applied sciences

MDPI

Article

Improved Phishing Attack Detection with Machine Learning: A Comprehensive Evaluation of Classifiers and Features

Sibel Kapan [1] and Efnan Sora Gunal [2,*]

[1] Department of Computer Engineering, Hacettepe University, 06800 Ankara, Türkiye; skapan@hacettepe.edu.tr
[2] Department of Computer Engineering, Eskisehir Osmangazi University, 26040 Eskisehir, Türkiye
* Correspondence: esora@ogu.edu.tr

Abstract: In phishing attack detection, machine learning-based approaches are more effective than simple blacklisting strategies, as they can adapt to new types of attacks and do not require manual updates. However, for these approaches, the choice of features and classifiers directly influences detection performance. Therefore, in this work, the contributions of various features and classifiers to detecting phishing attacks were thoroughly analyzed to find the best classifier and feature set in terms of different performance metrics including accuracy, precision, recall, F1-score, and classification time. For this purpose, a brand-new phishing dataset was prepared and made publicly available. Using an exhaustive strategy, every combination of the feature groups was fed into various classifiers to detect phishing websites. Two existing benchmark datasets were also used in addition to ours for further analysis. The experimental results revealed that the features based on the uniform resource locator (URL) and hypertext transfer protocol (HTTP), rather than all features, offered the best performance. Also, the decision tree classifier surpassed the others, achieving an F1-score of 0.99 and being one of the fastest classifiers overall.

Keywords: web security; phishing attack; machine learning; feature analysis

check for updates

Citation: Kapan, S.; Sora Gunal, E. Improved Phishing Attack Detection with Machine Learning: A Comprehensive Evaluation of Classifiers and Features. *Appl. Sci.* **2023**, *13*, 13269. https://doi.org/10.3390/app132413269

Academic Editor: Luis Javier Garcia Villalba

Received: 13 October 2023
Revised: 12 December 2023
Accepted: 14 December 2023
Published: 15 December 2023

1. Introduction

Currently, most companies and organizations provide fast and easy access to the services we use in daily life by digitizing them. However, this has brought the problem of data security. The personal information, financial information, and passwords used to access these services may lead to a data security problem. There are many different types of cyber attacks used to obtain personal and financial data. Phishing attacks, one of these types of attacks, is the stealing of users' personal and financial information through a fake website created by copying a legitimate and known website [1]. Phishing websites are designed by copying legal websites and creating similar ones. Hence, victims have no doubts about the website they entered, as there is little difference between a phishing website and the legitimate one.

The Anti-Phishing Working Group (APWG), which is supported by many IT security companies, is an organization studying the effects of phishing attacks [2]. Although the motivation and attack strategies of the attackers are known, a fully effective method to prevent phishing attacks has not been developed yet. In the third quarter of 2022, the APWG observed 1,270,883 total phishing attacks, a new record, and the worst quarter for phishing that the APWG has ever observed so far [3]. In this time interval, the APWG's founding member, OpSec Security found that phishing attacks against the financial sector, which includes banks, remained the largest set of attacks, accounting for approximately 23% of all phishing. The attacks against webmail and software-as-a-service providers remained steady, at approximately 17% of all attacks, while the attacks against retail/e-commerce sites fell to 4%, down from 14% in the first quarter. Phishing against social media companies trended downward to 11%, after fluctuating from 8% of all attacks in the fourth quarter of 2021 to 15% in the second quarter of 2022. Phishing against cryptocurrency targets fell from 4% in the second quarter to 2% in the third quarter [2,3]. Furthermore, phishing websites have been spreading by advertising on social media platforms or sending messages to predetermined victims on social media [4].

As the threat rapidly increases, various strategies have been developed to detect these attacks. The most common of those strategies, blacklisting, involves creating a list of known malicious websites that are known to be associated with phishing attacks [1]. When a user attempts to access a website, the system can check the domain of the website against the blacklist and block access if it appears on the list. Blacklisting can be an effective way to block known phishing websites, but it is not foolproof. New phishing websites are constantly being created, and it can be difficult to keep the blacklist up to date. Additionally, attackers can sometimes use domain names or websites that are not on the blacklist, making it harder to detect and block their attacks. On the other hand, machine learning involves using algorithms that can learn from and make predictions or decisions based on data [5,6]. In the context of phishing attack detection, a machine learning model could be trained on a dataset of known phishing and legitimate websites. The model would then be able to learn the characteristics that are commonly associated with phishing attacks and use this knowledge to make predictions about the likelihood that a new website is a phishing attack. Machine learning-based approaches to phishing attack detection can be more effective than blacklisting, as they can adapt to new types of phishing attacks and do not require manual updates to a blacklist. However, selected features and classification algorithms have a direct influence on attack detection performance, such that a poor choice of features and classifiers may degrade the performance significantly.

Therefore, this work thoroughly analyzes the contributions of various features and classification algorithms to detecting phishing attacks to find the best classifier and feature set for improved detection. For this purpose, a brand-new dataset was prepared and made publicly available at https://github.com/sibelkapan/phishing_dataset (accessed on 10 October 2023) for the research community. The dataset consists of 500 phishing and 500 legitimate websites represented with 25 features that are composed of uniform resource locator (URL), hypertext markup language (HTML), and hypertext transfer protocol (HTTP) attributes. Using an exhaustive selection strategy, every combination of these feature groups was then fed into various well-known classification algorithms, including k-nearest neighbors (k-NN), support vector machine (SVM), naive Bayes (NB), decision tree (DT), multi-layer perceptron (MLP), and stochastic gradient descent (SGD) [5,6], to detect phishing websites. While evaluating the effectiveness of the different features and classifiers, five different performance metrics, including accuracy, precision, recall, F1-score, and classification time, were utilized [5,6]. During the experiments, two existing datasets were also used in addition to the newly introduced dataset. Hence, the main contributions of our work can be summarized as follows:

- Introducing a new publicly available dataset for phishing attack detection,
- Comparing the performance of the features and finding the best feature subset for phishing attack detection in terms of different performance metrics,
- Comparing the performance of the classification algorithms and finding the best classifier for phishing attack detection in terms of different performance metrics.

