume of voice-mail phishing, commonly referred to as vishing. This method of phishing, utilizing voice messages, witnessed a continued rise, underscoring the adaptability and persistence of attackers in diversifying their tactics.

This paper offers valuable insights into improving the efficiency of classifiers for identifying phishing URLs. By conducting a thorough evaluation using different feature selection methods, including Gain Ratio, Information Gain, and Chi-square (χ^2), the paper highlights a significant enhancement in detection performance. The experimentation with reduced feature sets, specifically 17, 28, and 48 URL features, demonstrates notable improvements in accuracy, precision, recall, and F-measure across various machine learning classifiers. The following are the main findings of this paper,

1. The suggested method considerably increases the accuracy of phishing URL detection by accurately detecting and prioritizing crucial features, resulting in a more reliable and robust system for recognizing possible threats.
2. The Extra Trees classifier and the Chi-square ($\chi 2$) feature selection approach were used to detect phishing URLs optimally, using just 28 URL characteristics out of a total of 48.
3. Through the application of optimal feature selection, this work streamlines the detection process, reducing computational overhead and enhancing the efficiency of phishing URL detection systems.
4. By focusing on the most pertinent features, the approach contributes to the overall efficiency of phishing detection systems, ensuring a more rapid and resource-effective identification of phishing attempts.
5. Rigorous testing and systematic evaluation processes provide empirical evidence supporting the effectiveness of the proposed method. This validation enhances the credibility of the approach, ensuring its reliability across diverse phishing scenarios.

The rest of this paper is structured as follows: In Section 2, we have provided a concise overview of relevant prior research. Section 3 outlined the methodology involving feature selection techniques and supervised batch machine learning algorithms. The findings and performance comparisons with existing approaches from our experiments are detailed in Section 4. Section 5 gives Security Implications of the Proposed System. In Section 6, we have provided the limitations of our proposed approach. Our final conclusions are presented in Section 7.

2. Related Works

In the face of escalating cyber threats, the detection of phishing websites has become an area of intense research focus. Recent studies showcase a variety of innovative approaches that leverage advanced techniques, including feature selection and ensemble methods, to enhance the precision and adaptability of phishing detection systems.

Web phishing, a prevalent cyber threat, poses serious risks to the security of online users and organizations. As the landscape of phishing attacks continues to evolve, there is a pressing need for advanced and adaptive detection techniques.

M. Vijayalakshmi et al. have provided a comprehensive overview of existing research in web phishing detection, shedding light on the state-of-the-art methodologies, taxonomies, and future directions outlined by prominent scholars in the field [14].

Catal, Cagatay et al. have conducted a literature review with the aim of identifying, evaluating, and synthesizing findings on deep learning methodologies for detecting phishing, as documented in selected scientific publications. The analysis addressed nine particular research issues and provided a thorough review of the uses of deep learning algorithms in phishing detection, evaluating various aspects of their use. The review encompassed 43 journal articles sourced from electronic databases, offering insights into the defined research queries [15].

Basit, Abdul et al. have conducted an extensive literature review focusing on Artificial Intelligence (AI) techniques for the detection of phishing attacks. Machine Learning, Deep Learning, Hybrid Learning, and Scenario-based approaches are covered in the review. Additionally, they have offered a comparative analysis of various studies utilizing each AI technique, critically evaluating the strengths and weaknesses inherent in these methodologies [16].

Prabakaran, Manoj Kumar et al. have developed a unique technique that combines the capabilities of VAE with deep neural networks. In their suggested approach, the VAE model is used to automatically extract intrinsic information from raw URLs. This is accomplished by reconstructing the original input URL, which improves the identification of phishing URLs. The experimental phase includes crawling about 100,000 URLs from two openly accessible datasets: the ISCXURL2016 dataset and the Kaggle dataset. The results of the experiments confirmed the usefulness of the suggested paradigm. It outperformed the other models under evaluation, with a maximum accuracy of 97.45%. In addition, the model had a significantly quicker response time, clocking in at 1.9 seconds [17].

Tie Li et al. developed a unique method for detecting fraudulent URLs that incorporates linear and non-linear space transformation approaches. The authors developed a two-phase distance measure learning technique for linear transformation. Singular value decomposition was used first to create an orthogonal space, and then linear programming was used to find the best distance metric. The Nystro ¨m approach for kernel approximation was developed for non-linear transformation, and the updated distance metric was added using the radial basis function. This method attempted to capitalize on the advantages of both linear and non-linear transformations. A dataset of 33,1622 URLs, each with 62 characteristics, was obtained to validate the suggested feature engineering approaches. The experiments indicated a considerable improvement in the effectiveness and efficacy of particular classifiers such as KNN, SVM, and neural networks. The recognition rate of dangerous URLs using KNN increased from 68% to 86%, the linear SVM grew from 58% to 81%, and the Multi-Layer Perceptron improved from 63% to 82%. These results demonstrate the efficacy of the suggested mix of linear and non-linear space transformation approaches in enhancing the accuracy and efficiency of harmful URL recognition across several classifiers [18].

Gunikhan Sonowal et al. developed PhiDMA, a multiple-layer model for phishing detection. This novel PhiDMA model is composed of five layers, each of which contributes to its strong phishing detection capabilities. The layers are as follows: Automatic Upgrade Whitelisting, URL Features, Lexical Signature, String Matching, and Accessibility Score Comparison. An early implementation of the PhiDMA model has been created, complete with an accessible interface to provide accessibility for those with visual impairments. The study's findings demonstrate the PhiDMA model's usefulness, exhibiting its capacity to identify phishing websites with a high accuracy rate of 92.72%. This demonstrates the multilayer approach's promising potential for improving the accuracy and accessibility of phishing detection systems [2].
Abdul A. Orunsolu et al. developed an innovative machine learning-based prediction model to improve the efficacy of phishing prevention measures. The predictive model includes a Feature Selection Module, which is necessary for creating a powerful vector of features. Using an incremental component-based methodology, these attributes are methodically obtained from the URL, webpage properties, and webpage behavior. The predictive model is then supplied with the resulting feature vector. SVM and NB are used in this suggested system and trained on a 15-dimensional set of attributes. The tests were carried out on datasets including 2541 cases of phishing and 2500 instances of benign activities. The experimental results demonstrate exceptional performance metrics, with a minimal 0.04% False Positive rate and an amazing 99.96% accuracy for both SVM and NB predictive models when using a ten-fold cross-validation approach.

These findings highlight the suggested machine learning-based prediction model's noteworthy efficacy in developing powerful anti-phishing capabilities [4].

Weiping Wang et al. developed PDRCNN, a fast method for identifying phishing websites based merely on the website's URL. PDRCNN differs from prior approaches in that it does not need for the extraction of the target website's content and does not rely on third-party services. Encoding URL information into a two-dimensional tensor, which is then fed into an innovatively built deep learning neural network for categorization of the original URL, is the approach used. A bidirectional LSTM network is used to extract global features from the built tensor and assign string information to each

letter in the URL. Following that, a CNN is used to detect phishing by autonomously identifying crucial letters in the URL and compressing the retrieved features into a fixed-length vector space. PDRCNN outperforms utilizing each network individually because of the synergy of these two networks. The authors created a database with over 500,000 URLs from Alexa and PhishTank. The experimental results show that PDRCNN has a remarkable detection accuracy of 97% and an AUC value of 99%, outperforming the most advanced methods. Furthermore, with the trained PDRCNN model, the identification process is extremely fast, with an average detection time of only 0.4 ms per URL. This demonstrates the efficiency and usefulness of PDRCNN in detecting phishing websites [7].

JianTing Yuan et al. developed a novel collaborative neural network algorithm model that combines the attention procedure, a bidirectional independent recurrent neural network (Bi-IndRNN), and a capsule network (CapsNet). To train character- and word-level Uniform Resource Locator (URL) static embedding vector features, the model employs the word vector tool word2vec. At the same time, the programme extracts texture fingerprint features capable of detecting content differences across multiple harmful website URL binary files. These collected characteristics are then merged and sent into the joint neural network algorithm model. There are several steps to the procedure. The multihead attention process begins by adjusting weights and employing Bi-IndRNN to retrieve contextual semantic data. CapsNet is then used with dynamic routing to extract deep semantic information. Finally, for classification, a sigmoid classifier is used. The research uses a variety of ways to extract more extensive information from multiple viewpoints. When compared to other current methodologies, experimental findings show that the suggested strategy improves the categorization accuracy of harmful web page identification. This demonstrates the combined neural network algorithm's usefulness in enhancing the precision of recognizing fraudulent web sites [19].

Yazan Ahmad Alsariera et al. have introduced four meta-learner models, namely AdaBoost-Extra Tree (ABET), Bagging-Extra Tree (BET), Rotation Forest-Extra Tree (RoFBET), and LogitBoost-Extra Tree (LBET), all constructed using the extra-tree base classifier. These AI-based meta-learners were specifically developed and applied to datasets containing phishing websites, and their performances underwent thorough evaluation. The results of the studies showed that the suggested meta-learner models had a detection efficiency that regularly above 97%, with an extremely low false-positive rate of less than 0.028. Furthermore, the performance of these models outperformed that of existing machine learning-based phishing attack detection models. As a result, given their greater accuracy and efficacy in minimizing false positives, the study proposes using meta-learners as a preferable technique in the development of models for detecting phishing attempts [20].

