Turkish Journal of Science & Technology
16(2), 231-243, 2021

Research Paper

Detection of Phishing Attacks on Websites with Lasso Regression, Minimum Redundancy Maximum Relevance Method, Machine Learning Methods, and Deep Learning Model

Mesut TOĞAÇAR[1*]
[1] Department of Computer Technologies, Technical Sciences Vocational School, Fırat University, Elazığ, Turkey
[*1] mtogacar@firat.edu.tr

(Geliş/Received: 29/06/2021; Kabul/Accepted: 04/07/2021)

Abstract: Phishing attacks are malicious software designed to steal personal or public. These types of attacks generally use e-mail addresses or aim to impersonate web-based pages to trap users. In such applications, they use textual or visual-based attractive content to lure users into their network. The internet environment is a large network platform with billions of users, and on this platform, users must be able to safely conduct their transactions without being harmed. To ensure the security of web pages simultaneously on a platform with billions of users, artificial intelligence-based software has been developed recently and this situation continues. In this study, analyzes were performed using two datasets. The two datasets consist of a total of 12454 website content. The first dataset consists of 11054 websites and the second dataset consists of 1400 websites. The datasets are divided into two classes, "phishing" and "legitimate". The contributions of machine learning methods, deep learning models, and feature selection methods in detecting phishing attacks were analyzed. The best accuracy success rate for the first dataset was 97.26%. The best accuracy success rate for the second dataset was 94.76%. As a result, it has been observed that feature selection methods contribute to the experimental analysis in general.

Keywords: Phishing attack, deep learning, machine learning, feature selection.

Kement Regresyon, Minimum Yedeklilik Maksimum Alaka Yöntemi, Makine Öğrenimi Yöntemleri ve Derin Öğrenme Modeli ile Web Sitelerine Yönelik Oltalama Saldırılarının Tespiti

Öz: Kimlik avı saldırıları, kişisel veya genel olarak çalmak için tasarlanmış kötü amaçlı yazılımlardır. Bu tür saldırılar genellikle e-posta adreslerini kullanır veya kullanıcıları tuzağa düşürmek için web tabanlı sayfaları taklit etmeyi amaçlar. Bu tür uygulamalarda, kullanıcıları ağlarına çekmek için metin veya görsel tabanlı çekici içerikler tercih edilir. İnternet ortamı milyarlarca kullanıcısı olan büyük bir ağ platformudur ve bu platformda kullanıcıların işlemlerini güvenli bir şekilde zarar görmeden yapabilmeleri gerekmektedir. Milyarlarca kullanıcıya sahip bir platformda eş zamanlı olarak web sayfalarının güvenliğini sağlamak için son zamanlarda yapay zekâ tabanlı yazılımlar geliştirilmektedir. Bu çalışmada, iki veri seti kullanılarak analizler gerçekleştirilmiştir. İki veri seti toplam 12454 web sitesi içeriğinden oluşmaktadır. İlk veri seti 11054 web sitesinden ve ikinci veri seti 1400 web sitesinden oluşmaktadır. Veri kümeleri "oltalama" ve "legal" olmak üzere iki sınıfa ayrılır. Kimlik avı saldırılarını tespit etmede makine öğrenmesi yöntemleri, derin öğrenme modelleri ve özellik seçme yöntemlerinin katkıları analiz edildi. İlk veri seti için en iyi doğruluk başarı oranı %97.26'ydı. İkinci veri seti için en iyi doğruluk başarı oranı %94,76'ydı. Sonuç olarak, öznitelik seçme yöntemlerinin genel olarak deneysel analizlere katkı sağladığı gözlemlendi.

Anahtar kelimeler: Oltalama saldırısı, derin öğrenme, makine öğrenme, özellik seçimi.

1. Introduction

Phishing attacks are coded malicious software aimed at stealing users' personal information by using content that attracts users in the form of gifts, discounts, and awards through web applications or e-mails. Phishing attacks have an old history and are still used effectively by malicious people keeping up with technological developments [1,2]. Phishing attacks are an electronic fraud method that aims to drive users into their network by imitating corporate, commercial, or personal web addresses, e-mail addresses. The Internet is a large network used by billions of users around the world simultaneously. Thanks to this network, people have fast access to information, shopping, commerce, bill payment, etc. They can easily perform their operations. Security should be at the forefront as much as fast information exchange on the Internet. Precautionary internet protocols or security software have been developed. However, when technological developments are used by malicious people, this state of trust can be shaken [1,3]. Wandera stated in the Mobile Threat Landscape 2020 report that a new website is opened with phishing attacks every 20 seconds. In the same report, 87% of the attacks from mobile applications originate from messaging, social media, and game content web pages [4]. Anti-Phishing Working Group's 2020

* Corresponding author: mtogacar@firat.edu.tr. ORCID Number of author: [1] 0000-0002-8264-3899

study stated that the rate of phishing attacks with SSL protection is 75% [5]. Recently, most phishing attacks are carried out using the keyword "Novel Coronavirus". Google company blocked more than 240 million spam messages labeled "Coronavirus" in a week [6]. According to the security report published by IBM for 2020, it was reported that the health sector was negatively affected by phishing attacks the most [7].

Internet users' card information, password information, personal or public information, etc. Developing software-based systems that can simultaneously protect their security has become an indispensable need and studies on this are still ongoing. Recently, Artificial Intelligence (AI) - based software has been developed that can provide information security on a platform with billions of users [8]. If some of these studies are examined; Ozgur Sahingoz et al [9]. detected phishing attacks with machine learning methods in their study. They used the Natural Language Processing (NLP) technique on their websites with the approach they suggested. Then, they achieved a 97.98% success rate with the Random Forest (RF) method. Erzhou Zhu et al. [10] classified phishing attacks using the Decision Tree (DT) method and the Artificial Neural Network (ANN) model together in their study. Using the Optimal feature selection (OFS) algorithm with the approach they proposed, they increased productivity. The success of classification they obtained with the feature selection method without improving the data was 95.76%. S. Sountharrajan et al. [11] performed the classification process for phishing attacks of websites in their study. They used the Deep Boltzmann Machine (DBM) and the Stacked Auto-encoder (SAE) methods to select features in the dataset. Then, they trained the dataset with the deep learning model and their classification success was 94%. Ping Yi et al. [12] using Deep Faith Networks (DBNs), detected phishing attacks of websites on internet protocols (IP). They trained the features that websites use on IP data with the DBN model. They used Boltzmann machines for the classification method and achieved an 89.6% success rate. Mustafa Kaytan et al. [13] performed the classification process by using Extreme Learning Machines (ELM) to detect phishing attacks. They separated the dataset using the cross-validation method in the classification process and their classification accuracy success was 95.93%.

