Received: 15 September 2023 | Revised: 7 November 2023 | Accepted: 4 December 2023 | Published online: 4 December 2023

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2024, Vol. 2(2) 87–99
DOI: 10.47852/bonviewJDSIS32021736

BON VIEW PUBLISHING

A Comparative Analysis of Feature Eliminator Methods to Improve Machine Learning Phishing Detection

Jibrilla Tanimu[1], Stavros Shiaeles[2,*] and Mo Adda[1]

[1]*Department of Computing, University of Portsmouth, UK*

[2]*Centre for Cybercrime and Economic Crime, University of Portsmouth, UK*

Abstract: This machine learning (ML)-based phishing detection employs statistical models and algorithms to assess and recognize phishing attacks. These algorithms can learn patterns and features that distinguish between phishing and nonphishing attacks once they are trained on vast amounts of data from both types of cases. Phishing detection systems can quickly evaluate considerable data, identify possible phishing attempts, and warn users of potential dangers. ML-based phishing detection systems have the potential to continuously improve their accuracy over time through ongoing feature refinement, iterative model evaluation, and algorithm optimization. In contrast to conventional techniques, these systems offer a more effective and efficient approach to identifying and mitigating phishing attacks. This research critically analyzes existing literature on phishing detection, aiming to identify all proposed features and determine the critical ones necessary for accurate and fast phishing attack detection. By eliminating unnecessary overhead, this research enhances our understanding of feature eliminator methods and their role in improving ML-based phishing detection. The findings would contribute to the development of more robust cybersecurity measures to combat phishing attacks, as well as advance the field's knowledge and application of ML in detecting and mitigating such threats. The study highlights the importance of feature selection and optimization in achieving accurate and efficient phishing detection, ultimately strengthening the overall security posture of organizations and individuals against phishing attacks.

Keywords: binary visualization, phishing detection, spam, feature elimination, machine learning

1. Introduction

With the expansion of the internet, users from all demographics, including different age groups, genders, cultures, and businesses, have become more vulnerable to cyberattacks. This increase in internet usage has also led to a rise in the number of attackers and hackers who exploit its resources. These individuals employ various tactics to lure internet consumers into performing specific activities that enable attackers to obtain sensitive information and money.

Phishing is a cyberattack (Jamil et al., 2018) that employs deceptive tactics to deceive people into disclosing their personal information, such as passwords and credit card numbers. Attackers often impersonate reputable businesses and use bogus emails, websites, and messages to lure victims into revealing their data. These attacks can propagate malware, financial fraud, and theft of personal information. It is essential to be aware of these attacks and take steps to guard against them because of their potentially significant consequences. According to one study by Verma and Rai (2015), phishing is a common technique used to gather sensitive information on the internet, targeting individual users and businesses. As financial activities become increasingly digitized, differentiating between legitimate and nonlegitimate activities is crucial. Tanimu and Shiaeles (2022) defined phishing as social engineering attacks that leverage psychological manipulation of people and deceive them into disclosing confidential information. These factors explain the need for efficient and effective phishing detection. Thus, automating phishing detection using machine learning (ML) to detect phishing in real time is urgently needed. Moreover, using all available features can lead to overfitting and poor generalization, making feature elimination a critical step in ML.

Effective feature elimination (Almseidin et al., 2019; Li et al., 2017; Toolan & Carthy, 2010) can reduce the computational complexity of ML algorithms, leading to faster and more accurate predictions, which is particularly vital in phishing detection, where real-time detection is crucial for preventing the disclosure of sensitive information. Several feature elimination methods exist, including recursive feature elimination (RFE), univariate feature selection (UFS), and correlation-based feature selection. These methods demonstrate effectiveness in reducing the number of features without compromising the accuracy of the ML algorithm.

Also, FE aids in reducing the dimensionality of the input space by selecting a subset of features that are most discriminative and informative for distinguishing between legitimate and phishing emails. By eliminating irrelevant and redundant features (Onyema et al., 2022), the computational complexity and memory requirements of the detection system can be significantly reduced, leading to improved efficiency and faster processing times.

Furthermore, feature selection enhances the performance and accuracy of ML models by focusing on the most discriminative

*Corresponding author: Stavros Shiaeles, Centre for Cybercrime and Economic Crime, University of Portsmouth, UK. Email: stavros.shiaeles@port.ac.uk

features. By selecting the most relevant features, the models can capture the distinctive characteristics and patterns associated with phishing attacks, improving detection accuracy and reducing false positives and false negatives.

We propose using ML and feature selection to mitigate the problems because ML algorithms require large datasets to be effective, leading to a high-dimensional feature space (Glaser et al., 2020). In this research, we compare ML algorithms and an extensive feature set to optimize ML algorithms by reducing the features to the most effective ones to accelerate the ML procedure and reduce the resources needed for the classification.

This paper is organized as follows. Section 2 provides the related work on phishing detection. Section 3 presents the proposed method. Section 4 discusses the results and final features that perform efficiently in the experiment. Section 5 details the limitations of the proposed method. Next, Section 6 describes the contributions of the work to enhance a better understanding of the approach. Finally, Section 7 concludes the paper and provides directions for future work.

2. Literature Review

The authors of Phish-IDetector (Verma & Rai, 2015) described the phishing attack as a type of cyberattack that involves conning users into disclosing their sensitive information, such as passwords or credit card details. In their paper, the authors proposed an automatic phishing detection system called Phish-IDetector, based on analyzing the email message ID field. The system uses an ML algorithm to classify emails as phishing or legitimate based on features extracted from the message ID field. The authors evaluated the system on a dataset of real-world phishing emails and demonstrated its effectiveness in detecting phishing emails with high accuracy.

Almseidin et al. (2019) proposed a phishing detection system using ML and feature selection methods. They collected a dataset of legitimate and phishing emails and used various ML algorithms and feature selection methods to classify emails. The results revealed that the proposed system achieved high accuracy, with an average accuracy of 98.1%, and the feature selection methods improved the classifier performance. The authors concluded that their system could effectively detect phishing emails and be used as a stand-alone solution or integrated with existing email security systems.

Another similar approach was provided in Ali (2017), employing a phishing website detection method based on supervised ML with wrapper feature selection. The system uses a dataset of 1,100 URLs, half of which are phishing and half of which are legitimate, and applies four ML classifiers: random forest (RF), decision tree (DT), K-nearest neighbors (KNN), and naïve Bayes (NB). The features are selected using a wrapper feature selection method, and the classifiers are evaluated using the accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve. The results indicate that the RF classifier with the wrapper feature selection method outperforms the other classifiers with an accuracy of 99.64%. The study concludes that the proposed method can be a practical approach to detecting phishing websites.

Examining a different method to solve phishing attacks, other researchers (Kim et al., 2020) proposed a feature selection method based on the binary bat algorithm for phishing detection. They applied the method to a dataset of legitimate and phishing emails and extracted features related to the email header, URL, and content. The binary bat algorithm was used to select the most relevant features for classification. The results revealed that the proposed feature selection method achieved high accuracy in detecting phishing emails, with an average accuracy of 96.72%. The selected features also outperformed the original set of features regarding classification accuracy. The authors concluded that the binary bat algorithm could effectively select relevant features for phishing detection and can be applied to other classification tasks.