Ayman El Aassal et al. developed a new feature classification that emphasises the interpretation and function of each characteristic. Following that, they provide a revolutionary benchmarking framework called 'PhishBench.' This framework is a complete tool for carefully and comprehensively examining existing phishing detection capabilities. The capacity to execute assessments under same experimental settings, including uniform system specifications, datasets, classifiers, and evaluation criteria, distinguishes PhishBench. It is a pioneering effort in phishing benchmarking research, incorporating a thorough and methodical approach to feature comparison and assessment. The authors use PhishBench to evaluate phishing strategies described in the literature, using fresh and different datasets to determine the resilience and scalability of these tactics. The study investigates the effect of various dataset features on classification ability, such as shifting legitimate-to-phishing ratios and the extension of unbalanced datasets. The findings show that the uneven nature of phishing attempts has a substantial impact on the efficacy of detection systems, requiring researchers to consider this feature when proposing new approaches. Furthermore, the study emphasizes that retraining alone is insufficient to prevent new assaults; instead, the creation of unique features and methodologies is required to thwart attackers aiming to fool detection systems [21].

Shouq Alnemari et al. developed and tested four models to investigate the effectiveness of using ML techniques in the detection of phishing domains. In addition, the study does a comparison analysis, comparing the most accurate model among the four against current solutions in the literature. These models are based on approaches such as ANNs, support SVMs, DTs and RF. Furthermore, the URL's UCI phishing domains dataset is used as a baseline for a thorough evaluation of these models. The study's findings show that the model based on the random forest approach outperforms the other three strategies in terms of accuracy. It also outperforms previous solutions reported in the literature. This highlights the random forest approach's usefulness in improving the precision of phishing domain identification, as proven by its surpass in contrast to alternative machine learning approaches and established methodologies [22].

Yu-Hung Chen et al. developed an innovative machine learning architecture and a variety of learning algorithms to combine anti-phishing services to tackle cyber-phishing attacks. With the fast growth of information technology, the growing threat of cyber-phishing attacks, in which hackers attempt to steal vital personal information, has become a major problem in the field of information security. While user training, public awareness, and addressing fraudulent phishing all contribute to the prevention of phishing websites, recent research has primarily focused on preventing fraudulent phishing, frequently relying on manual identification methods that prove inefficient for real-time detection systems. This study utilizes statistical approaches such as ANOVA, X2, and information gain to evaluate traits and exclude those that are irrelevant. For training and assessing standard machine learning methods such as SVM with linear or the rbf kernels, LR, DT, and KNN, the top 28 most relevant features are selected. Furthermore, the study evaluates these algorithms by merging various classifiers such as the AdaBoost, bagging, and voting utilizing the ensemble learning idea. The results of this research show that the XGBoost model outperforms the other techniques investigated in the study, with a performance of 99.2%. This demonstrates the efficacy of the suggested machine learning system in providing powerful anti-phishing services [23].

Ali Aljofey et al. have developed a quick deep learning solution model that detects phishing based on website URLs using a character-level CNN. This paradigm, in particular, eliminates the requirement to retrieve material from the in- tended website or rely on third-party services. It adeptly gathers information and sequential patterns inside URL strings without the need for prior phishing expertise. Following that, it employs these sequential pattern characteristics to quickly categories the real URL. The study compares multiple classical machine learning models with deep learning models in the assessment phase, utilizing diverse feature sets such as hand-crafted features, character embedding, character-level TF-IDF, and character-level count vectors features. The experimental findings demonstrate the efficacy of the proposed model, with an accuracy of 95.02% on the study dataset and outstanding accuracy rates of 98.58%, 95.46%, and 95.22% on benchmark datasets. These findings suggest that the proposed model outperforms previous methods developed for phishing URL detection [24].

Zainab Alshingiti et al. demonstrated three distinct deep learning algorithms for detecting phishing URLs, comparing LSTM and CNN, and eventually an LSTM-CNN hybrid strategy. The experimental findings show that these suggested approaches are accurate, with CNN obtaining 99.2%, LSTM-CNN achieving 97.6%, and LSTM achieving 96.8%. In this study, the CNN-based system emerges as the optimum solution for phishing detection [25].

Peng Yang et al. have developed a multidimensional approach to phishing detection that leverages a fast deep learning- based algorithm. Initially, character sequence characteristics from the supplied URL are retrieved, allowing for quick categorization using deep learning. This phase acts alone, without the intervention of another party or any prior knowledge of phishing. In the following stage, multidimensional features are formed by combining URL statistics information, webpage code features, webpage text features, and the rapid classification outcome from deep learning. This method dramatically decreases the time necessary to identify and establish a threshold. Using a dataset of millions of phishing and valid URLs, the accuracy reached an astounding 98.99%, with a 0.59% false positive rate. Furthermore, the experimental findings show that by carefully modifying the threshold, detection efficiency may be enhanced even further. This approach delivers a complete and effective phishing detection mechanism, displaying great accuracy and flexibility in real-world circumstance [26].

Maria Carla Calzarossa et al. have made a substantial addition to the field of phishing website identification by developing an interpretable machine learning model. This model not only provides accurate forecasts of phishing instances, but it also provides explanations by identifying attributes most commonly connected with phishing websites. The authors describe a unique feature selection methodology based on Lorenz Zonoids, which reflect the multidimensional extension of the Gini coefficient. To demonstrate the efficacy of their solution, they apply it to an actual dataset comprising characteristics from both phishing and authentic websites [27].

Using ML approaches, Jian Mao et al. attempted to allow automatic phishing detection based on page layout. Their concept includes a learning-based aggregate analysis technique for determining page structure similarity, which is critical in identifying phishing pages. The authors then develop their method and evaluate four widely used machine learning classifiers. The evaluation focuses on accuracy and investigates the many aspects that influence the outputs of these classifiers [28].

Rundong Yang et al. developed an integrated strategy for detecting phishing websites that makes use of CNN and RF. This approach accurately validates the validity of URLs without the need for access to online content or the use of third-party services. To transform URLs into fixed-size matrices, the suggested approach uses character embedding. Following that, CNN models are used to extract features at various levels, which are subsequently classified using several RF classifiers. The final prediction results are determined using a winner-take-all strategy. The suggested model achieved a 99.35% accuracy rate on their dataset. Furthermore, using benchmark data, a detection rate of 99.26% was achieved, outperforming the performance of previous extreme models. This underscores the efficacy of the introduced approach in achieving robust and accurate phishing website detection [29].

Diana T. Mosa et al. did a thorough assessment focused on several modern machine-learning algorithms dedicated to tackling phishing challenges and obtaining accurate detection of various phishing assaults. The study makes use of a dataset including over 11,000 webpages from the Kaggle dataset. The influence of Thirty website elements was carefully investigated, as well as the resultant class labels indicating whether a website is a phishing one (1) or not (-1). Furthermore, the study examines confusion matrices for models used in machine learning such as NN, NB, and AdaBoost. These assessments yielded remarkable results: 90.23% for NN, 92.97% for Nave Bayes, and 95.43% for AdaBoost. These results highlight the effectiveness of the various machine-learning models in achieving high-quality accuracy in detecting and addressing different phishing attacks [30].

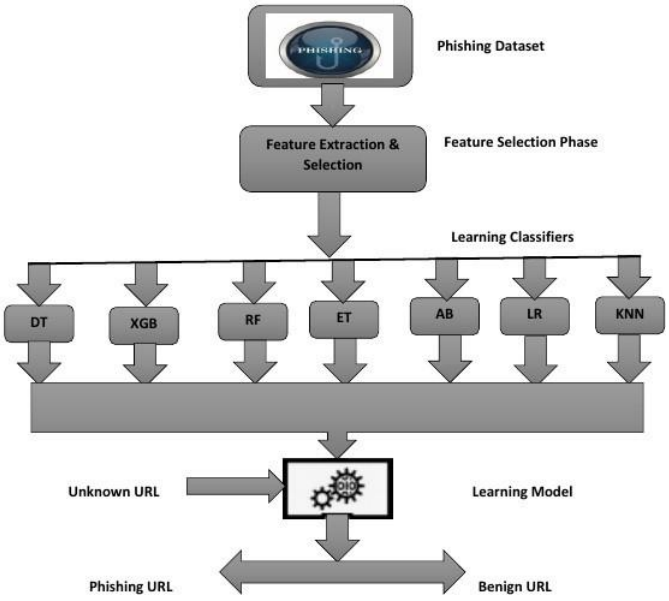

Fig. 2. Proposed Phishing URLs Detection Framework.

3. Methodology

Fig. 2 shows the proposed framework of the phishing URL detection system using feature selection techniques. We have used the Universiti Malaysia Sarawak, Malaysia Phishing and Legitimate Dataset [31]. This dataset consists of a set of 48 features has been extracted from a dataset comprising 5000 phishing web pages and an equal number of legitimate web pages. The dataset spans downloads conducted between January to May 2015 and from May to June 2017. The framework consists of feature extraction and selection phase, learning phase and detection phase.

