



INTELLIGENT METHODS IN CYBER DEFENCE: MACHINE LEARNING BASED PHISHING ATTACK DETECTION ON WEB PAGES

Remzi GÜRFİDAN*

Isparta University of Applied Sciences, Yalvaç VSTS, Department of Computer Programming, Isparta, Türkiye

Keywords

*Phishing Attack,
Cyber Security,
Machine Learning,
Web Site Phishing Attack
Detection,
Extra Trees.*

Abstract

Phishing attack on web pages is a type of malicious attack that aims to steal personal and sensitive information of internet users. Phishing attacks are usually conducted through various communication channels such as email, SMS, social media messages or websites. Users are directed to fake web pages of trusted organizations such as government agencies, banks, online shopping sites, etc. and asked to enter their personal information. These fake web pages may look remarkably like the original sites and are designed to mislead users. In this study, we used machine learning methods to detect the phishing attack threat of web pages and made significant progress in this area. Extensive analysis of six different machine learning algorithms showed that the Extra Trees algorithm yielded the most successful results. To further improve this success, we fine-tuned the Extra Trees algorithm and increased the correct classification success to 97.9%. In future studies, we would like to expand the dataset to include other machine learning methods to investigate the use of this technology in areas such as malware detection or the prevention of phishing attacks. This would be a crucial step towards providing more comprehensive protection in the field of cybersecurity.

SİBER SAVUNMADA AKILLI YÖNTEMLER: WEB SAYFALARINDA MAKİNE ÖĞRENİMİ TABANLI KİMLİK AVI TESPİTİ

Anahtar Kelimeler

*Phishing saldırısı,
Siber Güvenlik,
Makine Öğrenmesi,
Web Site Phishing Saldırı
Tespiti,
Extra Trees.*

Öz

Web sayfalarında ortalama saldırısı, internet kullanıcılarının kişisel ve hassas bilgilerini çalmayı amaçlayan kötü niyetli bir saldırı türüdür. Ortalama saldırıları genellikle e-posta, SMS, sosyal medya mesajları veya web siteleri gibi çeşitli iletişim kanalları aracılığıyla gerçekleştirilir. Kullanıcılar devlet kurumları, bankalar, çevrimiçi alışveriş siteleri gibi güvenilir kuruluşların sahte web sayfalarına yönlendirilir ve kişisel bilgilerini girmeleri istenir. Bu sahte web sayfaları orijinal sitelere oldukça benzeyebilir ve kullanıcıları yanıltmak için tasarlanmıştır. Bu çalışmada, web sayfalarının kimlik avı tehdidini tespit etmek için makine öğrenimi yöntemlerini kullandık ve bu alanda önemli bir ilerleme kaydettik. Altı farklı makine öğrenimi algoritmasının kapsamlı analizi, Extra Trees algoritmasının en başarılı sonuçları verdiğini gösterdi. Bu başarıyı daha da artırmak için Extra Trees algoritmasında ince ayarlar yaptık ve doğru sınıflandırma başarısını %97,9'a çıkardık. Gelecekteki çalışmalarda, bu teknolojinin kötü amaçlı yazılım tespiti veya kimlik avı saldırılarının önlenmesi gibi alanlarda kullanımını araştırmak için veri kümesini diğer makine öğrenimi yöntemlerini içerecek şekilde genişletmek istiyoruz. Bu, siber güvenlik alanında daha kapsamlı koruma sağlamaya yönelik çok önemli bir adım olacaktır.

Alıntı / Cite

Gürfidan, R., (2024). Intelligent Methods in Cyber Defence: Machine Learning Based Phishing Attack Detection on Web Pages, *Journal of Engineering Sciences and Design*, 12(2), 416-429.

Yazar Kimliği / Author ID (ORCID Number)

R. Gürfidan, 0000-0002-4899-2219

Makale Süreci / Article Process

Başvuru Tarihi / Submission Date	26.03.2024
Revizyon Tarihi / Revision Date	14.05.2024
Kabul Tarihi / Accepted Date	11.06.2024
Yayın Tarihi / Published Date	30.06.2024

* İlgili yazar / Corresponding author: remzigurfidan@isparta.edu.tr, +90-246-441-5300

INTELLIGENT METHODS IN CYBER DEFENCE: MACHINE LEARNING BASED PHISHING ATTACK DETECTION ON WEB PAGES

Remzi Gürfidan[†]

Isparta University of Applied Sciences, Yalvaç VSTS, Department of Computer Programming, Isparta, Türkiye

Highlights

- A method for detecting and preventing phishing attack threats in web pages is proposed.
- An alternative machine learning based cyber security tool is developed.
- An existing machine learning algorithm was fine tuned for the dataset to improve its performance.

Graphical Abstract

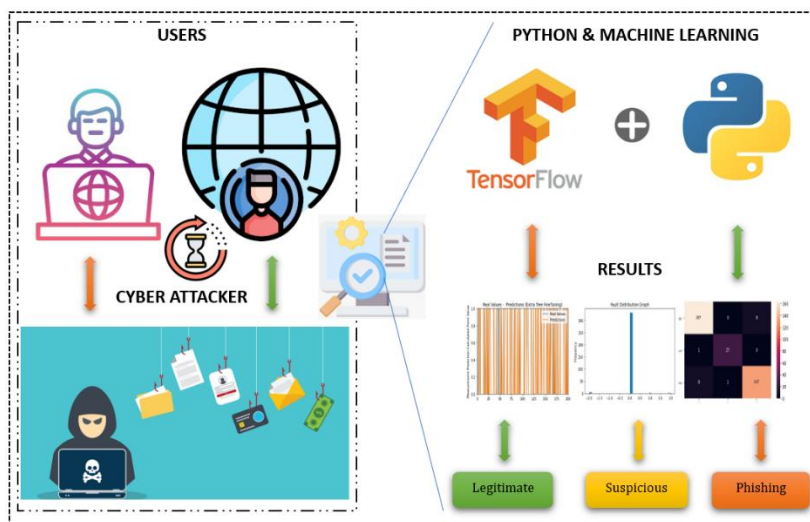


Figure. Graphical abstract of the process of the work

Purpose and Scope

Classifying, Detecting and Preventing Phishing Attacks on Web Pages.

