Journal of Cyber Security

Tech Science Press

DOI: 10.32604/jcs.2023.045579

ARTICLE

Empirical Analysis of Neural Networks-Based Models for Phishing Website Classification Using Diverse Datasets

Shoaib Khan, Bilal Khan, Saifullah Jan*, Subhan Ullah and Aiman

Department of Computer Science, City University of Science and Information Technology, Peshawar, Pakistan

*Corresponding Author: Saifullah Jan. Email: saifjan2@hotmail.com

Received: 31 August 2023 Accepted: 21 November 2023 Published: 28 December 2023

ABSTRACT

Phishing attacks pose a significant security threat by masquerading as trustworthy entities to steal sensitive information, a problem that persists despite user awareness. This study addresses the pressing issue of phishing attacks on websites and assesses the performance of three prominent Machine Learning (ML) models—Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM)—utilizing authentic datasets sourced from Kaggle and Mendeley repositories. Extensive experimentation and analysis reveal that the CNN model achieves a better accuracy of 98%. On the other hand, LSTM shows the lowest accuracy of 96%. These findings underscore the potential of ML techniques in enhancing phishing detection systems and bolstering cybersecurity measures against evolving phishing tactics, offering a promising avenue for safeguarding sensitive information and online security.

KEYWORDS

Artificial neural networks; phishing websites; network security; machine learning; phishing datasets; classification

1 Introduction

Phishing is an attack vector used by phishers to get access to sensitive information. This information can be a user's bank account, credit card number, personal information, etc. In phishing, the phishers directly attack the user by impersonating a legitimate authority. Phishing is not like your traditional hacker, which attacks the software, rather it is a form of social engineering. How can phishing be a security threat, even though many users know about the risks and potential impact? According to the Federal Bureau of Investigation (FBI), phishing was the most frequent sort of cybercrime in 2020, with phishing reports increasing 11 times over the previous year. According to the Internet Crime Complaint Centre (IC3), the number of phishing events nearly quadrupled from 114,702 in 2019 to 241,324 in 2020 [1]. In 2020, over 75% of organizations worldwide suffered some type of phishing assault. Another 35% were victims of spear phishing, and 65% were victims of Business Email Compromise (BEC) assaults [2]. The phishing attacks are classified into four categories that are:

Deceptive Phishing: Deceptive phishing is the most common type of phishing attack. As the name suggests, in deceptive phishing the phisher impersonates a known organization to gain access to sensitive information [3].

Spear Phishing: In spear phishing, phishers send emails to the victim, which contain malicious URLs. The URLs contain personal information about the victim [4].

Whale Phishing: In whale phishing, the phishers target top-level management of an organization like CEOs, business leaders, etc. In whale phishing, the attack aims to extract confidential information about the organization. The phishers may impersonate the CEO of the organization and gain access to the organization's confidential information [5].

URL Phishing: As the name suggests, In URL phishing the phishers use URLs to carry out phishing attacks. URL phishing is carried out using misspelled URLs, hidden links, or an email [6].

To solve the issue of phishing website classification, various studies have proposed different approaches with their strength and weaknesses. However, this study proposed a comparative analysis of different neural network-based models for phishing website classification. These models include Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM). The analyses are performed using some of the standard datasets taken from Kaggle and Mendeley repositories, respectively [6]. The performance of each employed model is compared based on accuracy, recall, precision, F-measure, and Matthew's Correlation Coefficient (MCC). Previous phishing detection machine learning models had issues with poor performance, low-quality datasets, and limited capacity to adjust to changing threats. It was frequently difficult for these models to obtain good recall, accuracy, precision, and F-measure. Their generalization was hampered by biases and restrictions in the dataset, and they were ill-prepared to deal with phishing strategies that were evolving quickly. We used high-quality datasets to compare machine learning models in our solution, which allows for better model selection, greater real-world applicability, and better flexibility to changing threats.