The newness of our approach lies in the integration of optimal feature selection techniques tailored specifically to enhance the accuracy and efficiency of phishing URL detection. While standard machine learning algorithms and feature selection methods are indeed widely used, our method uniquely combines multiple feature selection techniques—such as information gain, information gain ratio, and chi-square (χ^2)—to identify a highly relevant subset of URL features. By systematically evaluating and fine-tuning this feature subset across a diverse set of classifiers, we achieve notable performance gains that go beyond what is typically seen with conventional methods. Specifically, our approach achieves high detection accuracy while simultaneously reducing computational overhead, offering a more streamlined and effective solution compared to existing methodologies in phishing detection.

3.1 Feature Selection Techniques used for Phishing URLs Detection

We have used three feature selection techniques, Information Gain feature selection, Information Gain Ratio feature selection and Chi-square feature selection. We utilized the Weka implementation of the aforementioned feature selection techniques [32].

A. Information Gain Feature Selection Technique

The Information Gain Feature Selection approach is used in machine learning and data analysis to find and pick the most informative features in a dataset. It assesses a feature's usefulness by assessing its capacity to minimize uncertainty in predicting the target variable. The Information Gain is estimated based on the entropy of the dataset before and after considering a certain attribute [33].

The Information Gain (IG) for a particular feature X with respect to a target variable Y is typically calculated using the formula:

$$IG(X, Y) = H(Y) - H(Y \mid X) \tag{1}$$

Where:

- $H(Y)$ is the entropy of the target variable Y without considering any specific feature.
- $H(Y \mid X)$ is the conditional entropy of Y given the feature X.

Table 1. Feature selection using Information Gain on Phishing Dataset

Feature No.	Information Gain	Feature Name
F1	0.725991576	PctExtHyperlinks
F2	0.467560517	PctExtResourceUrls
F3	0.376117156	PctNullSelfRedirectHyperlinks
F4	0.306406845	PctExtNullSelfRedirectHyperlinksRT
F5	0.191403941	NumNumericChars
F6	0.181044924	FrequentDomainNameMismatch
F7	0.16957923	ExtMetaScriptLinkRT
F8	0.164486903	NumDash
F9	0.112567967	SubmitInfoToEmail
F10	0.094549902	NumDots
F11	0.092523917	PathLength
F12	0.083944032	QueryLength
F13	0.079981378	PathLevel
F14	0.079465026	InsecureForms
F15	0.075941377	UrlLength
F16	0.060312182	NumSensitiveWords
F17	0.048500065	NumQueryComponents
F18	0.040843554	PctExtResourceUrlsRT
F19	0.040474756	IframeOrFrame
F20	0.036994287	HostnameLength
F21	0.028318577	NumAmpersand
F22	0.026208142	AbnormalExtFormActionR
F23	0.021840282	UrlLengthRT
F24	0.021682746	NumDashInHostname
F25	0.017417149	IpAddress
F26	0.016142806	AbnormalFormAction
F27	0.015382134	EmbeddedBrandName
F28	0.014966024	NumUnderscore

The entropy $H(Y)\,Y$) is calculated as:
Where:

$$H(Y) = -\sum_{i} P(y_i) \cdot \log_2(P(y_i)) \tag{2}$$

- P (yi) is the probability of class yi in the target variable Y. The conditional entropy H (Y |X) is calculated as:

$$H(Y \mid X) = \sum_{j} P(x_j) \cdot H(Y \mid X = x_j) \tag{3}$$

Where:

- $P(x_j)$ is the probability of the occurrence of feature X having value x_j.
- $H(Y|X = x_j)$ is the entropy of the target variable Y given that feature X has the value x_j.

In practical terms, Information Gain is used to evaluate how well a particular feature separates or distinguishes different classes in the target variable, with higher values indicating more significant discriminatory power. The feature with the highest Information Gain is considered the most informative and is often selected for further analysis or model training. Table 1 shows the most relevant features selection using information gain. A threshold of 0.01 has been designated for the information gain feature selection approach. From the phishing dataset presented in Table 1, we have identified and chosen the 28 most pertinent features out of the total 48.

B. Information Gain Ratio Feature Selection Technique

The Information Gain Ratio (IGR) is an extension of Information Gain that takes into account the intrinsic information associated with a feature. It is calculated using the following formula [34]:

Table 2. Feature selection using Information Gain Ratio on Phishing Dataset

Feature No.	Information Gain Ratio	Feature Name
F1	0.24091	FrequentDomainNameMismatch
F2	0.2397	PctExtNullSelfRedirectHyperlinksRT
F3	0.2059	PctExtHyperlinks
F4	0.20314	SubmitInfoToEmail
F5	0.16735	PctNullSelfRedirectHyperlinks
F6	0.13887	IpAddress
F7	0.12721	InsecureForms
F8	0.11969	NumSensitiveWords
F9	0.11049	ExtMetaScriptLinkRT
F10	0.09815	NumHash
F11	0.09433	PopUpWindow
F12	0.08796	PctExtResourceUrls
F13	0.0832	NumDash
F14	0.07915	SubdomainLevelRT
F15	0.07683	TildeSymbol
F16	0.07609	AtSymbol
F17	0.07584	NumNumericChars

$$IGR(X,Y) = \frac{IG(X,Y)}{H(X)} \tag{4}$$

Where:
- $IG(X, Y)$ is the Information Gain between feature X and target variable Y.
- $H(X)$ is the entropy of feature X.

The entropy H(X) for a discrete feature X is calculated as:

$$H(X) = -\sum_{i} P(x_i) \cdot \log_2(P(x_i)) \tag{5}$$

Where:
- P (xi) is the probability of the occurrence of feature X having value xi.

In the context of Information Gain Ratio, the feature's intrinsic information is factored in to provide a normalized measure, making it useful when dealing with features with different scales or numbers of possible values. The Information Gain Ratio helps address the bias towards features with many distinct values that might have high Information Gain by virtue of their variability. Table 2 shows the most relevant features selection using information gain ratio. The information gain ratio feature selection method employs a threshold set at 0.07. From the phishing dataset illustrated in Table 2, 17 of the most significant features have been carefully chosen out of the total 48.

C. Chi-square (χ2) Feature Selection Technique

Chi-square (χ^2) Feature Selection is commonly used for assessing the independence between a categorical feature and a categorical target variable. The chi-square statistic is calculated using the following formula [35]:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{6}$$

Where:
- χ^2 is the chi-square statistic.
- O_i is the observed frequency for each category.
- E_i is the expected frequency for each category.

Table 3. Feature selection using Chi-square (χ^2) on Phishing Dataset

Feature No.	Chi-square (χ^2) Value	Feature Name
F1	7736.05208	PctExtHyperlinks
F2	5189.800092	PctExtResourceUrls
F3	3856.690874	PctNullSelfRedirectHyperlinks
F4	3602.756211	PctExtNullSelfRedirectHyperlinksRT
F5	2379.805674	NumNumericChars
F6	2238.732528	ExtMetaScriptLinkRT
F7	2152.552258	FrequentDomainNameMismatch
F8	1926.347548	NumDash
F9	1279.027571	SubmitInfoToEmail
F10	1212.619216	NumDots
F11	1144.429637	PathLength
F12	1006.970462	PathLevel
F13	1000.960019	InsecureForms
F14	984.913091	QueryLength
F15	969.381602	UrlLength
F16	713.119977	NumSensitiveWords
F17	633.160057	NumQueryComponents
F18	553.344548	IframeOrFrame
F19	511.19779	PctExtResourceUrlsRT
F20	437.491581	HostnameLength
F21	364.105132	NumAmpersand
F22	345.637576	AbnormalExtFormActionR
F23	300.427195	UrlLengthRT
F24	256.831745	NumDashInHostname
F25	210.462943	AbnormalFormAction
F26	201.043507	EmbeddedBrandName
F27	191.067488	NumUnderscore
F28	175.010175	IpAddress

The expected frequency E_i is calculated as:

$$E_i = \frac{(\text{row total} \times \text{column total})}{\text{grand total}} \tag{7}$$

After computing the chi-square statistic, it can be used to evaluate the independence between the feature and the target variable. Higher values of chi-square indicate a higher association between the feature and the target, and vice versa. In feature selection, features with higher chi-square values are considered more informative for predicting the target variable.

Table 3 shows the most relevant features selection using Chi-square (χ^2) Feature Selection. A threshold of 175 has been established for the Chi-square Feature Selection method. From Table 3, we have carefully chosen the 28 most pertinent features out of a total of 48 in the phishing dataset.

3.2 Machine Learning Classifiers used for Phishing URLs Detection

Several machine learning classifiers have been employed for phishing URLs detection, leveraging their ability to analyze patterns and features to distinguish between legitimate and phishing URLs. We have used 7 machine learning classifiers for phishing URLs detection like, Decision Tree, Random forest, Extra Trees, XGBoost, AdaBoost, Logistic Regression and K-Nearest Neighbors. We employed the implementations of the mentioned machine learning algorithms available in the scikit-learn machine learning library [36].

