Contents lists available at Dergipark
Journal of Scientific Reports-A

journal homepage: *https://dergipark.org.tr/tr/pub/jsr-a*

E-ISSN: 2687-6167 ***Number 56, March 2024***

RESEARCH ARTICLE

Receive Date: 19.02.2024 *Accepted Date: 18.03.2024*

Data correlation matrix-based spam URL detection using machine learning algorithms

Funda Akar[a,*]

[a]*Department of Computer Engineering, Erzincan Binali Yıldırım University, Erzincan 24002, Turkey, ORCID: 0000-0001-9376-8710*

Abstract

In recent years, the widespread availability of internet access has brought both advantages and disadvantages. Users now enjoy numerous benefits, including unlimited access to vast amounts of information and seamless communication with others. However, this accessibility also exposes users to various threats, including malicious software and deceptive practices, leading to victimization of many individuals. Common issues encountered include spam emails, fake websites, and phishing attempts. Given the essential nature of internet usage in contemporary society, the development of systems to protect users from such malicious activities has become imperative. Accordingly, this study utilized eight prominent machine learning algorithms to identify spam URLs using a large dataset. Since the dataset only contained URL information and spam classification, additional feature extractions such as URL length and the number of digits were necessary. The inclusion of such features enhances decision-making processes within the framework of machine learning, resulting in more efficient detection. As the effectiveness of feature extraction significantly impacts the results of the methods, the study initially conducted feature extraction and trained models based on the weight of features. This paper proposes a data correlated matrix approach for spam URL detection using machine learning algorithms. The distinctive aspect of this study lies in the feature extraction process applied to the dataset, aimed at discerning the most impactful features, and subsequently training models while considering the weighting of these features. The entire dataset was used without any reduction in data. Experimental findings indicate that tree-based machine learning algorithms yield superior results. Among all applied methods, the Random Forest approach achieved the highest success rate, with a detection rate of 96.33% for the non-spam class. Additionally, a combined and weighted calculation method yielded an accuracy of 94.16% for both spam and non-spam data.

Keywords: Classification; Machine Learning; Spam Detection; Tree-based Algorithms.

1. Introduction

The proliferation of the internet since the early 2000s has led to its widespread adoption among nearly all individuals in contemporary society. Through internet usage, users have access to vast knowledge and opportunities, with many relying on digital platforms for their daily activities, thus contributing to the increased utilization of the internet. However, this widespread adoption has also exposed users to various risks and harms, particularly through unconscious internet usage and the prevalence of fraudulent schemes. Many people have fallen victim to things like social engineering, fake websites, and email phishing. To reduce these risks, it is crucial to take preventive measures against threats such as spam emails and malicious URLs. Additionally, the increasing dependence on online transactions and the prevalence of technology, especially IoT technologies, in daily life increasingly raises issues related to web security and data protection. The use of malicious URL links causes a serious risk as it is one of the most common types of cyber-attacks and can allow unauthorized access to personal data [1].

Hence, this study was guided to define and solve the spam URL problem by using machine learning algorithms to analyze and classify internet content. In order to increase the efficacy of spam URL detection, various methods have been investigated to obtain the best results using different machine learning algorithms. The most important difference that distinguishes this study from others is the extraction of features from the data set and the training of models by taking into account the weight levels of these features.

Various studies have been conducted employing diverse methodologies targeting spam URLs, spam emails, and spam bots [2]–[8]. Numerous approaches have been employed for URL classification, culminating in the development of decision support systems and relevant applications. It is imperative to employ rigorous filtering mechanisms for URL classification, encompassing various web page advertisements and activities deemed hazardous [9]. Chen et al. utilize extensive URL databases or blacklists to detect malicious or phishing websites [10]. Frequently, attacks generate spam URLs resembling legitimate corporate websites through tactics such as social engineering. Malware is designed to disseminate through URLs, infecting computers and propagating the authors' software [11]. Frequently, attacks generate spam URLs resembling legitimate corporate websites through tactics such as social engineering. Malware is designed to disseminate through URLs, infecting computers and propagating the authors' software.

A relevant study analyzed the differences between benign and phishing URLs utilizing attributes such as URL and domain length. Through dissecting the structure of phishing URLs, domain registration, and the hosting machinery, the authors illustrated that domain names employed for phishing endeavors exhibit disparate lengths and locations [12]. Building upon domain behavior on phishing platforms, Ma et al. devised a model capable of identifying suspicious URLs with 99% accuracy, leveraging verbal and blacklisted host-based features [13].

Kwon et.al. proposed a different method for detecting spam URLs. Authors focused to domain independence, competitive robustness, and semi-supervised perception. They applied machine learning techniques in their investigation, which produced 96% recall and 70% accuracy results and obtained these results using Decision Tree, Logistic Regression and Support Vector Machine [14]. Takata et al. attempted to identify a direct download attack type by analyzing it as having the relevant code in the redirection [15]. Research was carried out by Almeida and Westphall to identify harmful URLs. The URLs constructed with identity theft in mind were found in the study. Generally, they stated that they were detected with success rates between 73-97% on URLs created for phishing purposes [16]. Manyumwa et.al. made multi-class classification on malicious URL detection. DMOZ, PhishTank, URLhaus, WEBSPAM datasets and XGBoost, Adaboost, LightGBM, CatBoost were used as machine learning methods in their study [11]. Rao and Pais proposed a different method for the detection of URLs developed for identity theft and tried to find the best result with many machine learning algorithms. In their study, they found the highest result with 99.31% accuracy [17]. Raj and Kang used many machine learning methods for spam URL detection in their study and the highest test accuracies were made with XGBoost and 87% success was found [18]. Another proposed system offers a dual-layered detection mechanism. Initially, URLs are categorized as either benign or malicious using a binary classifier. Subsequently, URL classes are further classified into five categories based on their features: benign, spam, phishing, malware, and defacement. In particular, we present findings on four ensemble learning methodologies, namely the ensemble of bagging trees (En_Bag) approach, the ensemble of k-nearest neighbor (En_kNN) approach, the ensemble of boosted decision trees (En_Bos) approach, and the ensemble of subspace discriminator (En_Dsc) approach. They also compare their En_Bag model with state-of-the-art solutions, demonstrating its superiority in both binary classification and multi-classification tasks, achieving accuracy rates of 99.3% and 97.92%, respectively [19].

Many studies have been done on URL spam detection before. This paper proposes a data correlated matrix approach for spam URL detection using machine learning algorithms. For this purpose, effective operations were carried out on the dataset to increase the success. First, a dataset containing large-scale data was provided in order to carry out the study. The dataset contains about 150.000 spam or non-spam URLs. Using feature engineering, the best way to approach the decision-making process was established. The features were systematically eliminated, considering their potential significance in spam detection within URLs. A correlation matrix was employed to identify the most influential features for learning. Utilizing this insight, an extensive classification process was undertaken with numerous machine learning algorithms to determine the optimal outcome. Finally, the findings were thoroughly discussed, providing insights into the efficacy of features in identifying spam URLs, thus culminating the study.