2. Related Work

In the literature, there are many studies to detect, prevent, and decrease the effects of phishing attacks. These studies have been divided into different categories [7]. To prevent attacks and reduce the impact area, the white–black list and heuristic methods have been widely used. These methods help the careless and unconscious user to be aware that the visited website is a phishing website and to prevent the user from visiting phishing websites. The main purposes of preventing access to a harmful website or warning the users are to minimize the material and moral damage caused by phishing websites and to ensure that legitimate websites belonging to institutions and organizations serve their users safely.

One of the most common attack detection methods, the white–black list, uses two different types of tagged website listings, legal and phishing, to identify whether a website is phishing or not [8]. The best examples of white–black list methods are the Google Safe Browsing application programming interface (API) [9] and the Netcraft AntiPhishing tool [10]. The white–black list method can reliably detect website on such a list. In a study using the blacklist method, web pages were scored using the URL, domain, and page contents of the website, and if the web page had a high score, it was blacklisted [11]. This method can detect phishing websites with a false positive rate below 0.1% and a false negative rate below 8%. However, the list methods cannot prevent zero-day attacks, which are new attacks that have not been used by attackers before [12]. Also, the rate of phishing detection depends on the update frequency of the lists. The lower the frequency, the lower the probability of attack detection.

Hence, the failure of the list methods to detect unknown or newly created phishing websites has led researchers to use heuristic methods [13]. Heuristic-based studies can be divided into several subcategories: content-based, rule-based, data mining- or machine learning-based, and hybrid. Furthermore, many researchers have investigated spam and phishing as well as their relationship [14–16]. The term frequency-inverse document frequency method used for detecting phishing websites is a content-based approach [17]. In another work, five different file-matching methods were used for classification using the similarity coefficient of websites [18]. The similarity of visual content uses screenshots, images, document object models, and CSS files of websites [19–21].

Studies using rule-based and machine learning-based methods are of great importance for the fast and effective detection of phishing attacks. Phishing websites were identified by Mohammad et al. [22] using the rule-based methods. The same authors then utilized 17 features with 4 categories in another work [23]. Eight features that have an insignificant effect on detecting phishing websites were removed from the dataset with the help of chi-square statistics so that an improved error rate was obtained using the C4.5 and other algorithms. Another effort on phishing attack detection employed artificial neural networks [24]. Basnet et al. [25] utilized 15 features together with the C4.5 and logistic regression classification algorithms. On the other hand, the results of the experiments and the difference between the performances of the algorithms were quite insignificant. Fette et al. [26] used the random forest algorithm for protection against phishing email attacks. In two different studies, fuzzy data mining techniques were used for identifying phishing websites that targeted internet banking [27,28]. In another study, the authors presented the cumulative distribution function gradient algorithm for feature selection in attack detection [29]. The features were extracted from the URL and HTML source code of the websites for creating a new dataset, and the random forest algorithm offered the best results. Sahingoz et al. proposed an anti-phishing system using seven different classification algorithms and natural language processing-based features [30]. In another effort to detect phishing sites, the authors designed a novel deep learning network formed by convolutional and multi-head self-attention layers [31]. Sonowal and Kuppusamy [32] introduced a multilayer model to detect phishing. The model incorporates five layers, including an auto-upgraded whitelist, URL features, lexical signature, string matching, and accessibility score comparison layers. Almomani et al. [33] used several machine learning algorithms and semantic features to detect phishing attacks. Bahaghighat et al. presented another phishing detection model utilizing traditional classification algorithms and feature transformation [34]. Adebowale et al. employed convolutional neural networks and long short-term memory networks to build a hybrid classification model for phishing website detection [35]. Additionally, several studies have conducted a comprehensive review of recent literature on phishing website detection [36–38].

3. Materials and Methods

In this section, our new dataset is introduced, the features are described, and the employed classifiers and performance metrics are briefly explained. A part of this work has been published in the Master of Science thesis by Kapan [39].

3.1. Dataset and Features

In our work, a new dataset was prepared to train and evaluate machine learning models for detecting phishing websites. The dataset has been made publicly available at https://github.com/sibelkapan/phishing_dataset (accessed on 10 October 2023) for the research community. While preparing the dataset, the URLs of legitimate and phishing websites were collected from the Alexa and PhishTank platforms, respectively.

Alexa is a resource that is widely used by researchers to collect legitimate websites [40]. It provides website rankings and analytics, and it also maintains a collection of websites including the most frequently used platforms, such as popular search engines and their services, social media platforms, financial sites, retail websites, and more. Additionally, the websites are further categorized according to their content. Hence, the ranking and analytics provided by this platform simply guide users to access legitimate sites. Due to the significant impact of phishing attacks on the finance and shopping sectors, URLs related to the subject were gathered using the keywords "finance" and "shopping". Furthermore, websites containing the keywords "login" and "update" were added to the dataset to include sites related to login and update functions, which are commonly exploited in phishing attacks. As a result, the legitimate class of the dataset encompasses not only widely visited legitimate websites but also those with lower traffic. Legitimate websites that could potentially be confused with phishing sites were also incorporated into the dataset.