In addition, researched phishing detection using a Bayesian classifier in data mining. Their research aimed to develop an effective ML model that could identify phishing emails with high accuracy. The authors used a dataset containing phishing and nonphishing emails, preprocessed by removing stop words and performing stemming. The dataset was split into training and testing sets for model evaluation. The NB classifier was applied to the dataset to classify emails as either phishing or nonphishing. The model achieved an accuracy of 98.3%, indicating that it was highly effective in detecting phishing emails. The authors also compared the performance of the NB classifier with other ML algorithms, such as DT and support vector machine (SVM), finding that NB performed better with high accuracy and precision in comparison with DT and SVM.

Another interesting tool is MP-Shield (Bottazzi et al., 2015), which focuses on a framework based on phishing attacks on mobile devices. The framework consists of a client-side application that runs on the mobile device and a server-side component that analyzes emails. The authors evaluated the framework on a dataset of phishing emails and demonstrated its effectiveness in detecting phishing emails on mobile devices.

Another study (Dada et al., 2019) reviewed the use of ML for email spam filtering, including detecting phishing emails. The authors identified various ML algorithms for phishing detection, including Bayesian networks, DT, and SVMs. The authors also identified several challenges associated with ML-based phishing detection, such as the need for large datasets, the problem of class imbalance and effectively differentiating between genuine and fraudulent emails that employ comparable content is crucial, necessitating the effective resolution of challenges to ensure the efficiency of ML in email spam filtering.

Khonji et al. (2013) conducted a literature survey of phishing detection techniques. The authors reviewed state-of-the-art phishing detection and identified various approaches, including rule-based, heuristic-based, and ML-based techniques. The authors noted that ML-based techniques have become increasingly popular recently due to their high accuracy and ability to adapt to new phishing attacks. The authors also identified several challenges associated with ML-based phishing detection, such as the need for large datasets and the problem of overfitting.

Another research by Tayyab and Masood (2019) provides a comprehensive review of existing research on using ML for phishing detection, focusing on using URLs and hyperlink information. The authors identified the critical features in phishing detection algorithms, such as lexical and syntactic features, and more advanced features, such as behavioral and semantic analysis. They also discussed the ML algorithms used in phishing detection, including DTs, SVMs, and artificial neural networks. In addition, Catal et al. (2022) focus on the applications of deep learning in phishing detection. Through a systematic literature review, the study highlights the potential of deep learning techniques in identifying and mitigating phishing attacks, underscoring the importance of advanced ML approaches in enhancing cybersecurity.

Moreover, Gualberto et al. (2020) proposed a method for enhancing the prediction rates of phishing detection using feature engineering and topic modeling techniques. The study used a dataset of 3500 phishing and legitimate URLs and applied a feature engineering process to extract relevant information, such as the

presence of specific characters or the URL length. They also used topic modeling to identify underlying URL themes that could indicate phishing. The results demonstrated that the proposed method achieved an accuracy of 98.55%, outperforming other traditional ML algorithms.

Furthermore, another study (Toolan & Carthy, 2009) combined the results of multiple ML classifiers to improve the accuracy of phishing detection. The approach involved extracting features from the HTML source code of a website and using those features as input to the classifiers. The authors evaluated their approach on a dataset of legitimate websites and known phishing websites, achieving an accuracy of 91.3%. Another similar research was conducted by Kumar Jain and Gupta (2018). Their approach involved extracting features from the website URL, content, and JavaScript code and using these as input to an ML classifier. The authors evaluated their approach on a dataset of legitimate websites and known phishing websites, achieving an accuracy of 98.2%. They also compared their approach to existing approaches and found that it outperformed them regarding the accuracy and false-positive and false-negative rates.

Similarly, a study (Thirumallai et al., 2020) implemented an ML-based phishing detection system combining three feature sets to achieve efficient classification and secure storage distribution for cloud or Internet of Things (IoT) applications. The system employs four ML models (RF, SVM, KNN, and multilayer perceptron) to classify phishing websites from legitimate ones. The three feature sets in the system are URL-based features, website content features, and website traffic features. The proposed system was assessed on two real-world datasets, and the results revealed that the RF algorithm outperforms other models with an accuracy of 99.8 and 99.6% on the two datasets, respectively.

In other research, features were used to mitigate phishing attacks (Nguyen et al., 2014) and adopted a combination of the three feature sets to achieve efficient classification and secure storage distribution for cloud or IoT applications. The three feature sets in the system are URL-based features, website content features, and website traffic features. The system employs four ML models (RF, SVM, KNN, and multilayer perceptron) to classify phishing websites from legitimate ones. The proposed system was assessed on two real-world datasets, and the results show that the RF algorithm outperforms other models with an accuracy of 99.8 and 99.6% on the two datasets, respectively.

Finally, Peng et al. (2018) proposed a phishing detection system that employs natural language processing (NLP) and ML techniques. The authors collected a dataset of phishing emails and analyzed them using NLP tools to extract essential features, such as sentence structure, sentiment, and keywords. They then used several ML algorithms, including logistic regression and RF, to classify emails as legitimate or phishing. The results indicated that the proposed system achieved a high accuracy of 97.8% in detecting phishing emails. The authors concluded that integrating NLP and ML can effectively detect phishing attacks and provide users with an additional layer of security.

In addition, ML-based phishing detection has been widely researched and applied in recent years, as presented in Table 1. Various ML techniques have been explored, including deep learning neural networks, DTs, and SVMs, with high accuracy in detecting phishing attacks. Compared to established techniques, such as rule- and signature-based systems, ML-based solutions are more reliable, scalable, and flexible in responding to changing phishing threats and managing extensive volumes of data.

However, significant limitations exist in ML-based phishing detection (Itoo et al., 2021). Success heavily depends on the size and quality of the training data, and unbalanced datasets can produce biased results. Moreover, the constantly changing nature of phishing attacks can make it difficult for ML-based systems to keep up with the latest threats.

Despite the limitations, the literature indicates that ML-based phishing detection is a promising approach with several advantages over conventional ones (Stojanović et al., 2021). Additional research is required to implement feature selection and identify significant features and methods for phishing detection to address these challenges and provide more efficient and effective solutions to mitigate the problems. Thus, this research is urgently needed to address the challenges mentioned in the literature, and using feature selection displays better efficacy in mitigating the current phishing threat.

Figure 1 presents the proposed model, which aims to provide phishing detection based on feature elimination using an ML approach to feature selection to identify the critical features or attributes for detecting phishing attacks. The goal of feature elimination is to reduce the number of features or attributes used in the model to improve its accuracy and efficiency, which can be done by removing features with a low correlation with the target variable or using statistical tests to determine which features most influence the performance. The selected features are used with other neural networks to train the phishing detection model. We remove less significant features from the dataset to improve visualization. Table 4 displays some results that have been achieved.