In this study, no new dataset was generated. Instead, models were trained by assigning weight values solely based on the significance of features, without any reduction or modification to the dataset; the entire dataset was utilized in its original form.

2. Material and methods

In order to detect spam URLs, first of all, it is necessary to know what the URL is and what parts it consists of (Fig. 1). URLs usually consist of domain and subdomain. Rules defined by a protocol are used to transfer data to the other party. URLs contain many numbers and letters and there can be redirects within the website in a structure similar to a folder within path.

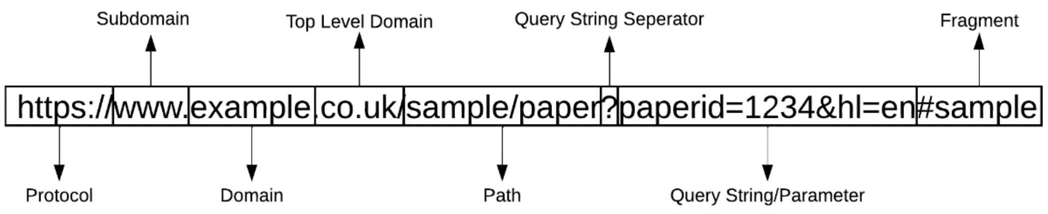

Fig 1. An example showing URL parts.

Data set from Kaggle was used in the study which contains a very large amount of data [20]. In this way, it will be better to test the transactions made. It is prepared to be given to machine learning algorithms in the most appropriate way by feature engineering with the acquired data. Data are based on general study opinion. Based on a general justification utilized in the literature search, these discrimination rates are split into 70% train and 30% test groups. Later, machine learning methods were made ready for implementation. The application is terminated by making spam URL classification. The flow chart of the processes performed in the study is shown in Fig. 2.

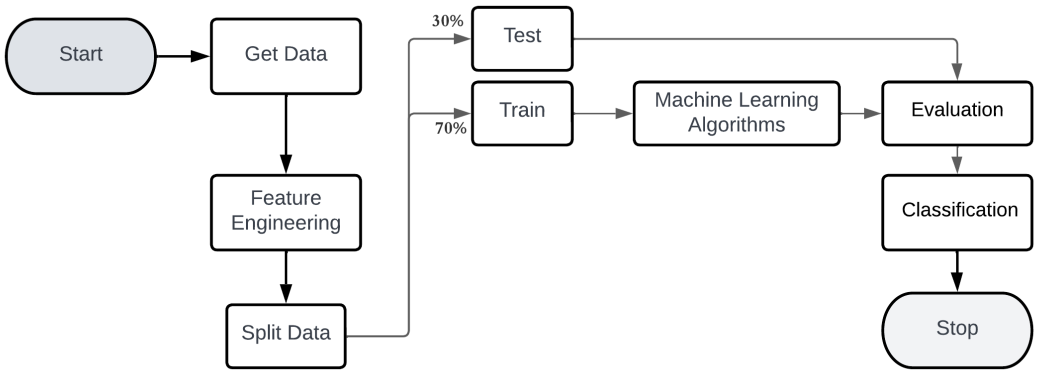

Fig. 2. Flowchart of the work.

2.1. Structure

The dataset, which has already labeled data, contains approximately 150.000 URLs [20]. Fig. 3 shows the spam and non-spam distribution of the URLs. Only URLs and their tags are included in the dataset. To determine whether it is spam, some pre-processing is required for this reason. 70% of the dataset is used for training and 30% for testing.

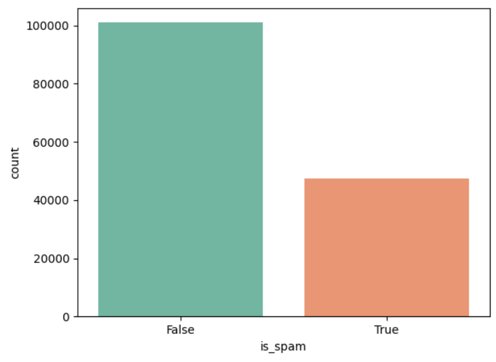

Fig. 3. Spam distribution of the dataset.

There are nearly 100.000 non-spam and 50.000 spam URLs in the dataset. This will result in varying weights for the values provided. The study utilized labeled values and did not use any missing data. Furthermore, the impact of weights on the outcomes remained unchanged. The outcomes of machine learning were included into the outcomes in the manner in which they were created.

2.2. Feature engineering

Since there is no information other than URL and label in the dataset, an extra feature must be added. Therefore, it is necessary to add data over numeric, special characters and letters in the dataset. Better results are expected after this procedure. Satisfactory results were found in the study, especially thanks to feature engineering part. Since the number of letters is very important in the transactions, first the number of letters and the length of the URL have been added. Special characters are also added one by one because there are many pieces. Since numerical data are also seen to be important, they are also indicated in the Table I.

Table 1. All features and descriptions in the dataset.

Features	Feature Description
length_url	Specifies the length of the URL.
num_digits	Shows the total number in the URL
num_letters	Total number of letters in URL
num_words	Total number of words in URL
with_https	Number of "https" in the URL
num_hashtag	Number of "#" in the URL
num_?	Number of "?" in the URL
num_/	Number of "/" in the URL
num_!	Number of "!" in the URL
num_-	Number of "-" in the URL
num_.	Number of "." in the URL
num_*	Number of "*" in the URL
num__	Number of "_" in the URL
num_%	Number of "%" in the URL
num_&	Number of "&" in the URL
inc_www	Whether there is “www” in the URL or not
inc_subscribe	Whether there is “subscribe” in the URL or not
inc_com	Whether there is “com” in the URL or not
inc_net	Whether there is “net” in the URL or not
inc_edu	Whether there is “edu” in the URL or not
inc_org	Whether there is “org” in the URL or not

Since machine learning algorithms were used many times while determining the features, all kinds of examinations were made. Every possible feature has been tried to be added until satisfactory results are obtained. It has been observed that the procedures performed directly affect the results. In this way, it has been tried to underline these important features while performing the classification processes. The degree of effect of the results was determined using the correlation matrix. The correlation matrix depicts the intra-grid structure of variables based on the correlation effect coefficient, which was a value between [-1, 1]; whichever number was closer to 1 resulted in more correlations between data components [21]. In this way, the effect of the added features can be seen clearly. Although some added features remained ineffective, they were still not removed. Because it has been seen that it has a small effect on performance. That's why every added feature is given to machine learning algorithms.

In Fig. 4 all features are given according to the feature importance and correlation matrix is given in Fig. 5. The length of the URL, the number of letters, the number of digits and the number of "-" are the four features having the most impact, as shown in figures.