Design/methodology/approach

To achieve the objectives, the usable dataset was chosen, and machine learning methods were utilised. In order to increase the success of the classification process, hyperparameter optimisation was performed and high accuracy was achieved.

Findings

The Extra Trees algorithm showed the most successful results with 88% successful classification. This rate was increased to 97% with fine tuning.

Research limitations/implications

This study is limited to the current dataset and the success of the machine learning methods used. In future studies, the dataset will be expanded, and different machine learning methods will be used to achieve higher success.

Social Implications

The findings of the present study are quite satisfactory. It is aimed to integrate the machine learning model into a software to be developed in the future and turn it into a real-time application. In this way, warning and prevention activities can be conducted when visitors visit sites that contain Phishing attack danger.

Originality

When this study is embedded and activated in a real-time application, it will be effective in protecting people with limited security knowledge about internet use and in the security of standard users. In addition, an alternative cyber security tool based on machine learning is proposed.

[†] Corresponding author: remzigurfidan@isparta.edu.tr, +90-246-441-5300

1. Introduction

Internet fraud is a type of offence that has emerged because of modern technological developments and aims to obtain personal information, material resources or property unfairly by abusing the online environment. Such offences are usually committed by hackers or other malicious persons and are conducted by various methods (Balogun, Akande, et al., 2021; Balogun, Mojeed, et al., 2021). Internet fraudsters operate by using various fraudulent tactics by gain phishing attacking or misleading the trust of users. One of the most common methods in this context is sending fake e-mails, a type of so-called "phishing". Phishing attack aims to capture users' personal information through fake emails, usually mimicking the name of an official organisation or service. These phony emails could include links in them that request personal data from recipients, such as credit card numbers, usernames, and passwords. Furthermore, the creation of fictitious websites is another often used technique in online fraud. Fraudsters can create fake websites by imitating a respectable and well-known company or organization in order to collect user data or install malicious software (malware). Often, these fake websites mimic real websites in an attempt to deceive users. Identity theft is another way that fraud is committed online. The act of getting someone's personal information—such as name, address, date of birth, or social security number—and using it for illegal activities is known as identity theft. Scammers can use this information to open false accounts, apply for credit cards, and commit other crimes. Ransomware, social engineering, and spoof websites are some other techniques used in online fraud. Every one of these methods puts users' privacy at risk and has the potential to cause serious ethical and financial problems. Because of this, users must be extremely cautious while using the Internet, only depend on reliable sources of information, and strictly adhere to online security recommendations (Wu et al., 2019).

Phishing attack is a cyber security risk that increases consumers' risk of having their information stolen (Mithra Raj & Arul Jothi, 2022). These attacks usually use bogus websites or communication methods and target the financial or personal information of their victims. Researchers and security experts have created a plethora of techniques to recognize Phishing attacks. In an attempt to differentiate between trustworthy and fake websites, users should first carefully review the URLs of any possibly dangerous websites (Jain & Gupta, 2019). Security measures such as SSL certificates and URLs beginning with the "https://" protocol are often absent from Phishing attack sites. Moreover, a thorough examination of Phishing attack websites might reveal grammatical and linguistic errors. Avoiding forms that ask for personal information is vital advice; a trustworthy website will usually provide choices for verification. Websites that request personal information from you through an information form out of the blue should cause you to be wary. Because Phishing attack websites sometimes conceal or offer erroneous information, it is especially important to verify the contact information.

People who fall victim to Phishing attack might face several risks, which could have serious consequences. The conditions surrounding people who fall victim to Phishing attack attempts may be examined from a number of perspectives, and the importance of this problem necessitates action from the individual as well as the greater society. First and foremost, there is a chance that victims of Phishing attack might lose money. Fraudsters can access victims' bank accounts, take credit card details, and use the personal information they have obtained for malicious purposes. The victims may suffer financial losses in addition to long-term financial difficulties as a result of this. In addition, Phishing attack victims could experience damage to their reputation. If the victims' compromised personal information is used maliciously, it might damage their reputation and cause a reduction in confidence in both personal and professional relationships. Phishing attacks directed at companies or professionals have the capacity to gravely damage the careers and commercial relationships of its victims. Victims of Phishing attack may also be at risk for personal safety breaches. If fraudsters use the personal information, they get to access other online accounts or services, victims' privacy can be compromised. This might make it more likely that victims would fall victim to Internet crimes like identity theft and exploitation of their personal data.

It's crucial for Internet users to recognize and avoid phishing attacks. To detect these hazards at the human and organizational levels, a variety of strategies can be used. Website Phishing attack detection is essential for user security and for encouraging responsible online behavior (Barraclough et al., 2021). Phishing attack detection on websites can help consumers recognize potential risks and increase their awareness of security concerns. This can assist users in identifying bogus websites or emails sent by fraudsters, protecting their personal information and promoting a secure online experience. Therefore, it's imperative that Internet users protect their personal information online and take preventative measures against Phishing attack schemes. Companies should also provide Phishing attack awareness training to their employees and take proactive measures to detect and prevent Phishing attack. This will ensure user security and strengthen the online environment's defenses against phishing attacks. The literature will be examined for research on Phishing attack detection and prevention, and the results will be incorporated in the second section of the study. The third section offers a thorough discussion of the data set that was used, along with an explanation of the machine learning techniques that were employed. The

outcomes and conclusions drawn from machine learning procedures are provided in the fourth part. The final part discusses how this study compares to the body of current literature and presents the study's findings.

The literature will be examined for studies on Phishing attack detection and prevention, and the results will be incorporated in the second section of the study. The third section offers a thorough discussion of the data set that was used, along with an explanation of the machine learning techniques that were employed. The outcomes and conclusions drawn from machine learning procedures are provided in the fourth part. The final section discusses how this study compares to the body of current literature and presents the study's findings.