This paper, using the website data where phishing attacks occurred, as input, is to detect with machine learning methods and designed a deep learning model. It was also to observe whether feature selection methods contributed to the classification process. The summary of the other section of the study is as follows; information about the data set used in the experimental analysis is given in the second section. Information about machine learning methods, deep learning models, and feature selection methods used in the experimental analysis is given in the third section. Information about the experimental analysis and results are included in the fourth section. The last two sections consisted of Discussion and Conclusion, respectively.

2. Datasets

The experiment of this study consists of the analysis of two datasets. The first dataset consists of two files with extensions "txt" and "CSV", containing 11055 website content. Each website has parameters containing 30 features, and the label groups of these parameters consist of {-1, 1} or {-1, 0, 1} values. Information about the 30 features used as parameter values is given in Table 1. The data consists of websites that are designed for the binary classification model and where phishing attacks occur and are legitimate. This dataset is made available to the public [14].

Table 1. Website parameters and values in the first dataset.

Feature no	Website parameters	Label Values	Feature no	Website parameters	Label Values
1	UsingIP	{-1, 0, 1 }	16	ServerFormHandler	{-1, 1 }
2	LongURL	{-1, 0, 1 }	17	InfoEmail	{-1, 1 }
3	ShortURL	{-1, 1 }	18	AbnormalURL	{-1, 1 }
4	Symbol@	{-1, 1 }	19	WebsiteForwarding	{-1, 0, 1 }
5	Redirecting//	{-1, 1 }	20	StatusBarCust	{-1, 1 }
6	PrefixSuffix-	{-1, 1 }	21	DisableRightClick	{-1, 1 }
7	SubDomains	{-1, 0, 1 }	22	UsingPopupWindow	{-1, 1 }
8	HTTPS	{-1, 0, 1 }	23	IframeRedirection	{-1, 1 }
9	DomainRegLen	{-1, 1 }	24	AgeofDomain	{-1, 1 }
10	Favicon	{-1, 1 }	25	DNSRecording	{-1, 0, 1 }
11	NonStdPort	{-1, 1 }	26	WebsiteTraffic	{-1, 1 }
12	HTTPSDomainURL	{-1, 1 }	27	PageRank	{-1, 1 }
13	RequestURL	{-1, 1 }	28	GoogleIndex	{-1, 0, 1 }
14	AnchorURL	{-1, 0, 1 }	29	LinksPointingToPage	{-1, 1 }
15	LinksInScriptTags	{-1, 0, 1 }	30	StatsReport	{-1, 1 }

A second dataset was used to confirm the success of the proposed approach. This dataset is made available to the public. It has two classes: phishing and legitimate. The second dataset consists of a total of 1401 websites, 901 of which are phishing and 500 are legitimate. The dataset contains 10 features for each website [15]. These features are given in Table 2. The file format of the dataset is "xlsx". For this study, we converted the file format of the second dataset to the "CSV" file type.

Table 2. Website parameters and values in the second dataset.

Feature no	Website parameters	Label Values	Feature no	Website parameters	Label values
1	Having @ Symbol	{-1, 1 }	6	No.of Dots	{-1, 1 }
2	Presence of IP Address	{-1, 1 }	7	No. of Hyphen in Host Address	{-1, 1 }
3	Length of URL	{-1, 1 }	8	"Email" Keyword	{-1, 1 }
4	No. of Slashes	{-1, 1 }	9	URLs do not have “https”	{-1, 1 }
5	Special Character	{-1, 1 }	10	Age of URL	{-1, 1 }

In all of the experimental analyzes, 30% of the datasets were separated as test data, and 70% as training data. In addition, the cross-validation method (k = 10) was chosen and applied in the classification of datasets.

3. Models, Methods, and Proposed Approach

3.1. Nearest Neighbor Method (kNN)

The kNN method is a machine learning approach that is included in the supervised learning model and, can solve the classification problem of input data in the algorithm model. In the classification process, features that are similar to data features are labeled in the same class. While calculating this, sample data features are randomly selected and the distances of other data features are calculated according to sample data features. As a result, for each feature, the number of nearest neighbor features are looked at (as many as k). Here, the parameter k is usually chosen values such as $\{2,3,5,7, ..\}$. The disadvantage of the kNN method is that it needs a memory requirement to keep distance information for each feature data [16]. For distance measurements between features; one of the "Euclidean", "Manhattan", "Minkowski" methods is preferred. With the Euclidean formula, the distance (d) is calculated according to Eq. (1). Here, the variables p and q represent features [17].

$$\mathrm{d}(\mathrm{p},\mathrm{q}) = \sqrt{(\mathrm{p}_1 - \mathrm{q}_1)^2 + (\mathrm{p}_2 - \mathrm{q}_2)^2} \tag{1}$$

In this study, the kNN method supported by the Sklearn library was used. The Euclidean method was used for distance measurement and k value was selected as three.

3.2. Decision Tree Method

The Decision Tree (DT) method, consisting of root nodes and leaf nodes; is a machine learning approach preferred in classification processes. In the classification process, nodes are divided into sub-nodes using recursive methods, and this situation continues until it does not affect the classification process [18]. In the classification process, the "Entropy" method called information gain measurement is used to distinguish the data features. The uncertainties of the data are measured with the entropy method and the data features are classified with the obtained probability values. The formula used for measuring entropy (E) is given in Eq. (2). While N variable expresses the data number in Eq. (2). P is i. refers to the probability value of the data [19].

$$E = -\sum_{i=1}^{N} P_i \log_2 P_i \tag{2}$$

In this study, other important parameters were preferred for the DT method; criterion value Gini was selected, maximum depth value was selected 30. Other parameter values are the default values accepted in the Sklearn library [20].

3.3. Random Forest Method

Random Forest (RF) method is a machine learning approach used in regression and classification processes. The RF method creates multiple sets of decision trees and then combines these clusters to make the classification process more accurate. It aims to bring together as many different decision trees as possible, thus creating a low-correlation forest community. In the classification process, random nodes are selected and the best node is chosen among randomly selected variables. The Gini parameter is used to measure the homogeneity of classes. If the Gini measurement value of the lower node is lower than the Gini measurement value of the parent node, the branch with the nodes is considered successful. The Gini measurement is calculated according to Eq. (3). All data is represented by variable N and selected data is represented by n. Also, the variable p_i represents the square of the result consisting of the division of the number of elements smaller than the selected data and the number of elements larger than itself [21,22].

$$Gini(N) = 1 - \sum_{i=1}^{n} {p_i}^2 \tag{3}$$

In this study, other important parameters were preferred for the RF method; criterion value Gini was selected, maximum depth value was selected 30. Other parameter values are the default values accepted in the Sklearn library.