Except for feature weighting, no data manipulation has occurred. No operations, such as addition or removal of data, have been conducted that could impact the outcomes. Any modifications to the dataset would inevitably influence the results. Machine learning methods were applied while strictly adhering to the original dataset. The considerable size disparity between non-spam and spam data underscores the significance of accurately detecting non-spam instances, given its twice as large representation, which substantially affects the success rates. Additionally, the augmentation of features has introduced supplementary inputs, crucial for providing a more comprehensive benchmark for machine learning evaluation. The procedures were designed with this consideration in mind, aiming to compare the effects of features on the results, thereby discerning the efficacy of the added features for achieving superior outcomes.

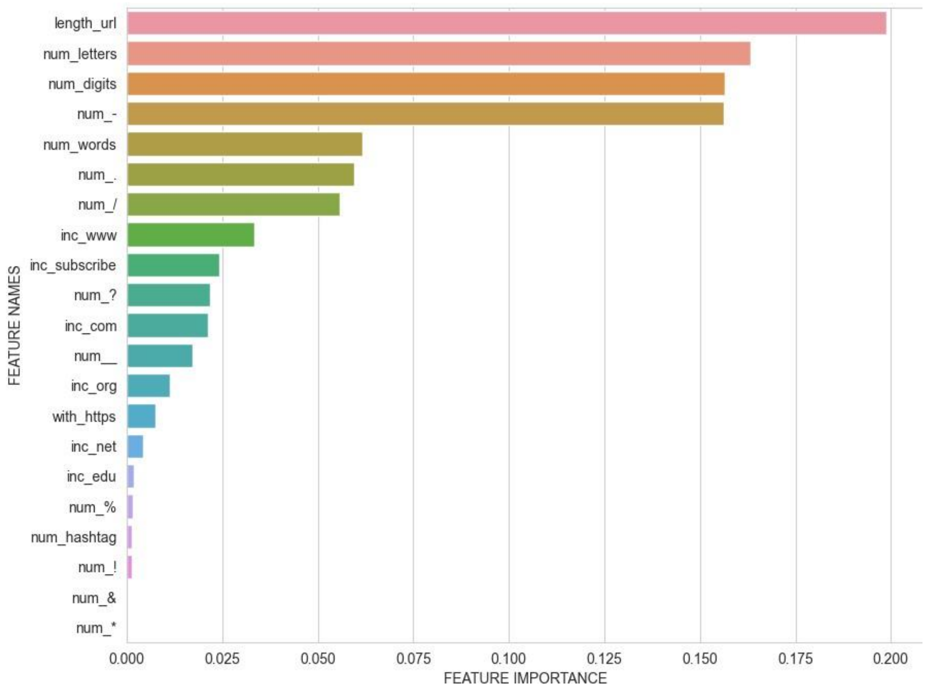

Fig. 4. Feature importance of dataset.

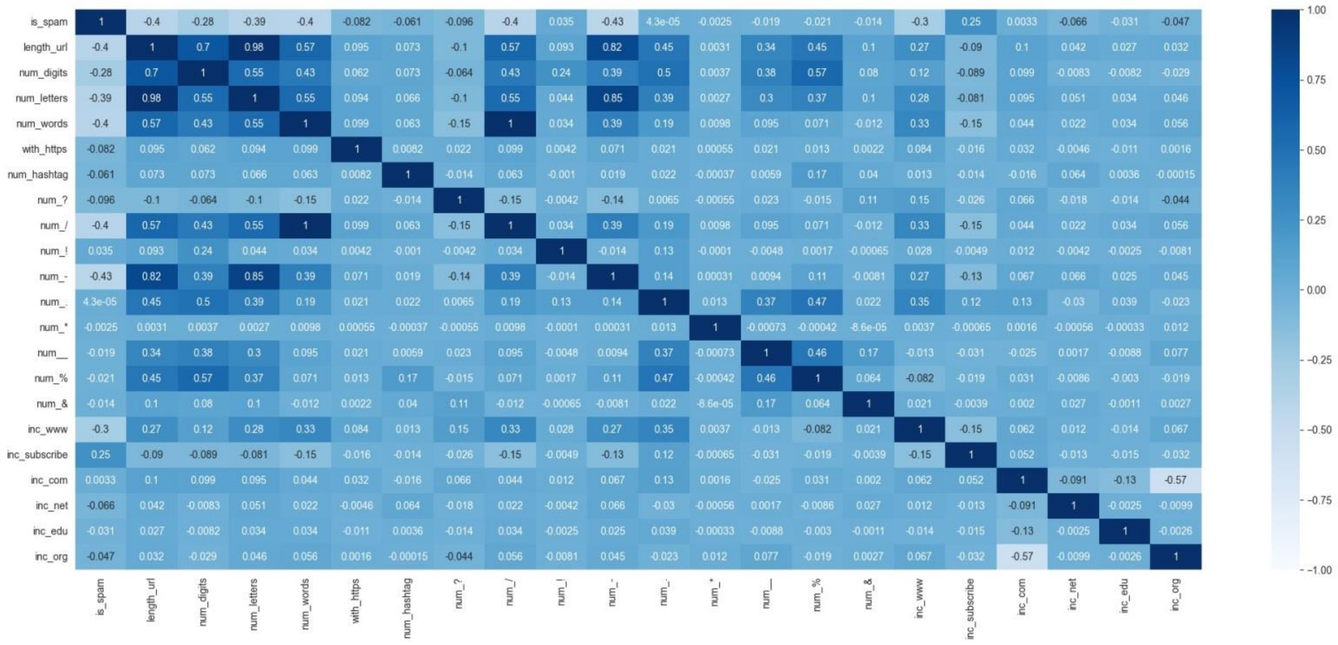
Fig. 5. Correlation matrix.

Correlation matrix was created when only features were detected, and additions were made. This clarifies which feature can match which feature. In addition to "https" and "http" has also been added in the study. However, "http" was excluded because it was not correlated with any value. If it is not removed, it will cause inaccuracies in the results. It is possible to make such a comparison for each added feature. It appears to be comparable to how significant features are in the correlation matrix. It appears that the correlational structure, or likeness to one another, increases with a higher dark blue degree. The conclusion to be made here is that this structure determines the quality of the added features. So, it is clearly seen which additional features will affect the success. Furthermore, features such as exclamation points and hashtags included in the URL have little effect. Nevertheless, the features in this structure were not removed to avoid difficulties in decision making. Because it was seen that when it was eliminated, the success rate of the experimental findings reduced. Therefore, it is important to add features by considering their advantages over each other.