2. Literature Survey

Using an advanced AC approach called Multi-Labelled Classifier Based Associative Classification (MCAC), Abdelhamid et al. looked into the online Phishing attack problem in their study to see whether it might be applied to the problem. Empirical findings utilizing authentic data gathered from various sources indicate that AC, particularly MCAC, identifies Phishing attack websites more accurately than other clever algorithms. Additionally, MCAC produces additional hidden information (rules) that are not discovered by other algorithms, enhancing classifier prediction performance. Using MCAC, an accuracy of about 94% was attained (Abdelhamid et al., 2014). Yi et al. largely uses a deep learning system to detect Phishing attack websites. The study begins by designing two sorts of web Phishing attack features: original and interactive elements. A detection model based on Deep Belief Networks (DBN) is presented in the ensuing section. The test, which uses actual IP streams from Internet service providers (ISPs), shows that the DBN-based detection model can achieve a true positive rate of around 90% and a false positive rate of about 0.6% (Yi et al., 2018). Ying and Xuhua suggest a novel technique that is independent of any particular Phishing attack application. The work's goal is to investigate anomalies in Web sites, namely the difference between a website's identity and its structural elements, as well as HTTP transactions. Approximately 88% success is achieved in detecting Phishing attack pages (Ying & Xuhua, 2006).

To identify online phishing attacks, Adeyemo et al. suggested employing ensemble-based Logistic Model Trees (LMT). To generate a single model tree, logistic regression and tree induction techniques are used in LMT. The testing results show that the suggested techniques, with at least 97.18% accuracy and area under curve (AUC) values of 0.996, are quite successful in identifying Phishing attack websites. Additionally, the suggested approaches perform better than a number of machine learning-based phishing attack models found in recent research. Thus, it is advised to use the provided methods to handle dynamic website phishing attacks (Adeyemo et al., 2021). Moghimi and Varjani's study introduced a brand-new rule-based technique for identifying Phishing attack scams in online banking. Two newly proposed feature sets are used for web page identification in the rule-based approach. The web pages were classified using the support vector machine (SVM) technique. Our tests demonstrate that the suggested model has an accuracy of only 0.86% for false negative alarms and 99.14% for true positives when it comes to identifying Phishing attack pages in online banking (Moghimi & Varjani, 2016). Convolutional neural networks (CNNs) are used in Yerima and Alzaylaee's high accuracy classification system, one can distinguish between authentic and fake websites. Their algorithm is trained on a dataset of 4,898 Phishing attack and 6,157 genuine websites. Our CNN-based algorithms have shown to be successful in recognizing unknown Phishing attack sites through extensive experiments. Furthermore, the CNN-based approach achieved an F1_Score of 0.976 and a Phishing attack detection rate of 98.2%, outperforming other machine learning classifiers evaluated on the same dataset (Yerima & Alzaylaee, 2020). In their work, Rashid et al. presented a successful machine learning-based Phishing attack detection method. Overall, the testing findings demonstrate that the suggested method performs best when combined with the Support vector machine classifier, correctly identifying 95.66% of Phishing attack and suitable websites with just 22.5% of the creative functionality needed. When compared to a set of common Phishing attack datasets from the "University of California Irvine (UCI)" archive, the suggested technique yields encouraging results (Rashid et al., 2020). Sahingoz et al. propose a real-time anti-Phishing attack system that employs NLP-derived features and seven distinct categorization approaches. A new dataset is generated and used to test experimental outcomes in order to gauge the system's performance. Based on comparison and experimental findings from several applicable classification methods, the Random Forest approach using only NLP-based characteristics performs best, detecting Phishing attack URLs with an accuracy of 97.98% (Sahingoz et al., 2019). Three deep learning-based methods for identifying Phishing attack websites were proposed by Alshingiti et al.: an LSTM-CNN based strategy, a CNN for comparison, and long short-term memory (LSTM) for detection. The accuracy of the suggested methodologies is demonstrated by the experimental findings, which are 99.2%, 97.6%, and 96.8% for CNN, LSTM-CNN, and LSTM, respectively (Alshingiti et al., 2023). Dhanavanthini ve Chakkravarthy uses recurrent neural networks (RNN) to deliver state-of-the-art accuracy in identifying harmful URLs. This effort aims to concentrate just on the content included in the URL, which speeds up the process and demonstrates how early detection of zero-day attacks is possible. Prior research examines URLs, traffic figures, and Internet content. The RNN in the paper is optimized to be used on small devices, such Raspberry Pis and mobile phones, without sacrificing inference time (Dhanavanthini & Chakkravarthy, 2023).

3. Material and Method

This section presents the purpose and basic mathematical calculations of the machine learning algorithms used in this study. In addition, the metrics used to determine the success of the trained machine learning models and the calculation methods of the metrics are shown. In the last sub-section of the chapter, the dataset used in this study and the features of the dataset are explained.

3.1. Extra Tree Algorithm

The Extra Trees (Extremely Randomized Trees) algorithm is a machine learning method specifically used to solve classification and regression problems. Extra Trees is a method based on decision trees. It uses many trees like the Random Forest algorithm as a working logic. In addition, unlike Random Forest, Extra Trees takes more randomness into account when constructing trees (Breiman, 2001; Geurts et al., 2006). G denotes the prediction tree. Here θ denotes a uniform independent distribution vector that is assigned before the growth of the tree. All trees are combined and averaged into a tree ensemble of $G(x)$, which is generated using the Breiman, 2001 equation (Equation 1) (Hammid et al., 2018).

$$G(x, \theta_1, \dots, \theta_2) = \frac{1}{2} \sum_{r=1}^R G(x, \theta_r) \quad (1)$$

GridSearchCV is a hyperparameter tuning method available in the scikit-learn library. It is used to experiment with various combinations of hyperparameters used to improve the performance of a model. By specifying a given hyperparameter space (parameter combinations), it evaluates the performance of the model for different combinations in that space and selects the hyperparameters that perform best. GridSearchCV tries to select the best hyperparameters by cross-validating over the specified hyperparameter combinations. In this study, the GridSearchCV method was applied to the most successful ExtraTree algorithm and the best values of the selected hyperparameters were determined. These best parameter values were then used to train the model. The tested and found hyperparameter values are shown in Table 2.