A. Decision Tree (DT)

The Decision Tree algorithm is a prominent machine learning technique widely employed for both classification and regression tasks. It operates by recursively partitioning the dataset into subsets based on the values of input features, creating a tree-like structure that facilitates decision-making. Decision trees have gained popularity due to their interpretability, simplicity, and applicability to diverse data types, making them a fundamental tool in the machine learning landscape. The initial node at the top of the tree, representing the primary decision or test based on a chosen feature. Subsequent nodes that follow the root, each corresponding to a decision or test on a specific feature. Connect the nodes and represent the possible outcomes of the decisions or tests. Terminal nodes at the end of the branches, containing the final predictions or classifications [37]. The algorithm selects the most informative feature to split the

dataset, aiming to create homogeneous subsets. Recursive partitioning process continues iteratively, creating a branching structure until a stopping criterion is met, such as a specified tree depth or minimum samples in a node. The final predictions or classifications are made at the leaf nodes, ensuring a clear and interpretable decision-making process.

B. Random Forest (RF)

The Random Forest algorithm is a powerful and versatile ensemble learning method that belongs to the family of tree-based algorithms. Known for its high predictive accuracy and robustness, Random Forest combines the strengths of multiple decision trees to make reliable predictions in both classification and regression tasks. Developed as an improvement over individual decision trees, this algorithm has become a popular choice across various domains in machine learning. Random Forest comprises a collection of decision trees, each trained on a different subset of the dataset. During the training process, a random subset of the features and data points is chosen for each decision tree, introducing diversity in the ensemble. For classification tasks, the algorithm aggregates the predictions of individual trees through voting, while for regression tasks, it averages the predictions [38]. Random Forest randomly selects subsets of the dataset with replacement, creating diverse training sets for each decision tree. At each node of a decision tree, a random subset of features is considered for splitting, reducing the correlation between individual trees. The final prediction is made by aggregating the outputs of all decision trees, either through majority voting or averaging.

C. Extra Tree (ET)

The Extra Trees algorithm, short for Extremely Randomized Trees, is a powerful ensemble learning technique that belongs to the family of tree-based algorithms. Similar to Random Forest, Extra Trees builds a collection of decision trees to enhance predictive performance in both classification and regression tasks. However, what sets Extra Trees apart is its additional layer of randomness during the tree-building process, making it a robust and efficient tool in machine learning. Like Random Forest, Extra Trees utilizes multiple decision trees in its ensemble. Extra Trees introduces extra randomness by selecting random features and thresholds for each split in the decision trees. The final prediction is made through the aggregation of the predictions from individual trees, often using majority voting for classification tasks or averaging for regression tasks [39]. Similar to Random Forest, Extra Trees creates multiple subsets of the dataset through bootstrapping, ensuring diversity in the training data for each tree. Extra Trees goes a step further by randomly selecting features and thresholds at each node during the decision tree construction, reducing the correlation between trees. The final prediction is obtained by combining the outputs of all decision trees in the ensemble.

D. XGBoost

XGBoost, short for eXtreme Gradient Boosting, is a powerful and efficient machine learning algorithm that belongs to the family of gradient boosting methods. Developed to improve upon traditional gradient boosting techniques, XG- Boost has gained widespread popularity and is recognized as a state-of-the-art algorithm in various machine learning competitions. It excels in both classification and regression tasks, providing high predictive accuracy and robustness. XGBoost is designed for efficiency, supporting parallel and distributed computing to handle large datasets and accelerate training. The algorithm employs pruning techniques to remove unnecessary branches from trees, further enhancing model efficiency and preventing overfitting. XGBoost includes built-in cross-validation capabilities, allowing for the assessment of model performance and hyperparameter tuning [40]. The algorithm starts with a single decision tree, predicting the target variable. The residuals (differences between predicted and actual values) are calculated for each data point. Additional decision trees are sequentially added to the ensemble, each focusing on reducing the remaining residuals. XGBoost optimizes the model parameters using gradient descent, minimizing a specific loss function.

E. AdaBoost

AdaBoost, short for Adaptive Boosting, is a powerful ensemble learning algorithm designed to improve the performance of weak learners by combining them into a robust and accurate model. Developed by Yoav Freund and Robert Schapire, AdaBoost belongs to the family of boosting algorithms and is widely used in classification tasks. Its adaptive nature focuses on assigning higher weights to misclassified instances, allowing subsequent weak learners to prioritize these instances during training. AdaBoost starts with a weak learner, often a simple model like a decision stump (a one-level decision tree), that performs slightly better than random chance. Instances in the dataset are assigned weights, with higher weights given to misclassified instances from previous iterations. Weak learners are trained sequentially, with each subsequent learner emphasizing the misclassified instances from the previous iterations. The final prediction is made through a weighted sum or voting mechanism, where the contribution of each weak learner is determined by its performance during training [41]. Each instance in the dataset is initially assigned equal weight. A weak learner is trained on the dataset, and its performance is assessed. Instances that are misclassified are assigned higher weights, emphasizing their importance in subsequent iterations. Sequential iterations are repeated for a predefined number of iterations or until a specified performance threshold is reached. The weak learners are combined into a strong model, and their individual contributions are weighted based on their performance.

F. Logistic Regression (LR)

Logistic Regression is a widely used machine learning algorithm employed for binary and multiclass classification tasks. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm. Developed for binary classification, it has been extended to handle multiple classes through techniques like one-vs-all or one-vs-one. Logistic Regression models the relationship between the independent variables and the probability of a particular outcome using the logistic function. The logistic function, represented by the sigmoid curve, is a crucial component of logistic regression. It transforms input values into a range between 0 and 1, representing the probability of the positive class. Logistic regression involves finding optimal weights and a bias term that define the decision boundary. The algorithm learns these parameters during training. The cost function, often the cross-entropy loss, measures the difference between the predicted probabilities and the actual class labels. The objective is to minimize this cost during training [42]. It initialize the weights and bias randomly or with predefined values. Compute the linear combination of input features, weights, and bias. Apply the sigmoid function to the linear combination to obtain probabilities between 0 and 1. Evaluate the difference between predicted probabilities and actual labels using the cost function. Adjust weights and bias to minimize the cost function using gradient descent or other optimization techniques. Iterative training: is performed until convergence, updating parameters to improve the model's predictive performance.

G. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a versatile and intuitive machine learning algorithm used for both classification and regression tasks. It is categorized as a lazy or instance-based learning algorithm, as it does not explicitly learn a model during training. Instead, KNN makes predictions based on the majority class or average value of the k-nearest data points in the feature space. KNN's simplicity and effectiveness make it a popular choice, particularly in scenarios where interpretability and ease of implementation are crucial. KNN relies on a distance metric, commonly Euclidean distance, to measure the similarity between data points in the feature space. The algorithm considers a specified number, k, of nearest neighbors to make predictions. These neighbors are determined based on their proximity to the target data point in the feature space. For classification tasks, the majority class among the k-nearest neighbors is assigned to the target data point. For regression tasks, the average value of the target variable among the k-nearest neighbors is used [43]. It stores the training dataset in memory. For a given test data point, calculate the distance to all data points in the training set using a specified distance metric. Identify the k-nearest neighbors based on the calculated distances. For classification, predict the majority class among the k-nearest neighbors. For regression, predict the average value of the target variable.

4. Experimental Results

This section shows the results of our research on the identification of phishing URLs and provides a thorough discussion of the findings. Our study was meant to evaluate the usefulness of various features and machine learning models in recognizing phishing URLs.

4.1 Dataset

We have used the Universiti Malaysia Sarawak, Malaysia Phishing and Legitimate URLs Dataset. The Phishing URLs Dataset comprises a total of 10,000 URL samples, specifically, 5,000 legitimate samples and 5,000 phishing samples. These samples within the dataset encompass all the essential elements utilized in various research studies documented in the literature. The dataset includes the necessary components employed in diverse research endeavors as outlined in existing literature [31].

The Universiti Malaysia Sarawak, Malaysia Phishing and Legitimate Dataset was chosen for its comprehensive collection of phishing and legitimate URLs, offering a structured base for developing and testing our phishing detection model. However, we acknowledge that the dataset has certain limitations. First, it may not capture the full diversity and complexity of real-world phishing attacks, which are constantly evolving with new tactics and variations. Additionally, potential biases in the dataset could arise if the URLs do not reflect the latest phishing trends or geographic-specific threats, as phishing tactics can vary by region and target audience. To address these limitations, future work will involve validating the model on additional, more diverse datasets, ideally updated regularly to reflect emerging phishing strategies. This approach would help to ensure that the model remains robust and effective in real-world applications, while minimizing any biases that could impact its generalizability.

4.2 Measures Used for Performance Evaluation of Learning Classifiers on Phishing URLs Dataset

We explored the performance of several machine learning models, including Decision Trees, XGBoost, Random Forest, Extra Trees, Logistic Regression, AdaBoost and K-Nearest Neighbors for the task of phishing URLs detection. We have used following measures to evaluate our proposed approach for phishing URLs detection [44].

- **Accuracy:** Accuracy evaluates the overall efficiency of a categorization model. It is the proportion of accurately predicted occurrences to the total number of examples in the dataset.

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

- **Precision:** Precision assesses the accuracy of the positive predictions made by the model. It is the ratio of true positive predictions to all positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

- **Recall:** The model's ability to properly identify positive cases is measured by recall. It is the proportion of genuine positive predictions to all occurrences of true positive prediction.

