Machine Learning-Based Effective Malicious Web Page Detection

Anıl Utku[1], Ümit Can[1]

[1]Department of Computer Engineering, Munzur University, Tunceli, Turkey
Corresponding Author: anilutku@munzur.edu.tr

Research Paper Received: 25.07.2022 Revised: 17.10.2022 Accepted: 17.10.2022

Abstract—The use of the Internet is becoming more and more widespread day by day, putting millions of users at risk of cyber-attacks. Internet usage has increased significantly and various cyber-attacks have been made through malicious websites. With these attacks, much information such as people's private information, bank information, and social information can be captured. Many methods have been developed to prevent cyber-attacks. In particular, methods that use machine learning methods other than traditional methods give more successful results. In this study, it has been tried to automatically detect malicious websites by using the URL properties of malicious websites. For this purpose, popular machine learning methods such as DT, kNN, LightGBM, LR, MLP, RF, SVM, and XGBoost were used. According to the experimental results, the RF algorithm achieved 96% accuracy.

Keywords—Malicious websites, cyber attacks, machine learning.

1. Introduction

Since its emergence, the Internet has allowed millions of people to interact with each other instantly, creating a network that connects billions of people around the world. At the same time, although this communication opportunity has positive aspects, it has many dark sides [1]. With the development and widespread use of internet technologies, cyber attackers have become an increasingly important security problem. Many types of malware, such as phishing and trojan horses, use internet addresses, ie URL addresses, as a tool for attack purposes. Due to the fact that the algorithms related to the generation of URL addresses have reached certain levels, new and many malicious URL addresses appear every day. Therefore, it is very important to identify these web links in order to prevent various network attacks and ensure network security [2], [3]. Internet users can do many important tasks such as shopping, banking, reservation, bill payment, obtaining information by using web pages, and they just browse the web pages without taking any action. In all these transactions, they may share sensitive data such as credit card information, address information, financial information, and personal information that they do not want others to receive. It is possible to access such sensitive information on the computer even if only these web pages are accessed without any sharing, and it is possible to seize or disrupt the system by taking advantage of the vulnerabilities [4]. In order to prevent this, it is necessary to ensure the security of the web page that is entered or to be entered. For this purpose, there are many studies in the literature using artificial intelligence methods to detect malicious websites.

Hou et al. [5] presented a machine learning-based method for detecting maliciously used web pages. Their study systematically analyzes the features of a malicious web page and offers important features for machine learning. According to the results they obtained, it was revealed that the methods they proposed were successful even in complex code structures. Ma et al.[6] on the other hand, detected malicious websites by using the lexical and host-based features of the URL. They achieved a successful result with an accuracy rate of 99%. Their methods have achieved very successful results. Zhang et al. [7] used URL information instead of the content of the web page to detect malicious web pages. Various methods have been used for URL feature selection and classification methods have been applied. Eshete [8], on the other hand, proposed a method that includes all of the static, dynamic, machine learning, evolutionary search and optimization approaches for the detection of malicious websites. They achieved very good results with the process they applied. Kazemian and Ahmed [9] argued that traditional methods detect malicious websites by looking at black lists, but these lists are not very effective, and they use K-Nearest Neighbor (kNN), Support Vector Machine (SVM) and Naive Bayes (NB) supervised methods and unsupervised methods such as K-Means and Affinity Propagation to detect malicious websites. The supervised learning method achieved an accuracy of 98%. Sahingoz et al. [10] proposed a method to detect phishing attacks using URL features. They used seven different classification algorithms and natural language processing features, and their proposed method achieved an accuracy rate of 97.98%. Liu and Lee [11], on the other hand, described three spam methods used by malicious sites; redirection spam, hidden Iframe spam, and content hiding spam. Then they proposed a new method based on Convolutional Neural Network (CNN). Their methods have achieved successful results. Li et al. [12] used classical machine learning methods and deep learning methods. Their study based on the detection of malicious X.509 certificate found on sites with bad content. They achieved 95.9% accuracy in ensemble methods and 98.2% accuracy in SVM-based methods. Malicious domain names and URLs are transmitted to different users via email and messages. Hence, [13] used machine learning techniques for malicious URL detection. In this study, the lexical features of the URL were used and good results were obtained with machine learning methods.