Table 2. Hyperparameters tried to be optimized and values tested

Parameters and Their Values Tested for Hyperparameter Optimization	n_estimators': [50, 100, 200] 'max_depth': [None, 10, 20, 30] 'min_samples_split': [2, 5, 10] 'min_samples_leaf': [1, 2, 4]
Hypermeter Values	max_depth': 20 'min_samples_leaf': 2 'min_samples_split': 2 'n_estimators': 100

3.2. K-Nearest Neighbours Algorithm

K-Nearest Neighbors (KNN) is an effective machine learning method that is preferred as a classification or regression solver. The algorithm uses the classes or values of the nearest neighbouring points to classify or predict a new data point. The basic principle of KNN proceeds by recognizing that data points with similar characteristics tend to have the same class or a similar value. Considering x and y as axis values, after calculating the distance, the input x is considered as the class value with the highest probability. This is calculated by Equation 2.

$$P(y = j|X = x) = \frac{1}{K} \sum_{i \in A} I(y^{(i)} = j) \quad (2)$$

3.3. Decision Tree Algorithm

A machine learning technique called the decision tree algorithm is used to predict and categorize a target variable's (dependent variable's) numerical value based on specific dataset feature values. Regression analysis using decision trees is a popular tool for identifying and visualizing patterns in data sets. Decision tree regression uses a set of criteria, like information gain, the Gini coefficient, or other measurements, to identify the optimal partition when splitting the dataset. To segment the dataset as efficiently as possible, a sequence of decisions must be made next. Decision tree regression can therefore be used to forecast the target variable and uncover intricate relationships within the dataset.

$$y(x) = f(x) - \sum_{i=1}^N c_i I(x \in R_i) \quad (3)$$

$y(x)$ is the estimated target variable value. x is the feature vector of the data point. $f(x)$ is the predicted value of the data point. N is the total number of nodes in the tree. C_i is the estimated value at the i -th node. $I(x \in R_i)$ is an indicator function that indicates whether the data point belongs to the i -th region (node). It takes the value 1 if x is in that region and 0 otherwise. R_i denotes a specific feature range of the i -th region (node).

3.4. Gradient Boosting Algorithm

Gradient Boosting is a machine learning algorithm used as a solver in classification and regression processes. This algorithm aims to create a strong learner by combining weak learners together. Gradient Boosting aims to combine weak predictors (usually decision tree type models) to create a strong prediction model. The basic principle of how this algorithm works is to correct the erroneous learning of the previous weak estimator by adding new estimators. This process affects the calculation of the weights, while the new values are determined by the loss function. Equation 4 is used for the overall model calculation.

$$\gamma_m = \arg \min_{\gamma} \sum_i^n L(y_i, F_{m-1}(x_i) + \gamma) \quad (4)$$

Here $i = 1 - n$ belongs to r_{ij} , where j represents the leaf. y is the observed value, γ is the predicted value

3.5. Random Forest Algorithm

Random Forest is a machine learning algorithm that is widely used especially in classification and regression problems. Random Forest can create a more powerful and generalizable model by combining multiple decision trees. When decision trees are configured for regression models, the average of the decision trees is the prediction value. Random Forest uses randomization to minimize the risk of overfitting. Random feature selections and random generation of data subsets make the model more diverse and generalizable. Mean square error value for Random Forest is calculated as in Equation 5.

$$RF_{MSE} = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2 \quad (5)$$

Where N is the number of data points, f_i is the value returned by the model and y_i is the actual value for data point i .

3.6. AdaBoost Algorithm

AdaBoost (Adaptive Boosting) is an ensemble learning algorithm for building strong models. AdaBoost aims to build a stronger model by combining weak models together. The AdaBoost algorithm is an algorithm that works on weights and each weak classifier is assigned a weight. Once a classifier is trained on the weighted training set, the weights of the misclassified examples are increased, and the next classifier is trained on this updated weighted data set. This process continues until a desired number of iterations or specific learning objective is reached. Equation 6 is used for the overall model calculation. The error rate is calculated by ϵ_t , that is, It shows how well the t 'th classifier is able to correct the errors made on the weighted training data set.

$$\epsilon_t = \frac{\sum_{i=1}^N w_{i,t} \cdot 1(h_t(x_i) \neq y_i)}{\sum_{i=1}^N w_i} \quad (6)$$

It's here, The 1 function is a function that indicates whether the expression in parentheses is true (1) or false (0).

3.7. Evaluation of the models

The performance of the model is measured using error metrics, which are employed to assess the effectiveness of machine learning algorithms. These metrics aid in evaluating the degree to which a model's predictions agree with actual values and its capacity for generalization.

Mean absolute error (MAE) is a metric that shows how close the predicted values are to the true values. This metric is calculated by Equation 7 (AlOmar et al., 2020; Hammid et al., 2018; Mishra et al., 2017).

$$MAE = \frac{1}{n} \sum_{r=1}^n |P_d^{r,m} - P_d^{r,c}| \quad (7)$$

To compare the prediction errors of several trained models, root mean square error, or RMSE, was selected. The model's ability to forecast absolute deviation is better the closer the RMSE value is to 0. Calculating the RMSE value is done using Equation 8 (AlOmar et al., 2020; Hammid et al., 2018; Mishra et al., 2017; Willmott & Matsuura, 2005).

$$RMSE = \sqrt{\frac{1}{n} \sum_{r=1}^n (P_d^{r,m} - P_d^{r,c})^2} \quad (8)$$

The coefficient of determination (R2) is used to estimate model efficiency and is calculated by Equation 9 (Hammid et al., 2018).

$$R^2 = 1 - \frac{\sum_{r=1}^n (P_d^{r,m} - P_d^{r,c})^2}{\sum_{r=1}^n (P_d^{r,m} - P_d^{r,m})^2} \quad (9)$$

MSE either assesses the quality of an estimator. The MSE metric is calculated by Equation 10.