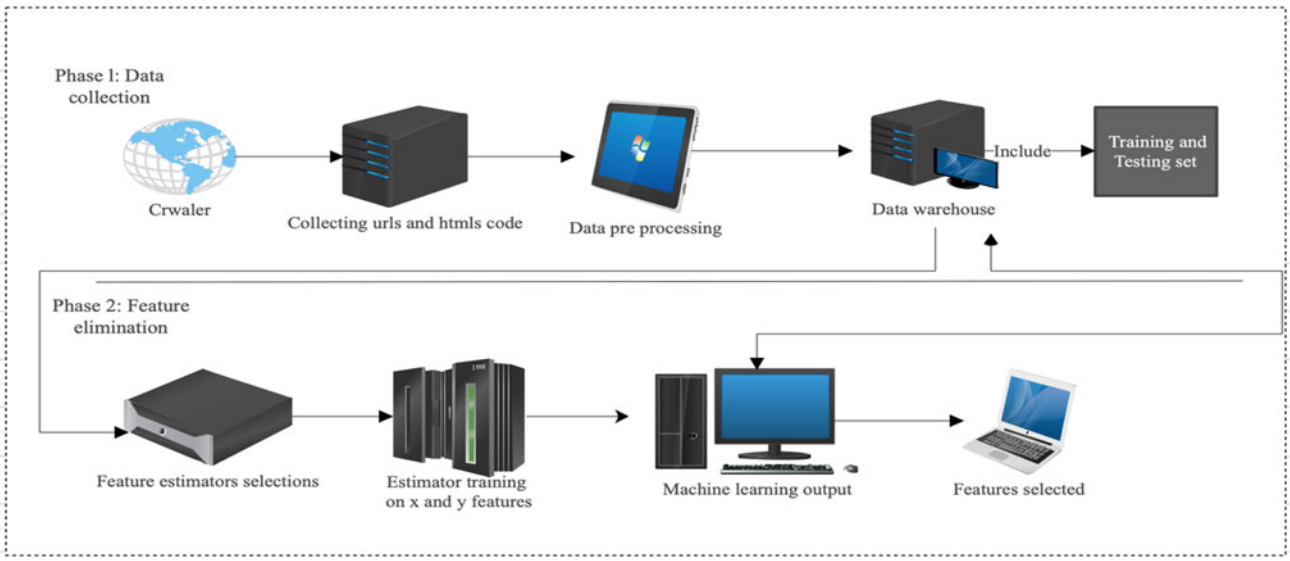
The next step involves analyzing the dataset, ensuring it is free from any inconsistencies or errors, and preparing it for use in training and testing the proposed model. Once the dataset is preprocessed correctly, we can proceed to the implementation stage using ML and neural network algorithms (Martin et al., 2011), such as DT, SVM, NB, 2D and 3D neural networks, and TensorFlow and recently proposed algorithms (e.g., artificial general intelligence). These innovative techniques are designed for greater efficiency and efficacy in achieving the objectives.

The first stage of the proposed model is to collect extensive data from PhishTank (https://phishtank.org/) using a crawler, which successfully crawled over 50,000 phishing websites and temporally stored them in the data warehouse. Second, the next stage is the sorting phase, including removing duplicate crawler data to avoid data redundancy and splitting the data into phishing and nonphishing. Third, the next stage aggregates the crawled data into the database for data manipulation and easy retrieval. Fourth, in the feature estimator selection, we employed various estimators that include RF, NB, DT, KNN, Adaboost, XGBoost, LR, SVM, and KMC to choose the most effective estimator on the x and y features; also at this stage, we tested some of the feature elimination methods that include chi-squared test, stepwise regression, correlation-based, RFE, personal correlation coefficient, decision-making, forward feature selection, and UFS. However, after thorough testing, we decided to drop the chi-squared test, stepwise regression, personal correlation, forward feature selection, and decision-making. These methods exhibited poor performance on our dataset and were also affected by multicollinearity, where the selected features showed a high correlation with each other. Fifth, the next stage utilizes both the estimator, feature eliminator method (chosen) and the phishing and nonphishing dataset in the data warehouse for the model to utilize and perform the final feature selection. Feature selection is conducted to remove less significant features from the proposed model and retain the most significant ones. Effective feature selection techniques utilized, including RFE, correlation-based feature selection, and UFS methods, were employed to expedite

Table 1
Comparative analysis of the literature

Ref	Accuracy	Limitation	Contribution	Method	Result
Almseidin et al. (2019)	Not specified	Limited description of the dataset	Proposed a model using machine learning and feature selection methods for phishing detection	Machine learning and feature selection	Achieved high accuracy in detecting phishing emails
Nakamura et al. (2013)	Not specified	Limited application to feature selection	Proposed a binary bat algorithm for feature selection	Binary bat algorithm	Outperformed other feature selection algorithms in experiments
Abunadi et al. (2013)	Not specified	Limited discussion of results	Proposed a feature extraction process for phishing detection	Feature extraction	Achieved high accuracy in detecting phishing emails
Toolan and Carthy (2010)	Not specified	Limited discussion of results	Proposed a feature selection method for spam and phishing detection	Feature selection	Achieved high accuracy in detecting phishing emails
Verma and Rai (2015)	Not provided	Limited discussion of results	Proposed the Phish-IDetector system based on message IDs for automatic phishing detection	Machine learning	A detection rate of 97.5% on a dataset of 900 phishing and legitimate emails
Lakshmi et al. (2021)	96.71%	Limited by dataset size	Proposed a system for phishing detection in web pages using supervised deep learning classification and the ADAM optimization technique	Machine learning	A detection rate of 96.71% for phishing web pages on a dataset of 10,000 web pages
Zaini et al. (2020)	97.1%	Limited dataset and variety	Proposed a system for phishing detection using machine learning classifiers	Machine learning	A detection rate of 97.1% on a dataset of 2,000 phishing and legitimate emails
Ali (2017)	99.1%	Limited by dataset size	Proposed a system for phishing website detection based on supervised machine learning with wrapper feature selection	Machine learning	A detection rate of 99.1% on a dataset of 1,000 phishing and legitimate websites
Hanus et al. (2022)	Not specified	Dataset bias, small sample size	Proposed a system in investigating the effectiveness of phishing training programs	Survey	Programs can significantly reduce the likelihood of employees falling for phishing emails
Alharbi et al. (2022)	Not specified	Small sample size	Proposed a system in investigating the awareness of phishing attacks among social media users	Survey	Awareness of phishing attacks and recommended the implementation of educational programs
Ayaburi and Andoh-Baidoo (2023)	Not specified	Limited generalizability	Proposed in investigating the role of reformulated locus of control in phishing susceptibility	Survey	Found that reformulated locus of control significantly influences phishing susceptibility
Siwakoti et al. (2023)	Not specified	Lack of specific implementation details	Provides an overview of IoT security vulnerabilities and attacks	Review	Summarizes IoT security vulnerabilities, attacks, and countermeasures
Shah et al. (2022)	96.5%	Assumes equal weight for all criteria	Proposes a fuzzy multi-criteria decision-making model for phishing website detection	Machine learning	ML can effectively detect phishing websites with a high level of accuracy
Schiller et al. (2023)	Not specified	Limited to e-mail phishing attacks only	Proposed an effective support system in preventing e-mail phishing attacks	Empirical study	Found that support systems significantly reduce the success rate of e-mail phishing attacks
Ansari et al. (2022)	Not specified	The study is limited to AI-based cybersecurity awareness training	Proposed a new approach to prevent phishing attacks using AI-based cybersecurity awareness training	Machine learning	The method reduces the success rate of e-mail phishing attacks

Figure 1
Proposed method



the selection process. These techniques aid in streamlining the model by focusing on the most relevant features and discarding the less informative ones. By leveraging these feature selection methods, the research ensures a more refined and optimized ML model for phishing detection. The utilization of RFE, correlation-based feature selection, and UFS contributes to the efficiency and effectiveness of the feature elimination process, facilitating improved accuracy and performance in detecting phishing attacks.