There are many studies in the literature on detecting malicious websites. These studies can generally be classified as URL information-based, website content-based and behavior-based studies [4]. In these studies, researchers used many methods such as heuristics, machine learning-based methods to detect malicious websites. In this study, it has been tried to determine whether a website is malicious or non-malicious by using URL information. Decision Tree (DT), kNN, Light Gradient Boosting (LightGBM), Logistic Regression (LR), Multilayer Perceptron (MLP), Random Forest (RF), SVM and eXtreme Gradient Boosting (XGBoost) algorithms were used to detect malicious web pages. The methods were compared using accuracy, precision, recall, and F-1 score values. Among these methods, the RF algorithm obtained the best result. In this study, a comparative analysis of machine learning methods is presented. Experimental results, as far as we know, are more successful than the results of studies in the literature.

2. Classification Models

- DT: Decision trees are a type of tree-shaped decision structure that is learned by induction from sample data whose classes are known, and one of the most used methods in solving classification problems is decision trees [14]. A decision tree is a structure used by dividing large amounts of records into very small groups of records by applying simple decision-making steps. With each successful division, the members of the result groups become much more similar to each other. In the structure of the decision tree, each node represents a feature [15].
- kNN: The grouping method proposed by Cover and Hart [16], in which the group where the sample data point is located and the nearest neighbor to this data point are determined according to the k value is called the kNN algorithm [16]. kNN algorithm is one of the most known and used algorithms among machine learning algorithms. Classification is made by using the closeness between a selected feature and the closest feature [17]. Despite its simplicity, kNN gives competitive results and in some cases even outperforms other complex learning algorithms. The most important advantage of this method is that successful studies can be performed in the classification process with multiple categorized data points [18].
- LR: Bivariate regression model, which is a basic linear regression model, gives the linear relationship between independent X variable and dependent Y variable. In regression analysis, independent variables can be quantitative or qualitative [19]. However, if the dependent variable is qualitative or quantitative, it also changes the quality of the regression analysis and the solution techniques, especially when the dependent variable is qualitative, the most used solution method is logistic regression analysis [20].
- MLP: It is one of the most commonly used artificial neural network models. MLP model is consists of three different layers: input layer, hidden layer and output layer. The input layer is the layer from which the data is read. Since each neuron represents a feature, it contains as many neurons as the number of features. The output layer is the layer where the classes are determined [21]. It is one of the most used methods in artificial neural networks. The most important advantage of MLP is its high learning potential, noise resistance, nonlinearity, fault tolerance and success in general tasks [22].
- RF: It is an ensemble learning method which can be defined as a collection of tree type classifiers. It is an improved version of the Bagging method by adding the randomness feature [23], [24]. Instead of branching each node using the best branch among all the variables, RF branches each node using the best randomly selected variables at each node. Each dataset is generated by displacement from the original dataset. Trees are then developed using random feature selection [25]. Developed trees are not pruned. This strategy makes RF accuracy unmatched. RF is also very fast, resistant to overfitting, and can work with as many trees as desired [26].
- SVM: It is an algorithm developed by Vapnik and Cortes [27]. It is a binary classifier with high generalization ability. Its greatest feature is that it finds a single global minimum without being stuck with a local minimum. It is used in data sets where the patterns between the variables are not known, and it is an educational

method used for classification or regression tasks.

- $LightGBM$: It is one of the next generation successful ensemble learning algorithms. It was developed by Microsoft in 2017. Another feature that distinguishes this method from other gradient boosting algorithms is the growth strategy it uses during the training of decision trees. While LightGBM uses the vertical growth strategy (leaf growth), other gradient boosting methods use the level-wise growth strategy [28]. Another important feature that makes LightGBM advantageous is the two new algorithms it contains that increase the processing speed [28].
- $XGBoost$: It is another successful ensemble learning algorithm and was proposed by Chen in 2016 [29]. It is based on the Classification and Regression Tree, redefines partition attributes and uses the minimization of the loss function to determine partition attributes. Its accuracy is higher than other models and the training cost is less than other models [30].

3. Building Malicious Web Page Detection Models

In recent years, personal information theft and financial losses have been experienced due to increasing cyber attacks in parallel with the increase in Internet usage. Cyber attacks can be carried out through phishing, spam mail and, malware. Websites, which are the most widely used elements of the Internet, have become the target of attackers. The exploration of malicious websites is of critical importance to ensure the security of institutions and individuals. The sooner a malicious website is detected, the faster the defense will be.

Many malicious attacks can be made on internet users through malicious websites. For example, they can infect their systems with malware or suffer a phishing attack. To deceive and ensnare users, attackers can create malicious websites with fake information and spread malicious advertisements. Apart from that, users' credentials can be stolen using vulnerable websites.