2.3. Machine learning algorithms

The scope of machine learning algorithms has been expanded to allow for a more thorough evaluation in this study. While evaluating machine learning in the study, both class-based achievements and weighted average outcomes were provided. Machine learning algorithms were completed with default values. If the algorithms are executed using parameters other than the default ones, better results are likely to be attained. The key purpose here is to determine which machine learning approach is the most likely to solve this problem. Because several machine learning algorithms are employed in the study, a comprehensive explanation of each approach is not provided here. The study's major goal is not to explain the methods, but to compare the success of the methods with each other. Relevant machine learning approaches have been investigated, and detailed descriptions of the methods may be found in the publications listed below. Machine learning algorithms used in the study:

- Logistic Regression (LR) [22]–[24]: Commonly used for classification problems, especially effective in binary classification tasks.
- Decision Tree Classifier (DTC) [25]–[27]: Widely applicable in various domains, used for both classification and regression tasks. Decision trees find applications in data mining, medicine, finance, and marketing.
- Random Forest Classifier (RFC) [28]–[30]: Used in a variety of application areas. Particularly effective for classification and regression problems with large datasets.
- Naive Bayes (NB) [31]–[33]: Especially popular in text classification tasks such as spam filtering and sentiment analysis. Also applicable to multi-class classification problems.

- K- Nearest Neighbor (KNN) [34], [35]: Used for classification and regression tasks based on spatial similarities. Applications include medical diagnosis in healthcare, customer segmentation in marketing, and more.
- XGBoost (XGB) [36], [37]: Widely applicable and particularly preferred for classification and regression tasks with high-dimensional datasets. Applications span across finance, medicine, natural language processing (NLP), and more.
- AdaBoost Classifier (ABC) [38]–[40]: Constructs a strong classifier by combining weak learners. Often used in areas like face recognition and speech classification.
- Gradient Boosting Classifier (GBC) [41], [42]: Based on gradient boosting principles and commonly used for classification and regression problems. Has a broad range of applications, including web page ranking and medical diagnosis.

Each algorithm has its strengths and weaknesses, so choosing the most suitable one depends on the context of the problem and characteristics of the dataset. Since many articles on how the machine learning methods employed work provide formulas and usage reasoning, only the scanned publications are shown in this study by citing the investigated materials. Because all of these strategies are included in the study and a multi-class dataset is used, both class-based and weighted overall performance results are provided.

2.4. Evaluation metrics

We require certain performance indicators to assess the efficacy of the machine learning techniques used in the study. It is possible to compare which approach performs better than which method in this way. The technique to be used may not always be the most accurate. There are several problem and outcome-oriented algorithms available.
The following metrics were utilized in the study:

$$Accuracy = \frac{TP + TN}{TP+FP+TN+FN} \tag{1}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

Accuracy (1) is defined as the quotient of correctly predicted results divided by total results, signifying overall performance. The F1-score represents the harmonic mean of precision and recall levels (2). Precision (3) represents the proportion of true positive results to other positive outcomes. The recall (4) value is defined as the ratio of true positive values to true negative and true positive values [41]. A Confusion Matrix must be created in order to calculate all of these parameters.

3. Results and discussion

As previously indicated, eight distinct machine learning methodologies were employed in the study. These methods yield a two-class result due to two labels in the dataset as spam or not spam. The outcomes are presented alongside the confusion matrix and ROC curves for comprehensive evaluation. The ROC curve can give us information about how accurate the operation is and how far we are from a successful method. Accuracy, F1-Score, Recall and Precision values are also given as in Section 2.4 for a better evaluation of the study. This will make it easier to compare the study to others and allow for a better evaluation of its effectiveness.

The results of Logistic Regression are presented in Fig. 6 and Table 2. It seems clear that non-spam data is better detected. According to the results obtained, it seems that Logistic Regression is not suitable for solving this problem. Although the ROC curve is not even close to 1, this machine learning method can be useful considering its speed.

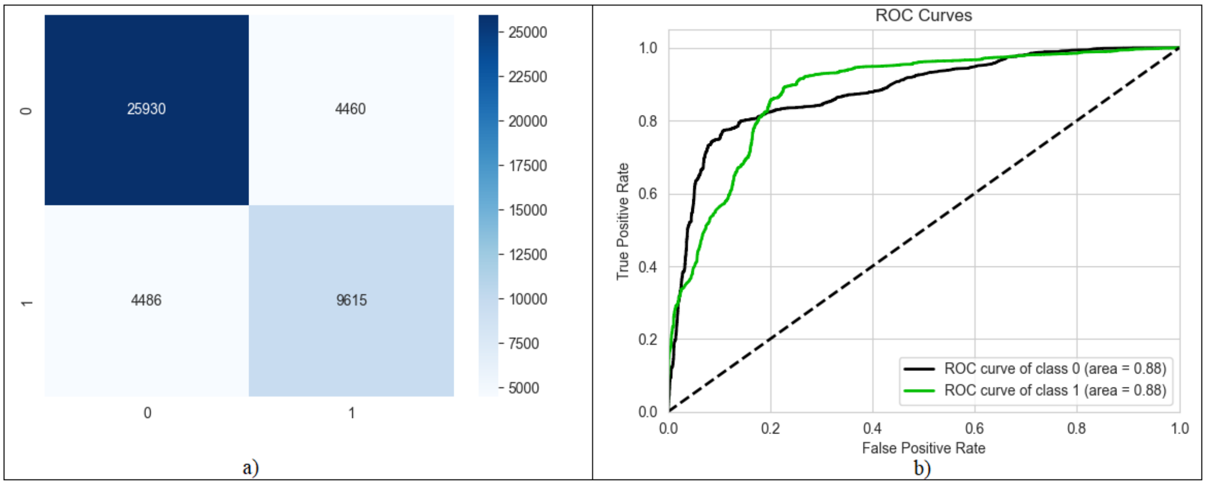

Fig. 6. (a) Confusion matrix; (b) ROC curve values of logistic regression.

Table 2. Evaluation metrics values of logistic regression.

Classes	Precision (%)	Recall (%)	F1-Score (%)	Support
Not spam (class 0)	85.25	85.32	85.29	30390
Spam (class 1)	68.31	68.19	68.25	14101

The results of the Decision Tree are presented in Fig. 7 and Table 3. It appears that non-spam data is more effectively detected. It is seen that Decision Tree generates a good result among the results obtained. ROC curve is very close to 1. As can be seen, the operations performed are very close to the truth. This method is very convenient to use because it is fast.

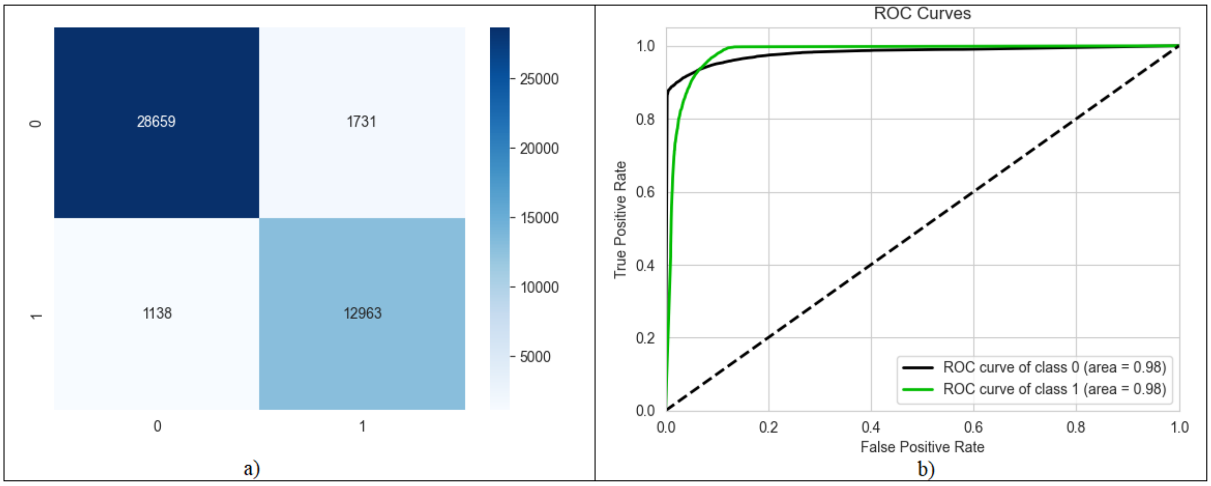

Fig 7. (a) Confusion matrix; (b) ROC curve values of decision tree.