3.4. Feature Selection Methods

The minimum Redundancy Maximum Relevance (mRMR) algorithm is a filtering method that tries to select the most relevant features among the features it obtains from the data. It performs a ranking among the features it associates and tries to minimize the weakest associated features (redundancy features). In other words; treats each feature as a separate overlap variable and measures the level of similarity between the two features using mutual information between them [23]. Here, f_i is defined as the variable representing the features, and the ordering of the features vectorially is in the form of $f_i = [\, {f_i}^1, {f_i}^2, {f_i}^3, \ldots, {f_i}^N]$. The $I(F_i, F_j)$ variable carries the mutual information values between the i and j features. The mRMR algorithm measures the similarity between two features and also measures the similarity between class tags for each feature. This situation is represented vectorially by the variable $h = [\, h^1, h^2, \ldots, h^N]$ and the measurement value is represented by the variable $I(H, F_i)$. Here S represents the selected set of features and $|S|$ represents the number of selected features. The best properties are achieved by meeting two conditions (minimum redundancy and maximum relevance). For the minimum redundancy, the formula in Eq. (4) is used, and for the maximum relevance, the formula in Eq. (5) is used. Combined combination of two conditions is given in Eq. (6) and Eq. (7) [24,25].

$$minW, W = \frac{1}{|S|^2} \sum_{F_i F_j \in S} I(F_i, F_j) \tag{4}$$

$$maxV, V = \frac{1}{|S|} \sum_{F_i \in S} I(F_i, H) \tag{5}$$

$$\max\,(V - W) \tag{6}$$

$$\max\,({}^{V}/_{W}) \tag{7}$$

The LASSO is preferred for the methods in which textual-based statistical calculations are performed and was created inspired by the linear regression method. This method minimizes the sum of square errors with an upper bound on the sum of the absolute feature values by processing the feature values obtained from the data. The formula used in the LASSO method is given in Eq. (8). The variable λ represents the amount of shrinkage. The variable t is inversely proportional to the variable λ, and as t becomes infinite, the problem becomes ordinary least squares. The variable β_j represents the coefficient and depends on the variable λ. In this case, features with a coefficient value equal to zero are extracted by the algorithm. The X and Y variables represent the features associated with the algorithm [26].

$$\min\left(\frac{\|Y - X\beta\|_2^2}{n}\right) ; \sum_{j=1}^{k}\|\beta\|_1 < t \quad (8)$$

In this study, the mRMR [25] and LASSO [27] feature selection algorithms were compiled in Python and used for two datasets. Among the 30 features in the first dataset, the feature with a low level of relation was extracted and classification was performed with 29 features. Among the 10 features in the second dataset, the feature with a low level of association was removed and classification was made with 9 features.

3.5. Deep Learning Model

In this study, the sequential model of the Keras library is used [28]. The designed model consists of Dense layers. Dense layers are regular layers and are called Dense because each neuron receives all the neurons from the previous layers as input [29]. For the first dataset, the number of input neurons of the experiment layer was 60 and the number of hidden neurons was chosen as 30. For the second dataset, the number of input neurons of the experiment layer was 30 and the number of hidden neurons was chosen as 10. The activation function was selected as ReLU. ReLU is a unit of functions that linearizes nonlinear input values [30]. The ReLU function takes $[0, +\infty]$ values and does not contain negative values. The Adam was chosen as the optimization method and the learning rate of this optimization method was selected 10^{-3}. The data showing the layer and parameter values of the designed model are given in Table 3. Also, the loss parameter of the designed model "mean_squared_error" was selected. After trying all the used parameters of this model, the ones showing the best success were preferred.

Table 3. Deep learning model layers were designed for this study.

Layer	Activation	Input Neuron	Hidden Neuron	Input Size
Dense	ReLU	Input dimensions	Input dimensions	Input dimensions
Dense	ReLU	Dataset #1: 60 Dataset #2: 30	Dataset #1: 30 Dataset #2: 10	-
Dense	ReLU	1	-	-
Output		ReLU		Two Classes

3.6. Proposed Approach

The proposed approach is aimed at successfully detecting phishing attacks on the website that have recently increased their effectiveness. Feature selection methods (mRMR, LASSO) were used to accomplish this. Among the parameter features of the websites, the most inefficient feature was removed and given as an input to the machine learning and deep model network. Thus, the detection of phishing attacks was achieved with a hybrid approach. The overall design of the proposed approach is shown in Fig. 1. The experimental analysis codes of this study were compiled in Python language [31].

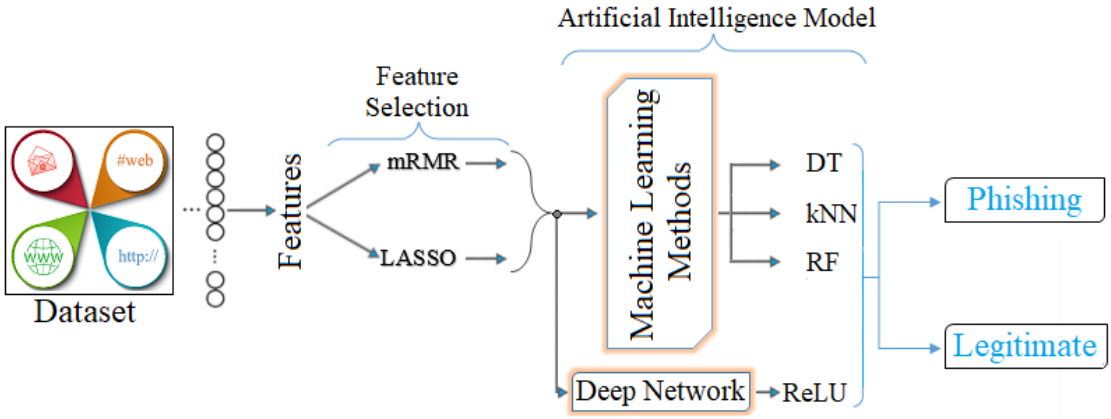

Figure 1. The general design of the proposed artificial intelligence supported approach.