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

- **F1-Measure (F1-Score):** The F1-measure is a fair assessment of the model's performance since it is an average of accuracy and recall.

$$F1 - Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{11}$$

- **False Positive Rate:** The false positive rate (FPR) is a measure used in statistics and machine learning to evaluate the performance of a binary classification model. It quantifies how often the model incorrectly predicts a positive outcome when the true outcome is negative.

$$\text{FPR} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}} \tag{12}$$

- **False Negative Rate:** The false negative rate (FNR) is a metric used to assess the performance of a binary classification model. It quantifies how often the model incorrectly predicts a negative outcome when the true outcome is actually positive.

$$\text{FNR} = \frac{\text{False Negatives (FN)}}{\text{False Negatives (FN)} + \text{True Positives (TP)}} \tag{13}$$

The confusion matrix yielded the values TP, TN, FP, and FN. These performance indicators, which offer a quantifiable assessment of accuracy, precision, recall, and overall performance, are frequently used to assess the success of classification algorithms.

4.3 Performance Evaluation of Machine Learning Classifiers on Phishing URLs Dataset using Full Feature Set

Table 4 presents the performance assessment of each machine learning classifier on the phishing URL dataset by employing the complete feature set consisting of 48 URLs features. Extra Trees (ET) achieved the highest accuracy (98.03%) and F1-measure (98.03%), with relatively low false positive (0.0132) and false negative rates (0.0234), making it the best performer overall. Random Forest (RF) closely followed, with an accuracy of 97.97% and a similarly low false positive rate (0.0172), but with a slightly longer test time. Decision Tree (DT) and XGBoost also performed well, with accuracies of 96.5% and 97.23%, respectively, balancing moderate training and testing times. K-Nearest Neighbors (KNN) had the lowest accuracy (85.63%) and higher false positive and false negative rates, indicating less reliability for this task. Additionally, Fig. 3 illustrates the ROC curve for the classifiers.

Table 4. Performance Evaluation of Machine Learning Classifiers on Phishing URLs Dataset using Full Feature Set (48 Features)

Classifier	Accuracy	Precision	Recall	F1-Measure	FPR	FNR	Train Time (sec.)	Test Time (sec.)
DT	96.5	96.51	96.5	96.5	0.035	0.043	0.0750	0.0028
RF	97.97	97.97	97.97	97.97	0.0172	0.0181	0.8843	0.049
ET	98.03	98.05	98.03	98.03	0.0132	0.0234	0.7933	0.0605
XGBoost	97.23	97.23	97.23	97.23	0.0238	0.0355	0.1160	0.0054
AdaBoost	97.07	97.07	97.07	97.07	0.0331	0.0301	0.5542	0.0345
LR	93.13	93.14	93.13	93.13	0.0702	0.066	0.1317	0.0037
KNN	85.63	85.75	85.63	85.62	0.167	0.098	0.0069	0.2201

ROC curves would typically illustrate the trade-off between the true positive rate and false positive rate for each classifier. Classifiers with ROC curves closer to the top-left corner of the plot demonstrate a higher capacity for distinguishing between phishing and legitimate URLs. Based on the table, Extra Trees and Random Forest would likely have ROC curves closest to the top-left, reflecting their superior performance in accuracy and low false positive rates. This analysis underscores how model selection impacts both accuracy and computational cost, with Extra Trees standing out as the most effective model when using the full feature set.

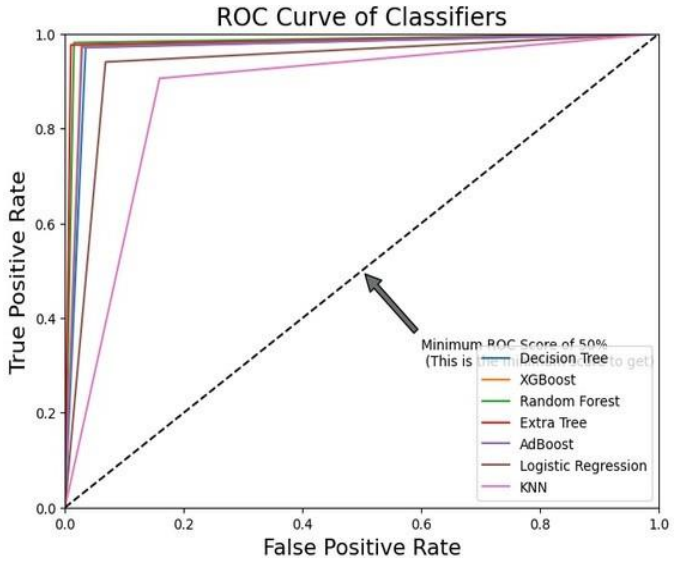

Fig. 3. ROC Curve of Classifiers using Full Feature Set (48 Features).

4.4 Performance Evaluation of Machine Learning Classifiers on Phishing URLs Dataset using Gain Ratio Feature Selection Method

In Table 5, the performance evaluation of individual machine learning classifiers on the phishing URL dataset is depicted, utilizing 17 URL features and the Gain Ratio Feature selection method. Decision Tree (DT)shows an improvement in accuracy, precision, recall, and F1-measure, achieving 96.77% across these metrics. Its false positive rate (0.0406) and false negative rate (0.0242) are moderate, with notably short training and testing times, making it an efficient choice. Random Forest (RF) and Extra Trees (ET) experienced slight decreases in accuracy (97.3% for RF and 97.4% for ET), but both maintained strong performance with low FPR and FNR values. Both classifiers also have moderate training and testing times, indicating they remain highly effective despite the feature reduction. XGBoost and AdaBoost Both saw slight decreases in performance metrics, with accuracies of 96.43% and 96.17%, respectively. However, XGBoost maintained a relatively low FPR (0.0318) and a fast testing time, making it efficient for real-time applications. Logistic Regression (LR) experienced a notable decrease in accuracy to 89.7%, with higher FPR and FNR values, indicating reduced effectiveness with the reduced feature set. K-Nearest Neighbors (KNN)improved in accuracy to 93%, showing an increase in performance from the full feature set. However, it has a higher FPR and FNR compared to most other models, with a significantly higher test time (0.2055 sec), indicating higher computational demand. Fig. 4 provides the ROC curve for the classifiers.

Table 5. Performance Evaluation of Machine Learning Classifiers on Phishing URLs Dataset using Gain Ratio Feature Selection Method

Classifier	Accuracy	Precision	Recall	F1-Measure	FPR	FNR	Train Time (sec.)	Test Time (sec.)
DT	96.77	96.77	96.77	96.77	0.0406	0.0242	0.02814	0.00230
RF	97.3	97.3	97.3	97.3	0.0223	0.0216	0.7139	0.0445
ET	97.4	97.4	97.4	97.4	0.0216	0.0216	0.6092	0.05450
XGBoost	96.43	96.43	96.43	96.43	0.0318	0.0302	0.0630	0.00325
AdaBoost	96.17	96.17	96.17	96.17	0.0365	0.0413	0.32721	0.02860
LR	89.7	89.76	89.7	89.7	0.1164	0.0728	0.10304	0.0030
KNN	93	93	93	93	0.0541	0.0735	0.0060	0.2055

With the Gain Ratio feature selection, ROC curves would illustrate the trade-off between true positive and false positive rates across classifiers, capturing their ability to distinguish between phishing and legitimate URLs. Classifiers like Extra Trees and Random Forest would likely have ROC curves closer to the top-left corner, indicating strong discriminatory ability with high true positive rates and low false positives even after feature reduction. Decision Tree would also show favorable ROC performance due to its improvements across all key metrics. Overall, the Gain Ratio feature selection method helped streamline performance, particularly enhancing computational efficiency while

preserving accuracy for some classifiers, with Extra Trees and Random Forest remaining top performers.

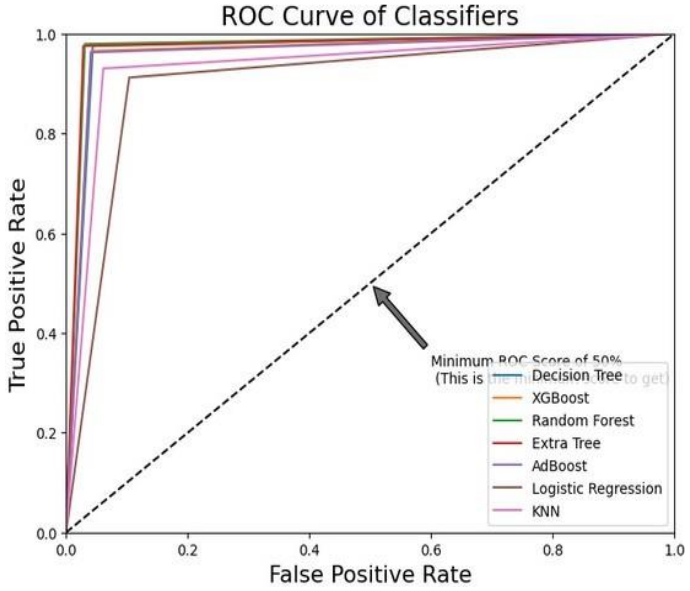

Fig. 4. ROC Curve of Classifiers using Gain Ratio Feature Selection Method

4.5 Performance Evaluation of Machine Learning Classifiers on Phishing URLs Dataset using Information Gain Feature Selection Method