URL-based, web-content-based, and script-based methods are mainly used for the detection of malicious web pages. URL-based methods are a safe and efficient approach as they can detect malicious URLs before users visit them. In this study, a URL-based malicious website detection system has been developed.

3.1. Data set

In this study, a dataset consisting of 1781 URL data, in which websites are labeled as malicious and non-malicious based on the application layer and network properties, was used. The dataset consists of malicious URL data obtained from platforms that blacklist web pages according to their malicious status, and URL data tagged as non-malicious. There are 1565 benign and 216 malicious websites in the dataset. The dataset used is publicly accessible via https://www.kaggle.com/datasets/xwolf12/malicious-and-benign-websites. The attributes in the dataset can be described as follows:

- URL: It is the anonymous ID of the URL analyzed in the study.
- URL_LENGTH: The number of characters in the URL.
- NUMBERSPECIALCHARACTERS: It is the number of special characters such as "/", "%", "#", "&", " defined in the URL. ", "=".
- CHARSET: It is a categorical value called a character set.
- SERVER: It is a categorical value and refers to

the operating system of the server from which the packet response is received.

- CONTENT_LENGTH: It refers to the content size of the HTTP header.
- WHOIS_COUNTRY: It is a categorical variable that represents the countries from which the server response is received.
- WHOIS_STATEPRO: The server response is a categorical variable that expresses the statuses received.
- WHOIS_REGDATE: It is a variable that provides the Whois server registration date and has date values in the format DD/MM/YYY HH:MM.
- WHOISUPDATEDDATE: It refers to the last update date of the server analyzed through Whois.
- TCPCONVERSATIONEXCHANGE: It refers to the number of TCP packets exchanged between the server and the honeypot client.
- DISTREMOTETCP_PORT: Indicates the number of detected ports that differ from TCP.
- REMOTE_IPS: It represents the total number of IPs connected to Honeypot.
- APP_BYTES: Indicates the number of bytes transferred.
- SOURCEAPPPACKETS: It refers to packets sent from Honeypot to the server.
- REMOTEAPPPACKETS: Indicates packets received from the server.
- APP_PACKETS: It is the total number of IP packets generated during the communication between the honeypot and the server.
- DNS_QUERY_TIMES: The number of DNS packets generated during communication between the Honeypot and the server.
- TYPE: It is a categorical variable that represents the type of web page being analyzed. Their values are 1 for malicious websites and 0 for non-malicious websites.

In the data preprocessing step, incorrect or missing fields were checked among the attributes in the dataset. This attribute has been omitted from the CONTENT_LENGTH column because it has 812 NULL values, which is almost half of the dataset. In addition, columns with categorical values such as WHOIS_COUNTRY and SERVER were removed from the dataset. The relationships among the features obtained after removing the highly correlated features are shown in Figure 1.

As seen in Figure 1, there is a strong relationship between URL_LENGTH, NUMBER_SPECIAL_CHARACTERS and CONTENT_LENGTH attributes and Type attribute.

3.2. Evaluation Metrics

The most frequently used metrics when measuring the success of classification methods are accuracy, precision, recall, and F-1 score metrics obtained from the confusion matrix. These metrics were also used in this study. The complexity matrix shown in Table 1 is a table used to show the classification performance of various classification algorithms on a dataset.

Table 1.
Confusion matrix.

	Actual values		
Prediction values		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

TP denotes values with a positive true value and is positively predicted by the classifier. FN represents values whose true value is negative but positively predicted by the classifier. FP denotes values whose true value is positive but negatively predicted by the classifier. TN, on the other hand, refers to