This work made a significant contribution to the world of cybersecurity by tackling the key issue of phishing assaults using Machine Learning (ML) techniques. Using real-world datasets from Kaggle and Mendeley repositories, the authors evaluated the performance of several ML models, including ANN, CNN, and LSTM, through extensive testing and analysis. To examine the study's success, we used a variety of assessment criteria such as accuracy, precision, recall, F-measure, and MCC. The study's findings show that ML-based phishing detection works effectively. Overall, the work advances the area of cybersecurity by demonstrating the potential of machine learning in improving phishing detection systems and increasing overall cyber defense. The importance of this work lies in the application of machine learning models to improve phishing detection. The article presents a possible path for strengthening cybersecurity safeguards against phishing attempts by comparing the models and assessing their effectiveness with real-world datasets. This discovery not only represents a possible paradigm change in cybersecurity tactics, but it also adds to the larger area of ML's use in cyber threat mitigation, implying greater resilience in online security systems.

The rest of the paper is organized as: Section 2 presents the literature review, Section 3 presents the experimental setup, Section 4 discusses the results obtained and finally, Section 5 concludes the study.

2 Related Work

The steps in the ML process include data collection, data preprocessing, data splitting, data feeding to the algorithm, and accuracy testing of the algorithm. The data in supervised learning are labeled. Studies of many kinds have been carried out to stop phishing attempts. These investigations

make use of ML algorithms, fuzzy logic, and surveys to discover fresh ways for phishers to acquire private data. The study that has been done by researchers to stop phishing is listed below.

Mohammad et al. [7] categorized several website elements which are based on characteristics from the domain, the address bar, abnormalities, HTML, and JavaScript. After which the dataset was sent to a classifier. These algorithms are Classification Based on Associations (CBA), Repeated Incremental Pruning to Produce Error Reduction (RIPPER), PRISM, and Decision Tree (DT). With a 5.76% average mistake rate, DT surpasses RIPPER, PRISM, and CBA, while RIPPER comes in second with a 5.94% average error rate. While PRISM recorded the greatest average error rate at 21.24%. Another method proposed by Mohammad et al. [8] was based on self-structuring neural networks, where they forecasted phishing websites. A testing accuracy of 92.18% was obtained. A method was proposed by Kaytan et al. [9], where they classified phishing websites using an Extreme Machine Learning (EML) algorithm. The authors suggested specific algorithms with specific rules for specific attributes. The average classification accuracy for the EML algorithm used in this investigation was 95.05%. Musa et al. [10] performed a comparative analysis on Extreme Gradient Boosting (XGBoost). Evaluation matrices used are precision, recall, F-measure, accuracy, and MCC. They got an accuracy of 97% on XGBoost. Chatterjee et al. [11] used a deep reinforcement algorithm in the detection of phishing websites. In reinforcement learning, the agent interacts with the environment and obtains states, and based on that it takes action. The evaluation matrices used in this research are precision, recall, accuracy, and F-measure. This deep reinforcement learning algorithm got an accuracy of 90%. Zhu et al. [12] presented a Neural Network (NN) model with optimal feature selection. The whole NN model's accuracy is 99.93%.