PhishTank is, on the other hand, a collaborative online platform and community-driven database dedicated to combating phishing attacks on the internet [41]. Operated by OpenDNS, a cybersecurity company, PhishTank allows users to submit and verify suspected phishing URLs. The platform leverages the collective intelligence of its user community to identify and catalog phishing websites. Users can report potential phishing sites they encounter, and others can validate these reports, helping to build a comprehensive and up-to-date database of known phishing URLs. Researchers and security professionals therefore use PhishTank as a resource to enhance phishing detection systems and protect users from falling victim to fraudulent online activities [23,25,27]. In our work, the URLs of phishing websites were collected from this platform, in a similar way used with Alexa.

The collected URLs from both platforms were then visited via the Firefox browser using the Selenium library [42], and 25 numerical features were extracted for each website. Websites with missing features were not included in the dataset. Ultimately, our dataset consists of a total of 500 phishing and 500 legitimate unique websites, each represented by a 25-dimensional feature vector. The features are categorized into three groups: URL, HTML, and HTTP. The URL features were collected directly from the URLs. The HTML features were extracted from the visited websites. The HTTP features were gathered from the HTTP response codes of the servers. All these features and feature groups are summarized in Table 1.

The choice of the meta information-driven URL, HTML, and HTTP features instead of the actual content of a website was mainly influenced by practical, technical, and ethical considerations. Scalability and speed, the adversarial nature of phishing attacks, privacy concerns, generalization across languages and cultures, and interpretability of features were among the main reasons behind this choice.

Analyzing the content of websites can be computationally expensive and time-consuming, especially when considering language complexity and typos. Phishing detection systems need to operate in real-time to effectively prevent users from accessing malicious sites. Therefore, utilizing meta information allows for faster processing and scalability.

Table 1. Feature groups and corresponding features for phishing website detection.

No	Feature	Feature Group	Description
1	Domain name similarity	URL	Similarity (based on Ratcliff-Obershelp's algorithm [43]) between the domain name of the visited website and the URL domain name obtained from Alexa or PhishTank
2	URL length		Number of all characters in a URL
3	HTTP protocol		HTTP protocol type: standard (0) or secure (1)
4	# '.' symbol		Number of dot symbols in a URL
5	# '/' symbols		Number of slash symbols in a URL
6	# '//' symbols		Number of double slash symbols in a URL
7	# '-' symbols		Number of dash symbols in a URL
8	# '_' symbols		Number of underscore symbols in a URL
9	# '=' symbols		Number of equal symbols in a URL
10	# '(' and ')' symbols		Number of parenthesis symbols in a URL
11	# '{' and '}' symbols		Number of curly bracket symbols in a URL
12	# '[' and ']' symbols		Number of square bracket symbols in a URL
13	# '<' and '>' symbols		Number of less than and greater than symbols in a URL
14	# '~' symbols		Number of tilde symbols in a URL
15	# '*' symbols		Number of asterisk symbols in a URL
16	# '+' symbols		Number of plus symbols in a URL
17	Inclusion of '@' symbol		URL includes an at symbol (1) or not (0)
18	Inclusion of IP address		URL includes an IP address (1) or not (0)
19	# <a> tags	HTML	Number of <a> tags in a website, used to create hyperlinks or anchor links, which is an essential element for linking one webpage to another, linking to different sections within the same page, or linking to external resources
20	# <input> tags		Number of <input> tags in a website, used to create various types of interactive form elements
21	# <button> tags		Number of <button> tags in a website, used to create a clickable button for triggering actions, submitting forms, or performing other interactive functions
22	# <link> tags		Number of <link> tags in a website, used to link external resources, such as stylesheets, icons, and other documents, to an HTML document
23	# <iFrame> tags		Number of <iFrame> tags in a website, used to embed an external resource, such as another HTML document, a video, or a web page, within the current document
24	HTTP response history	HTTP	HTTP response code returned by a server to indicate the outcome of a client's request made to the server.
25	Redirect		Website redirects to another site (1) or not (0), detected using HTTP redirection response codes

Also, phishers constantly evolve their tactics to evade detection mechanisms. Analyzing content might be more susceptible to evasion techniques, as phishers may modify content to mimic legitimate sites more effectively. Meta information, on the other hand, tends to be more stable and less prone to manipulation.

Analyzing the content of websites may also raise privacy concerns, especially if the analysis involves examining the actual text on the page. However, meta information analysis is less intrusive and avoids the potential ethical issues associated with examining the content of websites.

Additionally, meta information features can be more language-independent and culturally neutral. Phishing attacks can target users anywhere in the world, and focusing on meta information allows for the development of models that are effective across different languages and regions.

Moreover, meta information features are generally more interpretable. Understanding why a model classifies a website as phishing based on meta information is often simpler than interpreting decisions based on the subtle content of a website.

3.2. Classifiers

A classifier, or classification algorithm, is a type of machine learning algorithm that is designed to categorize input data into predefined classes or categories [5,6]. The goal of a classification algorithm is to determine a relationship, mapping input instances to a target class based on a set of labeled training instances. The instances are represented with relevant features. Once trained, the algorithm can then predict the class of new, unseen instances. Therefore, in our work, the input instances correspond to websites, the features correspond to URL-, HTML-, and HTTP-based features extracted from input instances as explained earlier, and the target class can be either phishing or legitimate.

Six well-known classifiers were employed in our work to categorize websites: SVM, k-NN, DT, SGD, NB, and MLP [5,6]. The attack detection performances of these classifiers, together with the above-mentioned feature sets, were comparatively evaluated throughout the experiments.