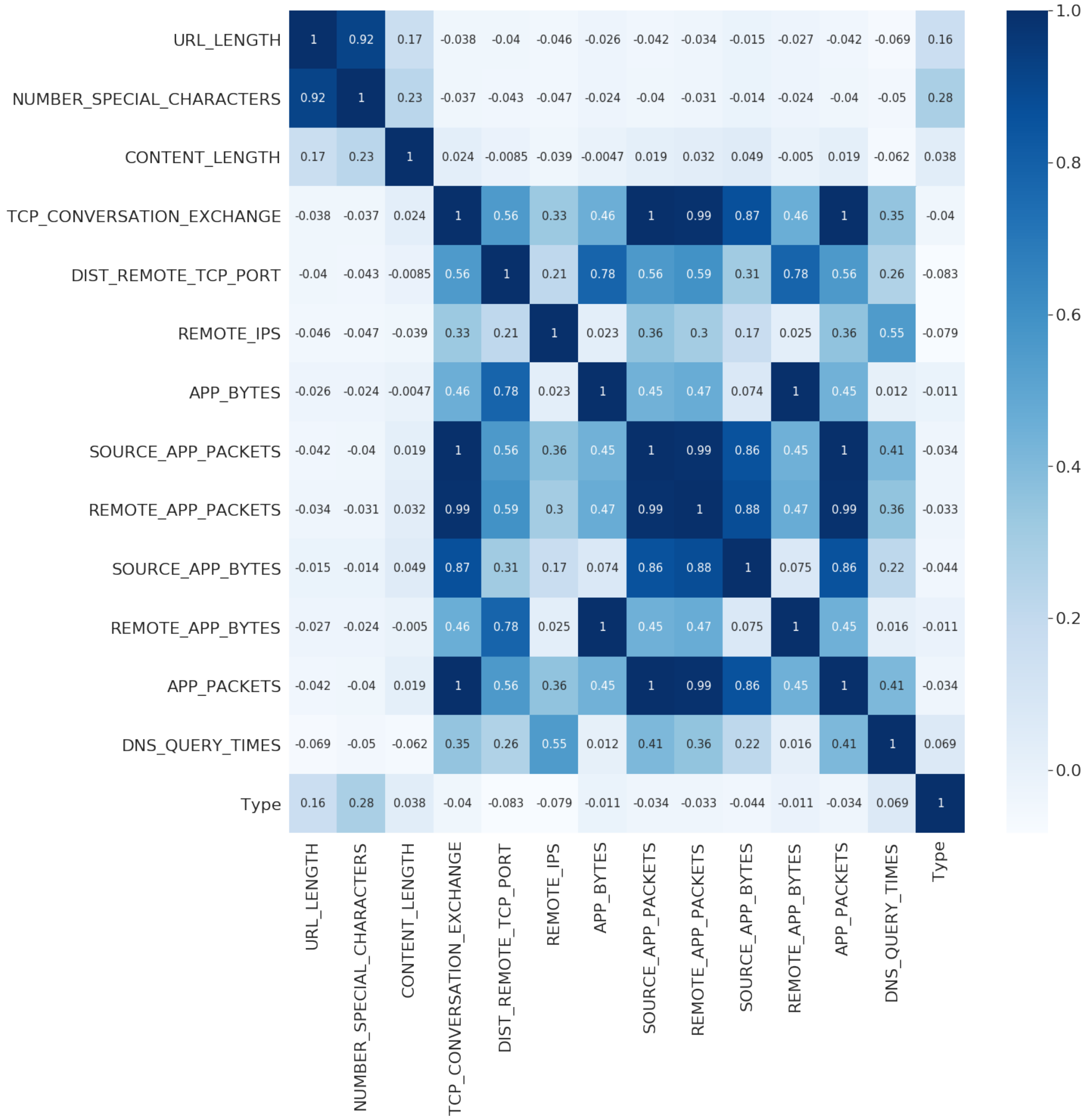

Figure 1. Relationships between attributes

values whose true value is negative and negatively estimated by the classifier. Using these values in the confusion matrix, accuracy, precision, recall, and F-1 score metrics are calculated. The accuracy metric is the most intuitive performance measure and is simply the ratio of correctly predicted observations to total observations. The accuracy metric is calculated using Eq. 1.

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

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Precision is calculated using Eq. 2.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall is the ratio of correctly predicted positive observations to all observations in the real class. Recall is calculated using Eq. 3.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The F-1 score is the weighted average of precision and recall. Therefore, this score takes into account both false positives and false negatives. Intuitively, accuracy is not as easy to understand, but an F-1 score is often more useful than accuracy, especially when there is an uneven class distribution. Accuracy works best if false positives and false negatives cost similarly. If the cost of false positives and false negatives is very different, it's better to look at both precision and recall. F-1 score is calculated with Eq. 4.

$$F - 1\, score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

4. Experimental Results

This study compared DT, kNN, LightGBM, LR, MLP, RF, SVM and XGBoost algorithms for detecting malicious web pages using accuracy, precision, recall, and F-1 score values.