4. Experimental Analysis and Results

Experimental analyzes were carried out using Python software codes. The Google Colab server was used for hardware requirements and for compiling the codes. Confusion matrix was used as a measurement in experimental analysis. The measurement metrics of the confusion matrix are; Specificity (Spe), Prediction (Pre), Recall, F-score (F-scr), and Accuracy (Acc). Metric results are calculated by equations between Eq. (9) and Eq. (13). Meanings of variables used in equations; True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) [32,33].

$$\text{Spe} = \frac{\text{TN}}{\text{TN+FP}} \tag{9}$$

$$\text{Pre} = \frac{\text{TP}}{\text{TP+FP}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{11}$$

$$\text{F-scr} = \frac{2\text{x (Recall x Pre)}}{\text{Recall+Pre}} \tag{12}$$

$$\text{Acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{13}$$

The experimental analysis consisted of two steps for each dataset. In the first step, the success of machine learning methods and designed deep network model is analyzed. In the second step, it was analyzed whether feature selection methods (Lasso and mRMR) contributed to the success achieved. In all steps of the experiment, the data sets were separated as 30% test data and analyzes were carried out. In addition, the validity of the analyzes was checked using the cross-validation method (choosing k = 10).

The first analysis was performed using 30% test data of the first dataset. Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 96.26% was achieved with the DT method; an accuracy success of 93.46% was achieved with the kNN method; an accuracy success of 96.83% was achieved with the RF method, and an accuracy of 95.57% was achieved with the deep network model. The confusion matrices of the first analysis are shown in Fig. 2. The second analysis was performed by separating the data with the cross-validation method of the first dataset. Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 95.90% was achieved with the DT method; an accuracy of 93.37% was achieved with the kNN method; an accuracy of 97.03% was achieved with the RF method and an accuracy of 95.50% with the deep network model. The accuracy graphs of the second analysis are shown in Fig. 4. The accuracy results of the second analysis and first analysis approximately corresponded to each other. In both analyzes, it was observed that the RF method was more successful. The third analysis was performed using 30% test data of the second dataset. Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 94.05% was achieved with the DT method; an accuracy success of 92.62% was achieved with the kNN method; an accuracy success of 94.05% was achieved with the RF method, and an accuracy of 93.81% was achieved with the deep network model. The confusion matrices of the third analysis are shown in Fig. 3. The fourth analysis was performed by separating the data with the cross-validation method of the second dataset. Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 94.18% was achieved with the DT method; an accuracy of 93.98% was achieved with the kNN method; an accuracy of 94.18% was achieved with the RF method and an accuracy of 94.18% with the deep network model. The accuracy graphs of the fourth analysis are shown in Fig. 5. The accuracy results of the third analysis and fourth analysis approximately corresponded to each other. In both analyzes, it was observed that the DT, RF, Deep Network methods were more successful. Detailed analysis results of the confusion matrices obtained from the first step of the experiment are given in Table 4.

In the second step of the experiment, the feature selection methods (Lasso, mRMR) made a classification by determining the most inefficient feature among the features obtained from datasets. Thus, 30 features in the first dataset were reduced to 29 features, and 10 features in the second dataset were reduced to 9 features. Then, the analyzes performed in the first step of the experiment were re-performed. The Lasso feature selection method was used for the fifth, sixth, seventh, eighth analyzes. In the ninth, tenth, eleventh, twelfth analyzes, the mRMR feature selection method was used. The fifth analysis was performed using 30% test data of the first dataset (29 featured). Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model.

An accuracy success of 96.50% was achieved with the DT method; an accuracy success of 93.76% was achieved with the kNN method; an accuracy success of 96.83% was achieved with the RF method, and an accuracy of 95.75% was achieved with the deep network model. The confusion matrices of the fifth analysis are shown in Fig. 6. The sixth analysis was performed by separating the data with the cross-validation method of the first dataset. Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 96.04% was achieved with the DT method; an accuracy of 93.29% was achieved with the kNN method; an accuracy of 97.05% was achieved with the RF method and an accuracy of 95.98% with the deep network model. The accuracy graphs of the sixth analysis are shown in Fig. 7. As a result of the fifth and sixth analyzes, it was observed that the Lasso method contributed to the classification success. Other than the kNN method, the success of other analysis results has increased. The seventh analysis was performed using 30% test data of the second dataset (9 featured). Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 94.05% was achieved with the DT method; an accuracy success of 92.86% was achieved with the kNN method; an accuracy success of 94.29% was achieved with the RF method, and an accuracy of 93.81% was achieved with the deep network model. The confusion matrices of the seventh analysis are shown in Fig. 8. The eighth analysis was performed by separating the data with the cross-validation method of the second dataset. Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 94.18% was achieved with the DT method; an accuracy of 92.84% was achieved with the kNN method; an accuracy of 94.29% was achieved with the RF method and an accuracy of 94.18% with the deep network model. The accuracy graphs of the eighth analysis are shown in Fig. 9. As a result of the seventh and eighth analyzes, it was observed that the Lasso method contributed to the classification success. The reduction of the number of features with the Lasso method was equivalent to the results obtained in previous analyzes or increased.

Table 4. Analysis results of datasets obtained by machine learning methods and deep network model (%).

Dataset	Test Rate	Feature	Model, Method	Spe	Pre	Recall	F-scr	Acc
Dataset #1	30%	30	DT	95.06	96.08	97.23	96.65	96.26
			kNN	92.29	93.83	94.40	94.11	93.46
			RF	95.67	96.56	97.77	97.16	96.83
			Deep Network	96.35	97.0	94.94	95.96	95.57
Dataset #2	30%	10	DT	99.30	98.26	83.09	90.04	94.05
			kNN	97.18	93.39	83.09	87.94	92.62
			RF	99.30	98.26	83.09	90.04	94.05
			Deep Network	99.30	98.25	82.35	89.60	93.81

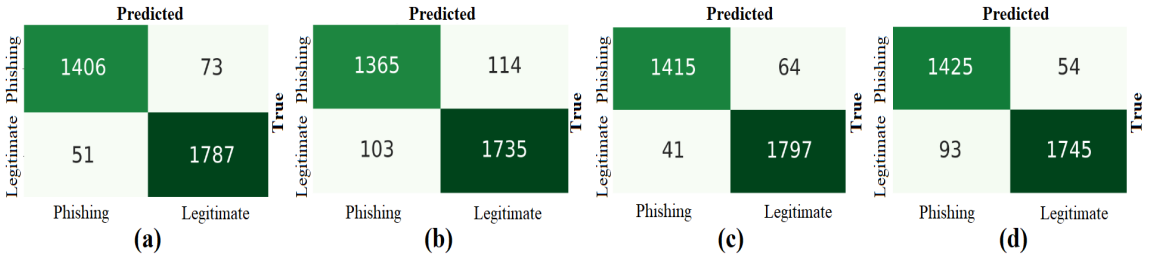

Figure 2. Confusion matrices were obtained from 30% test data of the first dataset.