$$MSE = \frac{1}{n} \sum_{r=1}^n (P_i - P'_i)^2 \quad (10)$$

3.8. Dataset Description

The dataset used for the training of machine learning algorithms in this study is the "Website Phishing Data Set", which is available on the Kaggle platform and is openly available to users (*Website Phishing Dataset*, n.d.). There are a total of 10 features in the dataset. This dataset was created by identifying distinctive characteristics of legal and Phishing attack websites and collecting 1353 different websites from different sources. Phishing attack websites were collected from the Phishtank data archive, a free community website where anybody can upload, verify, monitor, and exchange Phishing attack data. A PHP web program was used to collect real webpages from the Yahoo and starting point directories. After installing the PHP script in a browser, 548 trustworthy websites out of 1353 were gathered. 103 dubious URLs and 702 Phishing attack URLs were found. A website is deemed to be SUSPECTED if it is thought to include both valid and Phishing attack elements. This could indicate that the website is Phishing attack or legitimate. Table 3 displays the fields that were part of the data collection that was used.

Table 3. Site characteristics in the data set

URL Anchor	Request URL	SFH
URL Length	Having '@'	Prefix/Suffix
IP	Sub Domain	Web traffic
Domain age	Class	

The importance of each feature in the data set in Phishing attack detection was analysed by defining them as items. Afterwards, the effects and importance levels of the input features on the result in the current data set were calculated and transformed into the form shown in Table 4. In this way, the input features in the data set become much more meaningful and interpretable.

- **URL:** In phishing attacks, fake websites often use similar URL structures to real sites. Therefore, careful examination of the URL will help users to recognise fake sites.
- **Anchor:** In phishing attacks, malicious links are often disguised with misleading texts. Therefore, the texts of the links should be carefully examined and evaluated whether they are reliable.
- **Request URL:** In phishing attacks, malicious content and scripts are often loaded from external sources. Therefore, attention should be paid to whether the URLs requested by a web page are reliable.
- **SFH URL Length:** Same Origin Policy URL length should be checked if a web page redirects to resources that do not belong to its domain. This may indicate a potentially malicious redirect.
- **Having '@' (Email Spoofing):** In phishing attacks, fake email addresses and sender names are used to send credible-looking messages. Therefore, it is important to carefully examine email addresses and senders.
- **Prefix/Suffix IP:** In phishing attacks, misleading connections can be created using IP addresses. For this reason, it is necessary to carefully check IP addresses and determine whether they are reliable.

- **Sub Domain:** In phishing attacks, fake websites often use subdomains similar to the main domain name. Therefore, it is important to carefully examine subdomains.
- **Web Traffic:** In phishing attacks, websites with popular and heavy traffic may be more targeted. Therefore, the traffic of a website should be carefully evaluated.
- **Domain Age:** Newly created or recently registered domains may have been created to potentially be used for Phishing attack. Therefore, the age and registration process of a website should be considered.

Table 4. Graph of the influence values and importance levels of input features on the result in the data set

Dataset Coloumn	Feature Weight	Feature Weight Graph
having_IP_Address	0.1135	
web_traffic	0.0438	
id	-	
URL_of_Anchor	-0.0352	
age_of_domain	-0.0487	
URL_Length	-0.0594	
Request_URL	-0.1204	
SSLfinal_State	-0.3267	
popUpWidnow	-0.3707	
SFH	-0.4346	

PCA (Principal Component Analysis) stands for principal component analysis. PCA is a statistical technique used to understand the relationships between variables in multivariate data sets and to express the data set with fewer variables. The numerical results and visual example obtained when PCA is applied on the dataset are shown in Figure 6 and Table 5.

Table 5. The numerical results of PCA

Features and Metrics	Values	
Variance explanation percentage (PC1)	25.91%	
Variance explanation percentage (PC2)	16.15%	
Eigenvalue of PC1	2.333	
Eigenvalue of PC2	1.454	
Component Loadings	PC1	PC2
SFH	0.46	0.27
popUpWidnow	0.35	0.22
SSLfinal_State	0.36	0.00
Request_URL	0.25	0.38
URL_of_Anchor	0.31	0.38
web_traffic	-0.40	0.53
URL_Length	0.22	0.06
age_of_domain	0.38	-0.55
having_IP_Address	0.16	-0.01

- **Variance Explanation Percentage (PC1 and PC2):**

PC1 explains 25.91% and PC2 explains 16.15% of the total dataset. PC1 explains a larger proportion of the total variance, indicating that PC1 retains more information and contains more variability.

- **Eigenvalues:**

The eigenvalue of PC1 is 2.333 and the eigenvalue of PC2 is 1.454. Eigenvalues measure the variance of each component in the original data set. Components with larger eigenvalues retain more variance.

- **Component Loadings:**

Component loadings indicate the relationship of each component to the original variables. Positive or negative loadings indicate the direction of the relationship between variables. Loadings with large absolute values indicate that the variable plays an important role in the formation of the relevant component.

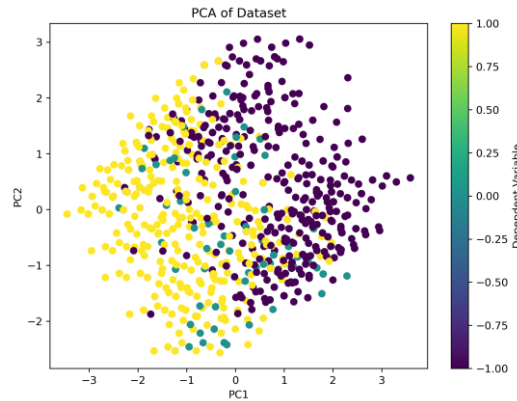


Figure 2. PCA of dataset

- Variables such as SFH, popUpWidnow, SSLfinal_State, Request_URL, URL_of_Anchor, age_of_domain have large loads in both components. These variables play an important role in the key components that make up the PCA results.
- Variables such as web_traffic, age_of_domain are positively related to PC1 and negatively related to PC2. These variables play an important role in determining the differences between PC1 and PC2.
- The loadings of variables such as having_IP_Address, SSLfinal_State are significantly less in PC2. These variables contribute less to the formation of PC2.