The dataset is divided into 70% training and 30% testing. 10% of the training data is reserved for validation. After obtaining the training, validation, and test datasets, the data were normalized. Validation data is used for the optimization of algorithm parameters. It is aimed to select the most suitable model parameters using validation data. Parameter analysis studies were carried out using GridSearchCV to determine the parameters of the applied machine learning algorithms. In the applied algorithms, cross validation was used to eliminate the overfitting problem and to increase the quality of the models created. Cross-validation allows the performance of the model to be tested before encountering high error rates on an as yet unseen test dataset. A value of k=10 was chosen for cross validation. All applied models were run on 10 different datasets randomly generated using cross validation, and the results obtained were averaged. Classification models were created using the obtained parameters and predictions were made. Precision, recall, accuracy, and F-1 score values were obtained by creating a confusion matrix according to the prediction results obtained. The flow diagram of the developed system is presented in Figure 2.

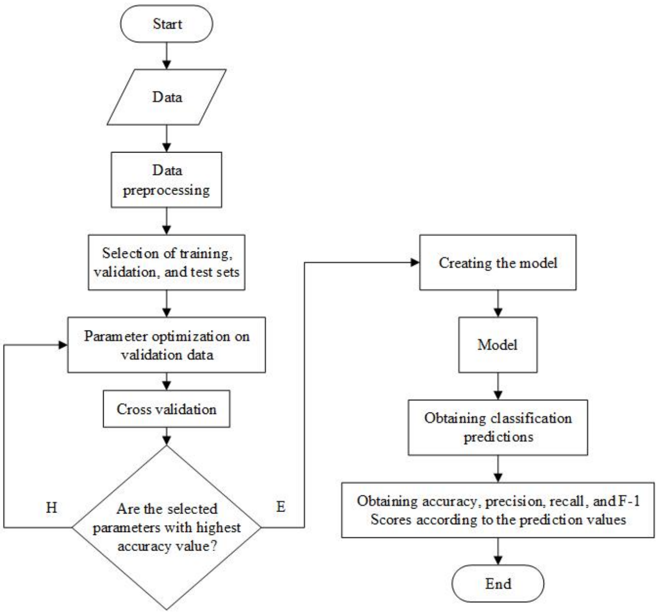
Start
Data
Data preprocessing
Selection of training, validation, and test sets
Parameter optimization on validation data
Cross validation
Are the selected parameters with highest accuracy value?
H
E
Creating the model
Model
Obtaining classification predictions
Obtaining accuracy, precision, recall, and F-1 Scores according to the prediction values
End

Figure 2. Flow chart of the developed system

The confusion matrix for DT is shown in Table 2.

As shown in Table 2, the number of correctly classified malicious web pages is 30, and the number of correctly classified non-malicious web pages is 440. DT correctly classified 470 web pages. DT misclassified 61 web pages.

The confusion matrix for kNN is shown in Table 3.

As shown in Table 3, the number of correctly clas-

Table 2.
Confusion matrix for DT.

	Actual values		
Prediction values		Non-malicious (0)	Malicious (1)
	Non-malicious (0)	440	22
	Malicious (1)	39	30

Table 3.
Confusion matrix for kNN.

	Actual values		
Prediction values		Non-malicious (0)	Malicious (1)
	Non-malicious (0)	444	18
	Malicious (1)	35	34

sified malicious web pages is 34, and the number of correctly classified non-malicious web pages is 444. kNN correctly classified 478 web pages. kNN misclassified 53 web pages.

The confusion matrix for LightGBM is shown in Table 4.

Table 4.
Confusion matrix for LightGBM.

	Actual values		
Prediction values		Non-malicious (0)	Malicious (1)
	Non-malicious (0)	450	12
	Malicious (1)	29	40

As seen in Table 4, the number of correctly classified malicious web pages is 40, and the number of correctly classified non-malicious web pages is 450. LightGBM correctly classified 490 web pages. LightGBM misclassified 41 web pages.

The confusion matrix for LR is shown in Table 5.

Table 5.
Confusion matrix for LR.

	Actual values		
Prediction values		Non-malicious (0)	Malicious (1)
	Non-malicious (0)	434	28
	Malicious (1)	45	24

As seen in Table 5, the number of correctly classified malicious web pages is 24, and the number of correctly classified non-malicious web pages is 434. LR correctly classified 458 web pages. LR misclassified 73 web pages.

The confusion matrix for MLP is shown in Table 6.

Table 6.
Confusion matrix for MLP.

	Actual values		
Prediction values		Non-malicious (0)	Malicious (1)
	Non-malicious (0)	440	23
	Malicious (1)	39	29

As seen in Table 6, the number of correctly classified malicious web pages is 29, and the number of correctly classified non-malicious web pages is 440. MLP correctly classified 469 web pages. MLP misclassified 62 web pages.

The confusion matrix for RF is shown in Table 7.