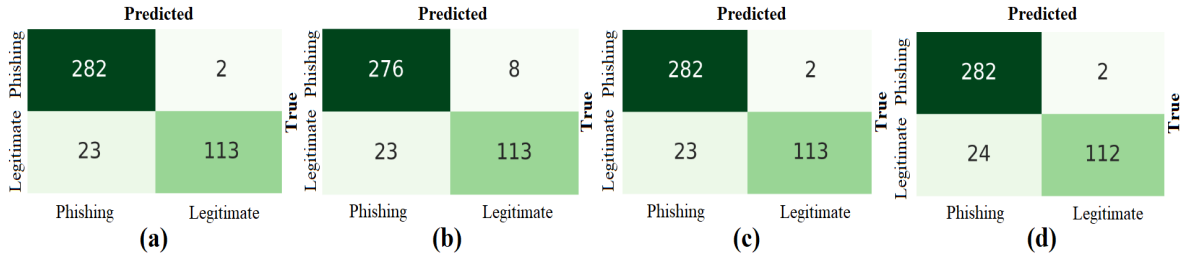

Figure 3. Confusion matrices were obtained from 30% test data of the second dataset.

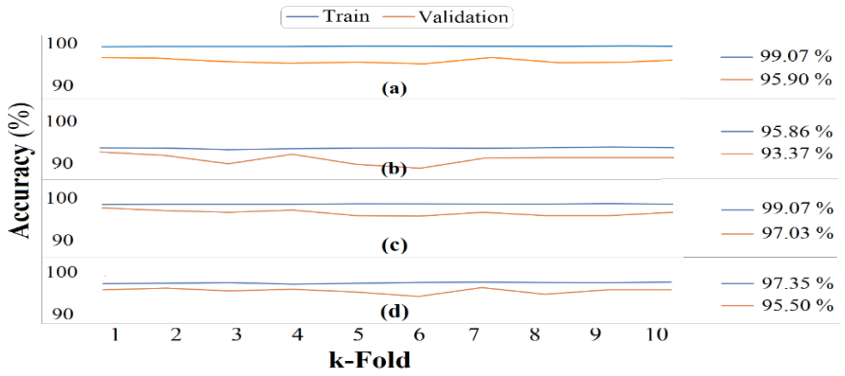

Figure 4. Success graphs and accuracy results were obtained from data separated by the cross-validation method of the first dataset.

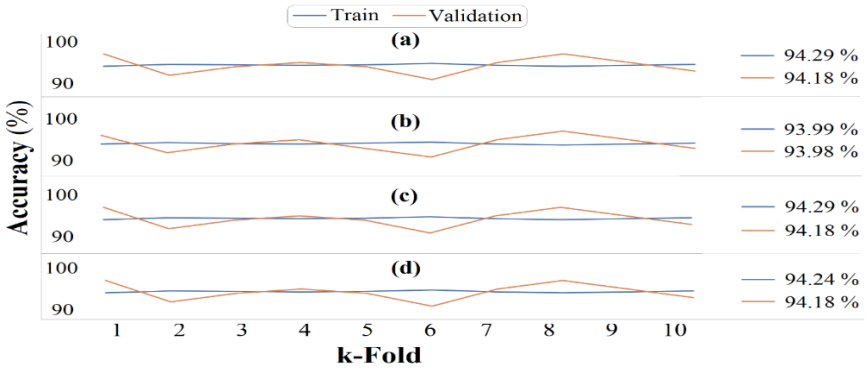

Figure 5. Success graphs and accuracy results were obtained from data separated by the cross-validation method of the second dataset.

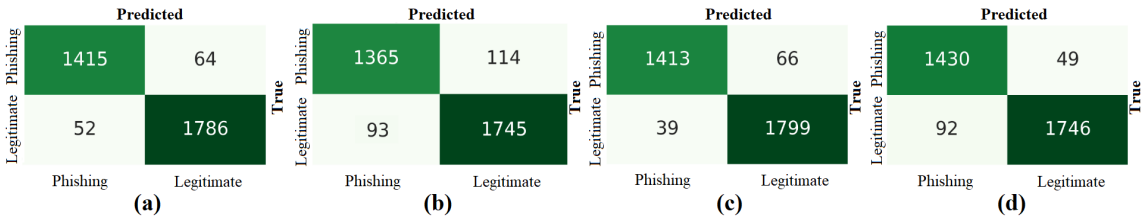

Figure 6. Confusion matrices were obtained after applying the Lasso feature selection method to 30% test data of the first dataset.

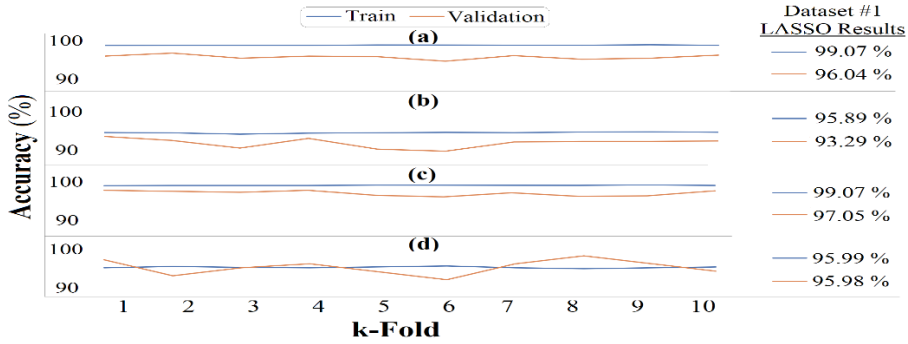

Figure 7. Success graphs and accuracy results were obtained from data separated by the cross-validation method after the Lasso feature selection method of the first dataset.

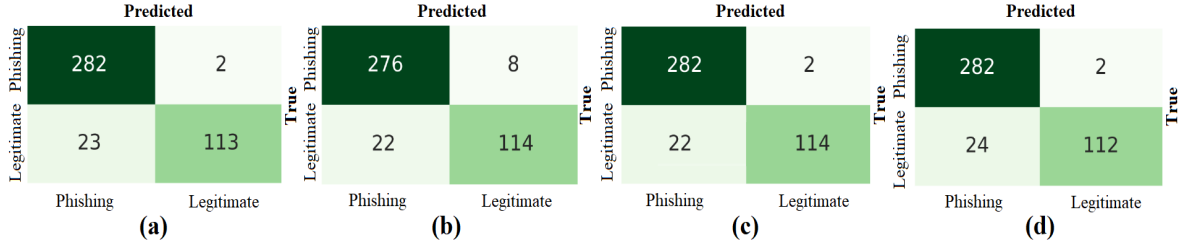

Figure 8. Confusion matrices were obtained after applying the Lasso feature selection method to 30% test data of the second dataset.