We aimed to collect all available features from the literature, assessing them against the datasets and all available ML algorithms to understand the critical features for phishing detection. The less significant features are dropped from the model to reduce the feedback provided to the ML and increase its performance and accuracy.

3. Implementation

The available dataset was divided into a training set comprising 54% of the data and a testing set comprising 46% of the data. Trade-off between training and testing: The 54%–46% ratio strikes a balance between having enough training data and sufficient testing data (Yao et al., 2022). This ratio is a commonly used and accepted practice in ML, providing a reasonable distribution that minimizes the risk of overfitting and allows for robust model evaluation. The training set was used to select and improve the model, including hyperparameter tuning, weight optimization, threshold setting, and feature selection. Hyperparameter tuning involves selecting the optimal values of the parameters that control the learning algorithm behavior (Ding et al., 2021). In contrast, weight optimization involves adjusting the internal model parameters to improve accuracy. Furthermore, the threshold setting involves determining the cutoff point for the output to balance precision and recall.

Moreover, feature selection identifies the most relevant features to include in the model, reducing overfitting and improving generalization (Samad & Gani, 2020). The following techniques are crucial for improving the performance of an ML model: carefully selecting the optimal hyperparameters, optimizing the internal weights, setting the appropriate threshold, and selecting the most relevant features.

These results were verified through 10-fold cross-validation. The testing set was used to evaluate the performance of models appropriately tuned and optimized during the earlier stages. Table 2 presents the results. The RF performed best among the tested models, whereas the SVM achieved the least significant performance.

3.1. Technology

Python (Guo et al., 2023) was explicitly selected due to its flexibility and compatibility in using packages, such as Pandas, MySQL connector for database connection, and other libraries. In this work, ML libraries, such as Pandas, NumPy, and Scikit-learn, were used. In addition, the pickle binary format was used to preserve and reuse already created models. Python programming was successfully executed on an Ubuntu 20.04 LTS with a two-core 8 GB RAM virtual machine.

3.2. Server implementation

The server required for the implementation of the model includes a high-performance computing environment with sufficient memory and storage capacity, in addition to ML libraries and packages for data processing, feature selection, and model training. Furthermore, the server also has the ability to handle large datasets, perform parallel processing, and provide robust security measures to protect sensitive data. Additionally, scalability is necessary to accommodate future growth and increased computational demands; this would utilize the feature listed in Table 2 for the feature elimination process.

We plotted the graphical representation of the current features achieved after the feature elimination phase to negate the inefficient features from the dataset, as illustrated in Figure 3. The list of features from the literature includes using the internet protocol (IP) addresses, long URLs, URLs with the "@" symbol, redirecting using the double solidus "//", URLs of anchors, links in <Mete>, adding prefixes or suffixes separated by the hyphen "-", subdomains with multiple

Table 2
Features adopted from the literature review

S/N	Features
1	Links in tags
2	Abnormal URLs
3	Age of domain
4	Port
5	Right click disabled
6	Pop up windows
7	Embedded brand name
8	Subdomain level
9	Redirect page
10	IP address
11	Pct Ext resource URLs
12	Insecure forms
13	Double slash redirecting
14	Frequent domain name mismatch
15	URL length RT
16	Ext meta script link RT
17	Using pop-up windows
18	Double slash in path
19	Missing title
20	Page rank
21	SSL final state
22	Fake link in status bar
23	Random string
24	Host name length
25	Query length
26	No HTTPS
27	Links pointing to a page
28	Num hash
29	IFrame or frame
30	Insecure forms

subdomains, <Script> and <Links>, tags, the server form handle, website forwarding, HTTPS socket layers, domain registration lengths, submitting information to email, abnormal URLs, website forwarding, status bar customization, the domain age, DNS recording, disabled right click, using pop-up windows, website traffic, page rank, Google index, statistical report-based feature, IFrame redirection, using a nonstandard port number, Numhash, and query length. A total of 47 features were initially identified, and the features with the most negligible significance were subsequently removed, resulting in the features listed in Table 2.

We executed the procedure delineated in the preceding section to conduct the intended study. We successfully amassed over 50,000 websites and URLs of both malicious and legitimate nature, each containing the subentities depicted in Figure 2. The dataset was subsequently partitioned into two distinct categories (legitimate and nonlegitimate) to train and evaluate the proposed model (Raschka et al., 2020).

Figure 2 illustrates the dataset obtained by crawling the PhishTank repository and storing it in a MySQL database. The first column represents the unique ID assigned to each crawled data, incremented by one for each entry. The second column contains the URLs of successfully crawled phishing websites, while the third column displays the HTML code of some of these phishing websites. All these entities collectively form the dataset used for training and testing the proposed model.

Figure 2
Sample of the dataset

ID	URL	HTML_Code
1	https://pancake-swapp.com/	<html lang="en-US"> <head itemscope itemtype="htt...
2	https://superapesclub.us/	<html class="h-100 qpfalxes idc0_335" lang="en"><h...
3	https://www.au.com/	<html prefix="og: http://ogp.me/ns#" class="tmpEle...
4	https://www.metamaskextension.one/?page=login	<html lang="en"> <head> <!-- Required meta t...
5	https://xb666815.com/	<html> <head> <meta charset="utf-8"> <m...
6	https://www.powr.io/form-build	<html lang='en'> <head> <link as='font' crosso...

Figure 3
Correlation matrix

3.3. Choosing the right classifier

Selecting the appropriate classifiers is a crucial component of creating a successful ML-based phishing detection system (Basit et al., 2020). Various classifiers are suitable for different data and their effects; thus, selecting a classifier can significantly change system performance. Observing the following conditions is crucial in selecting a classifier for the proposed model, providing guidelines for choosing the correct classifiers to work within the research.

- Resources needed for computation: Running some classifiers are expensive because they require considerable memory and computing power. When selecting a classifier, computing resources must be considered.
- Quantity and quality of training data: The classifier performance can be significantly influenced by the quantity and quality of training data. A sizeable and representative dataset is required to train the classifier properly.
- The complexity of problems. Some classifiers perform better with straightforward problems, whereas others perform better with more complicated problems.

Table 3
Classification results

Algorithms	Accuracy	Precision	Recall	F1 score
Random forest	97.1%	96.2%	90.2%	95.7%
Naïve Bayes	88.7%	88.4%	81.3%	88.1%
Decision tree	94.8%	92.5%	90.2%	92.5%
KNN	64.5%	61.8%	61.4%	62.0%
Adaboost	91.7%	90.3%	89.7%	90.2%
XGBoost	94.7%	94.1%	92.5%	94.0%
Logic regression	92.7%	90.8%	89.5%	90.9%
Support vector machine	56.0%	54.8%	52.1%	55.3%
K-means cluster	69.9%	68.7%	61.3%	67.5%

- Performance measures: Different classifiers may perform better or worse depending on the metrics to assess each classifier's performance.

To justify and select suitable classification algorithms for this research, we ran and tested all algorithms in Table 3. We implemented the list algorithms to understand the efficacy of each of them regarding the features (Table 3). This algorithm uniquely identifies the correlation and tendency regarding the importance of each classifier with the feature, making it more reliable and transparent for the model to use for the next stage, the feature selection stage, as indicated in Table 4.

To determine which performs best on this dataset, we also considered the computation efficiency, scalability and interpretability, which led to selecting DT, extreme gradient boosting (XGBoost), and RF among the rest of the classifiers. In addition, RF achieved an accuracy of 97.1%, with XGBoost achieving 94.7% accuracy and the DT classifier reaching 94.8% accuracy among the listed classifiers in the table, leading to the unbiased selection of the most effective and efficient classifiers in the dataset.