Table 6 illustrates the performance evaluation of individual machine learning classifiers on the phishing URL dataset, employing 28 URL features and the Information Gain Feature Selection Method. Extra Trees (ET) achieved the highest accuracy (98.17%) and F1-measure, along with the lowest FPR (0.0129), making it the top performer in this configuration. The model also demonstrated a moderate training and testing time, showing effective trade-offs. Random Forest (RF)close to ET, with a high accuracy of 98.1% and a low FPR of 0.0156. Although RF has a longer training time (0.9523 sec), its testing time remains efficient, making it another strong performer. Decision Tree (DT)improved slightly, achieving 96.93% accuracy with moderate FPR and FNR values, showing balanced performance and efficiency with low training and testing times. XGBoost and AdaBoost both models experienced a slight decrease in performance, with XGBoost achieving 97.1% accuracy and AdaBoost 96.67%. Both maintained reasonable FPR and FNR values, but AdaBoost had longer training and testing times. Logistic Regression (LR) accuracy dropped to 92.6%, with higher FPR and FNR values, indicating that it was less effective with this feature selection method. K-Nearest Neighbors (KNN) improved slightly to 86.93% accuracy but still lagged behind other models, with the highest FPR (0.174) and a long test time (0.2106 sec), making it less efficient and less reliable for phishing detection. Fig. 5 provides the ROC curve for the classifiers.

Table 6. Performance Evaluation of Machine Learning Classifiers on Phishing URLs Dataset using Information Gain Feature Selection Method

Classifier	Accuracy	Precision	Recall	F1-Measure	FPR	FNR	Train Time (sec.)	Test Time (sec.)
DT	96.93	96.93	96.93	96.93	0.0394	0.0424	0.0620	0.0024
RF	98.1	98.1	98.1	98.1	0.0156	0.0222	0.9523	0.0434
ET	98.17	98.17	98.17	98.17	0.0129	0.0261	0.7145	0.0546
XGBoost	97.1	97.1	97.1	97.1	0.0231	0.0293	0.08414	0.0040
AdaBoost	96.67	96.67	96.67	96.67	0.0299	0.0306	0.4923	0.03310
LR	92.6	92.67	92.6	92.6	0.0918	0.0620	0.1228	0.0033
KNN	86.93	87.13	86.93	86.92	0.174	0.0992	0.0048	0.2106

Applying Information Gain as a feature selection method likely improved the ROC curves for some classifiers, especially for those with increased accuracy and reduced FPR, such as Extra Trees and Random Forest. These classifiers would likely have ROC curves closer to the top-left corner, indicating strong true positive rates and low false positives. The ROC curve analysis for this table would suggest that Extra Trees and Random Forest remain highly effective at distinguishing phishing from legitimate URLs, showing minimal performance trade-offs after reducing the feature set. Overall, Information Gain feature selection yielded high performance for Extra Trees and Random Forest, enhancing the efficiency of phishing detection while maintaining accuracy.

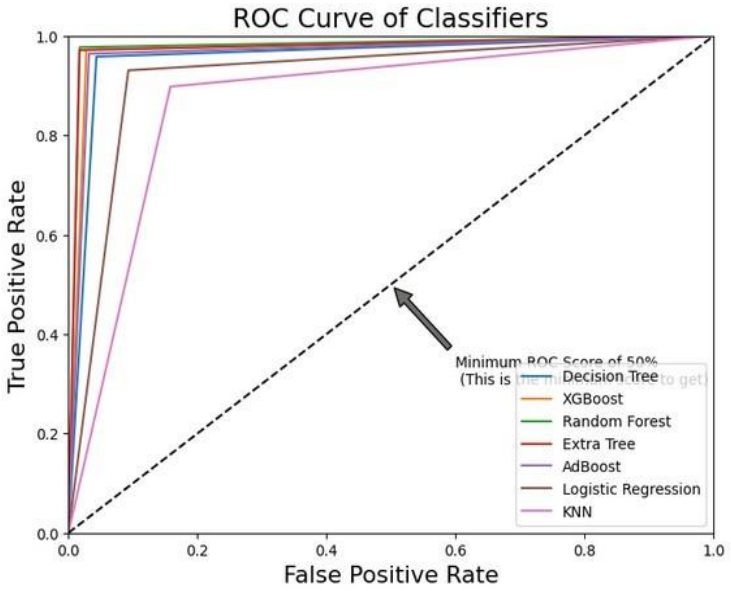

Fig. 5. ROC Curve of Classifiers using Information Gain Feature Selection Method

4.6 Performance Evaluation of Machine Learning Classifiers on Phishing URLs Dataset using Chi-square (χ2) Feature Selection Method

Table 7 presents the performance assessment of individual machine learning classifiers on the phishing URL dataset, employing 28 URL features and the Chi-square (χ^2) Feature Selection Method. Extra Trees (ET)achieves the highest accuracy (98.23%), with strong values across other metrics and a low FPR (0.0151). It also maintains efficient train and test times, making it a top choice for both effectiveness and efficiency. Random Forest (RF) close in performance to ET, with an accuracy of 98.03% and low FPR (0.0158). It has a longer train time (1.0713 sec), but maintains an efficient test time, making it effective for phishing detection. XGBoost and AdaBoost: Both classifiers improve under Chi-square feature selection, with XGBoost achieving 97.5% accuracy and AdaBoost reaching 97.1%. These models exhibit competitive FPR and FNR values, showing effectiveness in phishing detection, though AdaBoost has a relatively higher training time. Decision Tree (DT)also performs well, reaching 97.1% accuracy with low train and test times, making it suitable for phishing detection when quick training and prediction are desired. Logistic Regression (LR) shows improvement under Chi-square selection, achieving 93.47% accuracy. However, it has higher FPR and FNR values than the top performers, indicating limitations in phishing detection. K-Nearest Neighbors (KNN) though improved, KNN remains less effective, with an accuracy of 86.93% and the highest FPR (0.1753), suggesting it may not be as reliable in distinguishing phishing URLs. Moreover, Fig. 6 provides the ROC curve for the classifiers.

Table 7. Performance Evaluation of Machine Learning Classifiers on Phishing URLs Dataset using Chi-square (χ2) Feature Selection Method

Classifier	Accuracy	Precision	Recall	F1-Measure	FPR	FNR	Train Time (sec.)	Test Time (sec.)
DT	97.1	97.1	97.1	97.1	0.0408	0.0343	0.0605	0.0027
RF	98.03	98.03	98.03	98.03	0.0158	0.0168	1.0713	0.05120
ET	98.23	98.23	98.23	98.23	0.0151	0.0303	0.7832	0.06320
XGBoost	97.5	97.5	97.5	97.5	0.0342	0.0242	0.1139	0.0045
AdaBoost	97.1	97.1	97.1	97.1	0.0329	0.0242	0.5158	0.0387
LR	93.47	93.5	93.47	93.47	0.095	.0694	0.1795	0.0068
KNN	86.93	87.21	86.93	86.91	0.1753	0.0971	0.00614	0.2771

Applying Chi-square feature selection is expected to improve ROC curves for classifiers, especially those with increased accuracy and reduced FPR values, such as Extra Trees and Random Forest. These models likely exhibit ROC curves closer to the top-left corner, indicating high true positive rates and low false positives. The ROC curves for ET and RF would likely reflect their ability to distinguish phishing URLs effectively after feature selection, with steeper curves indicating robust classification. In summary, Chi-square feature selection enhances model performance for most classifiers, particularly Extra Trees and Random Forest, resulting in high accuracy, lower error rates, and efficient computational times, making them highly suitable for phishing detection applications.

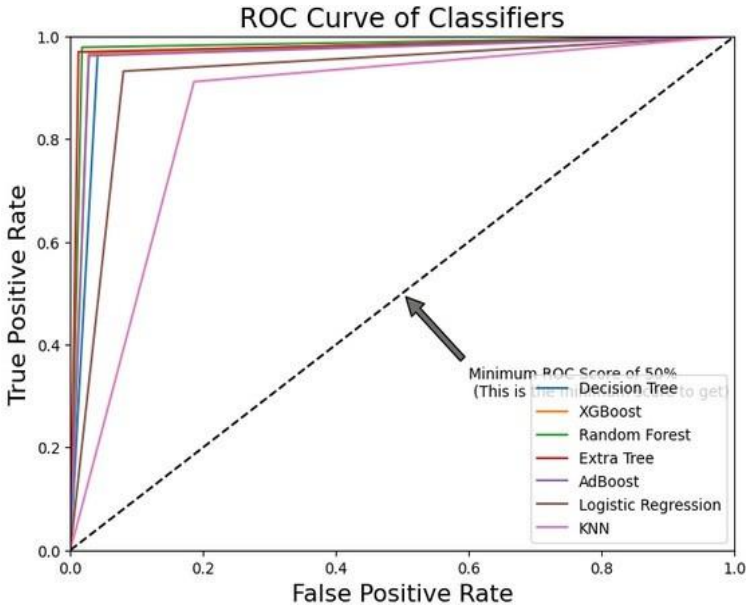

Fig. 6. ROC Curve of Classifiers using Chi-square ($\chi 2$) Feature Selection Method

4.7 Performance Comparisons with Existing Approaches

Table 8 presents performance comparisons with various existing approaches. Optimal detection of phishing URLs was attained by employing the Extra Trees classifier and the Chi-square feature selection method, utilizing only 28 URL features out of a total of 48. The comparison includes benchmarking against state-of-the-art existing approaches such as Cantina [45], Cantina+ [46], Jain [47], Xiang [48], Varshney [49], PhishStorm [50], PhishShield [51] and OFS-NN [52].