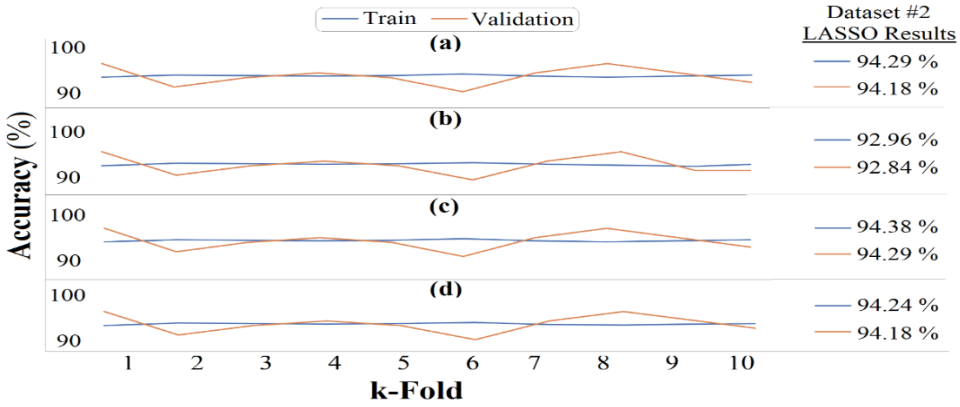

Figure 9. Success graphs and accuracy results were obtained from data separated by the cross-validation method after the Lasso feature selection method of the second dataset.

The ninth analysis was performed using 30% test data of the first dataset (29 featured). Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 96.59% was achieved with the DT method; an accuracy success of 93.76% was achieved with the kNN method; an accuracy success of 97.26% was achieved with the RF method, and an accuracy of 96.14% was achieved with the deep network model. The confusion matrices of the ninth analysis are shown in Fig. 10. The tenth analysis was performed by separating the data with the cross-validation method of the first dataset. Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 96.54% was achieved with the DT method; an accuracy of 93.41% was achieved with the kNN method; an accuracy of 96.96% was achieved with the RF method and an accuracy of 95.78% with the deep network model. The accuracy graphs of the ninth analysis are shown in Fig. 11. The validity of the accuracy results obtained from the ninth analysis was confirmed by the tenth analysis. The eleventh analysis was performed using

30% test data of the second dataset (9 featured). Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 94.29% was achieved with the DT method; an accuracy success of 92.62% was achieved with the kNN method; an accuracy success of 94.76% was achieved with the RF method, and an accuracy of 93.81% was achieved with the deep network model. The confusion matrices of the eleventh analysis are shown in Fig. 12. The twelfth analysis was performed by separating the data with the cross-validation method of the second dataset. Accuracy achievements were analyzed using machine learning methods (DT, kNN, RF) and deep network model. An accuracy success of 94.18% was achieved with the DT method; an accuracy of 93.98% was achieved with the kNN method; an accuracy of 94.18% was achieved with the RF method and an accuracy of 93.69% with the deep network model. The accuracy graphs of the twelfth analysis are shown in Fig. 13. As a result of the eleventh and twelfth analyzes, it was observed that the mRMR method contributed to the classification success.

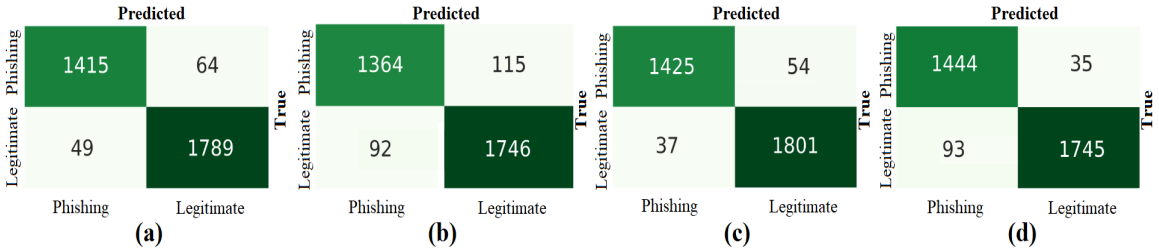

Figure 10. Confusion matrices were obtained after applying the mRMR feature selection method to 30% test data of the first dataset.

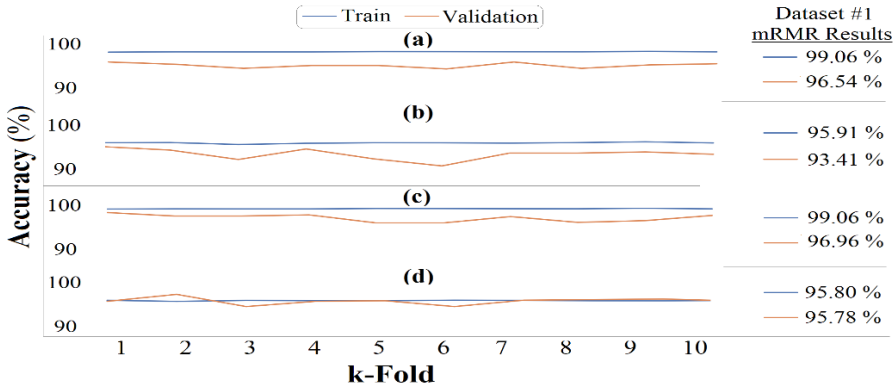

Figure 11. Success graphs and accuracy results were obtained from data separated by the cross-validation method after the mRMR feature selection method of the first dataset.

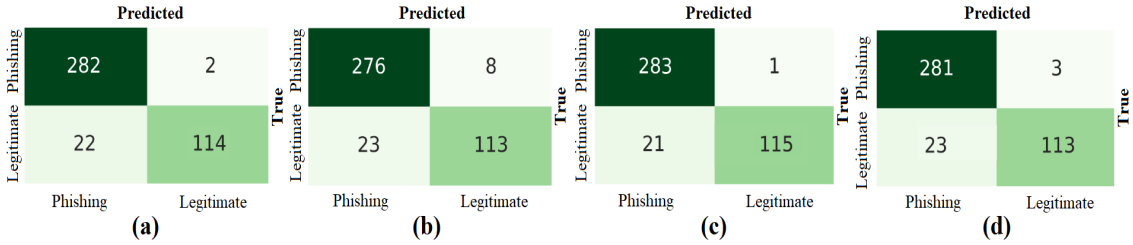

Figure 12. Confusion matrices were obtained after applying the mRMR feature selection method to 30% test data of the second dataset.

