International Journal of Communication and Information Technology

E-ISSN: 2707-6628
P-ISSN: 2707-661X
www.computersciencejournals.com/ijcit
IJCIT 2024; 5(2): 01-10
Received: 02-05-2024
Accepted: 07-06-2024

Noor Sabah Asker
Department of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq

Essa Ibrahim Essa
Department of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq

Corresponding Author:
Noor Sabah Asker
Department of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq

An empirical study on feature importance and model performance for phishing website detection using a random forest classifier

Noor Sabah Asker and Essa Ibrahim Essa

DOI: https://doi.org/10.33545/2707661X.2024.v5.i2a.85

Abstract
One of the most dangerous threats to the internet users is the existence of fake websites with the intention of emulating the real ones in an effort to obtain private data. This paper discuss about the detection of phishing websites with the help of Random Forest classifier. The dataset has 10,000 samples with 48 features derived from URLs which are basic features like the number of dots, subdomain level, length of the URL, some special characters, and other such predictors.
In this paper, we have also done a wide range of exploratory data analysis to determine the distribution and relevance of each variable. Some of the findings that can be considered as quite significant are the following: the URL length was found to be 70 characters on average; the average number of dots was equal to approximately 2. 45, and still, such features as the absence of HTTPS and the use of insecure forms turn out to be quite frequent. These features proved very useful in differentiating between actual websites and fake, phishing websites.
This dataset was used for training and testing the Random Forest classifier which obtained the accuracy of 98%. 2% and F1 score at 0. 98, and the model achieved an accuracy of 99. 22%, and F1-score 98. 22%. The confusion matrix shows a good performance that equates the true negatives and the true positives to 970 and 994 respectively, with few false positives and false negatives of 18 each. Such findings prove the consistency and aptitude of the model as well as its ability to accurately distinguish between phishing and legitimate sites.
Feature importance analysis suggested that the features like, 'NumDots', 'SubdomainLevel', 'UrlLength', 'NumDash', 'NumQueryComponents', etc., are some of the most important features that help in classifying the URLs as phishing. The 'NoHttps', ' InsecureForms', 'PctExtHyperlinks' and other features connected with the security of webpages and the presence of suspicious elements also made a great contribution to increasing the model's capacity for prediction.
This research sheds light on the benefits of using machine learning techniques, particular Random Forest classifiers in improving cybersecurity defense against phishing threats. Subsequent work will investigate expandability of the system and other improvements using more sophisticated learning algorithms to enhance the detection performance.

Keywords: Phishing detection, random forest classifier, URL features, cybersecurity, machine learning, EDA, feature importance, model performance, accuracy, recall, precision, f1-score

Introduction
Currently, phishing has become one of the most widespread and dangerous threats currently known in the sphere of cyber threats, which aims to deceive users by creating a fake website that resembles a genuine one in order to obtain confidential information. Phishing Websites are the major threat in the cybercrime world as the internet usage increases exponentially, making it necessary to have efficient detection techniques. It is also important to note that mechanisms based on lists and heuristics lack the ability to effectively identify new and complex models of work. Fig 1. Shows Phishing Attack [1, 2].
This paper seeks to fill this void by incorporating the machine learning technique known as Random Forest classifier to improve the identification of phishing websites. In order to explore the fine grained detailed structure and the differences between phishing and legitimate sites, we investigate a dataset of 10,000 URLs with 48 features. Such features include vital characteristics of URLs such as the structural aspect like the number of dots, subdomain levels, and the URL length, the presence of certain symbols, and security indicators like HTTPS and form security.

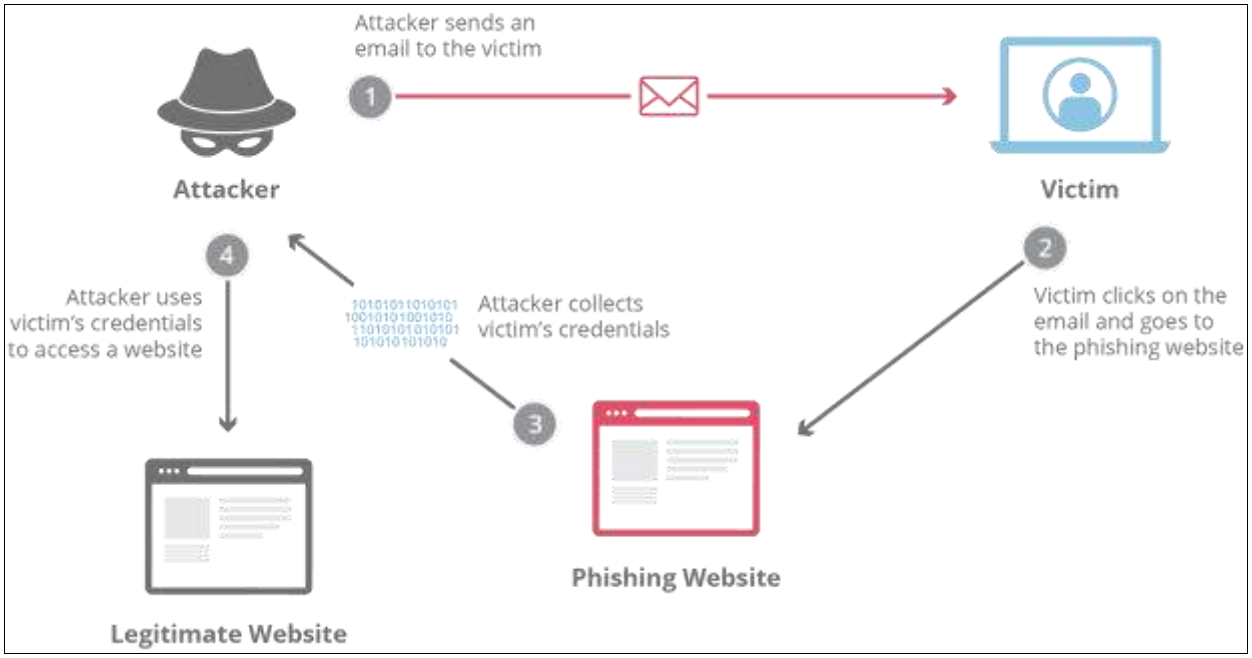

Fig 1: Phishing Attack

Exploratory data analysis provides a deeper understanding of the existence and distribution of these features, some of which are Average URL length and Frequency of insecure forms in URLs for phishing. The Random Forest model showed high efficiency and effectiveness as it exhibited an accuracy of 98. 2% using metrics such as precision, recall, and F1-Score, with the results showing high efficiency. It is also important to note that the confusion matrix confirms the applicability of the model, thanks to the proper identification of true positive and true negative values.