SVM aims to find a hyperplane that maximally separates the different classes in the data [5,6]. This is accomplished by finding the hyperplane that has the largest margin, which is the distance between the hyperplane and the closest data points from each class, known as the support vectors. This way, SVMs can generalize well to unseen data. In our experimental setup, we used a regularization parameter of 1 and the radial basis function (RBF) kernel.

k-NN is a type of instance-based, or lazy, learning algorithm [5,6]. It works by storing all the available data, and when a new data point needs to be classified or its value predicted, it looks at the k number of closest data points (based on some distance measure) and assigns the class or value that is most common among the k nearest neighbors. The k is user-defined, and it can be thought of as a tuning parameter. One of the main advantages of k-NN is its simplicity, as it requires little to no training, unlike many other classification algorithms. Also, it is considered robust to noisy data and insensitive to the scale of the data. In our experimental setup, we determined the optimal value of k as 5.

DT works by recursively partitioning the data into smaller subsets based on the values of the input features [5,6]. The idea is to create a tree-like model of decisions and their possible consequences so that at each internal node of the tree, a decision is made based on the values of one of the input features, and each leaf node represents a class label or a value, depending on the task. In our experimental setup, we used the Gini impurity as the split criterion.

NB is a probabilistic classifier that is based on Bayes' theorem, which is a mathematical formula used to calculate the probability of an event based on prior knowledge of conditions that might be related to the event [5,6]. In the context of a Naive Bayes classifier, the event is the class label of a new data point, and the prior knowledge is represented by the class labels and feature values of the training data. The "naive" part of the name refers to the assumption of independence between the features, meaning that the algorithm assumes that the presence or absence of one feature does not affect the presence or absence of another feature.

The SGD classifier is a type of linear classifier [5,6]. SGD is an optimization algorithm that is used to minimize the cost function of a model. The cost function represents how well the model fits the training data. The algorithm starts with an initial set of model parameters and iteratively updates them in the direction of the negative gradient of the cost function, using only a small subset of the training data at each iteration, which is why it is called stochastic (random sample). In our experimental setup, we utilized a hinge loss function with an l2 regularization term, set at a regularization strength of 0.0001, and executed training for a maximum of 1000 iterations.

The MLP classifier is a type of artificial neural network that consists of one or more layers of artificial neurons, also called perceptrons, which are connected by weighted links and are organized in a feed-forward architecture [5,6]. Each layer receives input from the previous layer and sends output to the next layer, and the last layer is the output layer, which produces the final class label or value predictions. An MLP is trained using a variant of the backpropagation algorithm, which is a supervised learning algorithm that uses the gradient descent optimization method to adjust the weights of the network. During training, the input data are propagated forward through the network, and the difference between the network's output and the true label or value is used to compute the gradients of the cost function with respect to the weights. Then, the gradients are used to update the weights. In our experimental configuration, we employed a neural network architecture with two hidden layers utilizing the rectified linear unit (ReLU) as the activation function. The optimization algorithm employed was the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm, coupled with a regularization term set at 0.00001. The learning rate was held constant throughout training, and the model underwent a maximum of 200 iterations.

3.3. Performance Metrics

We utilized five different performance metrics, including accuracy, precision, recall, F1-score, and classification time, to assess the effectiveness of phishing detection approaches [5,6]. These metrics are essential for understanding the strengths and weaknesses of a model in differentiating between the positive (phishing) and negative (legitimate) classes as well as attack detection speed.

Accuracy, precision, recall, and F1-score are calculated based on the four possible outcomes of a detection model's predictions: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). TPs correspond to the instances where the model correctly predicts the positive class. Hence, in the context of phishing website detection, a true positive occurs when the model correctly identifies a website as phishing, and indeed, the website is a phishing site. FPs correspond to the instances where the model incorrectly predicts the positive class. In other words, a false positive occurs when the model mistakenly identifies a legitimate website as phishing. TNs are the outcomes where the model correctly predicts the negative class. A true negative occurs when the model correctly identifies a website as legitimate, and indeed, the website is a legitimate one. FNs are the outcomes where the model incorrectly predicts the negative class. In other words, a false negative occurs when the model fails to identify a phishing website and incorrectly classifies it as legitimate.

Accuracy is a fundamental metric that gauges the overall correctness of a classification model. It quantifies the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances in the dataset, as formulated in (1). A high accuracy indicates a model's ability to make correct predictions across both positive (phishing) and negative (legitimate) instances.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{1}$$

Precision measures the accuracy of the positive predictions made by a model. It quantifies the ratio of true positive predictions to the total number of instances predicted as positive (both true positives and false positives), as formulated in Equation (2). Precision is particularly relevant, as it assesses the model's ability to avoid misclassifying legitimate websites as phishing. A high precision value implies a low false positive rate.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{2}$$

Recall, also known as sensitivity or the true positive rate, evaluates the model's capacity to identify all positive instances in the dataset. It measures the ratio of true positive predictions to the total number of actual positive instances (both true positives and false negatives), as formulated in Equation (3). A high recall is crucial, as it indicates the model's effectiveness in capturing a significant portion of actual phishing instances, minimizing false negatives.

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{3}$$

The F1 score is the harmonic mean of precision and recall, providing a balanced evaluation of a model's performance, as formulated in Equation (4). It addresses situations where high precision and high recall are both desired. The F1 score is particularly relevant, where achieving a balance between accurately identifying phishing websites and minimizing false positives is crucial.

$$\text{F1 score} = (2 \cdot \text{Precision} \cdot \text{Recall})/(\text{Precision} + \text{Recall}) \quad (4)$$

The final performance metric, classification time, simply refers to the duration (in seconds) it takes for a classifier to predict the category of a new instance (website) after it has been trained on a given dataset.