4. Findings and Results

Table 6 shows the error metrics and accuracy values obtained in the training of Decision Tree and Random Forest algorithms. In addition, the graph showing the similarity between the real values and the values classified by the trained model, the error distribution graphs in classification and the confusion matrix tables are shown.

Table 6. Decision Tree and Random Forest Algorithm performance

DECISION TREE ALGORITHM		RANDOM FOREST ALGORITHM	
Metric	Values	Metric	Values
MAE:	0.251	MAE:	0.211
MSE:	0.477	MSE:	0.379
RMSE:	0.691	RMSE:	0.615
Acc.:	0.862	Acc.:	0.871
Fault Distribution			

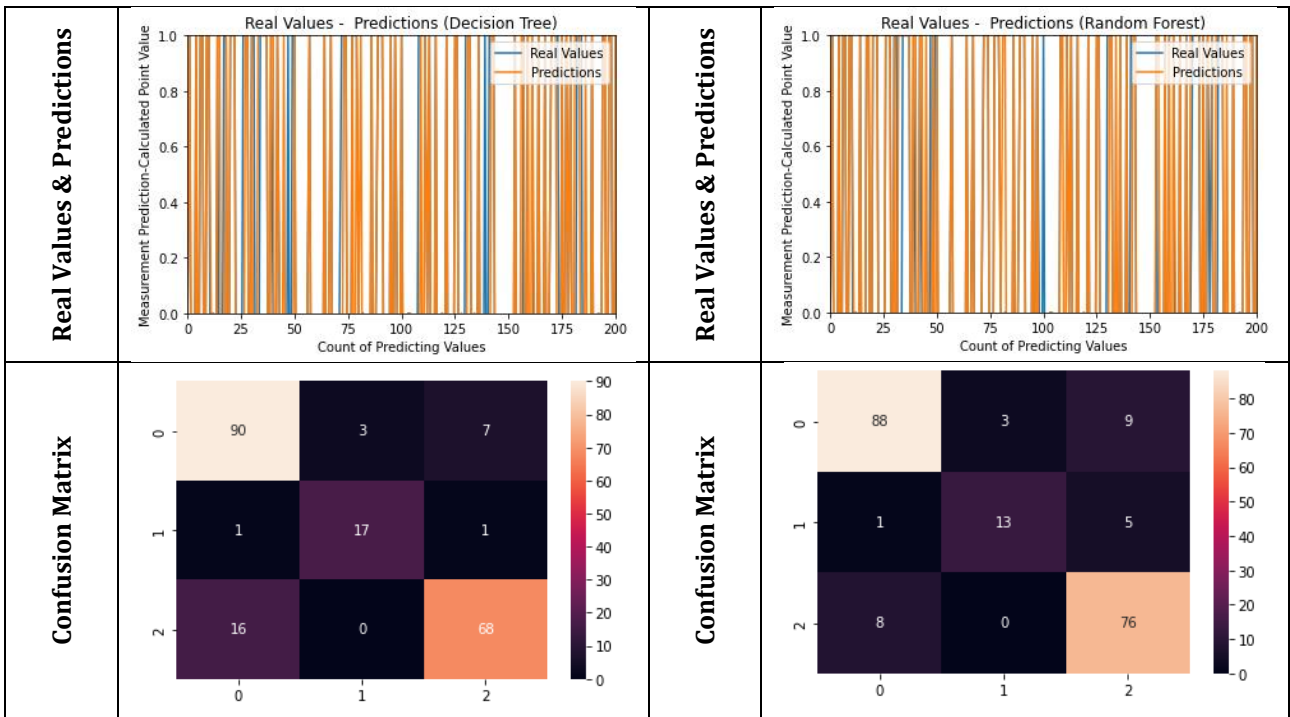


Table 7 shows the error metrics and accuracy values obtained in the training of AdaBoost Classifier and KNN algorithms. In addition, the graph showing the similarity between the real values and the values classified by the trained model, the error distribution graphs in classification and the confusion matrix tables are shown.

Table 7. AdaBoost Algorithm and KNN Algorithm performance

ADABOOST CLASSIFIER ALGORITHM		KNN ALGORITHM	
Metric	Values	Metric	Values
MAE:	0.285	MAE:	0.891
MSE:	0.463	MSE:	1.669
RMSE:	0.680	RMSE:	1.292
Acc.	0.802	Acc.	0.497
Fault Distribution		Fault Distribution	
Real Values & Predictions		Real Values & Predictions	

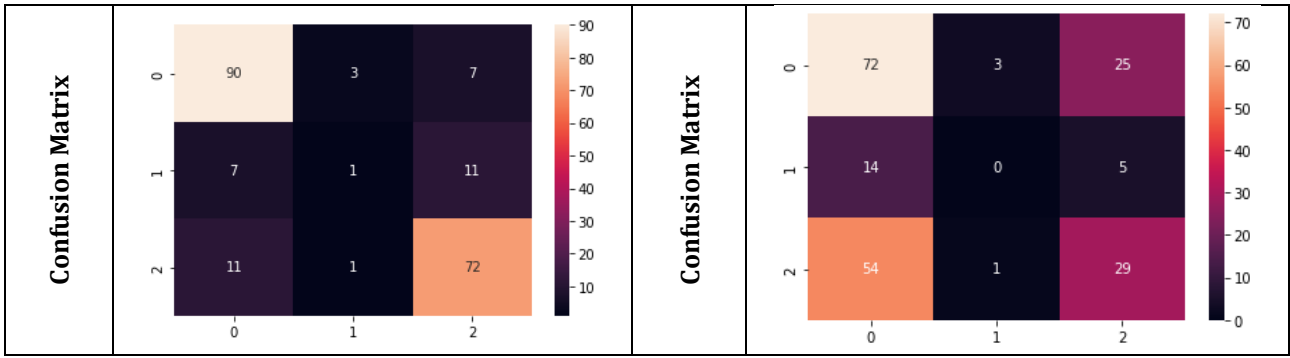


Table 8 shows the error metrics and accuracy values obtained in the training of Extra Trees and GradientBoosting algorithms. In addition, the graph showing the similarity between the real values and the values classified by the trained model, the error distribution graphs in classification and the confusion matrix tables are shown.