Accuracy (A): For all websites, the accuracy metric calculates the proportion of phishing and legitimate websites accurately detected:

$$\text{Accuracy} = \frac{N_{L\to L} + N_{P\to P}}{N_L + N_P} X100.$$

(Sahingoz et al., 2019)

Precision (P): This metric gauges the proportion of phishing websites correctly identified compared to those that are phishing:

$$\text{Precision} = \frac{N_{P\to P}}{N_{P\to P} + N_L \to P} X100.$$

(Carvalho et al., 2019)

Recall (R): It gauges the proportion of websites correctly labeled phishing versus legitimate:

$$\text{Recall} = \frac{N_{P\to P}}{N_{P\to P} + N_P \to L} X100.$$

(Tyagi et al., 2018)

F1 Score (F1): This metric is the harmonic mean of the accuracy and recall values:

$$F1 - Score = \frac{2\ X\ P\ X\ R}{P + R}.$$

(Abu-Nimeh et al., 2007)

3.3.1. Justification for not using other networks

Neural networks, such as backpropagation neural networks, radial basis function networks, and convolutional neural networks, are widely used for feature selection in ML tasks. While neural networks have many advantages, they also have limitations regarding feature selection. However, these networks have disadvantages that limit their usefulness in this research. One major disadvantage (Roohi & Phil, 2013) is their lack of interpretability, making it challenging to understand why certain features are included or excluded from the model. Additionally, these networks can be computationally expensive, require extensive training data (Dalgaty et al., 2021), and may suffer from overfitting. To avoid this compatibility challenge, we adopted the algorithms in Table 3. These methods improve the interpretability and performance of the proposed model while reducing overfitting.

3.4. Feature elimination

This stage introduced the most significant part of the research by examining well-known feature selection methods (Misra & Yadav, 2020), including RFE, linear feature elimination, and UFS. These methods have achieved optimal results based on the literature review, making it more important to investigate all approaches in the proposed model for a comprehensive comparison.

3.4.1. Recursive feature elimination

The RFE algorithm begins with all features in the training dataset and progressively removes them until the desired number of features remains. During each iteration, less critical features are eliminated based on their evaluated importance to accelerate the process (Bahl et al., 2019; Darst et al., 2018; Upadhyay et al., 2021). The relative importance of each feature can significantly change when evaluated over a new selection of features during the stepwise elimination process, especially for highly correlated features. The process is recursive, and a final ranking of features is generated using their elimination order (inverse). The feature selection process involves selecting the top n characteristics from this ranking.

Table 4
Recursive feature elimination results

Classifier	Decision trees		XGBoost		Random f.	
Features	Rank	Selection	Rank	Selection	Rank	Selection
Index	1	True	7	False	1	True
UsingIP	1	True	1	True	16	False
LongURL	4	False	22	False	6	True
Redirecting	1	True	1	True	1	True
PrefixSubffix	15	False	10	False	24	False
SubDomain	16	False	16	False	1	True
Symbol@	23	False	23	False	12	False
HTTPSDomain	1	True	1	True	21	False
DomainLengh	21	False	14	False	8	False
Favicon	13	False	17	False	23	False
RequestURL	14	False	14	False	11	False
AnchorURL	10	False	10	False	2	False
LinksInScriptTags	11	False	11	False	3	False
ServerFormHandler	8	False	14	False	10	False
AbnormalURL	2	False	4	True	20	False
WebsiteForwarding	25	False	25	False	4	False
StatusBarCust	5	False	5	False	25	False
DisableRightClick	12	False	17	False	27	False
UsingPopupWindow	26	False	19	False	18	False
IframeRedirection	24	False	15	False	22	False
AgeofDomain	17	False	20	False	7	False
DNSRecording	1	True	16	False	9	False
WebsiteTraffic	7	False	7	False	1	True
PageRank	18	False	18	False	14	False
GoogleIndex	22	False	1	True	12	False
LinksPointingToPage	6	False	6	False	1	True

3.4.2. RFE process

The following outline describes the RFE process used in this study:

Input:

X: as a dataset with n samples and m features

y: as a vector of n labels

Estimator: the ML algorithm that assigns the feature weights, as depicted in Figure 1.

Output:

Provide the subset of the features selected through the RFE process.

Steps:

- We initialized a set of features F with all m features.
- The model trains the estimator on the dataset (X, y) to obtain feature weights for each feature in F.
- Features are ranked in F based on their importance scores in ascending order (Table 3).
- Based on the step size, the model removes the lowest-ranked features from F until the desired number of features is reached.
- The steps are repeated two to four times for each subset of features until the desired number of features is reached.
- Finally, the subset of features with the highest cross-validation score is selected as the final set of features.

We achieved a significant result from the model used in RFE by ranking the most significant features as 1 and selecting features as true (Table 4). The redirecting feature was selected by all classifiers and ranked first by all the classifiers, which is the most comprehensive approach achieved in the feature elimination method. This method also demonstrates the ability of the selection process to select one feature using more than one classifier, showcasing the model strength in Table 4.

3.4.3. Correlation-based feature elimination

Correlation-based feature elimination is another crucial method adopted for the research that uses a filter strategy and is inconsequential to the preselected classification model. It analyzes feature subsets based on intrinsic data features, as the name suggests (correlations). The main objective is determining a feature subset with low feature-to-feature correlation, preventing redundancy, and high feature class correlation to preserve or boost predictive power (Rao & Pais, 2019).

$$Merit_x = \frac{\overline{kr}\mathrm{cf}}{\sqrt{k + k(k-1)\overline{rii}}}$$

where $Merit_x$ denotes the correlation between the summed component and outer variable, $\overline{rii}$ indicates the average inter-correlation between components, k represents the number of components, and $\overline{kr}\mathrm{cf}$ is the average correlation between the components and the outside variable.

This process loops back to the following best-unexpanded subset whenever a feature expansion does not improve. This algorithm explores the entire feature subset space without restrictions. Therefore, the extent of backtracking must be controlled. The feature subset produces the highest merit-up unit. The program returns the feature subset that produced the highest quality up to the last point. Unstacking the absolute values is essential to determine the mean value of the dataset, as indicated in Table 5.

Table 5
Unstacking values and taking the main

	LongURL	HTTPS	Subdomains	Redirecting//	UsingIP
LongURL	NaN	0.049033	0.004249	−0.080788	−0.052159
HTTPS	NaN	NaN	0.267531	−0.036536	0.071255
SubDomains	NaN	NaN	NaN	−0.043401	−0.080921
Redirecting//	NaN	NaN	NaN	NaN	0.397220
UsingIP	NaN	NaN	NaN	NaN	NaN

Correlation-based feature selection considerably cuts the number of features from 31 to 6 (including the subdomain, HTTPS, prefix suffix, long URL, index, and redirecting), drastically reducing the training and evaluation time from 30 to 5 s. More importantly, this processing step increases accuracy from 63 to 79% with 10-fold cross-validation. Correlation-based feature selection also improves the visualization of the feature by tabulating the correlation matrix (Figure 3), easing the interpretation and translation of the features.