Table 3. Evaluation metrics values of decision tree.

Classes	Precision (%)	Recall (%)	F1-Score (%)	Support
Not spam (class 0)	96.18	94.30	95.23	30390
Spam (class 1)	88.22	91.93	90.04	14101

Fig. 8 and Table 4 show the Random Forest's outcomes. Non-spam data appears to be better detected. Random Forest is the algorithm that produces the best outcomes out of all the results obtained. Because it is relatively slow, there may be a temporal contraction in larger data entries. As can be seen, ROC curve is very close to 1, the operations performed are very close to the truth indicating that the procedures were very accurate. This method is highly convenient to use because it is fast. In addition to its high results, the difference between the two classes was found smallest in this method among all methods. By cross validating the parameters of this method, which is often employed in studies, good results can be obtained. A sufficient number of parameters have been evaluated in the study and the best results have been tried to be obtained. Moreover, it is a frequently preferred algorithm due to the widespread usage of this method and the convenience of transportation.

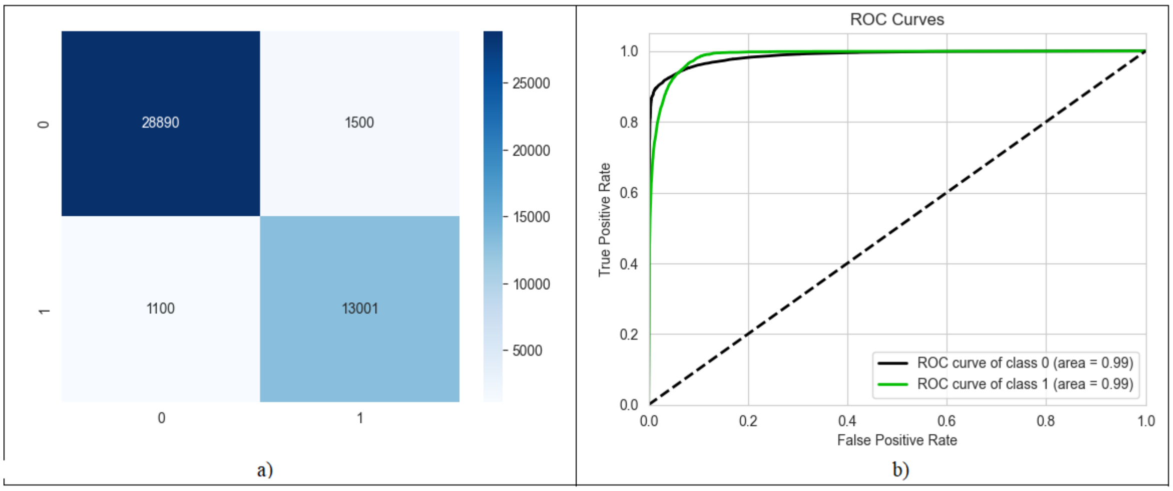

Fig. 8. (a) Confusion matrix; (b) ROC curve values of random forest.

Table 4. Evaluation metrics values of random forest.

Classes	Precision (%)	Recall (%)	F1-Score (%)	Support
Not spam (class 0)	96.33	95.06	95.69	30390
Spam (class 1)	89.66	92.20	90.91	14101

Fig. 9 and Table 5 present the findings of the Naive Bayes model. It is obvious that non-spam data is better detected but the results show that Naive Bayes is not an appropriate solution for this situation. As can be seen, ROC curve is not even close to 1. Nevertheless, this machine learning method can be useful considering that it is fast. Naive Bayes machine learning gets better results on mostly statistical data. Although the values used in this dataset are not statistical, they contain too many variables.

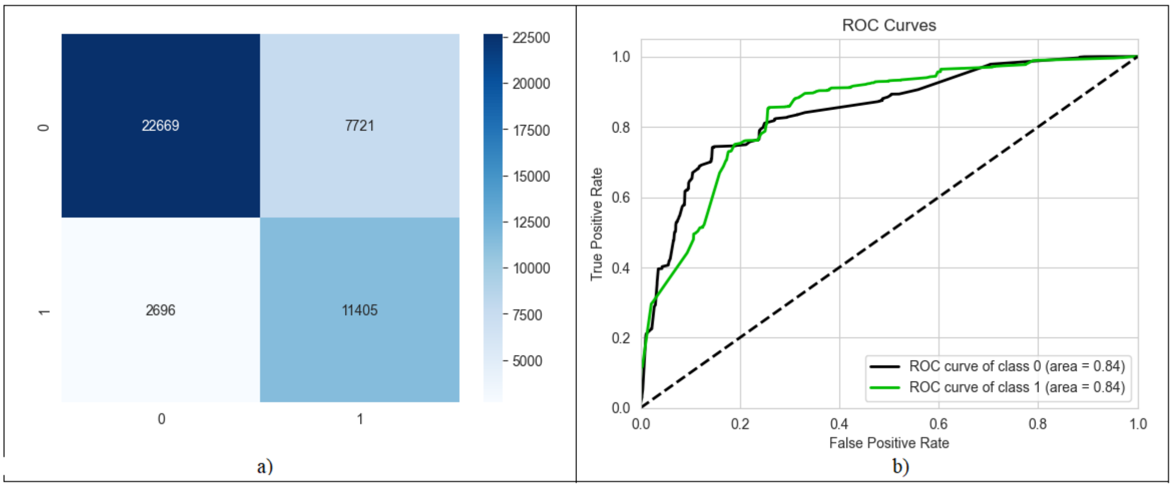

Fig. 9. (a) Confusion matrix; (b) ROC curve values of naïve bayes.

Table 5. Evaluation metrics values of naïve bayes.

Classes	Precision (%)	Recall (%)	F1-Score (%)	Support
Not spam (class 0)	89.37	74.59	81.32	30390
Spam (class 1)	59.63	80.88	68.65	14101

The K-Nearest Neighbor results are displayed in Fig. 10 and Table 6. Non-spam data is clearly detected better. The findings indicate that K-Nearest Neighbors is on the verge of solving this challenge. ROC curve has a value that close to 1. However, given how rapid this machine learning process is, it may be useful. K-Nearest Neighbor excel at classification via building neighborhood relations. Since several processes were performed by establishing correlation in this dataset it yielded good results.