Phishing website identification was carried out by Zamir et al. [13] utilizing a variety of ML methods. K-Nearest Neighbor (KNN), Naïve Bayes (NB), Random Forest (RF), Support Vector Machines (SVM), bagging, NN, and XGBoost are the methodologies employed. They attained a 97% accuracy rate for stacking (NN, RF, and bagging). Kumar et al. [6] performed a comparative analysis of ML algorithms for the detection of phishing websites. The classifiers used are RF, Logistic Regression (LR), Gaussian Naïve Bayes (GNB), DT, and KNN. The highest accuracy of 97% is achieved by KNN. A phishing website detection model based on a supervised ML algorithm was proposed by Ali [14]. In which he used wrapper features selection to increase the correct prediction rate. The measures used are true positive rate, true negative rate, correct classification rate, and geometric mean. The highest geometric mean is archived by RF, which is 0.971. In another research, Karabatak et al. [15] performed a comparative analysis between classifiers and feature selection algorithms. The highest accuracy was achieved by the Kstar algorithm with the feature selection algorithm Association Rule (AR1), which was 97.58%. A phishing website detection system was proposed by Yang et al. [16]. Depp learning algorithms they used are Recurrent Neural Network (RNN), RNN-RNN, CNN, CNN-CNN, LSTM, LSTM-LSTM, CNN-RNN, and CNN-LSTM. The highest accuracy is achieved by CNN-LSTM, which is 98.61%. Hadi et al. devised the Fast Associative Classification Algorithm (FACA) [17]. They employed the following algorithms: CBA, FACA, Classification based on Multiple Association Rules (CMAR), Enhanced Class Association Rule (ECAR), Multi-class Classification based on Association Rule (MCAR), and MCAR. By 1.5%, 0.2%, 1.8%, and 1.2%, respectively, the FACA beat the ECAR, MCAR, CMAR, and CBA. Aburrous et al. [18] performed classification mining techniques for the prediction of phishing websites. They also performed 2 experimental case studies in their research paper. In the first case study, they assigned female members to lure the staff members of the bank to take their personal banking accounts' usernames and passwords. They managed to deceive 16 members out of 50. In the second experimental case study, they engineered a replica of the Jordan Ahil Bank website. The website was designed to phish users and make them submit their credentials by sending

them fake emails. They deliberately put phishing features and factors to measure users' awareness. Other than that, the authors used four classification techniques DT, Partial Decision Trees (PART), RIPPER, and PRISM. The highest accuracy was achieved by PART which is 86.3%.

3 Experimental Setup

The methodology of this research begins with the selection of two datasets taken from Kaggle and Mendeley. Post selection of datasets, a preprocessing step is applied to each dataset for two main purposes, which are to replace the missing values and to convert string values to numerical values. The process of converting string to numerical is known as one-hot encoding [19]. ML methods are used on the dataset after preprocessing. This calls for the employment of ML methods like ANN, CNN, and LSTM. 10-fold cross-validation is used to check the results of the data utilized. After the prediction, an analysis of comparisons was done between all of the above strategies to see which one performed better. The overall methodology is presented in Fig. 1.

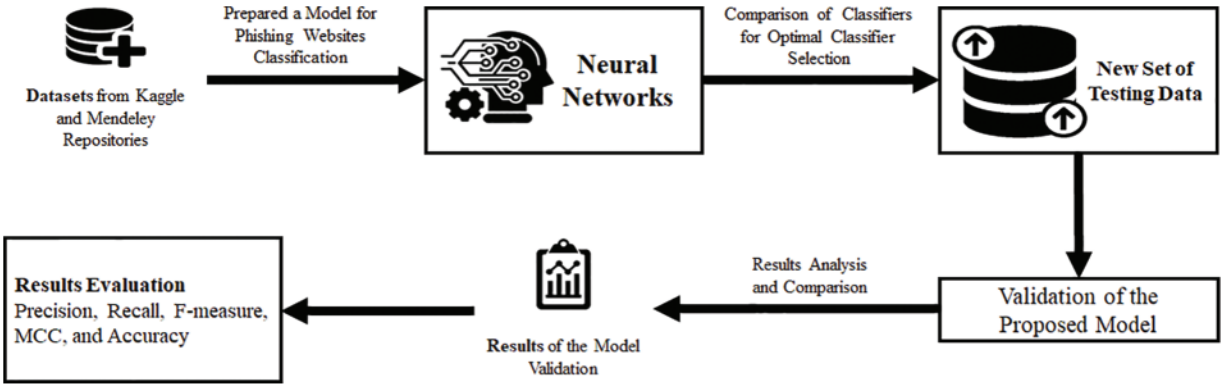

Figure 1: Our research methodology

Our methodology employs established neural network models and publicly available datasets, the study presents a comprehensive comparative analysis of these models in the context of phishing website classification. By systematically evaluating the performance of ANN, CNN, and LSTM on real-world datasets, this study offers valuable insights into the efficacy of these approaches for detecting phishing attempts. The study aims to provide a clear understanding of how these widely recognized models perform in a specific application area, offering a practical perspective for researchers and practitioners in the cybersecurity domain. Furthermore, this work demonstrates the potential of machine learning techniques in enhancing phishing detection systems, thus contributing to the broader goal of improving cyber defense mechanisms.