As seen in Table 7, the number of correctly classified malicious web pages is 48, and the number of correctly classified non-malicious web pages is 462. RF correctly classified 510 web pages. RF misclassified 21 web pages.

Table 7.
Confusion matrix for RF.

	Actual values		
Prediction values		Non-malicious (0)	Malicious (1)
	Non-malicious (0)	462	4
	Malicious (1)	17	48

The confusion matrix for SVM is shown in Table 8.

Table 8.
Confusion matrix for SVM.

	Actual values		
Prediction values		Non-malicious (0)	Malicious (1)
	Non-malicious (0)	459	4
	Malicious (1)	20	48

As can be seen in Table 8, the number of correctly classified malicious web pages is 48, and the number of correctly classified non-malicious web pages is 459. SVM correctly classified 507 web pages. SVM misclassified 24 web pages.

The confusion matrix for XGBoost is shown in Table 9.

Table 9.
Confusion matrix for XGBoost.

	Actual values		
Prediction values		Non-malicious (0)	Malicious (1)
	Non-malicious (0)	449	12
	Malicious (1)	30	40

As seen in Table 9, the number of correctly classified malicious web pages is 40, and the number of correctly classified non-malicious web pages is 449. XGBoost correctly classified 489 web pages. XGBoost misclassified 42 web pages.

Comparative experimental results for DT, kNN, LightGBM, LR, MLP, RF, SVM and XGBoost according to accuracy, precision, recall, and F-1 score values are shown in Table 10 and Figur3.

Table 10.
Comparative experimental results.

Model	Accuracy	Precision	Recall	F-1 score
DT	0,885	0,918	0,952	0,934
kNN	0,900	0,926	0,961	0,943
LightGBM	0,922	0,939	0,974	0,956
LR	0,862	0,906	0,939	0,922
MLP	0,883	0,918	0,950	0,933
RF	**0,960**	**0,964**	**0,991**	**0,977**
SVM	0,954	0,958	0,991	0,974
XGBoost	0,920	0,937	0,973	0,654

As shown in Table 10, RF has more successful results than other models compared. For RF, the accuracy value is 0.960, the precision value is 0.964, the recall value is 0.991, and the F-1 score is 0.977. SVM, XGBoost and LightGBM are the models with the most successful results after RF. For SVM, the accuracy value is 0.954, the precision value is 0.958, the recall value is 0.991, and the F-1 score is 0.974. For XGBoost, the accuracy value is 0.920, the precision value is 0.937, the recall value is 0.973, and the F-1 score is 0.954. The accuracy value for LightGBM is 0.922, the precision value is 0.939, the recall value is 0.974, and the F-1 score is 0.956.

As can be seen in Table 10 and Figure 3, RF showed a better classification performance in malicious web page detection compared to other models. SVM, XGBoost, and LightGBM are the models with the most successful results after RF. On the

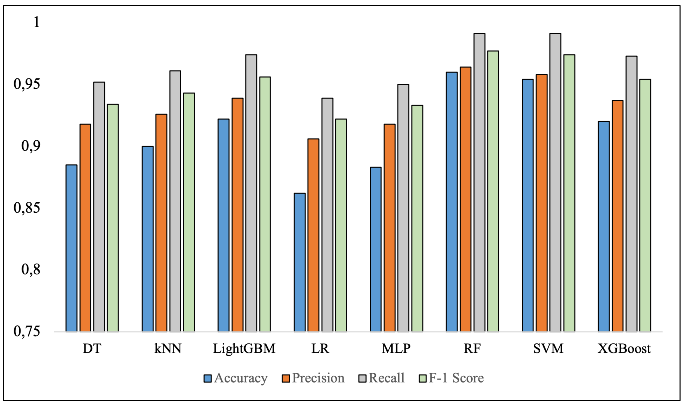

Figure 3. Comparative experimental results

other hand, LR showed unsuccessful classification performance in detecting malicious web pages.

The experimental results of the studies in the literature using the same dataset as this study are shown in Table 11.

Table 11.
Experimental results of studies in the literature.

Author(s)	Year	Model	Accuracy
Aljabri et al. [31]	2022	NB	%96
Labhsetwar et al. [32]	2021	RF	%92
Alkhudair et al. [33]	2020	RF	%95
Panischev et al. [34]	2020	RF	%95
Sandag et al. [35]	2018	kNN	%95

As seen in Table 11, the accuracy rate of the study by [31] is the same as the accuracy rate obtained in this study. In the study conducted by [31], 0.9564 precision, 0.9225 recall, and 0.9391 F1-Score values were obtained. This study obtained 0.964 precision, 0.991 recall, and 0.977 F1-Score values. Experimental results showed that this study had more successful results than studies in the literature.