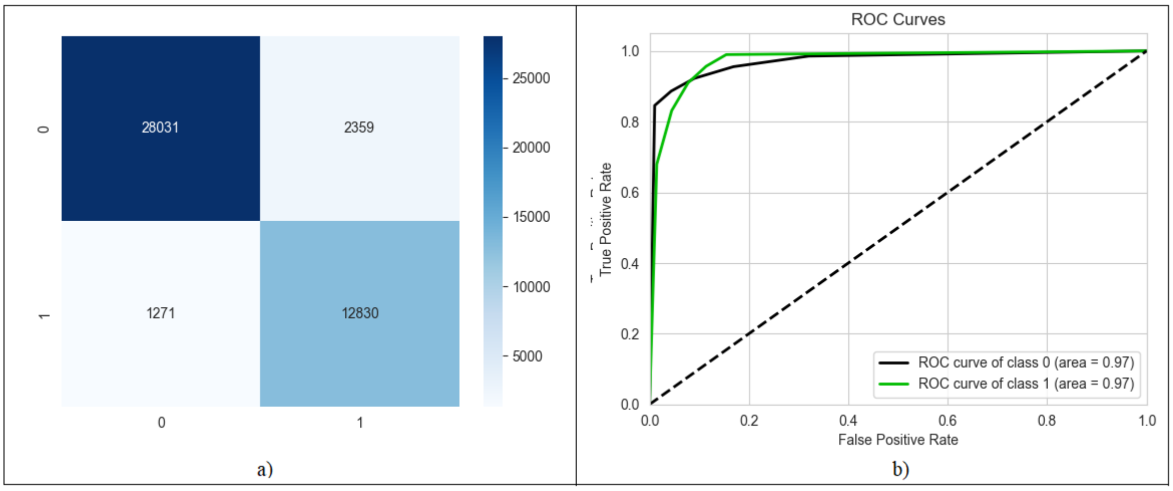

Fig. 10. (a) Confusion matrix; (b) ROC curve values of k-nearest neighbor.

Table 6. Evaluation metrics values of k-nearest neighbor.

Classes	Precision (%)	Recall (%)	F1-Score (%)	Support
Not spam (class 0)	95.66	92.24	93.92	30390
Spam (class 1)	84.47	90.99	87.61	14101

The XGBoost findings are displayed in Fig. 11 and Table 7. It is evident that non-spam data is better identified. According to the results, XGBoost is on the verge of resolving this issue. ROC curve value is close to 1. It might be beneficial, though, considering how fast this machine learning process works. XGBoost is frequently favoured in competitions and might be beneficial, though, considering how fast this machine learning process works. Aside from that, scholars prefer it because of the high worth of the results. It has a graph that is a little far from satisfactory.

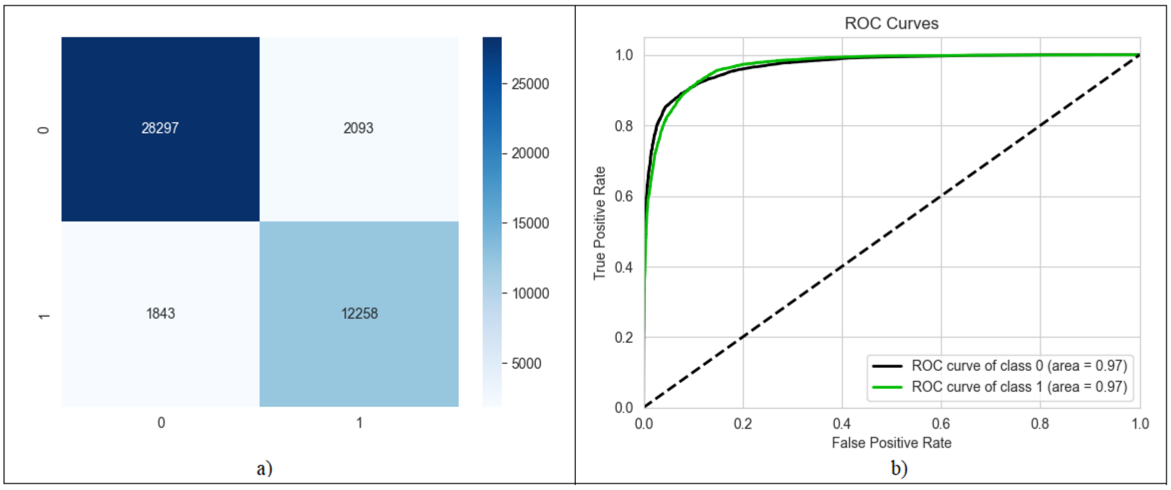

Fig. 11. (a) Confusion matrix; (b) ROC curve values of XGBoost.

Table 7. Evaluation metrics values of XGBoost.

Classes	Precision (%)	Recall (%)	F1-Score (%)	Support
Not spam (class 0)	93.89	93.11	93.50	30390
Spam (class 1)	85.42	86.93	86.17	14101

The results of AdaBoost are shown in Fig. 12 and Table 8. As seen, non-spam data is clearly detected better. The outcomes suggest that AdaBoost is close to resolving this issue. ROC curve value is slightly close to 1. However, given how rapid this machine learning process is, it may be useful. AdaBoost is frequently favored in contests and also popular among academics due to its high-value outcomes. It did not, however, provide very good remedies to this challenge. Although it is commonly used in stepped constructions, it did not produce the expected results when tackling this challenge. It is also thought that it can produce superior outcomes with an expanded parameter network.

Gradient Boosting results are given in Fig. 13 and Table 9. It is clear that non-spam data is better detected. According to the results obtained, it is seen that Gradient Boosting is comes very near to resolving this problem. ROC curve has a value slightly close to 1. However, considering that this machine learning method is fast, it can be useful. AdaBoost is often chosen in contests and also frequently preferred by researchers due to its high-value results. However, it did not provide very good solutions to this problem. Although it is mostly preferred in stepped structures, it could not give the expected results in solving this problem. It is also thought that it can give better results with an expanded parameter network.