3.4.4. Univariant feature selection

The strength of the association between each feature and the response variable is assessed using the UFS method. These techniques are straightforward to use and comprehend, making them excellent for better-comprehending data (Bergholz et al., 2008) (but not necessarily for optimizing the feature set for better generalization). The chi-square is an option for UFS, as shown:

$$x^2 = \sum_{i=1}^{n} \left(\frac{O_i - E_i}{E_i} \right) 2.$$

If a feature is independent of the target, it is expected to be uninformative for classifying an observation. In the equation above, $\boldsymbol{O_i}$ denotes the observation in the class, and $\boldsymbol{E_i}$ is the expected observation in class i, if no relationship exists between the feature and target. To use x2 for the feature selection, we calculate xx2 between each feature and target by selecting the desired number of features with the x2 score. It is frequently the first calculation performed on the data because it is quick and straightforward, and Figure 4 presents the influence of UFS. The correlation and p-value for the correlation are computed using Scipy's Pearson's *r* approach to approximate the likelihood that an uncorrelated system would have a correlation value of this magnitude.

Figure 4
Univariate score

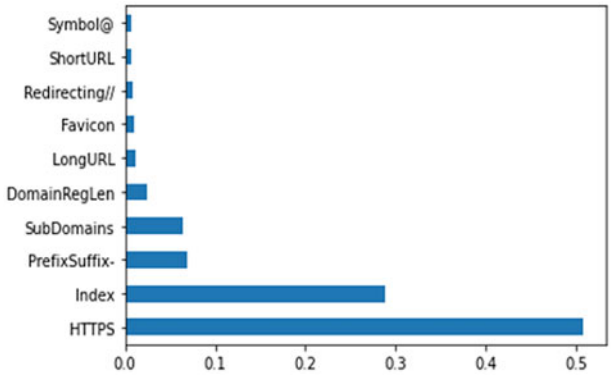


Table 6 demonstrates that, besides the univariate score, the model could distinguish features based on their selection or nonselection. The RF classifier attained the highest feature selection rate of 91%, whereas the DT classifier achieved the least significant rate of 77%. These results were obtained using the same classifiers that achieved significant results in Table 3, which were employed in the model optimization phase. Additionally, UFS and correlation-based feature selection methods were significantly enhanced when using the RFE approach, which outperformed the other two methods. However, both feature selection techniques played a critical role in the selection of final features (Moedjahedy et al., 2022), with ML classifiers employed as evaluation functions to assess the quality of all subsets. The final features were determined based on these evaluations, as presented in Section 4.

Table 6
Univariant feature selection results

Classifiers	Support vector machine	Random forest	Decision tree
Classification accuracy after univariate feature selection	0.868	0.914	0.778
Classification accuracy without selecting features	0.789	0.868	0.568

These findings highlight the superior performance of RF in both scenarios, indicating its suitability for the dataset and its ability to achieve high accuracy in both feature-selected and non-feature-selected settings.

4. Results and Discussion

In this research, we evaluated three feature selection methods, RFE, UFS, and correlation feature selection, to determine their effectiveness in selecting the most relevant features for phishing detection using ML algorithms. The experiments were performed on a dataset, and various classifiers were trained and tested using the selected features.

The results revealed that all three methods effectively reduced the feature space and improved classifier performance. The RFE method achieved the highest accuracy, with an average of 97.2%, followed by UFS, with an average of 91.4%, and correlation feature selection, with an average of 79.9%. The UFS and correlation methods are less computationally intensive and more suitable for larger datasets. Although they achieved slightly lower accuracy than RFE, they still demonstrated significant performance improvements compared to using all features.

The results suggest that feature selection is essential in developing effective phishing detection systems (Adebowale et al., 2019; Almseidin et al., 2019). It can improve classifier performance, reduce computational overhead, and detect real-time phishing attacks. The selection of the feature selection method depends on the dataset size and system-specific requirements.

The major cornerstone of this work is producing a model to determine the best features among a list. The model successfully identified the most effective features, ranking the redirecting feature as the most critical among all selected classifiers. The proposed model was effective with the provided data, iterating through the features and selecting the most efficient ones based on their importance. The top-ranked feature was considered the most significant feature with a selection of "true," whereas other selections were marked false and ranked lower. Table 4 indicates that less efficient features should be avoided in research. The list of the selected features is provided below.

- Using IP,
- Long URL,
- Subdomain,
- Redirecting,
- Google index,
- Links pointing to pages,
- HTTPS domain.

5. Limitations of the Feature Elimination

One of the significant limitations of the proposed model is that it can be computationally expensive, especially when dealing with extensive datasets. The RFE method adopted in this research is widely used in feature selection, but it can be slow and requires considerable computation.

Another limitation is that some techniques, such as UFS and correlation-based feature selection, may not always capture essential features. Moreover, UFS only considers the relationship between each feature and the target variable, whereas correlation-based feature selection only considers the relationship between each pair of features. Thus, essential features may be missed, or redundant features may be selected.

6. Optimizing the Model for Efficiency

Table 7
Time needed to detect phishing

Optimization/Training method	Time to detect phishing/spam (second)
Baseline model	3.2
Deep learning	1.2
Hyperparameter turning	1.8
Feature selection	2.5

Table 7 provides a comparison of the time required to detect phishing or spam websites, considering different optimization and training methods. Each row in the table represents a specific method used to improve the detection process, while the corresponding value in the "Time to Detect Phishing/Spam (seconds)" column indicates the average time taken by that method to identify and classify potentially malicious content.

Baseline Model: This refers to the initial or standard method used for detection without any specific optimization or advanced training techniques, taking an average of 3.2 s.

Feature Selection: In this case, a feature selection process was implemented to reduce the dimensionality of the data, resulting in a reduced detection time of 2.5 s.

Hyperparameter Tuning: By optimizing the hyperparameters of the detection model, the time needed to identify phishing or spam content was further reduced to 1.8 s.

Deep Learning: Utilizing deep learning techniques, such as neural networks, led to faster detection, with an average time of 1.2 s.

Table 7 highlights how different optimization and training methods impact the efficiency of phishing and spam detection, ultimately reducing the time required for accurate identification.

7. Research Contribution

The presented model offers improved efficiency and effectiveness in data analysis, particularly when dealing with input datasets containing a multitude of features. This model excels in generating precise and concentrated representations of the underlying data by eliminating irrelevant or duplicated features, as demonstrated by Yin et al. (2017). Additionally, feature selection aids in the identification of pivotal variables, facilitating more informed decision-making and a deeper comprehension of intricate phenomena. In summary, the utilization of ML algorithms for feature selection proves to be an invaluable asset in the realm of data analysis, holding the promise of advancing insights across diverse domains.

8. Conclusion and Future Work

In summary, the pivotal role of feature selection in ML-based phishing detection cannot be overstated. This research was driven by the goal of discerning critical features, gleaned from a comprehensive review of existing literature, to facilitate robust experiments and assemble pertinent elements for effective phishing detection. Our experiments encompassed an array of feature selection methodologies, culminating in the application of three highly effective techniques: RFE, UFS, and correlation feature selection.

The outcomes of this study, as delineated in Section 4, underscore the significance of feature selection. RFE and UFS emerged as standout performers, adept at reducing the feature space while upholding exceptional accuracy. In contrast, the correlation-based feature selection exhibited limited efficacy in culling irrelevant attributes. These insights are invaluable, constituting a substantial stride toward the development of an efficient anti-phishing system. Moreover, this research has substantially enriched our understanding of feature selection techniques.