Table 8. Extra Trees and GradientBoosting Algorithm performance

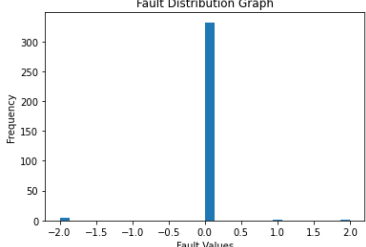
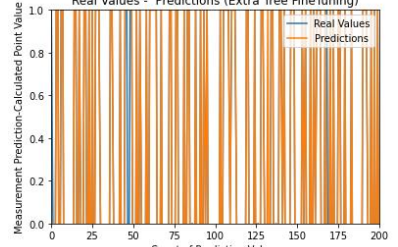
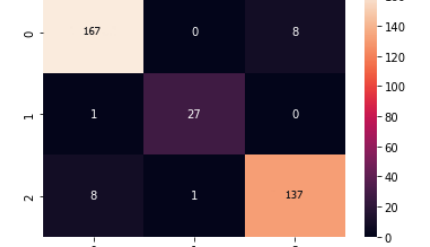
EXTRATREES ALGORITHM		GRADIENTBOOSTING ALGORITHM																																	
Metric	Values	Metric	Values																																
MAE:	0.197	MAE:	0.197																																
MSE:	0.364	MSE:	0.315																																
RMSE:	0.603	RMSE:	0.561																																
Acc.	0.886	Acc.	0.862																																
Fault Distribution		Fault Distribution																																	
Real Values & Predictions		Real Values & Predictions																																	
Confusion Matrix	<table border="1"> <tr><th></th><th>0</th><th>1</th><th>2</th></tr> <tr><th>0</th><td>90</td><td>2</td><td>8</td></tr> <tr><th>1</th><td>0</td><td>15</td><td>4</td></tr> <tr><th>2</th><td>9</td><td>0</td><td>75</td></tr> </table>		0	1	2	0	90	2	8	1	0	15	4	2	9	0	75	Confusion Matrix	<table border="1"> <tr><th></th><th>0</th><th>1</th><th>2</th></tr> <tr><th>0</th><td>92</td><td>3</td><td>5</td></tr> <tr><th>1</th><td>2</td><td>6</td><td>11</td></tr> <tr><th>2</th><td>7</td><td>0</td><td>77</td></tr> </table>		0	1	2	0	92	3	5	1	2	6	11	2	7	0	77
	0	1	2																																
0	90	2	8																																
1	0	15	4																																
2	9	0	75																																
	0	1	2																																
0	92	3	5																																
1	2	6	11																																
2	7	0	77																																

In Table 9, the performance of the Extra Trees algorithm was increased by fine tuning after training. The hyperparameter trials of the fine-tuning process and the hyperparameter values with the best results are shown

in Table 9. The fine-tuning parameters shown in Table 9 were tested and the parameter values at which the training started are shown. The last column of Table 9 shows the best performance values of the final test.

Table 9. Extra Trees fine tuning process and values

EXTRATREES ALGORITHM FINE TUNING PROCESS AND RESULTS					
Fine Tuning Model's Result		Hyperparameter Fine Tuning Parameters	Hyperparameter Fine Tuning Values	Hyperparameter Fine Tuning Parameters	Hyperparameter Fine Tuning Best Values
Metrics	Values	max_depth	None, 10, 20, 30	max_depth	30
MAE:	0.038	min_samples_leaf	1, 2, 4	min_samples_leaf	1
MSE:	0.073	min_samples_split	2, 5, 10	min_samples_split	5
RMSE:	0.271	n_estimators	50, 100, 200	n_estimators	100
Accuracy:	0.979	-	-	-	-

5. Discussion and Conclusions

Detection and prevention of Phishing attacks is an area that has attracted the attention of researchers and has been extensively studied. Table 10 shows the comparison table of this study with similar studies. In the table, the researcher, the year of the study, the preferred machine learning method, the success value obtained, and the data set studied are clearly shown. It is clear from this comparison that experiments with different machine learning methods have been conducted in the current study and improvement studies have been carried out in this area. This detail makes this study stand out from the others. In addition, the success rate obtained is acceptable compared to other studies.

Table 10. Comparative comparison table with similar studies in the literature

Works	Year	Dataset	Algorithm	Metrics
Ying and Xuhua	2006	Random page pool	SVM	Accuracy: 88%
Abdelhamid et al.	2014	Website history	MCAC	Accuracy: 94%
Moghimi and Varjani	2016	Yahoo (PhishTank)	SVM	Accuracy: 99.1%
Yi et al.	2018	Website traffic flows	DBN	Accuracy: 90%
Sahingoz et al.	2019	Own dataset	RF	Accuracy: 97.9%
Yerima and Alzaylae	2020	Benchmarked dataset	CNN	F1Score: 97.6%
Rashid et al.	2020	Google dataset	SVM	Accuracy: 95.6%
Adeyemo et al.	2021	Phishing datasets in UCI	LMT	Accuracy: 97.1%
This Work	2024	Website Phishing Data Set	Extra Trees	Accuracy: 97.9%

In this study, we investigated the effectiveness of machine learning methods for detecting Phishing attack threats on web pages. After six distinct machine learning algorithms were examined, the Extra Trees method was found to have the greatest success rate. We made some important adjustments to this approach in order to improve its effectiveness even further. Our performance tweaks to the Extra Trees algorithm significantly improved its ability to recognize phishing attacks. These results highlight the importance of developing machine learning techniques to provide a more effective protection against the dynamic environment of Phishing attack assaults. In order to further boost success in this sector, we plan to focus on certain techniques in our upcoming work. These involve expanding our dataset and adopting a more multifaceted strategy by combining several machine learning algorithms. By using larger and more diverse datasets, we will increase the generalization ability of our model and conduct a comprehensive study to better understand the advantages of different approaches. We'll also turn at larger-scale applications to assess how well our model performs in practical situations. This will allow us to

evaluate how well our developed approaches work in actual applications. Consequently, our study has shown how machine learning algorithms can detect Phishing attack dangers on websites with ease and success. The fact that the Extra Trees approach has been successfully applied shows that it has the potential to be a more effective tool for detecting assaults of this nature. Our next research will expand on these tactics in order to progress this field and keep people' online environments safer.