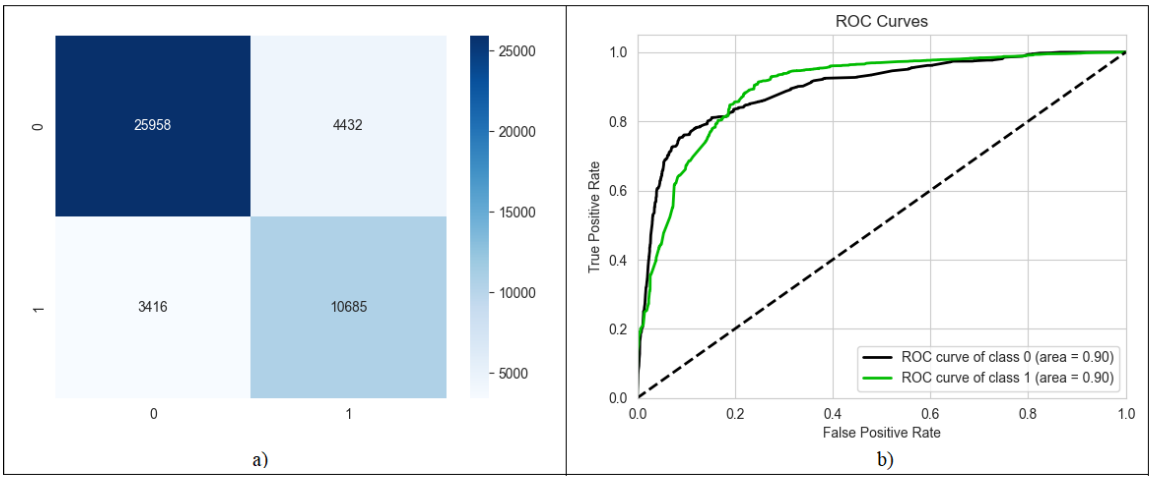

Fig. 12. (a) Confusion matrix; (b) ROC curve values of AdaBoost.

Table 8. Evaluation metrics values of AdaBoost.

Classes	Precision (%)	Recall (%)	F1-Score (%)	Support
Not spam (class 0)	88.37	85.42	86.87	30390
Spam (class 1)	70.68	75.77	73.14	14101

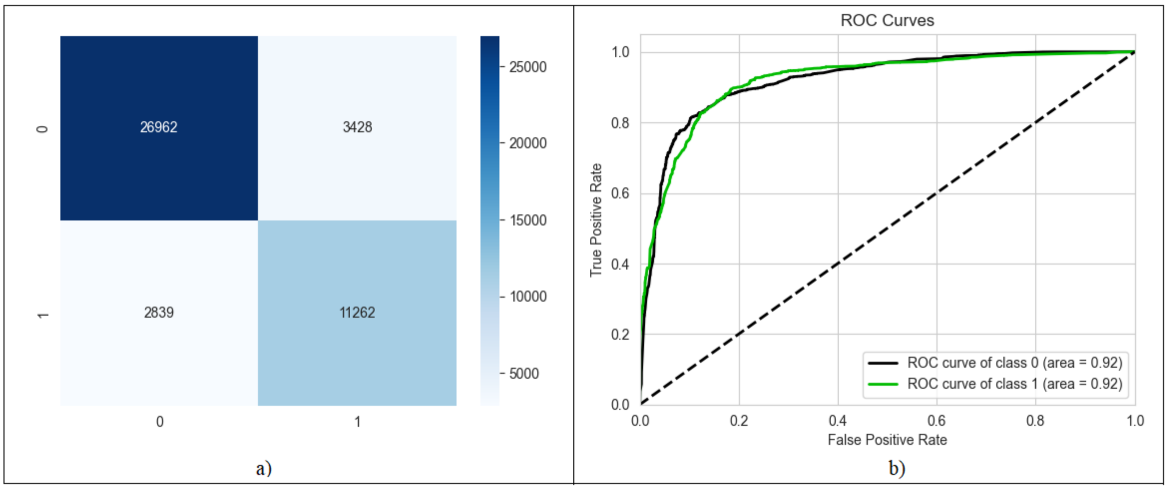

Fig. 13. (a) Confusion matrix; (b)ROC curve values of gradient boosting.

Table 9. Evaluation metrics values of gradient boosting.

Classes	Precision (%)	Recall (%)	F1-Score (%)	Support
Not spam (class 0)	90.47	88.72	89.59	30390
Spam (class 1)	76.66	79.87	78.23	14101

The weighted results of all the mentioned methods are shown in Table 10. Random Forest Classifier produces the best results with a success rate of 94.16% was achieved in all results. The most substantial aspect that distinguishes this study from other studies in the literature is that the most preferred and most successful machine learning methods are preferred and compared. Another prominent property of this study is that it worked with a large data set. Since it contains very high data, the results obtained are at a very satisfactory level. The main purpose of the study is to get impeccable results even with such large data. By using numerous machine learning methods, it has been determined which method can achieve better result. Random Forest method gave the best results in solving this problem, and the Decision Tree method gave the closest result. Additionally, the outcomes will be superior if a tree-based machine learning approach is used to solve such problems.

Table 10. Weighted metrics of the machine learning methods

ML Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression (LR)	79.89	79.88	79.89	79.89
Decision Tree (DT)	93.55	93.66	93.55	93.59
Random Forest (RF)	**94.16**	**94.22**	**94.16**	**94.18**
Naïve Bayes (NB)	76.59	79.95	76.59	77.30
K-Nearest Neighbor (KNN)	91.84	92.11	91.84	91.92
XGBoost (XGB)	91.15	91.20	91.15	91.17
AdaBoost (ABC)	82.36	82.76	82.36	82.52
Gradient Boosting (GBC)	85.91	86.10	85.91	85.99

4. Conclusion

Working with large datasets often yields more consistent results. For this purpose, it is essential to have a large-scale dataset. Accurate results are often difficult to find on small-scale datasets. Because if the weights of the data in the datasets are not determined very convenient, it will not be possible to get an efficiency from the operations performed. On the contrary, datasets created with very well selected data can also give very effective results with machine learning. Therefore, a large-scale dataset was selected in the study and extra features have been added to the data to be taught in machine learning. In this manner, machine learning would be able to make decisions while considering additional characteristics. The study demonstrates that machine learning algorithms of the tree-based typically produce favourable outcomes. This highlights how the decision mechanism provides appropriate responses for tree architectures. The Random Forest approach found a detection success of 96.33% for the highest non-spam class. Random Forest Classifier produced the best results in the study with 94.16%, 94.22%, 94.16% and 94.18% success was achieved in Accuracy, Recall, Precision and F1-Score values respectively for both spam and non-spam URL detection using combined and weighted results.

When the application is compared with other studies, the most striking difference is that a very large-scale dataset was selected. While working with big data, machine learning algorithms created can prevent high successes that may occur by chance. In this way, it was preferred to choose a large-scale and high-data dataset while selecting the dataset. In addition, most popular eight machine learning methods were used for comparison. As a result of the research such a comprehensive study was not found by the authors. The study provides convenience when determining which approach will be better when many algorithms are used. One of the conclusions that can be drawn from this study is that it has been clearly shown that tree-based classifiers give better results in multi-criteria decision making. There is a potential that the study will provide information for others to use in their future research.

Acknowledgements

There is no conflict of interest with any person or institution in the prepared article.

References

[1] R. S. Arslan, “Kötücül Web Sayfalarının Tespitinde Doc2Vec Modeli ve Makine Öğrenmesi Yaklaşımı,” *European Journal of Science and Technology*, no. 27, pp. 792–801, 2021, doi: 10.31590/ejosat.981450.