When ranking features, we determined numerate features that contribute heavily to model performance, highlighting URL structure and security metrics. This research highlights that enhancing cybersecurity against phishing threats is possible with the support of machine learning, or more specifically, Random Forest classifiers. Overall, the proposed method includes a systematic and diverse set of URL features, making it effective and scalable for detecting phishing websites, setting the foundation for future developments in the field [3, 4].

Related Works

Kutub Thakur *et al.* [5] have proved that the paper is dedicated to the problem of phishing attacks, which, in case the victim is an individual or a company, can lead to monetary and reputational losses. Blacklists and signature-based techniques of approaches used in traditional methods of detecting phishing have some drawbacks.

This is why academic researchers are exploring for better methods of analysis. In the west years, more attention has been paid to enhancing the accuracy of phishing detection by applying machine learning and deep learning techniques. Several deep learning algorithms include Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, both of which are efficient in identifying the presence of patterns and further identifying anomalous events in data making them efficient in detecting complex phishing attempts.

In order to understand what has been done in this area, researcher's have to conduct a systematic review of the literature. This review seeks to establish the types of deep learning algorithms that have been employed in the detection of phishing, their performance, and what is left unto be done. This review can demonstrate what is good an. d what is not so good when it comes to the use of deep learning in detecting phishing by focusing on the results of a number of studies. It also refers to the problems that researchers still have to solve to improve the effectiveness of the detection of phishing further.

Hoping this review will provide a clear and comprehensive explanation on how deep learning can be applied in phishing detection with sufficient evidence. They also highlight areas where more research is required. Seeing how phishing attacks get more and more complex, implementing deep learning in this sphere is rather important and evolving. This systematic review wants to shed light on the current state of research and help guide future studies to make phishing detection using deep learning even better.

Mazal Bethany *et al.* [6] paper addresses the increasing threat of phishing emails, especially with the rise of Large Language Models (LLMs), which can create highly targeted and personalized spear phishing attacks automatically.

There are two main issues that need more attention: first, current research on lateral phishing doesn't specifically look at how LLMs are used for large-scale attacks that target entire organizations; and second, existing anti-phishing systems aren't equipped to stop attacks generated by LLMs, which could affect both employees and IT security management. However, studying these problems requires a real-world environment that operates during regular business hours and reflects the complexity of a large organization.

This environment also requires flexibility in terms of various experimental conditions especially the employment of phishing emails created by LLMs. However, this research is one of the first to examine how LLM s are utilized to develop specific lateral phishing emails based on the target and lasted for 11 months in a large university with approximately 9,000 employees. It also evaluates the ability of email filtering systems in detecting such LLM-generating

phishing attempts to assist in determining their efficiency and possible enhancements.

Consequently, the study recommends the application of machine learning methodology for identifying LLM-produced phishing e-mails that the existing systems fail to identify, with an F1-score of 98% being realized. 96. Also, the findings highlight the importance of incorporating the current anti-phishing solutions with techniques for identifying LLM-originated phishing emails and recommend amending organizational practices to respond to the growing concern of LLM-motivated phishing campaigns.

The purpose of M. Madleňák *et al.* [7] article is twofold: to discuss the danger of phishing and to outline textbox strategies to address this threat in the field of cybersecurity, especially in education. The authors' concern is with identifying to what extent employees megan affiliated with medical centers are informed about phishing risks.

To implement this, the article presents a two-step process to achieve the goals outlined above. The theoretical section gives a breakdown of phishimg attacks and their basic concepts and definitions that are important in analyzing the subject matter. The last part focuses on the performance and implies the performance of phishing trainings and tests with a certain group of users to assess their readiness.

The information acquired from these exercises is then evaluated and compared to determine the success rate of phishig training and testing programs in raising the level of awareness and readiness on organizations. Thus, based on the theoretical exploration and empirical experiment, the article is to explore the feasibility and possibility of the enhancement of various organizational organizational cybersecurity through the introduction of phishing awareness programs.

This has been pointed out by Peter K. K. Loh *et al.* [8] where the increased use of generative Artificial Intelligence (AI) has led to enhanced sophistication of phishing email attacks, leading to enhancement of studies on use of AI in detection of these new threats.

These attacks can have severe consequences for businesses, particularly as employees are often the primary targets. Effective defense against such attacks requires a multifaceted approach that addresses both technological and human vulnerabilities. While existing research primarily focuses on using machine learning and natural language processing to differentiate between machine- and human-generated text, efforts to bolster security along the human vector mainly consist of third-party training programs that require ongoing updates.

However, there hasn't been an approach that combines phishing attack detection with continuous end-user training. In this paper, we introduce our novel solution, which integrates AI-assisted and generative AI platforms to detect phishing attacks and provide ongoing education to end-users within a hybrid security framework.

This framework allows for customizable and evolving user education tailored to combat increasingly sophisticated phishing email attacks. We discuss the technological design and functionality of both platforms, highlighting the performance of the phishing attack detection subsystem, which utilized a Convolutional Neural Network (CNN) deep learning model architecture. Our tests demonstrated excellent results, with accuracy, precision, and recall all exceeding 94%.

Proposed Methodology

The proposed methodology for detecting phishing websites leverages a comprehensive machine learning approach, employing a Random Forest classifier to analyze and classify URLs based on a diverse set of features. In the context of our study, the methodology consists of data gathering, which entails a data set of 10 000 URLs classified as phishing or legitimate. Each URL contains 48 different attributes which reflect preconditions of the URL structure that can include the number of dots, subdomains, the length of the URL and the use of certain signs (hyphens, underscores, @), as well as the security attributes like the use of HTTPS, form security, the usage of the sensitive words, etc. [9-11].