Conflict of Interest

No conflict of interest was declared by the author.

References

- Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection based Associative Classification data mining. *Expert Systems with Applications*, 41(13), 5948–5959. <https://doi.org/10.1016/j.eswa.2014.03.019>
- Adeyemo, V. E., Balogun, A. O., Mojeed, H. A., Akande, N. O., & Adewole, K. S. (2021). Ensemble-Based Logistic Model Trees for Website Phishing Detection. *Communications in Computer and Information Science*, 1347, 627–641. https://doi.org/10.1007/978-981-33-6835-4_41/TABLES/6
- AlOmar, M. K., Hameed, M. M., & AlSaadi, M. A. (2020). Multi hours ahead prediction of surface ozone gas concentration: Robust artificial intelligence approach. *Atmospheric Pollution Research*, 11(9), 1572–1587. <https://doi.org/10.1016/j.apr.2020.06.024>
- Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. *Electronics* 2023, Vol. 12, Page 232, 12(1), 232. <https://doi.org/10.3390/ELECTRONICS12010232>
- Balogun, A. O., Akande, N. O., Usman-Hamza, F. E., Adeyemo, V. E., Mabayoje, M. A., & Ameen, A. O. (2021). Rotation Forest-Based Logistic Model Tree for Website Phishing Detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12957 LNCS, 154–169. https://doi.org/10.1007/978-3-030-87013-3_12/TABLES/10
- Balogun, A. O., Mojeed, H. A., Adewole, K. S., Akintola, A. G., Salihu, S. A., Bajeh, A. O., & Jimoh, R. G. (2021). Optimized Decision Forest for Website Phishing Detection. *Lecture Notes in Networks and Systems*, 231 LNNS, 568–582. https://doi.org/10.1007/978-3-030-90321-3_47/TABLES/7
- Barracough, P. A., Fehringer, G., & Woodward, J. (2021). Intelligent cyber-phishing detection for online. *Computers & Security*, 104, 102123. <https://doi.org/10.1016/j.cose.2020.102123>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324/METRICS>
- Dhanavanthini, P., & Chakkravarthy, S. S. (2023). Phish-armor: phishing detection using deep recurrent neural networks. *Soft Computing*, 1–13. <https://doi.org/10.1007/S00500-023-07962-Y/TABLES/2>
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42. <https://doi.org/10.1007/S10994-006-6226-1/METRICS>
- Hammid, A. T., Sulaiman, M. H. Bin, & Abdalla, A. N. (2018). Prediction of small hydropower plant power production in Himreen Lake dam (HLD) using artificial neural network. *Alexandria Engineering Journal*, 57(1), 211–221. <https://doi.org/10.1016/j.aej.2016.12.011>
- Jain, A. K., & Gupta, B. B. (2019). A machine learning based approach for phishing detection using hyperlinks information. *Journal of Ambient Intelligence and Humanized Computing*, 10(5), 2015–2028. <https://doi.org/10.1007/S12652-018-0798-Z/TABLES/6>
- Mishra, G., Sehgal, D., & Valadi, J. K. (2017). Quantitative Structure Activity Relationship study of the Anti-Hepatitis Peptides employing Random Forests and Extra-trees regressors. *Bioinformatics*, 13(3), 60. <https://doi.org/10.6026/97320630013060>
- Mithra Raj, M., & Arul Jothi, J. A. (2022). Website Phishing Detection Using Machine Learning Classification Algorithms. *Communications in Computer and Information Science*, 1643 CCIS, 219–233. https://doi.org/10.1007/978-3-031-19647-8_16/TABLES/8
- Moghimi, M., & Varjani, A. Y. (2016). New rule-based phishing detection method. *Expert Systems with Applications*, 53, 231–242. <https://doi.org/10.1016/j.eswa.2016.01.028>
- Rashid, J., Mahmood, T., Nisar, M. W., & Nazir, T. (2020). Phishing Detection Using Machine Learning Technique. *Proceedings - 2020 1st International Conference of Smart Systems and Emerging Technologies, SMART-TECH 2020*, 43–46. <https://doi.org/10.1109/SMART-TECH49988.2020.00026>
- Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 117, 345–357. <https://doi.org/10.1016/j.eswa.2018.09.029>
- Website Phishing Dataset. (n.d.). Retrieved March 19, 2024, from <https://www.kaggle.com/datasets/ahmednour/website-phishing-data-set/data>
- Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1), 79–82. <https://doi.org/10.3354/CRO30079>
- Wu, C. Y., Kuo, C. C., & Yang, C. S. (2019). A Phishing Detection System based on Machine Learning. *Proceedings - 2019 International Conference on Intelligent Computing and Its Emerging Applications, ICEA 2019*, 28–32. <https://doi.org/10.1109/ICEA.2019.8858325>
- Yerima, S. Y., & Alzaylaee, M. K. (2020). High Accuracy Phishing Detection Based on Convolutional Neural Networks. *ICCAIS 2020 - 3rd International Conference on Computer Applications and Information Security*. <https://doi.org/10.1109/ICCAIS48893.2020.9096869>

- Yi, P., Guan, Y., Zou, F., Yao, Y., Wang, W., & Zhu, T. (2018). Web phishing detection using a deep learning framework. *Wireless Communications and Mobile Computing, 2018*. <https://doi.org/10.1155/2018/4678746>
- Ying, P., & Xuhua, D. (2006). Anomaly based web phishing page detection. *Proceedings - Annual Computer Security Applications Conference, ACSAC*, 381–390. <https://doi.org/10.1109/ACSAC.2006.13>