3.1 Datasets

On datasets acquired from the Kaggle [20] and Mendeley [21] repositories, the ML classification approaches are applied. The Mendeley repository's dataset has 10,000 cases and 48 characteristics, whereas the Kaggle repository's dataset has 11,055 examples and 30 features. The datasets are detailed in subsequent.

3.1.1 Kaggle

The attributes from Kaggle are categorized into four types of features [22].

***1. Address Bar-Based Features*:** Address bar-based features contain features that are extracted from the address bar of a website as shown in Table 1.

Table 1: Address bar features

SN	Variables	Description	Data type
1	IP_address	Instead of a domain name IP address is used	Boolean
2	Long URL	Using long URL	−1, 0, 1
3	Shortening service	Using URL shortening services	Boolean
4	At symbol	'@' symbol in URL	Boolean
5	Double slashes	'//' symbol in URL	Boolean
6	Prefixes or suffixes	Suffix or prefix in domain	Boolean
7	Having subdomain	Using subdomain	−1, 0, 1
8	HTTPS	Using HTTPS	−1, 0, 1
9	Domain registration length	Domains with an expiry date of less than one year	Boolean
10	Favicon	Favicon that is loaded from the external domain	Boolean
11	Non-standard port	The use of non-standard port	Boolean
12	HTTPS token	Using HTTPS token is the URL	Boolean

2. Abnormal-Based Features: Table 2 presents the abnormal-based features which are related to the hostname in the URL or are related to the IP address of the website.

Table 2: Abnormal based features

SN	Variables	Description	Data type
13	Request URL	Request URLs percentage from external domains	−1, 0, 1
14	URL of anchor	Percentage of URLs of anchor tag from external domains	−1, 0, 1
15	Links in tags	Percentage of links in tags	−1, 0, 1
16	Server form handler	SFHs that contain an empty string or "about:blank"	−1, 0, 1
17	Information to email	Information is submitted to an email	Boolean
18	Abnormal_URL	URL does not have the host	Boolean

***3. HTML and JavaScript-Based Features*:** This category contains features that are extracted from the HTML and Java scripts of a webpage as presented in Table 3.

Table 3: HTML and JavaScript-based features

SN	Variables	Description	Data type
19	Website forwarding	Number of redirect pages	−1, 0, 1
20	Status bar customization	"onMouseOver" changes the status bar	Boolean
21	Disable right click	Check if right click is disabled or not	Boolean

(Continued)

Table 3 (continued)

SN	Variables	Description	Data type
22	Using pop-up window	Having a pop-up window that contains text fields	Boolean
23	IFrame	Using iframe	Boolean

***4. Domain-Based Features*:** Table 4 shows domain-based features which contain features that are extracted from WHOIS and Alexa databases.

Table 4: Domain-based features

SN	Variables	Description	Data type
24	Domain age	Age of the domain	Boolean
25	DNS record	Having DNS record	Boolean
26	Traffic	Website traffic	−1, 0, 1
27	Page_rank	Page rank of the website	Boolean
28	Google index	Webpage indexed by Google	Boolean
29	Links	Links pointing to the web page	−1, 0, 1
30	SR based feature	The statistical-reports-based feature checks the host in top phishing IPs	Boolean

3.1.2 Mendeley

The dataset from the Mendeley repository contains 10,000 instances and 48 attributes. Out of 10000 instances, 5000 belong to phishing websites and 5000 belong to legitimate websites. The features from the Mendeley dataset are categorized into 6 types of features [23,24].

1. Symbol-Based Features: Table 5 shows the symbol-based features that use characters, symbols, and sensitive words in the URL of the webpage.