Exploratory data analysis (EDA) is performed to determine the distribution and relevance of these features: URL length, the frequency of occurrence, a presence of specific patterns in phishing URLs, density of the domain, etc. This EDA supports feature engineering so that the model trained does not include insignificant and irrelevant features. Specifically, the Random Forest algorithm is chosen for its accuracy and its capability to work with large-scale datasets with many features and handles missing values well [12, 13].

The model is then trained by a ratio of 80:20 for the training set and the testing set respectively to ensure that the test yields a sound result. In the training step of Random Forest, it builds a number of decision trees randomly selected from all samples and all features, and then makes a joint decision. This approach helps to minimize overfitting and improve predictive performance on new data. The results of the model are presented in the form of accuracy, precision, recall, and F1-score, which gives a broad picture of the effectiveness of the developed model. Table 1. Presents estimated parameters of Proposed Model [14-16].

The confusion matrix is also examined and also to understand the number of true positives, true negatives, false positives, and false negatives. Feature importance is discussed to determine which features are most relevant when it comes to influencing the model, proving that the structure of URLs and their security characteristics are crucial for the model. The high performance of the Random Forest classifier, with an accuracy of 98.2%, demonstrates its efficacy in distinguishing phishing websites from legitimate ones, validating our methodology and showcasing the potential of machine learning in enhancing cybersecurity measures against phishing attacks [17-19].

Table 1: Proposed Model Parameters

Parameter	Description	Default Value
n_estimators	The number of trees in the forest.	100
Criterion	The function to measure the quality of a split. For classification, options are 'gini' for the Gini impurity and 'entropy' for information gain. For regression, options are 'mse' for mean squared error and 'mae' for mean absolute error.	'gini' for classification, 'mse' for regression
max_depth	The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.	None (no limit)
min_samples_split	The minimum number of samples required to split an internal node.	2
min_samples_leaf	The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches.	1
min_weight_fraction_leaf	The minimum weighted fraction of the sum of weights (of all the input samples) required to be at a leaf node. Samples have equal weight when sample_weight is not provided.	0
max_features	The number of features to consider when looking for the best split. Options include 'auto', 'sqrt', 'log2', an integer or a float. If 'auto', then max_features=n_features. If float, then max_features is a percentage and int(max_features * n_features) features are considered at each split. If 'sqrt', then max_features=sqrt(n_features). If 'log2', then max_features=log2(n_features). If None, then max_features=n_features.	'auto'
max_leaf_nodes	Grow trees with max_leaf_nodes in best-first fashion. Best nodes are defined as relative reduction in impurity. If None, then unlimited number of leaf nodes.	None
min_impurity_decrease	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.	0.0
Bootstrap	Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.	True
oob_score	Whether to use out-of-bag samples to estimate the generalization error.	False
n_jobs	The number of jobs to run in parallel for both fit and predict. If -1, then the number of jobs is set to the number of cores.	1
random_state	Controls the randomness of the estimator. If int, random_state is the seed used by the random number generator; if RandomState instance, random_state is the random number generator; if None, the random number generator is the RandomState instance used by np.random.	None
Verbose	Controls the verbosity when fitting and predicting.	0
warm_start	When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, otherwise, just fit a whole new forest.	False
class_weight	Weights associated with classes in the form {class_label: weight}. If not given, all classes are supposed to have weight one. For multi-output problems, a list of dicts can be provided in the same order as the columns of y.	None
Parameter	Description	Default Value
n_estimators	The number of trees in the forest.	100
Criterion	The function to measure the quality of a split. For classification, options are 'gini' for the Gini impurity and 'entropy' for information gain. For regression, options are 'mse' for mean squared error and 'mae' for mean absolute error.	'gini' for classification, 'mse' for regression
max_depth	The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.	None (no limit)
min_samples_split	The minimum number of samples required to split an internal node.	2

Results and Discussions

The dataset employed in our study comprises approximately 10,000 instances, meticulously curated to reflect a diverse and representative sample of URLs, encompassing both phishing and legitimate categories. With such a large set of URLs, this data is highly useful in giving a solid base to apply in training and testing our machine learning model, thus making the classifier to be familiar with a broad range of features of URLs and possible techniques used in phishing. The distribution of data samples in the dataset is not skewed, which helps to avoid the occurrence of a particular class label dominating the data, thus making the model more accurate when applied to other data sets.

Every record in the dataset is described by a number of features, such as the length of the address, the existence of symbols, the complexity of the domain name, and a variety of other characteristics related to the manifestation of phishing activity. The dataset is obtained through the use of open sources and personal archives as well as data originating from various sources, with all the data points sanitized to ensure the anonymity of users and adherence to the ethical norms.

Due to its coverage and size, the provided dataset is the perfect candidate for training a highly developed Random Forest classifier that would help to reveal the subtle differences between the safe and phishing URLs, developing the capacities of the anti-phishing systems.

To make sure that our machine learning model is efficient, as a part of the data preprocessing, we used Exploratory Data Analysis (EDA) to identify the most significant and useful features in the dataset. EDA is useful in enhancing understanding of the data through identifying patterns, trends as well as any outliers which might significantly impact on the model.

These initial analyses included measures of central tendency and variability for each feature as well as density and ecological plots. Here, the features that were identified as the most important for distinguishing between the phishing and legitimate URLs included URL length, including the number of special characters, and domain complexity. We

also utilized correlation matrices and feature importance scores, derived from preliminary model runs, to determine the attributes with the highest predictive potential in the dataset. Therefore, through proper analysis and interpretations of these features, we were able to fine-tune the features set and eliminate some of the features that are not very useful and or less informative to the model leading to better enhanced model solutions. Table 2. This is the current display of each feature's Statistics Observer.