5. Conclusions

Recently, with the increase in the number of transactions made online, fraudulent websites have been created by attackers to imitate trusted websites in order to obtain the private information of users and institutions. Known as phishing, these processes are carried out by creating fake websites or sending spam emails. In addition to internet users, many institutions are also victims of phishing and suffer great material and moral losses. These malicious websites and emails have become the most popular and easy way of data theft.

This study aimed to detect malicious web pages with URL-based machine learning methods. For this purpose, DT, kNN, LightGBM, LR, MLP, RF, SVM and XGBoost algorithms were compared using accuracy, precision, recall, and F-1 score values. Experimental results have shown that RF has better classification performance in malicious web page detection compared to other models compared. SVM, XGBoost and LightGBM are the models with the most successful results after RF.

The fact that RF has more successful results than DT can be interpreted as RF preventing overlearning by using more than one tree. RF consists of randomly generated decision trees. Each node in the decision tree runs on a random subset of features to calculate the output. RF then combines the outputs of the individual decision trees to create the final output.

The fact that RF has more successful results than kNN can be interpreted with the size of the dataset. kNN is resistant to noisy training data and is effective in the case of a large number of training samples. However, kNN requires the value of the k parameter, which expresses the number of nearest neighbors, and the distance function to be used.

The RF has more successful results than Light-

GBM can be interpreted as parameter setting for RF is easy and RF is resistant to parameter changes.

The fact that RF has more successful results than LR can be interpreted by the non-linear nature of the features in the dataset. LR is a linear classifier while RF is a nonlinear classifier.

The fact that RF has more successful results than MLP can be interpreted as RF working more successfully on tabular data such as audio, image, and text data. RF and MLP are different kinds of algorithms. RF is a collection of decision trees. Each decision in the group processes the tree sample and estimates the output label. Decision trees in the community are independent. Each can guess the final answer. MLP is a network of interconnected neurons. Neurons cannot function without other neurons and are interconnected. They are usually grouped into layers and process the data in each layer and forward to the next layers. The last layer of neurons is the decision maker. RF can only work with tabular data. However, MLP can work with many different data types.

The fact that RF has more successful results than SVM can be interpreted as the presence of categorical and numerical features in the dataset. SVM calculates the distance between different points in a multidimensional space. RF, on the other hand, can handle high dimensional spaces as well as a large number of training examples.

The algorithms compared in this study have a successful classification performance in malicious website prediction. Detection of malicious websites is becoming more important in an increasingly digital world. In terms of phishing and data theft, the idea of preventing attacks by using artificial intelligence methods comes to the fore. The results obtained in this study are promising in terms of adapting the malicious website prediction problem to real-world applications.

Authors Contributions

AU: Conceptualization, methodology, writing original draft, data analysis, and model coding. UC: Writing original draft, investigation, data analysis, data collection, and programming support. All authors read and approved the final manuscript

References

[1] U. Can and B. Alatas, “Cyberbullying and cyberstalking on online social networks,” in *Securing Social Networks in Cyberspace*. CRC Press, 2021, pp. 141–162.

[2] R. S. ARSLAN, “Kötücül url filtreleme için derin öğrenme modeli tasarımı,” *Avrupa Bilim ve Teknoloji Dergisi*, no. 29, pp. 122–128, 2021.

[3] S. He, B. Li, H. Peng, J. Xin, and E. Zhang, “An effective cost-sensitive xgboost method for malicious urls detection in imbalanced dataset,” *IEEE Access*, vol. 9, pp. 93 089–93 096, 2021.

[4] A. Sirageldin, B. B. Baharudin, and L. T. Jung, “Malicious web page detection: A machine learning approach,” in *Advances in Computer Science and its Applications*. Springer, 2014, pp. 217–224.

[5] Y.-T. Hou, Y. Chang, T. Chen, C.-S. Laih, and C.-M. Chen, “Malicious web content detection by machine learning,” *Expert Systems with Applications*, vol. 37, no. 1, pp. 55–60, 2010.

[6] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Learning to detect malicious urls,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1–24, 2011.

[7] W. Zhang, Y.-X. Ding, Y. Tang, and B. Zhao, “Malicious web page detection based on on-line learning algorithm,” in *2011 International Conference on Machine Learning and Cybernetics*, vol. 4. IEEE, 2011, pp. 1914–1919.