4. Results and Discussion

In the experimental work, the contributions of different classifiers and feature sets to phishing website detection performance were examined using an exhaustive search strategy. As mentioned earlier, the dataset consists of the features from URL, HTML, and HTTP groups. The exhaustive search method [44] was used on these groups, and the dataset was divided into seven subsets based on every possible combination of the feature groups: URL, URL + HTML, URL + HTTP, HTML, HTML + HTTP, HTTP, and URL + HTML + HTTP features. With the help of these combinations, the performances of the different feature groups and classification algorithms, their relationships with each other, and their distinctiveness have been revealed.

The experiments were conducted using a computer equipped with an Intel (R) Core (TM) i5-2430M CPU @ 2.40 GHz and 8 GB of RAM. The Python programming language and Scikit-learn library [45] were used for the implementation of all the methods.

The classification time (in seconds), accuracy, precision, recall, and F1-score for each case are comparatively presented in Table 2, where the best value for each aspect is indicated in bold. Accordingly, the highest F1 score (0.99) was achieved with the URL + HTTP feature set and the DT classifier. On the other hand, the URL or URL + HTTP features with the NB classifier offered the lowest F1 score (0.53). The highest accuracy (0.99) was obtained using the URL + HTTP feature set and the DT classifier, whereas the URL or URL + HTTP features with the NB classifier provided the lowest accuracy (0.67). The URL + HTTP feature set and the DT classifier offered the highest precision (0.99), while the HTML feature set and the NB classifier achieved the lowest precision (0.68). In the case of recall performance, there were many leaders. The URL + HTTP or HTTP or URL + HTML + HTTP feature set with the SVM classifier, the HTTP feature set with the SGD classifier, the HTTP feature set with the NB classifier, the URL + HTTP or HTTP feature set with the MLP classifier, the URL + HTTP feature set with the k-NN classifier, and the HTTP feature set with the DT classifier provided the highest recall (0.99). On the other hand, the URL or URL + HTML or URL + HTTP feature set with the NB classifier achieved the lowest recall value (0.37).

Considering the classification time, the URL feature set with the NB classifier was the fastest model, with a classification time of 0.01 s, whereas the HTML + HTTP feature set with the MLP classifier was the slowest model, with a classification time of 1.00 s.

Ideally, an attack detection framework should yield not only the highest possible detection rate but also the lowest possible detection time. Hence, an analysis was also conducted to reveal the fastest classifier among the ones providing the highest F1 scores. The result of this analysis is visualized in Figure 1. As shown in this figure, DT not only offers the best F1 score but is also one of the fastest classifiers. On the other hand, the other relatively fast classifiers, such as SVM, SGD, NB, and k-NN, failed to provide an F1 score that was as high as that of the DT classifier. Also, MLP, the slowest classifier overall, could not beat the F1 score of the DT classifier.

Table 2. Detection statistics for each combination of the feature sets and classifiers.

Classifier	Feature Set	# Features	Time	Accuracy	Precision	Recall	F1
SVM	URL	18	0.03	0.91	0.91	0.91	0.91
	URL + HTML	23	0.03	0.94	0.93	0.95	0.94
	URL + HTTP	20	0.03	0.97	0.96	**0.99**	0.97
	HTML	5	0.03	0.85	0.81	0.91	0.86
	HTML + HTTP	7	0.03	0.92	0.88	0.97	0.93
	HTTP	2	0.02	0.93	0.88	**0.99**	0.93
	URL + HTML + HTTP	25	0.05	0.98	0.97	**0.99**	0.98
SGD	URL	18	0.03	0.84	0.86	0.81	0.84
	URL + HTML	23	0.03	0.92	0.91	0.93	0.92
	URL + HTTP	20	0.02	0.97	0.97	0.97	0.97
	HTML	5	0.02	0.84	0.84	0.85	0.84
	HTML + HTTP	7	0.03	0.91	0.88	0.95	0.92
	HTTP	2	0.03	0.92	0.87	**0.99**	0.93
	URL + HTML + HTTP	25	0.03	0.98	0.98	0.97	0.98
NB	URL	18	**0.01**	0.67	0.93	0.37	0.53
	URL + HTML	23	0.03	0.68	0.95	0.37	0.54
	URL + HTTP	20	0.02	0.67	0.93	0.37	0.53
	HTML	5	0.02	0.75	0.68	0.94	0.79
	HTML + HTTP	7	0.02	0.81	0.75	0.95	0.84
	HTTP	2	0.03	0.92	0.87	**0.99**	0.93
	URL + HTML + HTTP	25	0.03	0.68	0.95	0.38	0.54
MLP	URL	18	0.61	0.95	0.92	0.97	0.95
	URL + HTML	23	0.08	0.93	0.90	0.96	0.93
	URL + HTTP	20	0.12	0.98	0.97	**0.99**	0.98
	HTML	5	0.17	0.84	0.83	0.87	0.85
	HTML + HTTP	7	1.00	0.91	0.88	0.95	0.92
	HTTP	2	0.08	0.92	0.87	**0.99**	0.93
	URL + HTML + HTTP	25	0.06	0.96	0.97	0.95	0.96
k-NN	URL	18	0.05	0.93	0.89	0.98	0.93
	URL + HTML	23	0.05	0.92	0.91	0.93	0.92
	URL + HTTP	20	0.05	0.97	0.94	**0.99**	0.97
	HTML	5	0.05	0.85	0.85	0.85	0.85
	HTML + HTTP	7	0.05	0.93	0.91	0.96	0.94
	HTTP	2	0.05	0.92	0.87	0.98	0.92
	URL + HTML + HTTP	25	0.05	0.97	0.95	0.98	0.97
DT	URL	18	0.02	0.92	0.93	0.92	0.92
	URL + HTML	23	0.02	0.91	0.90	0.92	0.91
	URL + HTTP	20	0.03	**0.99**	**0.99**	0.98	**0.99**
	HTML	5	0.02	0.81	0.80	0.83	0.82
	HTML + HTTP	7	0.02	0.91	0.90	0.92	0.91
	HTTP	2	0.03	0.93	0.89	**0.99**	0.94
	URL + HTML + HTTP	25	0.03	0.96	0.98	0.94	0.96