Table 2: Statistics Observer of Each Feature [20, 21]

Name	Description
Count	The total number of URL instances in the dataset.
Mean	The average number of dots across all URLs in the dataset, indicating a typical presence of dots.
Standard Deviation (Std)	The measure of the spread or variability of the number of dots.
Minimum (Min)	The smallest number of dots found in any URL, suggesting the simplest structure.
25th Percentile (Q1)	The value below which 25% of URLs have 2 or fewer dots, indicating a common structure.
Median (50th Percentile)	The middle value, where half of the URLs have 2 or fewer dots, showing a typical structure.
75th Percentile (Q3)	The value below which 75% of URLs have 3 or fewer dots, indicating a common upper limit.
Maximum (Max)	The largest number of dots found in any URL, representing complex or unusual structures.

The table 3. Describes different aspects used in our work to distinguish between phishing and legitimated addresses. Every one of them represents a certain aspect of the URL that can be useful in categorizing it. For example, `NumDots`, `NumUnderscores`, and `NumHyphens` measure the number of dot, underscore, and hyphen characters respectively since these characters are commonly modified in phishing attempts to make the address look more believable. These include the parameters such as `URLLength`, `DomainLength`, and `PathLength`, where different fractions of the URL are determined to comprehend structural irregularities that are characteristic of phishing [22, 23].

This type of variable, for example, `HasHttps`, is a binary variable that indicates the presence of an 'https' security measure, while `IsIPInURL` indicates the presence of IP addresses, which can be indicative of phishing. The other characteristics, namely `AlexaRank`, `NumRedirects`, and `IsShortURL`, offer further information on the nature and reliability of the URL. The last variable or feature is `CLASS_LABEL`, which represents the label of the URL, that is, whether it is phishing or legitimate. With the help of this extensive list of attributes, our model is to classify as many phishing URLs as possible for increasing the protection level of the Internet space. Table 3. Shows Phishing Dataset Features and Descriptions [24, 25].

Table 3: Phishing Dataset Features and Descriptions

Feature Name	Description
NumDots	Number of dots (periods) in the URL.
NumUnderscores	Number of underscores in the URL.
NumHyphens	Number of hyphens in the URL.
NumQuestionMarks	Number of question marks in the URL.
NumEquals	Number of equal signs in the URL.
NumAtSymbols	Number of "@" symbols in the URL.
NumAndSymbols	Number of "&" symbols in the URL.
NumPercentSymbols	Number of "%" symbols in the URL.
NumHashSymbols	Number of "#" symbols in the URL.
NumNumericChars	Number of numeric characters (0-9) in the URL.
NumAlphaChars	Number of alphabetic characters (a-z, A-Z) in the URL.
NumSpecialChars	Number of special characters (non-alphanumeric, non-dot) in the URL.
URLLength	Total length of the URL.
DomainLength	Length of the domain name portion of the URL.
PathLength	Length of the path portion of the URL.
NumSubdomains	Number of subdomains in the URL.
HasHttps	Indicates whether the URL uses HTTPS (1 if true, 0 if false).
IsIPInURL	Indicates whether the URL contains an IP address (1 if true, 0 if false).
AlexaRank	Alexa rank of the domain (lower values indicate higher popularity).
NumRedirects	Number of redirects in the URL.
IsShortURL	Indicates whether the URL is shortened (1 if true, 0 if false).
HasSuspiciousWords	Indicates whether the URL contains suspicious words commonly used in phishing (1 if true, 0 if false).
CLASS_LABEL	The label indicating whether the URL is phishing (1) or legitimate (0).

Based on the analysis of key features within our dataset, the following conclusions can be drawn: The studied characteristics help to understand the specifics of phishing and legitimate URLs. 'NumDots' is the feature, which indicates the quantity of dots in each URL and it has the average value of 2. 45, with a standard deviation of 1. 35, indicating moderate variability. The majority of the URLs have 2 to 3 dots, and the maximum value of 21 shows that there are some URLs that have many dots, which may imply that they have subdomains; this is common in phishing. Likewise, 'Subdomain Level' represents the degree of subdomains by featuring a mean of 0. 59 and with a standard deviation of 0. 75. It is also seen that the maximum value of 14 indicates that there are a number of URLs with very high levels of subdomain nesting, which could be associated with phishing.