Table 5: Symbol-based features

SN	Variables	Description	Data type
1	NumDots	Dots in URL	Integer
2	NumDash	Dashes in URL	Integer
3	AtSymbol	Having '@' symbol in URL	Boolean
4	TildeSymbol	Having '∼' symbol in URL	Boolean
5	NumUnderscore	Number of underscores in URL	Integer
6	NumPercent	Number of percent ('%') symbols in URL	Integer
7	NumAmpersand	Number of ampersand ('&') symbols in URL	Integer
8	NumHash	Number of hash ('#') symbols in URL	Boolean
9	NumNumericChars	Number of numeric characters in URL	Integer

(Continued)

Table 5 (continued)

SN	Variables	Description	Data type
10	DoubleSlashInPath	Double slashes ('//') in path of URL	Boolean
11	NumSensitiveWords	Number of sensitive words in URL	Integer

2. Webpage URL-Based Features: The webpage url-based features category contains structural characteristics of the webpage URL. The features in this category are used to appear as legitimate websites and are presented in Table 6.

Table 6: Webpage URL-based features

SN	Variables	Description	Data type
12	SubdomainLevel	Level of subdomain in URL	Integer
13	PathLevel	Depth of the path in URL	Integer
14	UrlLength	URL length	Integer
15	NumQueryComponents	Number of query components in URL	Integer
16	NoHttps	Not having https in URL	Boolean
17	RandomString	Having Random String in URL	Boolean
18	HostnameLength	Hostname length in URL	Integer
19	PathLength	Path length in URL	Integer
20	QueryLength	Query length in URL	Integer

3. Domain-Based Features: This category contains features that are used to target phishing patterns that obfuscate the domain name segment of the webpage URL, these features are presented in Table 7.

Table 7: Domain-based features

SN	Variables	Description	Data type
21	NumDashInHostname	Number of dashes in the hostname	Integer
22	IpAddress	Using IP address in URL	Boolean
23	DomainInSubdomains	Checks for TLD or ccTLD in URL of the webpage	Boolean
24	DomainInPaths	Checks for TLD or ccTLD in the path of URL of the webpage	Boolean
25	HttpsInHostname	Checks HTTPS in the URL of the webpage	Boolean
26	EmbeddedBrandName	Check if the brand name is present in the subdomains and URL of the webpage	Boolean
27	FrequentDomainName Mismatch	Compare the most frequent domain name in the HTML content of the webpage with the domain name in the URL of the webpage	Boolean

4. Content URL-Based Features: Table 8 shows the content URL-based features that contain features that use URLs like hyperlinks and, resource links to mislead users and pull in resources.

Table 8: Content URL-based features

SN	Variables	Description	Data type
28	PctExtHyperlinks	Percentage of the external hyperlinks in HTML content of the webpage	Float
29	PctExtResourceUrls	Percentage of the external resource URLs in HTML content of the webpage	Float
30	ExtFavicon	Checks for external favicon	Boolean
31	ExtFormAction	Checks if the action attribute of the form tag contains a URL from an external domain	Boolean
32	PctNullSelfRedirectHyperlinks	Percentage of hyperlink fields containing the null, self-redirect value	Float
33	FakeLinkInStatusBar	Check if the HTML content of the webpage contains the "onMouseOver" command	Boolean

5. HTML Content-Based Features: Table 9 presents the HTML content-based feature contains features that have miscellaneous phishing characteristics, present in the HTML content of the webpage.

Table 9: HTML content-based features

SN	Variables	Description	Data type
34	InsecureForms	Check the action attribute in the form tag for the URL without https protocol	Boolean
35	RelativeFormAction	Checks the action attribute in the form tag for relative URL	Boolean
36	AbnormalFormAction	Checks the action attribute of the form tag for "#", "about:blank", empty string, or "javascript:true"	Boolean
37	Right_Click_Disabled	Check if right click is disabled or not	Boolean
38	PopupWindows	Check if the webpage contains any pop-ups	Boolean
39	SubmitInfoToEmail	Checks in the HTML of the webpage contain "mailto" function	Boolean
40	IframeOrFrame	Checks in the HTML of the webpage contains iframe or frame	Boolean
41	Missing_Title	Check the HTML for the title tag	Boolean
42	Images_Only_In_Form	Checks the forms in HTML for images only no text	Boolean

6. Correlated-Based Features: This group contains correlated features of other groups mentioned above and is presented in Table 10.