To further assess the effectiveness of the utilized classification algorithms on attack detection, we also tested them on two additional phishing datasets. The first dataset was proposed by Chiew et al. [29] while the second one is hosted at the UCI Machine Learning Repository [46]. Both datasets are publicly available. Since these datasets contain different numbers of samples than ours, 500 phishing and 500 legitimate website samples were randomly selected from each dataset for a fair evaluation. At this stage, feature selection was not applied, and the datasets were divided into training (70%) and test (30%) sections. The test results are listed comparatively in Table 3 to provide a comparative analysis of those classifiers applied to different datasets, shedding light on the strengths and weaknesses of the classifiers with different sets of features and instances. The performance metrics for comparison were chosen as accuracy, precision, recall and F1 score. As shown in the table, the number of features of the datasets was variable, ranging from 25 to 48 features, indicating different levels of complexity in the feature space. Across all classifiers, the performance metrics varied based on the dataset and the number of features considered. However, SVM and DT consistently demonstrated robust performances across different datasets and feature dimensions, suggesting their suitability for phishing detection tasks.

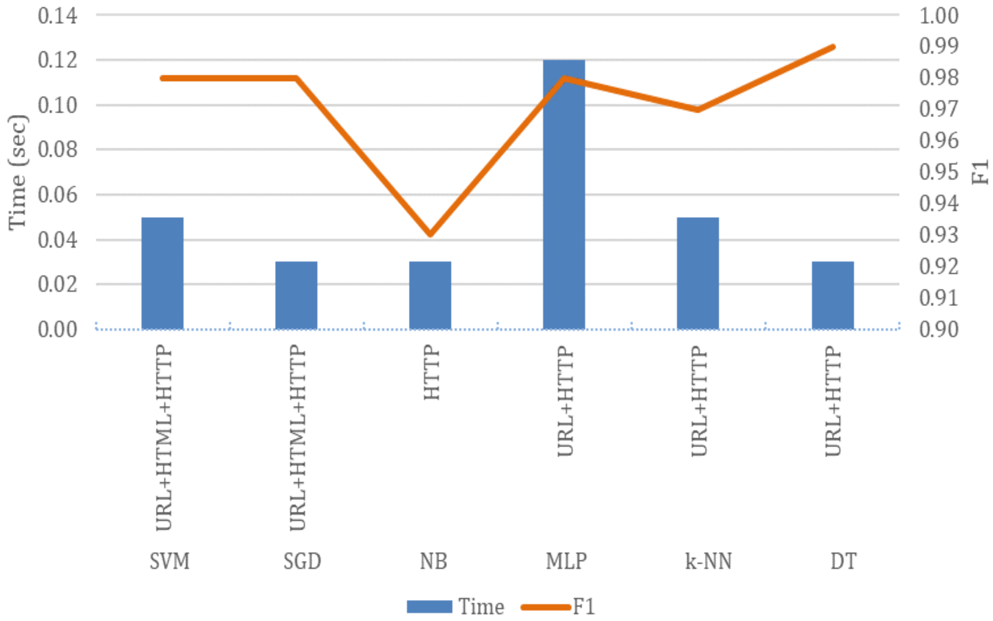

Figure 1. The classification times vs. the highest F1 scores achieved using each classifier and the corresponding feature subsets.

Table 3. Comparison of our dataset with benchmark datasets.

Classifiers	Dataset	# Features	Accuracy	Precision	Recall	F1
SVM	[29]	48	0.91	0.90	0.91	0.91
	[46]	30	0.87	0.87	0.87	0.87
	Our dataset	25	0.98	0.97	0.99	0.98
SGD	[29]	48	0.89	0.90	0.88	0.89
	[46]	30	0.84	0.90	0.75	0.82
	Our dataset	25	0.98	0.98	0.97	0.98
NB	[29]	48	0.81	0.99	0.62	0.76
	[46]	30	0.68	0.97	0.38	0.55
	Our dataset	25	0.68	0.99	0.38	0.54
MLP	[29]	48	0.86	0.86	0.86	0.86
	[46]	30	0.84	0.84	0.83	0.84
	Our dataset	25	0.96	0.97	0.95	0.96
k-NN	[29]	48	0.85	0.85	0.86	0.85
	[46]	30	0.88	0.88	0.89	0.88
	Our dataset	25	0.97	0.99	0.98	0.97
DT	[29]	48	0.93	0.93	0.93	0.93
	[46]	30	0.90	0.95	0.84	0.89
	Our dataset	25	0.96	0.98	0.94	0.96

The experimental results obtained in our work were also compared to earlier works in the literature. The comparison is summarized in Table 4, where the best classification models and accuracies are listed. As shown in the table, our work stands out among the cited studies in phishing attack detection, showcasing a significant dataset size of 1000 instances and utilizing various classifiers, resulting in an impressive accuracy of

0.99. The proposed work is evidently effective in distinguishing between legitimate and phishing instances within the provided dataset. The competitive accuracy score positions the proposed work at a level equivalent to or even surpassing other studies. The robust performance of the proposed work suggests the efficacy of the appropriate combination of the features and classifiers in the context of phishing detection.

Table 4. Comparison of the experimental results with earlier works.