As we gaze toward the future, it is evident that there is more work to be done. Leveraging the selected features from this study, such as IP, long URL, subdomain, redirections, Google index, links pointing to a page, and HTTPS, holds promise for creating a vast corpus of over 60,000 HTML codes for advanced phishing classification. This endeavor not only aims to enhance prediction accuracy but also promises to expedite the detection process, aligning with the relentless evolution of phishing tactics in the digital landscape.

In conclusion, this research acts as a stepping stone, illuminating the path toward more robust and efficient phishing detection systems. The journey of innovation and improvement in the realm of cybersecurity is ongoing, and this study stands as a testament to our commitment to staying ahead of emerging threats in the digital age.

Funding Support

This project was supported by the Petroleum Technology Development Funds (PTDF) Nigeria. The opinions expressed in this work are those of the author and do not necessarily represent the views of the organization concerning their findings.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in [PhishTank] at https://phishtank.org.

References

Abunadi, A., Akanbi, O., & Zainal, A. (2013). Feature extraction process: A phishing detection approach. In *2013 13th International Conference on Intelligent Systems Design and Applications,* 331–335. https://doi.org/10.1109/ISDA.2013.6920759

Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A comparison of machine learning techniques for phishing detection. In *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit,* 60–69.

Adebowale, M. A., Lwin, K. T., Sánchez, E., & Hossain, M. A. (2019). Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text. *Expert Systems with Applications*, *115*, 300–313. https://doi.org/10.1016/j.eswa.2018.07.067

Alharbi, A., Alotaibi, A., Alghofaili, L., Alsalamah, M., Alwasil, N., & Elkhediri, S. (2022). Security in social-media: Awareness of phishing attacks techniques and countermeasures. In *2nd International Conference on Computing and Information Technology*, 10–16. https://doi.org/10.1109/ICCIT52419.2022.9711640

Ali, W. (2017). Phishing website detection based on supervised machine learning with wrapper features selection. *International Journal of Advanced Computer Science and Applications*, *8*(9). https://doi.org/10.14569/ijacsa.2017.080910

Almseidin, M., Abu Zuraiq, A. A., Al-kasassbeh, M., & Alnidami, N. (2019). Phishing detection based on machine learning and feature selection methods. *International Journal of Interactive Mobile Technologies*, *13*(12), 171–183. https://doi.org/10.3991/ijim.v13i12.11411

Ansari, M. F., Sharma, P. K., & Dash, B. (2022). Prevention of phishing attacks using AI-based cybersecurity awareness training. *International Journal of Smart Sensor and Adhoc Network*, *3*(3), 6. https://doi.org/10.47893/ijssan.2022.1221

Ayaburi, E. W., & Andoh-Baidoo, F. K. (2023). How do technology use patterns influence phishing susceptibility? A two-wave study of the role of reformulated locus of control. *European Journal of Information Systems*, 1–21. https://doi.org/10.1080/0960085x.2023.2186275

Bahl, A., Hellack, B., Balas, M., Dinischiotu, A., Wiemann, M., Brinkmann, J., . . . , & Haase, A. (2019). Recursive feature elimination in random forest classification supports nanomaterial grouping. *NanoImpact*, *15*, 100179. https://doi.org/10.1016/j.impact.2019.100179

Basit, A., Zafar, M., Javed, A. R., & Jalil, Z. (2020). A novel ensemble machine learning method to detect phishing attack. In *IEEE 23rd International Multitopic Conference,* 1–5 https://doi.org/10.1109/INMIC50486.2020.9318210

Bergholz, A., Paaß, G., Reichartz, F., & Strobel, S. (2008). Improved phishing detection using model-based features. In *Conference on Email and Anti-Spam 2008*.

Bottazzi, G., Casalicchio, E., Cingolani, D., Marturana, F., & Piu, M. (2015). MP-Shield: A framework for phishing detection in mobile devices. In *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing,* 1977–1983. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.293

Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, *8*(8), 832. https://doi.org/10.3390/electronics8080832

Catal, C., Giray, G., Tekinerdogan, B., Kumar, S., & Shukla, S. (2022). Applications of deep learning for phishing detection: A systematic literature review. *Knowledge and Information Systems*, *64*(6), 1457–1500. https://doi.org/10.1007/s10115-022-01672-x

Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: Review, approaches and open research problems. *Heliyon*, *5*(6), e01802. https://doi.org/10.1016/j.heliyon.2019.e01802

Dalgaty, T., Miller, J. P., Vianello, E., & Casas, J. (2021). Bio-inspired architectures substantially reduce the memory requirements of neural network models. *Frontiers in Neuroscience*, *15*, 612359.

Darst, B. F., Malecki, K. C., & Engelman, C. D. (2018). Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC Genomic Data*, *19*(1). 65. https://doi.org/10.1186/s12863-018-0633-8

Ding, X., Liu, B., Jiang, Z., Wang, Q., & Xin, L. (2021). Spear phishing emails detection based on machine learning. In *IEEE 24th International Conference on Computer Supported Cooperative Work in Design*, 354–359. https://doi.org/10.1109/CSCWD49262.2021.9437758

Glaser, J. I., Benjamin, A. S., Chowdhury, R. H., Perich, M. G., Miller, L. E., & Kording, K. P. (2020). Machine learning for neural decoding. *eNeuro*, *7*(4), 1–16. https://doi.org/10.1523/ENEURO.0506-19.2020

Gualberto, E. S., de Sousa, R. T., Vieira, T. P. D. B., da Costa, J. P. C. L., & Duque, C. G. (2020). From feature engineering and topics models to enhanced prediction rates in phishing detection. *IEEE Access*, *8*, 76368–76385. https://doi.org/10.1109/ACCESS.2020.2989126

Guo, J., Ye, A., Wang, X., & Guan, Z. (2023). OpenSeesPyView: Python programming-based visualization and post-processing tool for OpenSeesPy. *SoftwareX*, *21*, 101278. https://doi.org/10.1016/j.softx.2022.101278

Hanus, B., Wu, Y. A., & Parrish, J. (2022). Phish me, phish me not. *Journal of Computer Information Systems*, *62*(3), 516–526. https://doi.org/10.1080/08874417.2020.1858730

Itoo, F., Meenakshi, & Singh, S. (2021). Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, *13*(4), 1503–1511. https://doi.org/10.1007/s41870-020-00430-y

Jamil, A., Asif, K., Ghulam, Z., Nazir, M. K., Mudassar Alam, S., & Ashraf, R. (2018). Mpmpa: A mitigation and prevention model

for social engineering based phishing attacks on Facebook. In *IEEE International Conference on Big Data*, 5040–5048. https://doi.org/10.1109/BigData.2018.8622505

Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: A literature survey. *IEEE Communications Surveys & Tutorials*, *15*(4), 2091–2121. https://doi.org/10.1109/SURV.2013.032213.00009

Kim, B., Lee, D. Y., & Kim, B. (2020). Deterrent effects of punishment and training on insider security threats: A field experiment on phishing attacks. *Behaviour & Information Technology*, *39*(11), 1156–1175. https://doi.org/10.1080/0144929X.2019.1653992

Kumar Jain, A., & Gupta, B. B. (2018). Towards detection of phishing websites on client-side using machine learning based approach. *Telecommunication Systems*, *68*, 687–700. https://doi.org/10.1007/s11235-017-0414-0

Lakshmi, L., Reddy, M. P., Santhaiah, C., & Reddy, U. J. (2021). Smart phishing detection in web pages using supervised deep learning classification and optimization technique ADAM. *Wireless Personal Communications*, *118*(4), 3549–3564. https://doi.org/10.1007/s11277-021-08196-7

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys*, *50*(6), 94. https://doi.org/10.1145/3136625

Martin, A., Anutthamaa, N. B., Sathyavathy, M., Saint Francois, M. M., & Venkatesan, P. (2011). A framework for predicting phishing websites using neural networks. *International Journal of Computer Science Issues*, *8*(2), 330–336.