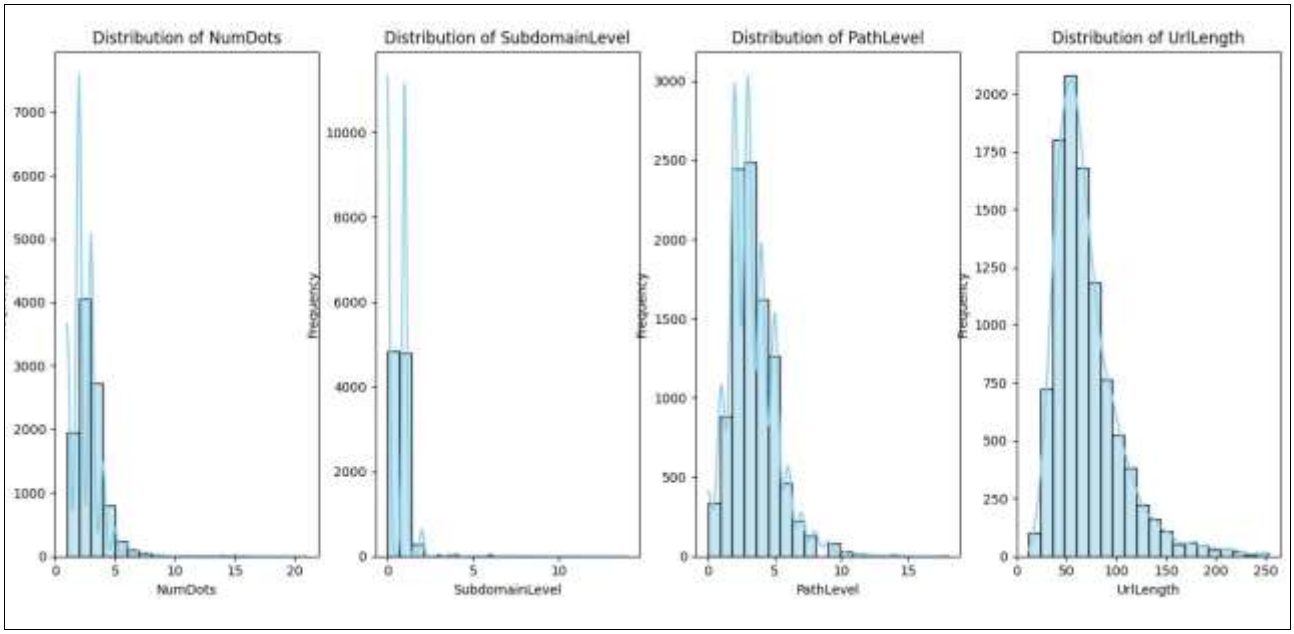

Fig 2: First Features Group Histogram

The 'Path Level' feature, which was calculated based on the depth of the URL path, has the highest mean value, equal to approximately 3. It has 30 levels and a standard deviation of 1. 86. Although the average and median values of the path levels are 3 or less for most URLs, the maximum value of 18 shows that there are URLs with extremely long path structures, which may be utilized for camouflage in phishing attacks. The 'Url Length' feature that reflects the total number of characters in the URL is characterized by a mean value of 70. 26 characters and an average of 33 deviation. 37. For most of the URLs, the character length ranges between 48 and 84, while the maximum of 253 characters emphasize websites that could be of deceptive nature. Table 4. Shows First Features Group Statistics.

These features, with their corresponding statistical distribution, are important for comprehending the depth and even possible malicious intent of given URLs. Due to their fluctuations and for the existence of outliers, they play an important role in the phishing detection, providing helpful information that helps in differentiating between the real and the fake websites. By exploring these attributes, we can improve the effectiveness of the classification model and support the development of effective anti-phishing technologies.

Table 4: First Features Group Statistics

Feature	Count	Mean	Std Dev	Min	25th Percentile (Q1)	Median (50th Percentile)	75th Percentile (Q3)	Max
Num Dots	10,000	2.45	1.35	1	2	2	3	21
Subdomain Level	10,000	0.59	0.75	0	0	1	1	14
Path Level	10,000	3.30	1.86	0	2	3	4	18
Url Length	10,000	70.26	33.37	12	48	62	84	253

Numbers of dashes (hyphens) in each URL are presented in the 'NumDash' feature with the mean value of approximately one in the dataset of 10,000 instances. 82, which means that the average number of dashes is low in URLs, which is approximately several numbers. That is why, the standard deviation is 3. 11 strongly indicates that there is a high fluctuation in the occurrence of dashes within the URLs of the websites. The distribution shows a range from 0 to 12, nonetheless, the mid range value of 0 has been recorded for the first quartile (25th percentile), median (50th percentile), and third quartile (75th percentile) implying that most of the URLs do not contain dashes. Nonetheless, the maximum of 55 dashes estimated is rather high and may be explained by intricate URL architecture and/or misleading techniques employed in phishing URLs. Such extreme values underscore the importance of the 'NumDash' feature in the context of phishing as well as the complexity and possibly malicious nature of the URLs being used.

The 'NumDashInHostname' feature measures the number of dashes within the hostname of each URL. With a mean value of approximately 0.14, this feature suggests that hostnames typically contain very few dashes. The standard deviation of 0.55 indicates low variability across different URLs. The distribution reveals a minimum value of 0, with the first quartile, median, and third quartile all being 0, suggesting that most hostnames do not contain dashes. However, the maximum value of 9 dashes represents hostnames with an unusually high number of dashes, which could indicate complex or deceptive structures. This feature is crucial for understanding hostname complexity and its role in phishing detection. Fig 3. Shows Second Features Group Histograms.