Study	Dataset Size	Classification Method	Accuracy
Proposed work	500 legitimate, 500 phishing	DT	0.99
[17]	100 legitimate, 100 phishing	TF-IDF	0.95
[24]	600 legitimate, 800 phishing	Neural network	0.92
[25]	24086 legitimate, 16797 phishing	Rule-based	0.99
[26]	6950 legitimate, 860 phishing	PILFER	0.99
[29]	5000 legitimate, 5000 phishing	Random forest	0.94
[32]	995 legitimate, 667 phishing	Multi-layer filters	0.92

5. Conclusions

Phishing attacks are a significant threat to the security of organizations and individuals, and machine learning can be a powerful tool for detecting them. Classifiers and features are essential components of this process. By selecting the most appropriate classifier and evaluating the relevance of different features, it is possible to improve the performance of phishing detection.

In this work, we comprehensively evaluated different classifiers as well as features to find out their contributions to the performance of phishing detection in terms of several metrics such as accuracy, recall, precision, F1 score, and detection time. We also introduced a new and publicly available phishing dataset for the research community. According to our evaluations, the URL + HTTP feature set paired with the DT classifier appeared as the top performer, achieving the highest F1 score (0.99), accuracy (0.99), and precision (0.99). On the contrary, the NB classifier with several feature sets exhibited the lowest F1 score (0.53), precision (0.68), and accuracy (0.67). Additionally, the recall performances varied across the classifiers, with SVM consistently providing the highest recall (0.99) most of the time. The analysis of classification time revealed that the DT classifier yielded not only the best F1 score but also one of the shortest classification times, while NB was the fastest with the URL feature set (0.01 s), and MLP was the slowest with the HTML + HTTP feature set (1.00 s). Further testing on other benchmark datasets reaffirmed the robust performances of DT and SVM, positioning them as reliable choices for phishing detection tasks.

With the increasing prevalence of phishing attacks, it is crucial that organizations invest in the development of robust and effective phishing detection systems. Based on our findings, we can conclude that organizations can improve their ability and speed to detect and respond to phishing attacks and better protect themselves and their customers from this threat by utilizing machine learning with appropriate classifiers and feature sets.

In future work, other possible features and classifiers can be analyzed. Also, a cost analysis can be performed by considering the effects of the collection speed of the features.

Author Contributions: Conceptualization, E.S.G.; methodology, S.K. and E.S.G.; software, S.K.; validation, S.K. and E.S.G.; data curation, S.K.; writing—original draft preparation, S.K. and E.S.G.; writing—review and editing, E.S.G.; supervision, E.S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/sibelkapan/phishing_dataset/.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Asiri, S.; Xiao, Y.; Alzahrani, S.; Li, S.; Li, T. A survey of intelligent detection designs of HTML URL phishing attacks. *IEEE Access* **2023**, *11*, 6421–6443. [CrossRef]
2. APWG Anti-Phishing Working Group. Available online: https://apwg.org (accessed on 10 October 2023).
3. APWG Phishing Activity Trends Report Q3. 2022. Available online: https://apwg.org/trendsreports (accessed on 10 October 2023).
4. Tinubu, C.O.; Falana, O.J.; Oluwumi, E.O.; Sodiya, A.S.; Rufai, S.A. PHISHGEM: A mobile game-based learning for phishing awareness. *J. Cyber Secur. Technol.* **2023**, *7*, 134–153. [CrossRef]
5. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
6. Zhou, Z.H. *Machine Learning*; Springer Nature: Berlin/Heidelberg, Germany, 2021.
7. Khonji, M.; Iraqi, Y.; Jones, A. Phishing detection: A literature survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 2091–2121. [CrossRef]
8. Mohammad, R.M.; Thabtah, F.; Mccluskey, L. Tutorial and critical analysis of phishing websites methods. *Comput. Sci. Rev.* **2015**, *17*, 1–24. [CrossRef]
9. Google Safe Browsing API. Available online: https://developers.google.com/safe-browsing/v4 (accessed on 10 October 2023).
10. Netcraft Anti-Phishing Toolbar. Available online: https://www.netcraft.com/apps (accessed on 10 October 2023).
11. Whittaker, C.; Ryner, B.; Nazif, M. Large-scale Automatic Classification of Phishing Pages. In Proceedings of the 17th Network & Distributed System Security Symposium, San Diego, CA, USA, 28 February–3 March 2010; pp. 1–14.
12. Jain, A.K.; Gupta, B.B. A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterp. Inf. Syst.* **2022**, *16*, 527–565. [CrossRef]
13. Qabajeh, I.; Thabtah, F.; Chiclana, F. A recent review of conventional vs. automated cyber-security anti-phishing techniques. *Comput. Sci. Rev.* **2018**, *29*, 44–55. [CrossRef]
14. Moore, T.; Clayton, R.; Stern, H. Temporal Correlations between Spam and Phishing Websites. In Proceedings of the 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats, Boston, MA, USA, 21 April 2009; pp. 1–8.
15. Thomas, K.; Grier, C.; Ma, J.; Paxson, V.; Song, D. Design and Evaluation of a Real-Time URL Spam Filtering Service. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 22–25 May 2011; pp. 447–462.
16. Gangavarapu, T.; Jaidhar, C.D.; Chanduka, B. Applicability of machine learning in spam and phishing email filtering: Review and approaches. *Artif. Intell. Rev.* **2020**, *53*, 5019–5081. [CrossRef]
17. Zhang, Y.; Hong, J.; Cranor, L. CANTINA: A Content Based Approach to Detecting Phishing Web Sites. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 639–648.
18. Wardman, B.; Stallings, T.; Warner, G.; Skjellum, A. High-Performance Content Based Phishing Attack Detection. In Proceedings of the eCrime Researchers Summit, San Diego, CA, USA, 7–9 November 2011; pp. 1–9.
19. Zhang, H.; Liu, G.; Chow, T.; Wenyin, L. Textual and visual content-based anti-phishing: A Bayesian approach. *IEEE Trans. Neural Netw.* **2011**, *22*, 1532–1546. [CrossRef]
20. Li, Y.; Xiao, R.; Feng, J.; Zhao, L. A semi-supervised learning approach for detection of phishing webpages. *Optik* **2013**, *124*, 6027–6033. [CrossRef]
21. Mao, J.; Tian, W.; Li, P.; Wei, T.; Liang, Z. Phishing-alarm: Robust and efficient phishing detection via page component similarity. *IEEE Access* **2017**, *5*, 17020–17030. [CrossRef]
22. Mohammad, R.M.; Thabtah, F.; Mccluskey, L. An Assessment of Features Related to Phishing Websites Using an Automated Technique. In Proceedings of the IEEE International Conference for Internet Technology and Secured Transactions, London, UK, 10–12 December 2012; pp. 492–497.
23. Mohammad, R.M.; Thabtah, F.; Mccluskey, L. Intelligent rule-based phishing websites classification. *IET Inf. Secur.* **2014**, *8*, 153–160. [CrossRef]
24. Mohammad, R.M.; Thabtah, F.; Mccluskey, L. Predicting phishing websites based on self-structuring neural network. *Neural Comput. Appl.* **2014**, *25*, 443–458. [CrossRef]
25. Basnet, R.B.; Sung, A.H.; Liu, Q. Rule-Based Phishing Attack Detection. In Proceedings of the International Conference on Security and Management, The World Congress in Computer Science, Computer Engineering and Applied Computing, London, UK, 18–21 July 2011.
26. Fette, I.; Sadeh, N.; Tomasic, A. Learning to Detect Phishing Emails. In Proceedings of the 16th ACM International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 649–656.
27. Aburrous, M.R.; Hossain, A.; Dahal, K.; Thabatah, F. Modelling Intelligent Phishing Detection System for E-banking Using Fuzzy Data Mining. In Proceedings of the IEEE International Conference on CyberWorlds, Washington, DC, USA, 7–11 September 2009; pp. 265–272.
28. Aburrous, M.R.; Hossain, A.; Dahal, K.; Thabatah, F. Intelligent phishing detection system for e-banking using fuzzy data mining. *Expert Syst. Appl.* **2010**, *37*, 7913–7921. [CrossRef]