Table 8. Performance Comparisons with Existing Approaches

Approach	Accuracy	Precision	Recall	F1-Measure
Cantina	96.9	21.2	89	34.2
Cantina+	95.5	96.4	96.4	96.4
Jain	89.4	99.3	86.1	92.2
Xiang	95.5	95.7	90	92.8
Varshney	93.6	72.3	99.5	83.7
PhishStorm	94.9	98.4	91.3	94.7
PhishShield	96.6	99.9	97.1	98.5
OFS-NN	99.3	96.9	95.9	96.4
Our-approach	**98.23**	**98.23**	**98.23**	**98.23**

Our Approach stands out with a balanced high performance, achieving an accuracy, precision, recall, and F1-Measure of 98.23% across all metrics. This consistency suggests a well-optimized detection capability, with reliable performance across different evaluation criteria. OFS-NN this method achieves the highest accuracy at 99.3%, although it has slightly lower precision, recall, and F1-Measure than our approach, indicating a strong but potentially slightly less balanced detection model. PhishShield this model is also highly effective, with an accuracy of 96.6% and a notably high precision of 99.9%, suggesting excellent specificity in detecting phishing URLs. Cantina shows a marked imbalance between precision (21.2%) and recall (89%), indicating it frequently identifies phishing URLs but is less reliable in the accuracy of its predictions. Varshney exhibits a very high recall of 99.5%, indicating it is highly sensitive to phishing detection but has comparatively lower precision (72.3%), which could mean a higher rate of false positives. Cantina+ and PhishStorm both methods demonstrate well-rounded performance across all metrics, with Cantina+ achieving a precision, recall, and F1- Measure around 96.4%, while PhishStorm achieves a precision of 98.4% and recall of 91.3%. Table 8 highlights that while some existing approaches excel in specific metrics (e.g., OFS-NN in accuracy, PhishShield in precision, and Varshney in recall), our approach provides a high, balanced performance across all metrics (98.23%). This balance indicates that our model is reliable and effective for phishing URL detection without sacrificing precision, recall, or F1-Measure.

5. Security Implications of the Proposed System

The proposed system for enhanced phishing URL detection using feature selection and machine learning approaches presents several important security implications, particularly in the context of handling adversarial attacks and adaptive phishing strategies. Here are some key considerations,

- Robustness Against Adversarial Attacks: Attackers may try to craft URLs that are specifically designed to mislead machine learning models. The system should incorporate techniques to detect adversarial examples by continuously updating its training dataset with newly observed phishing patterns. This could involve retraining the models periodically or using adversarial training techniques to improve resilience.
- Adaptation to Evolving Phishing Techniques: The system should incorporate mechanisms for dynamic learning, where the model can adapt to new phishing strategies as they emerge. This could involve online learning techniques or active learning approaches that allow the model to update its understanding based on new data.
- Feature Update Mechanism: The feature selection methods should be periodically reassessed to ensure that they remain relevant in identifying new phishing tactics. Incorporating feedback loops from user interactions and newly identified phishing attempts can help refine the feature set.
- User Behavior Monitoring: By integrating user behavior analytics, the system can identify anomalies that suggest phishing attempts, even if the URLs themselves appear legitimate. This is particularly useful in adaptive phishing attacks, where the attacker might impersonate trusted entities.
- Hybrid Approach: Combining multiple machine learning algorithms and feature selection techniques can create a more robust detection mechanism that is less susceptible to specific weaknesses of individual models. This ensemble approach can improve overall accuracy and resilience against various phishing tactics.

By addressing these implications, the proposed system can enhance its effectiveness in combating phishing threats, particularly in an ever-evolving threat landscape characterized by increasingly sophisticated attacks. Continuous improvement, user engagement, and a holistic approach to security will be essential to maintaining a robust defense against phishing.

6. Limitations

This paper presents valuable insights and advancements in detecting phishing URLs; however, several limitations should be acknowledged,

- Dataset Limitations: The effectiveness of machine learning models often relies on the quality and diversity of the training dataset. The dataset used in this study is limited in size or lacks diverse examples of phishing URLs, it may not fully capture the variety of phishing tactics employed by attackers.
- Feature Selection Limitations: The chosen feature selection methods may not capture all relevant characteristics of phishing URLs. Some important features may be overlooked, leading to potential gaps in detection capabilities.
- Model Generalization: There is a risk of overfitting the machine learning models to the training data, especially if the models are too complex relative to the dataset. This may lead to poor generalization to unseen phishing URLs in real-world scenarios.
- Performance Variability: The performance of machine learning algorithms can vary based on the specific characteristics of the data. Some classifiers may perform exceptionally well on specific types of phishing URLs but fail to generalize across the entire spectrum of phishing techniques.

7. Conclusion

This paper has provided insights into enhancing the effectiveness of classifiers in identifying phishing URLs. Through meticulous evaluation using various feature selection methods, such as Gain Ratio, Information Gain, and Chi-square (χ^2), demonstrated the notable improvement in detection performance. The experimentation with reduced feature sets, specifically 17, 28, and 48 URL features, showcased significant enhancements in accuracy, precision, recall, and F-measure for diverse machine learning classifiers, including Decision Tree, Random Forest, and KNN. Optimal detection of phishing URLs was attained by employing the Extra Trees classifier and the Chi-square feature selection method, utilizing only 28 URL features out of a total of 48. This underscores the importance of judiciously selecting relevant features, optimizing the detection capabilities of these classifiers while minimizing system overhead. This work underscore the practical applicability of feature selection techniques in refining phishing URL detection systems, thereby contributing to the on- going efforts in cybersecurity. The outcomes not only validate the efficacy of optimal feature selection but also provide a foundation for further exploration and refinement of techniques in the ever-evolving landscape of online security.

The future work should focus on several key areas to further improve the effectiveness and applicability of the proposed detection system. Firstly, expanding and diversifying the dataset will be essential, as collecting more real-world phishing examples can enhance the model's ability to generalize across various phishing tactics. Implementing dynamic feature selection methods that adapt to emerging threats, as well as incorporating behavioral features such as user interaction patterns, can significantly improve detection accuracy.

References

[1] Safi A, Singh S., A systematic literature review on phishing website detection techniques. *Journal of King Saud University-Computer and Information Sciences*. 2023 Jan 11.

[2] Sonowal G, Kuppusamy KS. PhiDMA–A phishing detection model with multi-filter approach. *Journal of King Saud University-Computer and Information Sciences*. 2020 Jan 1;32(1):99-112.

[3] Li T, Kou G, Peng Y. Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods. *Information Systems*. 2020 Jul 1; 91:101494.

[4] Orunsolu AA, Sodiya AS, Akinwale AT. A predictive model for phishing detection. *Journal of King Saud University- Computer and Information Sciences*. 2022 Feb 1;34(2):232-47.

[5] Salahdine F, El Mrabet Z, Kaabouch N. Phishing Attacks Detection a Machine Learning-Based Approach. *In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) 2021*. Dec 1 (pp. 0250-0255). IEEE.

[6] Al-Ahmadi S. PDMLP: phishing detection using multilayer perceptron. *International Journal of Network Security & Its Applications (IJNSA)*. Vol. 2020;12.

[7] Wang W, Zhang F, Luo X, Zhang S. PDRCNN: Precise phishing detection with recurrent convolutional neural networks. *Security and Communication Networks*. 2019 Oct 29; 2019:1-5.

[8] Yi P, Guan Y, Zou F, Yao Y, Wang W, Zhu T. Web phishing detection using a deep learning framework. *Wireless Communications and Mobile Computing*. 2018 Sep 26;2018.

[9] Rahim, M. S., Anwar, F., Saeed, M., & Saleem, K. Phishing detection based on features selection using rough set theory. *PLoS One*. 2019, 14(8), e0218167.

[10] Verma, A., & Ko, R. K. L. Machine learning models for phishing website detection: A survey. *Journal of Network and Computer Applications*. 2020, 166, 102687.

[11] Alazab, M., Layton, R., & Jurdak, R. Phishing in international waters: Exploring the cross-national patterns of phishing delivery and victimization. *Computers & Security*. 2018, 77, 165-175.

[12] Mohammad, A., Thabtah, F., & McCluskey, T. Phishing detection: A recent review and new approaches. *Journal of Information Security and Applications*. 2017, 38, 102-120.

[13] APWG Phishing Activity Trends Report 2nd quarter 2023. Available online, https://docs.apwg.org/ reports/apwg_trends_report_q2_2023.pdf?.

[14] Vijayalakshmi M, Mercy Shalinie S, Yang MH, U RM. Web phishing detection techniques: a survey on the state-of- the-art, taxonomy and future directions. *IET Networks*. 2020 Sep;9(5):235-46.