Misra, P., & Yadav, A. S. (2020). Improving the classification accuracy using recursive feature elimination with cross-validation. *International Journal on Emerging Technologies*, *11*(3), 659–665.

Moedjahedy, J., Setyanto, A., Alarfaj, F. K., & Alreshoodi, M. (2022). CCrFS: Combine correlation features selection for detecting phishing websites using machine learning. *Future Internet*, *14*(8), 229. https://doi.org/10.3390/fi14080229

Nakamura, R. Y. M., Pereira, L. A. M., Rodrigues, D., Costa, K. A. P., Papa, J. P., & Yang, X. S. (2013). Binary bat algorithm for feature selection. In X. S. Yang, Z. Cui, R. Xiao, A. H. Gandomi & M. Karamanoglu (Eds.), *Swarm intelligence and bio-inspired computation: Theory and applications* (pp. 225–237. Elsevier. https://doi.org/10.1016/B978-0-12-405163-8.00009-0

Nguyen, L. A. T., To, B. L., Nguyen, H. K., & Nguyen, M. H. (2014). A novel approach for phishing detection using URL-based heuristic. In *International Conference on Computing, Management and Telecommunications*, 298–303. https://doi.org/10.1109/ComManTel.2014.6825621

Onyema, E. M., Kumar, M. A., Balasubaramanian, S., Bharany, S., Rehman, A. U., Eldin, E. T., & Shafiq, M. (2022). A security policy protocol for detection and prevention of internet control message protocol attacks in software defined networks. *Sustainability*, *14*(19), 11950. https://doi.org/10.3390/su141911950

Peng, T., Harris, I., & Sawa, Y. (2018). Detecting phishing attacks using natural language processing and machine learning. In *IEEE 12th International Conference on Semantic Computing*, 300–301. https://doi.org/10.1109/ICSC.2018.00056

Rao, R. S., & Pais, R. (2019). Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Computing and Applications*, *31*, 3851–3873. https://doi.org/10.1007/s00521-017-3305-0

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*, *11*(4), 193. https://doi.org/10.3390/info11040193

Roohi, F., & Phil, M. (2013). Neuro fuzzy approach to data clustering: A framework for analysis. *European Scientific Journal*, *9*(9), 183–192.

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

Samad, D., & Gani, A. G. (2020). Analyzing and predicting spear-phishing using machine learning methods. *Multidisciplinary Sciences*, *10*(4), 262–273. https://doi.org/10.35925/j.multi.2020.4.30

Schiller, K., Adamsky, F., & Benenson, Z. (2023). Towards an empirical study to determine the effectiveness of support systems against e-mail phishing attacks. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 288. https://doi.org/10.1145/3544549.3585658

Shah, R. K., Hasan, M. K., Islam, S., Khan, A., Ghazal, T. M., & Khan, A. N. (2022). Detect phishing website by fuzzy multi-criteria decision making. In *1st International Conference on AI in Cybersecurity,* 1–8. https://doi.org/10.1109/ICAIC53980.2022.9897036

Siwakoti, Y. R., Bhurtel, M., Rawat, D. B., Oest, A., & Johnson, R. (2023). Advances in IoT security: Vulnerabilities, enabled criminal services, attacks and countermeasures. *IEEE Internet of Things Journal, 10*(13), 11224–11239. https://doi.org/10.1109/JIOT.2023.3252594

Stojanović, B., Božić, J., Hofer-Schmitz, K., Nahrgang, K., Weber, A., Badii, A., …, & Runevic, J. (2021). Follow the trail: Machine learning for fraud detection in Fintech applications. *Sensors*, *21*(5), 1594. https://doi.org/10.3390/s21051594

Tanimu, J., & Shiaeles, S. (2022). Phishing detection using machine learning algorithm. In *IEEE International Conference on Cyber Security and Resilience,* 317–322. https://doi.org/10.1109/CSR54599.2022.9850316

Tayyab, S., & Masood, A. (2019). A review: Phishing detection using URLs and hyperlinks information by machine learning approach. *International Journal of Computer Science and Mobile Computing*, *8*(3), 345–351.

Thirumallai, C., Mekala, M. S., Perumal, V., Rizwan, P., & Gandomi, A. H. (2020). Machine learning inspired phishing detection (PD) for efficient classification and secure storage distribution (SSD) for cloud-IoT application. In *IEEE Symposium Series on Computational Intelligence*, 202–210. https://doi.org/10.1109/SSCI47803.2020.9308183

Toolan, F., & Carthy, J. (2009). Phishing detection using classifier ensembles. In *eCrime Researchers Summit*, 1–9. https://doi.org/10.1109/ECRIME.2009.5342607

Toolan, F., & Carthy, J. (2010). Feature selection for spam and phishing detection. In *eCrime Researchers Summit*, 1–12. https://doi.org/10.1109/ecrime.2010.5706696

Tyagi, I., Shad, J., Sharma, S., Gaur, S., & Kaur, G. (2018). A novel machine learning approach to detect phishing websites. In *5th International Conference on Signal Processing and Integrated Networks*, 425–430. https://doi.org/10.1109/SPIN.2018.8474040

Upadhyay, D., Manero, J., Zaman, M., & Sampalli, S. (2021). Intrusion detection in SCADA based power grids: Recursive feature elimination model with majority vote ensemble algorithm. *IEEE Transactions on Network Science and Engineering*, *8*(3), 2559–2574. https://doi.org/10.1109/TNSE.2021.3099371

Verma, R., & Rai, N. (2015). Phish-idetector: Message-id based automatic phishing detection. In *12th International Joint Conference on e-Business and Telecommunications,* 427–434.

Yao, Y., Sullivan, T., Yan, F., Gong, J., & Li, L. (2022). Balancing data for generalizable machine learning to predict glass-forming ability of ternary alloys. *Scripta Materialia*, *209*, 114366. https://doi.org/10.1016/j.scriptamat.2021.114366

Yin, Z., Wang, Y., Liu, L., Zhang, W., & Zhang, J. (2017). Cross-subject EEG feature selection for emotion recognition using transfer recursive feature elimination. *Frontiers in Neurorobotics*, *11*, 19. https://doi.org/10.3389/fnbot.2017.00019

Zaini, N. S., Stiawan, D., Razak, M. F. A., Firdaus, A., Din, W. I. S. W., Kasim, S., & Sutikno, T. (2020). Phishing detection system using machine learning classifiers. *Indonesian Journal of Electrical Engineering and Computer Science*, *17*(3), 1165–1171. https://doi.org/10.11591/ijeecs.v17.i3.pp1165-1171

How to Cite: Tanimu, J., Shiaeles, S., & Adda, M. (2024). A Comparative Analysis of Feature Eliminator Methods to Improve Machine Learning Phishing Detection. *Journal of Data Science and Intelligent Systems*, *2*(2), 87–99. https://doi.org/10.47852/bonviewJDSIS32021736