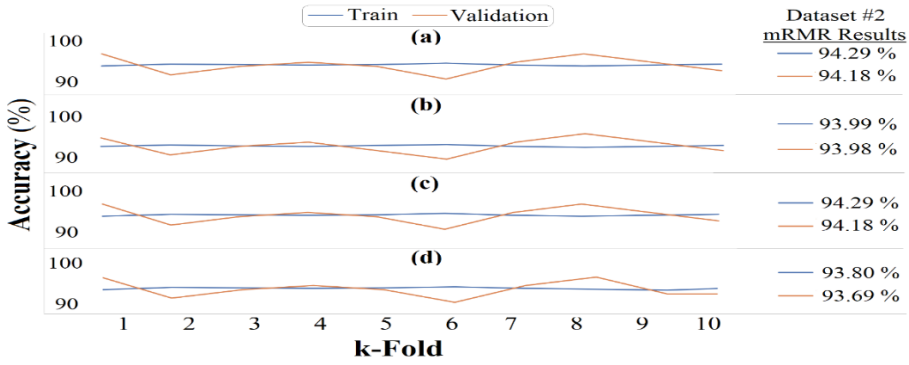

Figure 13. Success graphs and accuracy results were obtained from data separated by the cross-validation method after the mRMR feature selection method of the second dataset.

Detailed results of the analyzes performed in the second step of the experiment are given in Table 5. These results showed us the efficiency of feature selection methods in two datasets. It was also observed that the mRMR method was more effective than the Lasso method.

Table 5. Analysis results were obtained from 30% of test data after the feature selection methods are applied to datasets (%).

Dataset	Test Rate	Method, Total Feature	Model, Method	Spe	Pre	Recall	F-scr	Acc
Dataset #1	30%	LASSO / 29	DT	95.67	96.54	97.17	96.85	96.50
			kNN	93.87	93.62	93.74	92.95	93.76
			RF	95.54	96.46	97.88	97.16	96.83
			Deep Network	96.69	97.27	94.99	96.12	95.75
Dataset #2	30%	LASSO / 9	DT	99.30	98.26	83.09	90.04	94.05
			kNN	97.18	93.44	93.03	94.85	92.86
			RF	98.28	99.30	95.52	95.92	94.29
			Deep Network	99.30	98.25	82.35	89.60	93.81
Dataset #1	30%	mRMR / 29	DT	96.55	95.67	96.60	96.16	96.59
			kNN	93.82	92.22	93.70	92.95	93.76
			RF	97.09	96.35	97.28	96.91	**97.26**
			Deep Network	98.03	97.63	95.99	95.76	96.14
Dataset #2	30%	mRMR / 9	DT	98.28	99.30	95.52	95.92	94.29
			kNN	97.18	93.39	83.09	87.94	92.62
			RF	99.14	99.65	96.11	96.26	**94.76**
			Deep Network	97.41	98.94	94.92	95.58	93.81

5. Discussion

In this paper, phishing attacks against websites that threaten information security were detected. The approach suggested in my study has been successful in detecting e-fraud. Among the limited aspects of my proposed approach; it can be shown that the number of features of the dataset is not too much. This situation limited the success of the classification process. But, perhaps a combination of natural language processing methods for features in datasets would increase success. Because the datasets had information such as the domain names of the websites. I did not use this information in the suggested approach. In the proposed approach, feature selection methods affected the results positively. The success graph showing this performance situation is shown in Fig. 14. It was observed in Fig. 14 that the Lasso and mRMR method effectively increase the success. In addition, we used

the dataset training-test and cross-validation methods to reliable the proposed approach. As a result, I have seen through experimental analysis that our approach applies to the detection of phishing attacks.

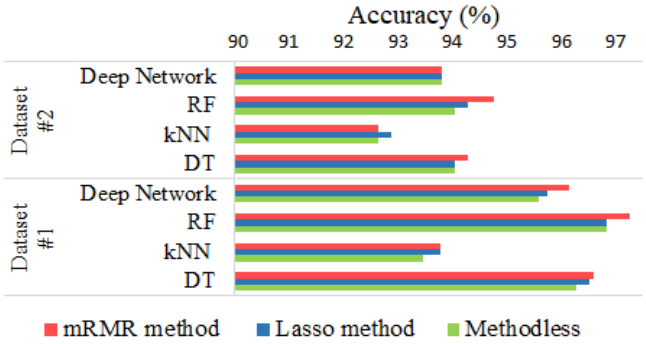

Figure 14. Bar graph showing the performance of the proposed approach on datasets.

6. Conclusion

This paper analyzed the AI-assisted detection of phishing attacks on websites. In the internet environment with billions of users, such attacks are frequently seen and the AI-based approaches are needed to minimize these attacks [34]. The experimental analysis of this study shows that the AI-based approaches are successful in the detection of phishing attacks. In the analysis of this study, the AI-based machine learning methods (DA, kNN, RF) were used to perform the classification process. In addition, the designed deep network model has also been used in the training and classification process of datasets. Feature selection methods (Lasso, mRMR) made the most important contribution to the proposed approach. The best performance results in both datasets were achieved with RF - mRMR methods. Analysis results of the first dataset with RF-mRMR methods; specificity success was 97.09%, prediction success was 96.35%, recall success was 97.28%, f-score success was 96.91% and Accuracy success was 97.26%. Likewise, analysis results of the second dataset with RF-mRMR methods; specificity success was 99.14%, prediction success 99.65%, recall success 96.11%, f-score success 96.26% and accuracy success 94.76%.

In the next study, the proposed approach for phishing attacks in web applications is planned to be used together with meta-heuristic optimization methods.

Funding

There is no funding source for this article.

Ethical approval

This article does not contain any data, or other information from studies or experimentation, with the involvement of human or animal subjects.

Conflicts of interest

The author declares that there is no conflict of interest related to this paper.

References

[1] S.S.M. Motiur Rahman, T. Islam, M.I. Jabiullah, PhishStack: Evaluation of Stacked Generalization in Phishing URLs Detection, Procedia Comput. Sci. 167 (2020) 2410–2418. doi:https://doi.org/10.1016/j.procs.2020.03.294.

[2] H. Önal, Phishing (Oltalama) Saldırısı Nedir? | BGA Security, BGA Secur. (2021). https://www.bgasecurity.com/2019/09/phishing-oltalama-saldirisi-nedir/ (accessed June 10, 2021).

[3] D. Goel, A.K. Jain, Mobile phishing attacks and defence mechanisms: State of art and open research challenges, Comput. Secur. 73 (2018) 519–544. doi:https://doi.org/10.1016/j.cose.2017.12.006.