[15] Catal C, Giray G, Tekinerdogan B, Kumar S, Shukla S. Applications of deep learning for phishing detection: a systematic literature review. *Knowledge and Information Systems*. 2022 Jun;64(6):1457-500.

[16] Basit A, Zafar M, Liu X, Javed AR, Jalil Z, Kifayat K. A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommunication Systems*. 2021 Jan; 76:139-54.

[17] Prabakaran MK, Meenakshi Sundaram P, Chandrasekar AD. An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders. *IET Information Security*. 2023 May;17(3):423-40.

[18] Li T, Kou G, Peng Y. Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods. *Information Systems*. 2020 Jul 1; 91:101494.

[19] Yuan J, Liu Y, Yu L. A novel approach for malicious URL detection based on the joint model. *Security and Communication Networks*. 2021 Dec 13; 2021:1-2.

[20] Alsariera YA, Adeyemo VE, Balogun AO, Alazzawi AK. Ai meta-learners and extra-trees algorithm for the detection of phishing websites. *IEEE access*. 2020 Aug 3; 8:142532-42.

[21] El Aassal A, Baki S, Das A, Verma RM. An in-depth benchmarking and evaluation of phishing detection research for security needs. *IEEE Access*. 2020 Jan 28; 8:22170-92.

[22] Alnemari S, Alshammari M. Detecting phishing domains using machine learning. *Applied Sciences*. 2023 Apr 7;13(8):4649.

[23] Chen YH, Chen JL. Ai@ ntiphish—machine learning mechanisms for cyber-phishing attack. *IEICE Transactions on Information and Systems*. 2019 May 1;102(5):878-87.

[24] Aljofey A, Jiang Q, Qu Q, Huang M, Niyigena JP. An effective phishing detection model based on character level convolutional neural network from URL. *Electronics*. 2020 Sep 15;9(9):1514.

[25] Alshingiti Z, Alaqel R, Al-Muhtadi J, Haq QE, Saleem K, Faheem MH. A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. *Electronics*. 2023 Jan 3;12(1):232.

[26] Yang P, Zhao G, Zeng P. Phishing website detection based on multidimensional features driven by deep learning. *IEEE access*. 2019 Jan 11; 7:15196-209.

[27] Calzarossa MC, Giudici P, Zieni R. Explainable machine learning for phishing feature detection. *Quality and Reliability Engineering International*. 2023.

[28] Mao J, Bian J, Tian W, Zhu S, Wei T, Li A, Liang Z. Phishing page detection via learning classifiers from page layout feature. *EURASIP Journal on Wireless Communications and Networking*. 2019 Dec;2019(1):1-4.

[29] Yang R, Zheng K, Wu B, Wu C, Wang X. Phishing website detection based on deep convolutional neural network and random forest ensemble learning. *Sensors*. 2021 Dec 10;21(24):8281.

[30] Mosa DT, Shams MY, Abohany AA, El-kenawy ES, Thabet M. Machine Learning Techniques for Detecting Phishing URL Attacks. *CMC-COMPUTERS MATERIALS & CONTINUA*. 2023 Jan 1;75(1):1271-90.

[31] Phishing Dataset – A Phishing and Legitimate Dataset for Rapid Benchmarking. Available online, https://www. fcsit.unimas.my/phishing-dataset.

[32] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*. 2009 Nov 16;11(1):10-8.

[33] Qu K, Xu J, Hou Q, Qu K, Sun Y. Feature selection using Information Gain and decision information in neighbor- hood decision system. *Applied Soft Computing*. 2023 Mar 1; 136:110100.
[34] Prasetiyo B, Muslim MA, Baroroh N. Evaluation of feature selection using information gain and gain ratio on bank marketing classification using naive bayes. *In Journal of physics: conference series 2021*. Jun 1 (Vol. 1918, No. 4, p. 042153). IOP Publishing.
[35] Zhai Y, Song W, Liu X, Liu L, Zhao X. A chi-square statistics based feature selection method in text classification. *In 2018 IEEE 9th International conference on software engineering and service science (ICSESS) 2018*. Nov 23 (pp. 160-163). IEEE.
[36] scikit-learn Machine Learning in Python. Available online,https://scikit-learn.org/stable/.
[37] Rokach L, Maimon O. Decision trees. *Data mining and knowledge discovery handbook*. 2005:165-92.
[38] Breiman L. Random forests. *Machine learning*. 2001 Oct; 45:5-32.
[39] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Machine learning*. 2006 Apr; 63:3-42.
[40] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining 2016*. Aug 13 (pp. 785-794).
[41] Schapire RE. Explaining AdaBoost. *In Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik 2013*. Oct 9 (pp. 37-52). Berlin, Heidelberg: Springer Berlin Heidelberg.
[42] Stoltzfus JC. Logistic regression: a brief primer. *Academic emergency medicine*. 2011 Oct;18(10):1099-104.
[43] Peterson LE. K-nearest neighbor. *Scholarpedia*. 2009 Feb 21;4(2):1883.
[44] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. *Information processing & management*. 2009 Jul 1;45(4):427-37.
[45] Zhang Y, Hong JI, Cranor LF. Cantina: a content-based approach to detecting phishing web sites. *In Proceedings of the 16th international conference on World Wide Web 2007*. May 8 (pp. 639-648).
[46] Xiang G, Hong J, Rose CP, Cranor L. Cantina+ a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*. 2011 Sep 1;14(2):1-28.
[47] Jain AK, Gupta BB. A novel approach to protect against phishing attacks at client side using auto-updated white-list. *EURASIP Journal on Information Security*. 2016 Dec; 2016:1-1.
[48] Xiang G, Hong JI. A hybrid phish detection approach by identity discovery and keywords retrieval. *In Proceedings of the 18th international conference on World wide web 2009*. Apr 20 (pp. 571-580).
[49] Varshney G, Misra M, Atrey PK. A phish detector using lightweight search features. *Computers & Security*. 2016 Sep 1; 62:213-28.
[50] Marchal S, Fran ̧ois J, State R, Engel T. PhishStorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management*. 2014 Dec 4;11(4):458-71.
[51] Rao RS, Ali ST. PhishShield: a desktop application to detect phishing webpages through heuristic approach. *Procedia Computer Science*. 2015 Jan 1; 54:147-56.
[52] Zhu E, Chen Y, Ye C, Li X, Liu F. OFS-NN: an effective phishing websites detection model based on optimal feature selection and neural network. *IEEE Access*. 2019 Jun 4; 7:73271-84.

Authors' Profiles

Dharmaraj R. Patil holds a Ph.D. in computer engineering from Kavayitri Bhahinabai Chaudhari North Maharashtra University, Jalgaon, Maharashtra, India, and a master's degree in computer science and engineering from Government College of Engineering, Aurangabad, Maharashtra, India. He is an associate professor at the R.C. Patel Institute of Technology in Shirpur, Maharashtra, India, in the department of computer engineering. He has been a teacher for twenty years. Web mining, intrusion detection, and web security are his areas of interest in research. He has numerous papers published in journals and international/national conferences.

Rajnikant B. Wagh holds a Ph.D. in computer engineering from Kavayitri Bhahinabai Chaudhari North Maharashtra University, Jalgaon, Maharashtra, India. He is an associate professor at the R.C. Patel Institute of Technology in Shirpur, Maharashtra, India, in the department of computer engineering. His areas of interest are Data Mining, Cryptography, Web Personalization and Recommender Systems. He has published many papers in international/national conferences and journals.

Vipul D. Punjabi is a dedicated scholar and educator currently pursuing a Ph.D. in Computer Science and Engineering. He completed his Master of Technology in Information Technology from Technocrats Institute of Technology, Bhopal, affiliated with RGPV University, in 2013. His journey in the field of computer engineering began with his Bachelor of Engineering degree, which he completed in 2006 from R.C. Patel Institute of Technology, Shirpur, driven by his passion for both learning and teaching, Vipul serves as an Assistant Professor at R.C. Patel Institute of Technology, Shirpur, where he imparts knowledge and mentors aspiring engineers. With his academic achievements and professional experience, Vipul continues to contribute significantly to the field of computer science, both through his research endeavors and his dedication to shaping

the next generation of technologists.

Shailendra M. Pardeshi is a research scholar at Oriental University's Department of Computer Science & Engineering (CSE) in Indore, Madhya Pradesh, India. In 2012, He received M. Tech. degree in CSE from TIT under RGPV in Bhopal, India. In 2006, he earned a Bachelor Degree B.E. from R C Patel Institute of Technology in Shirpur, Maharashtra, India. He had published various papers in journals and conferences included in Scopus, WOS, IEEE, Springer and reputed indexing. He also applied for a design patent. In 2022, he received the Best Researcher honour. His current areas of study include Emotion and sentiment and analysis through the use of machine learning (ML) techniques.

How to cite this paper: Dharmaraj R. Patil, Rajnikant B. Wagh, Vipul D. Punjabi, Shailendra M. Pardeshi, "Enhanced Phishing URLs Detection using Feature Selection and Machine Learning Approaches", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.14, No.6, pp. 48-67, 2024. DOI:10.5815/ijwmt.2024.06.04