29. Chiew, K.L.; Tan, C.L.; Wong, K.; Yong, K.S.; Tiong, W.K. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Inf. Sci.* **2019**, *484*, 153–166. [CrossRef]
30. Sahingoz, O.K.; Buber, E.; Demir, O.; Diri, B. Machine learning based phishing detection from URLs. *Expert Syst. Appl.* **2019**, *117*, 345–357. [CrossRef]
31. Xiao, X.; Zhang, D.; Hu, G.; Jiang, Y.; Xia, S. CNN–MHSA: A convolutional neural network and multi-head self-attention combined approach for detecting phishing websites. *Neural Netw.* **2020**, *125*, 303–312. [CrossRef] [PubMed]
32. Sonowal, G.; Kuppusamy, K.S. PhiDMA—A phishing detection model with multi-filter approach. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 99–112. [CrossRef]
33. Almomani, A.; Alauthman, M.; Shatnawi, M.T.; Alweshah, M.; Alrosan, A.; Alomoush, W.; Gupta, B.B. Phishing website detection with semantic features based on machine learning classifiers: A comparative study. *Int. J. Semant. Web Inf. Syst.* **2022**, *18*, 1–24. [CrossRef]
34. Bahaghighat, M.; Ghasemi, M.; Ozen, F. A high-accuracy phishing website detection method based on machine learning. *J. Inf. Secur. Appl.* **2023**, *77*, 103553. [CrossRef]
35. Adebowale, M.A.; Lwin, K.T.; Hossain, M.A. Intelligent phishing detection scheme using deep learning algorithms. *J. Enterp. Inf. Manag.* **2023**, *36*, 747–766. [CrossRef]
36. Basit, A.; Zafar, M.; Liu, X.; Javed, A.R.; Jalil, Z.; Kifayat, K. A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommun. Syst.* **2021**, *76*, 139–154. [CrossRef] [PubMed]
37. Abdillah, R.; Shukur, Z.; Mohd, M.; Murah, T.M.Z. Phishing classification techniques: A systematic literature review. *IEEE Access* **2022**, *10*, 41574–41591. [CrossRef]
38. Safi, A.; Singh, S. A systematic literature review on phishing website detection techniques. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *5*, 590–611. [CrossRef]
39. Kapan, S. Analysis of the Features Used in Detecting Phishing Attacks by Machine Learning. Master's Thesis, Eskisehir Osmangazi University, Eskisehir, Türkiye, 2021.
40. Kirda, E. Getting Under Alexa's Umbrella: Infiltration Attacks Against Internet Top Domain Lists. In Proceedings of the 22nd International Information Security Conference, New York, NY, USA, 16–18 September 2019.
41. PhishTank. Available online: https://www.phishtank.com (accessed on 10 October 2023).
42. Selenium Web Driver. Available online: https://www.selenium.dev (accessed on 10 October 2023).
43. Ratcliff, J.W.; Metzener, D. Pattern matching: The gestalt approach. *Dr. Dobb's J.* **1988**, *13*, 46.
44. Bal, S.; Sora Gunal, E. The impact of features and preprocessing on automatic text summarization. *Rom. J. Inf. Sci. Technol.* **2022**, *25*, 117–132.
45. Scikit-Learn Library. Available online: https://scikit-learn.org/stable/index.html (accessed on 10 October 2023).
46. UCI Machine Learning Repository, Phishing Websites Data Set. 2015. Available online: https://archive.ics.uci.edu/ml/datasets/phishing+websites (accessed on 10 October 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.