[2] D. Sahoo, C. Liu, and S. C. H. Hoi, “Malicious URL Detection using Machine Learning: A Survey,” *ArXiv*, vol. abs/1701.0, 2017.

[3] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi, “Detecting spam blogs: A machine learning approach,” *Proceedings of the National Conference on Artificial Intelligence*, vol. 2, pp. 1351–1356, 2006.

[4] F. O. Catak, K. Sahinbas, and V. Dörtkarde\cs, “Malicious URL detection using machine learning,” *Artificial intelligence paradigms for smart cyber-physical systems*, IGI Global, pp. 160–180, 2021.

[5] A. Begum and S. Badugu, “A study of malicious url detection using machine learning and heuristic approaches,” *Advances in Decision Sciences, Image Processing, Security and Computer Vision*, Springer, pp. 587–597, 2020.

[6] S. Kumar, X. Gao, I. Welch, and M. Mansoori, “A machine learning based web spam filtering approach,” *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016, pp. 973–980.

[7] P. Parekh, K. Parmar, and P. Awate, “Spam URL detection and image spam filtering using machine learning,” *Computer Engineering*, 2018.

[8] M. Aljabri *et al.*, “Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions,” *IEEE Access*, vol. 10, no. October, pp. 121395–121417, 2022, doi: 10.1109/ACCESS.2022.3222307.

[9] I. Hernández, C. R. Rivero, D. Ruiz, and R. Corchuelo, “CALA: ClAssifying Links Automatically based on their URL,” *Journal of Systems and Software*, vol. 115, pp. 130–143, 2016.

[10] C.-M. Chen, J.-J. Huang, and Y.-H. Ou, “Efficient suspicious URL filtering based on reputation,” *Journal of Information Security and Applications*, vol. 20, pp. 26–36, 2015.

[11] T. Manyumwa, P. F. Chapita, H. Wu, and S. Ji, “Towards Fighting Cybercrime: Malicious URL Attack Type Detection using Multiclass Classification,” *2020 IEEE International Conference on Big Data (Big Data)*, pp. 1813–1822, 2020.

[12] D. K. McGrath and M. Gupta, “Behind Phishing: An Examination of Phisher Modi Operandi.,” *LEET*, vol. 8, p. 4, 2008.

[13] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Identifying suspicious URLs: an application of large-scale online learning,” *Proceedings of the 26th annual international conference on machine learning*, pp. 681–688, 2009.

[14] H. Kwon, M. B. Baig, and L. Akoglu, “A domain-agnostic approach to spam-url detection via redirects,” *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 220–232, 2017.

[15] Y. Takata, M. Akiyama, T. Yagi, T. Hariu, and S. Goto, “Minespider: Extracting urls from environment-dependent drive-by download attacks,” *2015 IEEE 39th Annual Computer Software and Applications Conference*, vol. 2, pp. 444–449, 2015.

[16] R. Almeida and C. Westphall, “Heuristic phishing detection and URL checking methodology based on scraping and web crawling,” *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 1–6, 2020.

[17] R. S. Rao and A. R. Pais, “Detection of phishing websites using an efficient feature-based machine learning framework,” *Neural Computing and Applications*, vol. 31, no. 8, pp. 3851–3873, 2019.

[18] R. Raj and S. S. Kang, “Spam and Non-Spam URL Detection using Machine Learning Approach,” *2022 3rd International Conference for Emerging Technology (INCET)*, pp. 1–6, 2022.

[19] Q. Abu Al-Haija and M. Al-Fayoumi, “An intelligent identification and classification system for malicious uniform resource locators (URLs),” *Neural Computing and Applications*, vol. 35, no. 23, pp. 16995–17011, 2023, doi: 10.1007/s00521-023-08592-z.

[20] Kaggle, “Spam URLs Classification Dataset.” https://www.kaggle.com/datasets/shivamb/spam-url-prediction.

[21] A. Hmimou and others, “On the computation of the correlation matrix implied by a recursive path model,” *2020 IEEE 6th International Conference on Optimization and Applications (ICOA)*, pp. 1–5, 2020.

[22] S. Sperandei, “Understanding logistic regression analysis,” *Biochemia medica*, vol. 24, no. 1, pp. 12–18, 2014.

[23] D. Maulud and A. M. Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140–147, 2020.
[24] J. Chen *et al.*, "A comparison of linear regression, regularization, and machine learning algorithms to develop Europe-wide spatial models of fine particles and nitrogen dioxide," *Environment international*, vol. 130, p. 104934, 2019.
[25] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
[26] Y. K. Qawqzeh, M. M. Otoom, and F. Al-Fayez, "A Proposed Decision Tree Classifier for Atherosclerosis Prediction and Classification," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 19, no. 12, pp. 197–202, 2019.
[27] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.
[28] L. Breiman, "Random forests; uc berkeley tr567," *University of California: Berkeley, CA, USA*, 1999.
[29] J. R. Quinlan, *C4. 5: programs for machine learning*, Elsevier, 2014.
[30] L. Breiman, J. Friedman, C. Stone, and R. Olshen, "Classification and regression trees (crc, boca raton, fl)," 1984.
[31] I. Rish and others, "An empirical study of the naive Bayes classifier," *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41–46, 2001.
[32] E. Frank and R. R. Bouckaert, "Naive bayes for text classification with unbalanced classes," *Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings 10*, pp. 503–510, 2006.
[33] Ö. Şahinaslan, H. Dalyan, and E. Şahinaslan, "Naive bayes sınıflandırıcısı kullanılarak youtube verileri üzerinden çok dilli duygu analizi," *Bilişim Teknolojileri Dergisi*, vol. 15, no. 2, pp. 221–229, 2022.
[34] Y. Wu, K. Ianakiev, and V. Govindaraju, "Improved k-nearest neighbor classification," *Pattern recognition*, vol. 35, no. 10, pp. 2311–2318, 2002.
[35] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN Model-Based Approach in Classification," *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pp. 986–996, 2003.
[36] T. Chen *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.
[37] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interactive Learning Environments*, pp. 1–20, 2021.
[38] T.-K. An and M.-H. Kim, "A new diverse AdaBoost classifier," *2010 International conference on artificial intelligence and computational intelligence*, vol. 1, pp. 359–363, 2010.
[39] X. Li, L. Wang, and E. Sung, "AdaBoost with SVM-based component classifiers," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 5, pp. 785–795, 2008.
[40] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost-teaching AdaBoost to generalize better," *Graphicon*, vol. 12, no. 5, pp. 987–997, 2005.
[41] J. Son, I. Jung, K. Park, and B. Han, "Tracking-by-segmentation with online gradient boosting decision tree," *Proceedings of the IEEE international conference on computer vision*, pp. 3056–3064, 2015.
[42] S. Peter, F. Diego, F. A. Hamprecht, and B. Nadler, "Cost efficient gradient boosting," *Advances in neural information processing systems*, vol. 30, 2017.