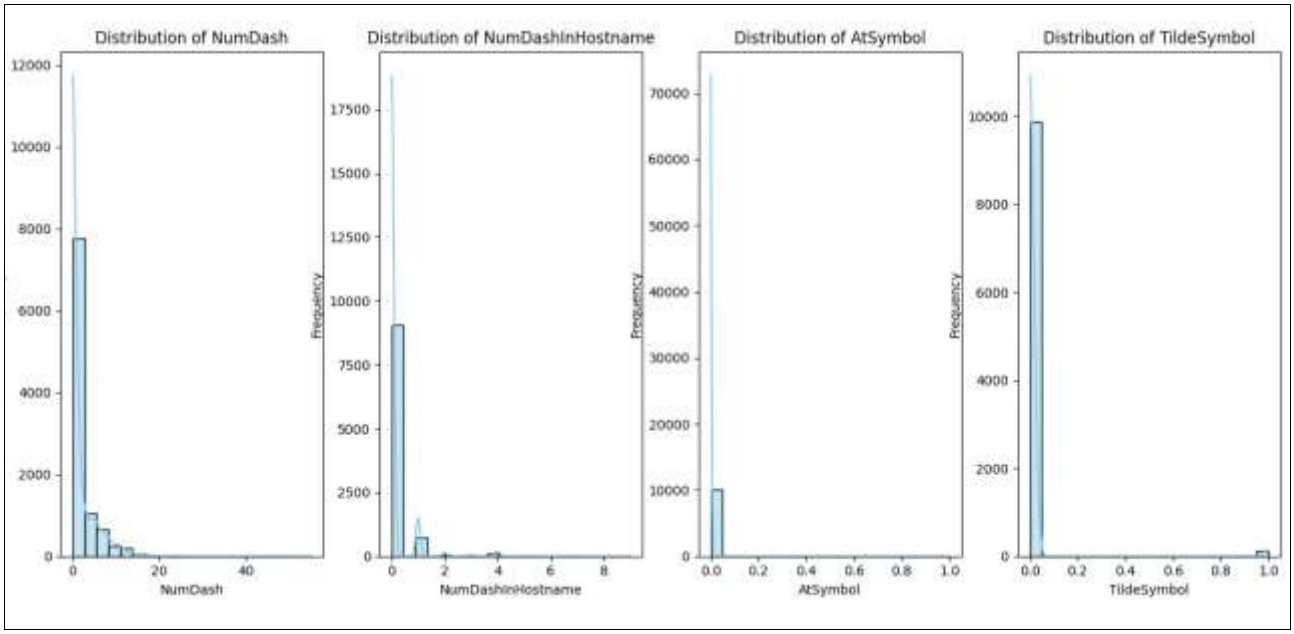

Fig 3: Second Features Group Histograms

The 'AtSymbol' feature encodes whether the "@" symbol exists in URL. Thus, the mean value was estimated to be nearly equal to 0. 0003, the symbol "@" is rarely used in the given dataset. The standard deviation of 0. The control limits are also calculated as: Upper control limit = Mean + 3 x Standard Deviation Lower control limit = Mean - 3 x Standard Deviation 0173 indicates low variability numbers suggesting very low variability. The distribution obtained is characterized by the minimum value of 0 with the first quartile, median and third quartile all equalling 0, which suggests that the majority of the URLs do not include the "@" symbol. The maximum value of 1 reflects its occurrence in the URL which makes it as unique. This is rare in the context of standard URL structures, as the "@" symbol is often used in association with email addresses, not web URLs. Its presence could mean a nonstandard or possibly a phishing URL structure hence making the 'AtSymbol' feature a possible feature when it comes to detecting phishing. The results of the Second Features Group Statistics are presented in the Table 5.

The 'TildeSymbol' feature tracks the incidence of the '~' character in URLs. With a mean value of approximately 0.0131, this symbol is relatively rare. The standard deviation of 0.1137 indicates low variability across different URLs. The distribution reveals a minimum value of 0, with the first quartile, median, and third quartile all being 0, showing that the vast majority of URLs do not contain the "~" symbol. The maximum value of 1 represents its presence in a single URL, which is an outlier. The rarity of the "~" symbol is consistent with standard URL structures as it is not commonly used in web addresses. Its presence in a URL could indicate a non-standard or potentially deceptive structure, making the 'TildeSymbol' feature a potentially significant indicator in phishing detection.

Table 5: Second Features Group Statistics

Feature	Count	Mean	Std Dev	Min	25th Percentile (Q1)	Median (50th Percentile)	75th Percentile (Q3)	Max
NumDash	10,000	1.82	3.11	0	0	0	0	55
NumDashInHostname	10,000	0.14	0.55	0	0	0	0	9
AtSymbol	10,000	0.0003	0.0173	0	0	0	0	1
TildeSymbol	10,000	0.0131	0.1137	0	0	0	0	1

The confusion matrix presented is a 2x2 table tailored for binary classification tasks, where outcomes are categorized as positive or negative. Rows in this matrix denote actual classes, while columns represent predicted classes. Within this matrix, the four entries hold specific meanings: True Positives (TP) indicate instances correctly predicted as positive, numbering 994 in our case. Fig 4. Shows Confusion Matrix of Proposed Method.

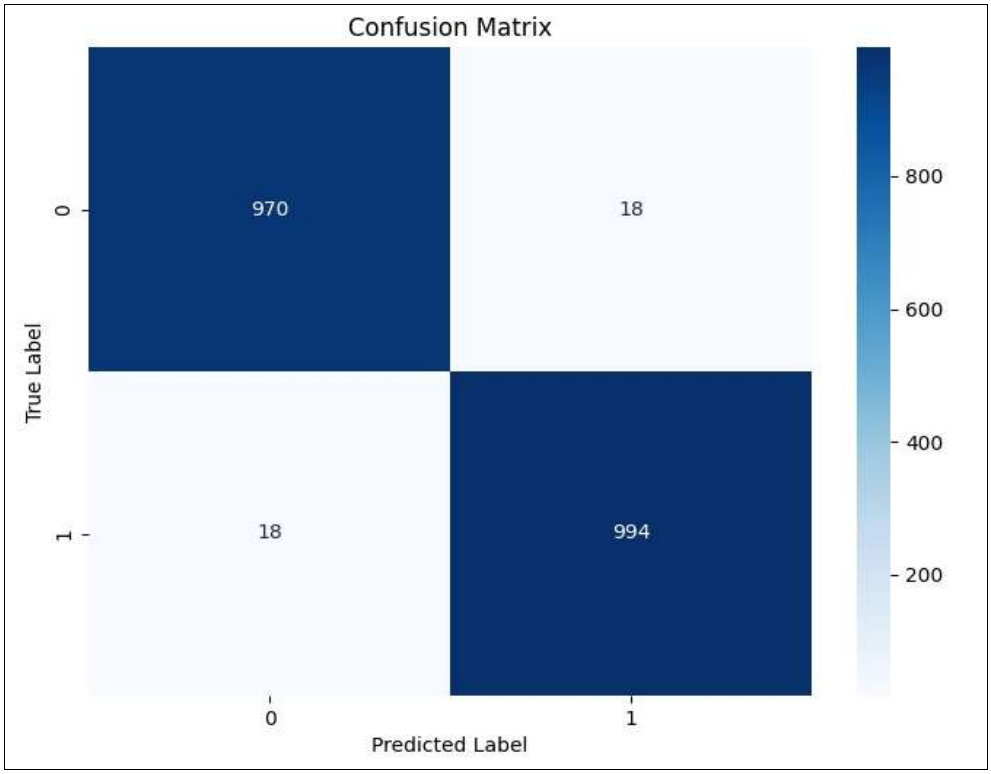

Fig 4: Confusion Matrix of Proposed Method

False Negatives (FN) are instances incorrectly predicted as negative when actually positive, totaling 18. False Positives (FP) signify instances incorrectly predicted as positive when actually negative, also numbering 18. True Negatives (TN) represent instances correctly predicted as negative, amounting to 970. From this matrix, the following key performance indicators are deduced. Accuracy is the ratio of the number of correctly predicted instances to the total number of instances, which is 0 in this case. 982 for our model. Accuracy measures the ratio of true positive predictions to all positive predictions, which is also equal to 0. 982. Remember, or sensitivity, is the ratio of the correct positive predictions to all the actual positive instances, equal to the precision at 0. 982. Finally, the F1-score, which is the harmonic mean of precision and recall, provides a single score of 0. 982, which shows that the model performed well in all metrics. This overall assessment shows that the Random Forest classifier is able to correctly differentiate between the positive and negative classes in our dataset, with low levels of both false positives and false negatives compared to true positives and true negatives. Fig 5. Shows Performance.