[4] WANDERA, Mobile Threat Landscape Report 2020 | Wandera, 2020. https://www.wandera.com/mobile-threat-landscape/ (accessed August 18, 2020).
[5] APWG, Phishing Activity Trends Report Q1 2020, 2020. www.apwg.org.
[6] Phishing Statistics: The 29 Latest Phishing Stats to Know in 2020 - Hashed Out by The SSL Store™, Hashedout. (2021). https://www.thesslstore.com/blog/phishing-statistics-latest-phishing-stats-to-know/ (accessed June 10, 2021).
[7] M. Abdelhamid, The Role of Health Concerns in Phishing Susceptibility: Survey Design Study, J Med Internet Res. 22 (2020) e18394. doi:10.2196/18394.
[8] J. Chen, C. Su, Z. Yan, AI-Driven Cyber Security Analytics and Privacy Protection, Secur. Commun. Networks. 2019 (2019) 1859143. doi:10.1155/2019/1859143.
[9] O.K. Sahingoz, E. Buber, O. Demir, B. Diri, Machine learning based phishing detection from URLs, Expert Syst. Appl. 117 (2019) 345–357. doi:https://doi.org/10.1016/j.eswa.2018.09.029.
[10] E. Zhu, Y. Ju, Z. Chen, F. Liu, X. Fang, DTOF-ANN: An Artificial Neural Network phishing detection model based on Decision Tree and Optimal Features, Appl. Soft Comput. 95 (2020) 106505. doi:https://doi.org/10.1016/j.asoc.2020.106505.
[11] S. Sountharrajan, M. Nivashini, S.K. Shandilya, E. Suganya, A.B. Banu, M. Karthiga, Advances in Cyber Security Analytics and Decision Systems, 2020. doi:10.1007/978-3-030-19353-9.
[12] P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang, T. Zhu, Web Phishing Detection Using a Deep Learning Framework, Wirel. Commun. Mob. Comput. 2018 (2018) 4678746. doi:10.1155/2018/4678746.
[13] M. Kaytan, D. Hanbay, Effective Classification of Phishing Web Pages Based on New Rules by Using Extreme Learning Machines, Anatol. J. Comput. Sci. 2 (2017) 15–36. https://dergipark.org.tr/download/article-file/333655.
[14] E. Chand, Phishing website Detector, Kaggle. (2021). https://www.kaggle.com/eswarchandt/phishing-website-detector (accessed June 7, 2021).
[15] A. J., Phishing Websites Detection, Kaggle. (2020). https://www.kaggle.com/akshaya1508/phishing-websites-detection? (accessed August 19, 2020).
[16] C. Sitawarin, D. Wagner, On the Robustness of Deep K-Nearest Neighbors, in: 2019 IEEE Secur. Priv. Work., 2019: pp. 1–7. doi:10.1109/spw.2019.00014.
[17] A. Niwatkar, Y.K. Kanse, Feature Extraction using Wavelet Transform and Euclidean Distance for speaker recognition system, in: 2020 Int. Conf. Ind. 4.0 Technol., 2020: pp. 145–147. doi:10.1109/I4Tech48345.2020.9102683.
[18] S. Zhang, Z. Shi, G. Wang, R. Yan, Z. Zhang, Groundwater radon precursor anomalies identification by decision tree method, Appl. Geochemistry. 121 (2020) 104696. doi:https://doi.org/10.1016/j.apgeochem.2020.104696.
[19] Y. Wang, S.-T. Xia, J. Wu, A less-greedy two-term Tsallis Entropy Information Metric approach for decision tree classification, Knowledge-Based Syst. 120 (2017) 34–42. doi:https://doi.org/10.1016/j.knosys.2016.12.021.
[20] Scikit-learn kütüphanesi, Scikit Learn. (2021). https://scikit-learn.org/stable/supervised_learning.html#supervised-learning (accessed August 20, 2020).
[21] C. Aldrich, Process variable importance analysis by use of random forests in a shapley regression framework, Minerals. 10 (2020) 1–17. doi:10.3390/min10050420.
[22] H. Han, X. Guo, H. Yu, Variable selection using Mean Decrease Accuracy and Mean Decrease Gini based on Random Forest, in: 2016 7th IEEE Int. Conf. Softw. Eng. Serv. Sci., 2016: pp. 219–224. doi:10.1109/ICSESS.2016.7883053.
[23] M. Toğaçar, B. Ergen, Z. Cömert, Detection of lung cancer on chest CT images using minimum redundancy maximum relevance feature selection method with convolutional neural networks, Biocybern. Biomed. Eng. (2019). doi:https://doi.org/10.1016/j.bbe.2019.11.004.
[24] M. Cıbuk, U. Budak, Y. Guo, M.C. Ince, A. Sengur, Efficient deep features selections and classification for flower species recognition, Measurement. 137 (2019) 7–13. doi:https://doi.org/10.1016/j.measurement.2019.01.041.
[25] Hanchuan Peng, Python binding to mRMR Feature Selection algorithm, (n.d.). https://github.com/fbrundu/pymrmr.
[26] V. Fonti, Feature Selection using LASSO, VU Amsterdam. (2017) 1–26. doi:10.1109/access.2017.2696365.
[27] Feature selection - LASSO, Scikit-Learn. (2020). https://scikit-learn.org/stable/modules/feature_selection.html (accessed August 21, 2020).
[28] The Sequential model, Keras. (2020). https://keras.io/api/models/sequential/ (accessed August 21, 2020).
[29] Y. Yang, S. Liu, Non-porous thin dense layer coating: Key to achieving ultrahigh peak capacities using narrow open tubular columns, Talanta Open. 1 (2020) 100003. doi:https://doi.org/10.1016/j.talo.2020.100003.
[30] D. Zou, Y. Cao, D. Zhou, Q. Gu, Gradient descent optimizes over-parameterized deep ReLU networks, Mach. Learn. 109 (2020) 467–492. doi:10.1007/s10994-019-05839-6.
[31] S.A. Khan, Phishing Websites Classification using Deep Learning, GitHub. (2020). https://github.com/sohailahmedkhan173/Phishing-Websites-Classification-using-Deep-Learning (accessed August 9, 2020).
[32] D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, BMC Genomics. 21 (2020) 6. doi:10.1186/s12864-019-6413-7.
[33] M. Toğaçar, B. Ergen, Deep Learning Approach for Classification of Breast Cancer, in: 2018 Int. Conf. Artif. Intell. Data Process., 2018: pp. 1–5. doi:10.1109/idap.2018.8620802.
[34] T. Lin, D.E. Capecci, D.M. Ellis, H.A. Rocha, S. Dommaraju, D.S. Oliveira, N.C. Ebner, Susceptibility to Spear-Phishing Emails: Effects of Internet User Demographics and Email Content, ACM Trans. Comput. Hum. Interact. 26 (2019) 32. doi:10.1145/3336141.