Table 10: Correlated-based features

SN	Variables	Description	Data type
43	SubdomainLevelRT	The correlation of subdomain level	−1, 0, 1
44	UrlLengthRT	The correlation of the length of the URL	−1, 0, 1
45	PctExtResourceUrlsRT	The correlation of the percentage of external URL	−1, 0, 1
46	AbnormalExtFormActionR	The correlation of abnormal actions in the form	−1, 0, 1
47	ExtMetaScriptLinkRT	The correlation of meta script link	−1, 0, 1
48	PctExtNullSelfRedirectHyperlinksRT	The correlation of null self-redirect hyperlinks	−1, 0, 1

3.2 Training and Testing

Data splitting into training and testing is done using the 10-fold cross-validation approach. K-fold cross-validation has been a widely accepted technique in recent years [25]. The dataset is partitioned into K number of folds in K-fold. A distinct subgroup of data is tested in each round when training is utilized with the remaining subgroups. This procedure is repeated until each split-up group of data is used for testing in comparison to the other subgroups as training [26].

3.3 Techniques Evaluated

The ML classification techniques used in this research are ANN, CNN, and LSTM. A brief discussion of each technique is presented subsequently.

3.3.1 Artificial Neural Network

ANN is an ML algorithm inspired by biological neurons. It is built on a network of artificial neurons. An artificial neural network's nodes are linked together, and each connection is allocated a weight based on its strength. ANN has three levels. An input layer, a concealed layer, and an output layer [27]. It is made up of linked nodes, or "neurons," that are organized into three layers: input, hidden, and output. In a feedforward network, information goes through each neuron, which evaluates incoming data, assigns weights and biases, and transfers the output to the next layer. Neurons employ activation functions to provide nonlinearity within the network, allowing it to record complicated data linkages. During training, the network uses optimization algorithms to adjust its weights and biases to minimize the difference between predicted and actual outputs, allowing it to learn and make accurate predictions or classifications in a variety of tasks such as image recognition, language processing, and decision-making.

The ANN for phishing website classification processes begins with data collection and preprocessing, followed by the ANN architecture setup. The model is then trained via forward and backward propagation, with weights repeatedly adjusted to minimize the loss. Following training, the model's performance is assessed using testing data and fine-tuned:

1. **Data Collection:**
   Obtain labeled datasets containing features from both phishing and legitimate websites.
2. **Data Preprocessing:**

- Normalize and preprocess data (scaling, encoding).
- Split data into training and testing sets.

3. **Model Architecture Setup:**
   - Initialize input layer nodes based on the number of features.
   - Add hidden layers with nodes and choose activation functions (ReLU).
   - Initialize an output layer with a single node (binary classification) and use sigmoid activation.
4. **Training Process:**
   - Initialize weights randomly.
   - Perform forward propagation.
   - Compute the weighted sum of inputs and apply activation functions for each layer.
   - Calculate the loss (binary cross-entropy).
   - Perform backpropagation.
   - Calculate gradients of loss concerning weights.
   - Update weights using optimization algorithms.
5. **Model Evaluation:**
   - Use the trained model to predict the testing dataset.
   - Calculate metrics like accuracy, precision, recall, F1-score, and MCC.
6. **Fine-tuning and Validation:**
   - Adjust hyperparameters based on validation performance.
   - Utilize techniques (L2 regularization) to prevent overfitting.