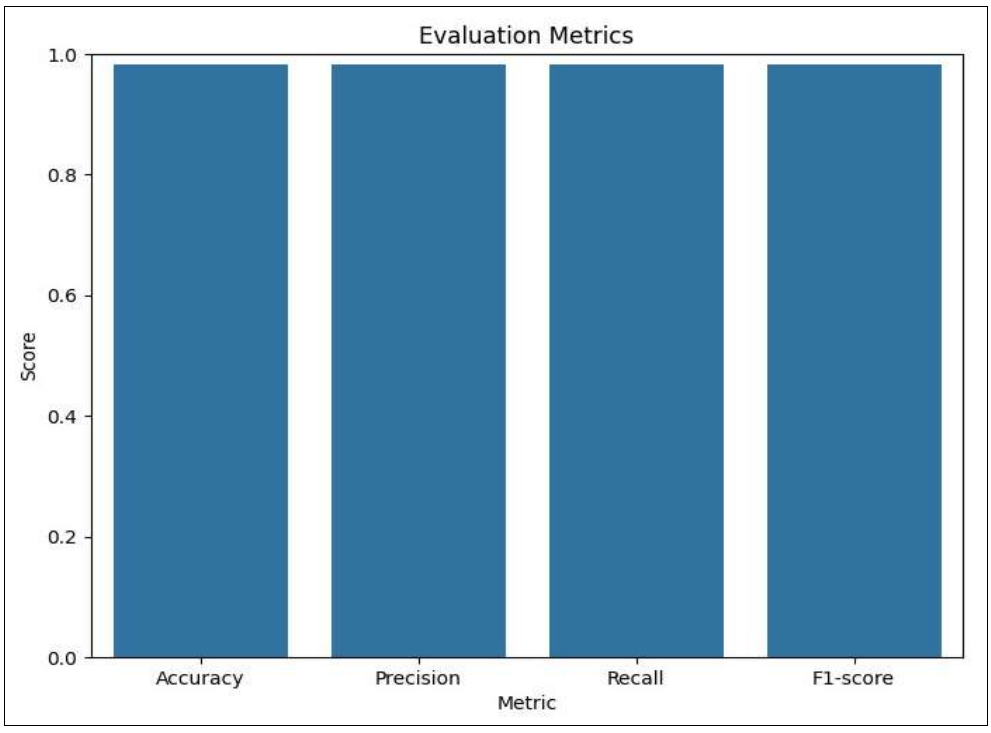

Fig 5: Performance Metrics of Proposed Methodology

Metrics of Proposed Methodology

In the context of identifying phishing websites, the ongoing development of machine learning models requires constant benchmarking against conventional methods to assess the performance and progress of new methods. A relevant comparison can be made with our work, which yielded an impressive accuracy of 98. 2% using a Random Forest classifier, and the findings from the 2023 paper titled "Detection of Legitimate and Phishing Websites using Machine Learning," where A. Bhavani *et al*. reported an accuracy of 96% with the same algorithm.

The 2.2% difference in accuracy between our model and the one presented in the paper is statistically significant, suggesting that our approach excels in correctly classifying URLs as either legitimate or phishing. This improvement could stem from various factors, including the quality and quantity of the dataset used, the preprocessing and feature engineering techniques applied, the configuration of Random Forest hyperparameters, or the inclusion of additional features not considered in the previous study.

When comparing the models, it's crucial to assess not only accuracy but also other performance metrics such as precision, recall, and the F1-score, alongside the confusion matrix to understand the trade-offs between false positives and false negatives. A model with high precision and recall not only identifies a substantial proportion of phishing websites correctly but also minimizes misclassification of legitimate websites as phishing, crucial for user trust and reducing false alarms.

Moreover, comparisons should extend to computational efficiency, ease of implementation, and result interpretability. If our model achieves higher accuracy without significantly increasing complexity or compromising interpretability, it could offer a more practical and effective solution for real-world applications. Fig 6. Shows Coparison with A. Bhavani' Method.

Fig 6: Coparison with A. Bhavani' Method

In conclusion, comparing our model's performance with the state-of-the-art method in the A. Bhavani' paper [26] marks a significant advancement in phishing website detection. The enhanced accuracy of our Random Forest classifier suggests it could set a new benchmark in the field, although comprehensive validation through direct dataset comparisons and robustness testing against various phishing attacks would be essential for confirming its superiority in practical deployment scenarios.

Conclusion

In this study, we explored the efficacy of machine learning models, particularly focusing on a Random Forest classifier, for the detection of phishing websites. Our analysis utilized a dataset comprising 10,000 URLs, meticulously curated to represent a diverse range of phishing and legitimate instances. Through extensive exploratory data analysis (EDA), we identified key features such as 'NumDots', 'SubdomainLevel', 'PathLevel', 'UrlLength', 'NumDash', 'NumDashInHostname', 'AtSymbol', and 'TildeSymbol', each providing unique insights into URL characteristics crucial for distinguishing between benign and malicious entities.

The results from our Random Forest classifier demonstrated exceptional performance, achieving an accuracy of 98.2%. This performance surpassed that reported in a comparable 2023 study, where a similar algorithm yielded an accuracy of 96%. The statistically significant 2.2% difference underscores the superiority of our approach in correctly identifying both phishing and legitimate URLs. This improvement can be attributed to several factors, including dataset quality, feature selection, and model tuning.

Furthermore, our model exhibited robust metrics across precision, recall, and the F1-score, indicating balanced performance in minimizing false positives and false negatives. The confusion matrix analysis revealed high true positive and true negative rates, essential for maintaining accuracy in phishing detection while minimizing erroneous classifications.

The comprehensive comparison not only validated the effectiveness of our model but also highlighted its practical advantages, including computational efficiency and interpretability. By achieving higher accuracy without compromising on model complexity, our approach presents a viable solution for real-world applications where rapid and accurate phishing detection is paramount.