[8] B. Eshete, “Effective analysis, characterization, and detection of malicious web pages,” in *Proceedings of the 22nd International Conference on World Wide Web*, ser. WWW ’13 Companion. New York, NY, USA: Association for Computing Machinery, 2013, pp. 355–360.

[9] H. B. Kazemian and S. Ahmed, “Comparisons of machine learning techniques for detecting malicious webpages,” *Expert Systems with Applications*, vol. 42, no. 3, pp. 1166–1177, 2015.

[10] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from urls,” *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.

[11] D. Liu and J.-H. Lee, “Cnn based malicious website detection by invalidating multiple web spams,” *IEEE Access*, vol. 8, pp. 97 258–97 266, 2020.

[12] J. Li, Z. Zhang, and C. Guo, “Machine learning-based malicious x. 509 certificates’ detection,” *Applied Sciences*, vol. 11, no. 5, p. 2164, 2021.

[13] A. S. Raja, R. Vinodini, and A. Kavitha, "Lexical features based malicious url detection using machine learning techniques," *Materials Today: Proceedings*, vol. 47, no. 1, pp. 163–166, 2021.

[14] SPSS, *AnwerTree Algorithm Summary*. USA: SPSS White Paper, 1999.

[15] J. Sun and H. Li, "Data mining method for listed companies' financial distress prediction," *Knowledge-Based Systems*, vol. 21, no. 1, pp. 1–5, 2008.

[16] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[17] M. Khan, Q. Ding, and W. Perrizo, "k-nearest neighbor classification on spatial data streams using p-trees," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 517–528.

[18] E. Erdem and F. Bozkurt, "A comparison of various supervised machine learning techniques for prostate cancer prediction," *Avrupa Bilim ve Teknoloji Dergisi*, no. 21, pp. 610–620, 2021.

[19] C. Mood, "Logistic regression: Why we cannot do what we think we can do, and what we can do about it," *European Sociological Review*, vol. 26, no. 1, pp. 67–82, 2010.

[20] S. Domínguez-Almendros, N. Benítez-Parejo, and A. R. Gonzalez-Ramirez, "Logistic regression models," *Allergologia et Immunopathologia*, vol. 39, no. 5, pp. 295–305, 2011.

[21] Y. Canbay, A. İsmetğlu, and P. Canbay, "Covid-19 hastalığının teşhisinde derin öğrenme ve veri mahremiyeti," *Mühendislik Bilimleri ve Tasarım Dergisi*, vol. 9, no. 2, pp. 701–715, 2021.

[22] H. Faris, I. Aljarah, N. Al-Madi, and S. Mirjalili, "Optimizing the learning process of feedforward neural networks using lightning search algorithm," *International Journal on Artificial Intelligence Tools*, vol. 25, no. 06, p. 1650033, 2016.

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, 2016.

[25] M. Mursalin, Y. Zhang, Y. Chen, and N. V. Chawla, "Automated epileptic seizure detection using improved correlation-based feature selection with random forest classifier," *Neurocomputing*, vol. 241, pp. 204–214, 2017.

[26] H. Chen, Z. Lin, H. Wu, L. Wang, T. Wu, and C. Tan, "Diagnosis of colorectal cancer by near-infrared optical fiber spectroscopy and random forest," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 135, pp. 185–191, 2015.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[29] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 785–794.

[30] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[31] M. Aljabri, F. Alhaidari, R. M. A. Mohammad, S. Mirza, D. H. Alhamed, H. S. Altamimi, S. M. Chrouf *et al.*, "An assessment of lexical, network, and content-based features for detecting malicious urls using machine learning and deep learning models," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[32] S. R. Labhsetwar, P. A. Kolte, and A. S. Sawant, "Rakshanet: Url-aware malicious website classifier," in *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*. IEEE, 2021, pp. 308–313.

[33] F. Alkhudair, M. Alassaf, R. U. Khan, and S. Alfarraj, "Detecting malicious url," in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*. IEEE, 2020, pp. 1–5.

[34] O. Y. Panischev, E. N. Ahmedshina, D. V. Kataseva, A. Katasev, and A. Akhmetvaleev, "Creation of a fuzzy model for verification of malicious sites based on fuzzy neural networks," *International Journal of Engineering Research and Technology*, vol. 13, no. 12, pp. 4432–4438, 2020.

[35] G. A. Sandag, J. Leopold, and V. F. Ong, "Klasifikasi malicious websites menggunakan algoritma k-nn berdasarkan application layers dan network characteristics," *CogITo Smart Journal*, vol. 4, no. 1, pp. 37–45, 2018.