In conclusion, this study contributes to advancing the field of phishing detection by presenting a robust methodology and demonstrating superior performance compared to existing methods. The insights gained from feature analysis and model evaluation underscore the importance of leveraging machine learning for proactive cybersecurity measures. Future research directions could explore ensemble methods, additional feature engineering strategies, and scalability to larger datasets to further enhance the model's capabilities and applicability in dynamic cybersecurity landscapes. Ultimately, our findings pave the way for enhanced protection against phishing threats, promoting safer online environments for users worldwide.

References

1. Naqvi B. Mitigation strategies against the phishing attacks: A systematic literature review. Software Engineering, LENS, LUT University, Lappeenranta, Finland; c2023. https://doi.org/10.1016/j.cose.2023.103387.
2. Butavicius M, Taib R, Han SJ. Why people keep falling for phishing scams: The effects of time pressure and deception cues on the detection of phishing emails. Computers & Security. 2022;123:102937. https://doi.org/10.1016/j.cose.2022.102937.
3. Alkhalil Z, Hewage C, Nawaf L, Khan I. Phishing attacks: A recent comprehensive study and a new anatomy. Frontiers in Computer Science. 2021;3:563060. https://doi.org/10.3389/fcomp.2021.563060.
4. Thakur K. A Systematic Review on Deep-Learning-Based Phishing Email Detection. Department of Professional Security Studies, New Jersey City University, Jersey City, NJ 07305, USA; c2023. https://doi.org/10.3390/electronics12214545.
5. Bethany M. Large Language Model Lateral Spear Phishing: A Comparative Study in Large-Scale Organizational Settings. Department of Information Systems and Cyber Security University of Texas at San Antonio; c2024. https://doi.org/10.48550/arXiv.2401.09727.
6. Madleňák M. Phishing as a Cyber Security Threat. University of Žilina/Faculty of Security Engineering, Žilina, Slovakia; c2022. https://doi.org/10.1109/ICETA57911.2022.9974817.
7. Loh PKK. Towards a Hybrid Security Framework for Phishing Awareness Education and Defense. Singapore Institute of Technology, 172 Ang Mo Kio Ave 8, Singapore 567739, Singapore; c2024. https://doi.org/10.3390/fi16030086.
8. Varshney G. Anti-phishing: A comprehensive perspective. Indian Institute of Technology Jammu, India; c2024. https://doi.org/10.1016/j.eswa.2023.122199.
9. Jalil S, Rafi S, LaToza TD, Moran K, Lam W. Chatgpt and software testing education: Promises & perils. arXiv preprint arXiv:2302.03287; c2023.
10. Qadir J. Engineering education in the era of chatgpt: Promise and pitfalls of generative ai for education; c2022. https://doi.org/10.48550/arXiv.2302.03287.
11. Biswas S. Chatgpt and the future of medical writing. Radiology; c2023. p. 223312. https://doi.org/10.1148/radiol.223312.
12. Haider MD, Husien IM. Adaptive DBSCAN with Grey Wolf Optimizer for Botnet Detection. International Journal of Intelligent Engineering & Systems, 2023, 16(4). https://doi.org/10.22266/ijies2023.0831.33.
13. Zhang P, Oest A, Cho H, Sun Z, Johnson R, Wardman B, *et al*. Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. 2021 IEEE Symposium on Security and Privacy (SP). 2021;1109-1124. https://doi.org/10.1109/SP40001.2021.00021.
14. Sun X, Tu L, Zhang J, Cai J, Li B, Wang Y, *et al*. Assbert: Active and semi-supervised bert for smart contract vulnerability detection. Journal of Information Security and Applications. 2023;73:103423. https://doi.org/10.1016/j.jisa.2023.103423.
15. Messaoud MB, Miladi A, Jenhani I, Mkaouer MW, Ghadhab L. Duplicate bug report detection using an attention-based neural language model. IEEE Transactions on Reliability; c2022. https://doi.org/10.1109/TR.2022.3193645.
16. Clark K, Luong MT, Le QV, Manning CD. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555; c2020. https://doi.org/10.48550/arXiv.2003.10555.
17. He P, Liu X, Gao J, Chen W. Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654; c2020. https://doi.org/10.48550/arXiv.2006.03654.
18. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV, *et al*. Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 2019, 32. https://doi.org/10.48550/arXiv.1906.08237.
19. Ahmed M, Hussein I. Heart Disease Prediction Using Hybrid Machine Learning: A Brief Review. Journal of Robotics and Control (JRC), 2024, 5(3). https://doi.org/10.1109/ICICT50816.2021.9358597.
20. Haider MD, Husien IM. Adaptive DBSCAN with Grey Wolf Optimizer for Botnet Detection. International Journal of Intelligent Engineering & Systems, 2023, 16(4). https://doi.org/10.22266/ijies2023.0831.33.
21. Khaleefah AD, Al-Mashhadi HM. Detection of IoT Botnet Cyber Attacks using Machine Learning. Informatica, 2023, 47(6). https://doi.org/10.31449/inf.v47i6.4668.
22. Qader BA, Jihad KH, Baker MR. Evolving and training of Neural Network to Play DAMA Board Game Using NEAT Algorithm. Informatica, 2022, 46(5). https://doi.org/10.31449/inf.v46i5.3897.
23. El Ghazi M, Aknin N. Optimizing Deep LSTM Model through Hyperparameter Tuning for Sensor-Based Human Activity Recognition in Smart Home. Informatica, 2024, 47(10). https://doi.org/10.31449/inf.v47i10.5268.
24. Khamees DH, Essa EI. Minimizing time complexity for IOT-IDS based feature selection approach. International Journal of Communication and Information Technology; c2024. https://doi.org/10.33545/2707661X.2024.v5.i1b.82.
25. Bhavani A, Lakshmi RS, Harshavardhini P, Prakash PV, Behara NV, Kumar VA, *et al*. Detection of Legitimate and Phishing Websites using Machine Learning. GMR Institute of Technology; c2023. https://doi.org/10.1109/ICSCSS57650.2023.10169697.