3.3.2 Convolutional Neural Network

CNN belongs to a class of neural networks that are commonly used for the recognition of visual imagery. CNN has an extraction feature that makes it different from other neural networks. The hidden layer of CNN consists of a convolution layer that convalesces an input into numerous kernels. And a pooling layer that downsizes each feature map to a smaller matrix [28]. It is made up of several layers, including convolutional layers, pooling layers, and fully linked layers. CNNs employ convolutional filters to glide over input pictures, allowing them to capture spatial hierarchies and learn significant characteristics at different sizes. Pooling layers reduce data dimensionality, which reduces computing effort and aids in feature translation invariance. The CNN's fully linked layers at the end interpret and forecast the learnt information. CNNs excel in picture classification and object identification because of their capacity to automatically learn and represent crucial properties from input data.

CNNs automatically learn important information in the context of phishing website categorization using convolutional and pooling layers, which are capable of capturing both local and global patterns. Non-linearity is introduced using activation functions, which improves the model's ability to recognize complicated interactions. Fully linked layers improve the learned characteristics even more for the final categorization decision. Because of its capacity to handle sophisticated visual patterns, CNNs have demonstrated great effectiveness in image-based applications. However, developing an ideal CNN and fine-tuning its parameters are critical for getting accurate and robust phishing detection findings. The steps are as follows:

1. **Input Data:**
   - Extracted features are provided as input to the CNN.
2. **Convolutional Layers:**
   - Multiple convolutional layers are employed to detect low-level to high-level features in the input data.

- Each layer uses learnable filters to convolve and produce feature maps that highlight relevant patterns.

3. **Activation Function:**
   - After convolution, an activation function is applied element-wise to introduce non-linearity.
   - This enables the network to capture complex relationships and add flexibility to the learned features.
4. **Pooling Layers:**
   - Pooling layers perform downsampling by selecting the most important information from feature maps.
   - Max pooling, for instance, selects the maximum value in each region, reducing spatial dimensions.
5. **Flattening:**
   - The pooled feature maps are flattened into a 1D vector to be processed by the subsequent fully connected layers.
6. **Fully Connected Layers:**
   - These layers learn intricate relationships in the flattened features through weighted connections.
   - Neurons in this layer are fully connected to the previous layer, allowing them to consider the global context.
7. **Output Layer:**
   - The final layer produces the classification output, indicating whether the website is phishing or legitimate.
   - Activation functions are used to convert raw scores into class probabilities.

3.3.3 Long Short-Term Memory

LSTM is an ML algorithm that belongs to RNN architecture. RNNs can learn long-term dependencies and can work well on a large variety of problems [29]. Memory cells process and store information at varied time intervals, a forget gate governs the flow of information into and out of the cells, input and output gates control memory updating, and a cell state conveys information across time steps. Because of this intricate structure, LSTMs are capable of mitigating the vanishing gradient problem in training RNNs, making them extremely capable of learning and remembering patterns in sequences, such as in natural language processing tasks, time series analysis, and any sequential data where context preservation is critical.

The LSTM network is well-suited for sequential data tasks such as phishing website categorization, which includes URLs, HTML, or characteristics for sequential pattern capture. Because of gating mechanisms that restrict information flow over time steps, LSTMs excel at learning from past data. The embedding layer converts categorical data to continuous forms, allowing for efficient LSTM functioning. The network learns deep sequential correlations for exact predictions using LSTM and fully linked layers. For maximum phishing detection performance, architecture, and hyperparameter tuning remain critical. The steps are detailed as:

1. **Input Data:**
   - Sequential website data such as URL, HTML, or extracted features are provided as input to the LSTM network.
2. **Input Embedding:**
   - Raw categorical data are converted into continuous vectors.

3. **LSTM Layers:**
   - Multiple LSTM layers process sequential data and capture temporal dependencies.
   - LSTM units maintain cell states and hidden states to remember and forget information selectively.
4. **Output from Last Step:**
   - The output from the last LSTM step or the aggregated sequence information contains learned features.
5. **Fully Connected Layers:**
   - These layers take the LSTM outputs as input and learn intricate relationships in the sequential features.
6. **Output Layer:**
   - The final layer produces the classification output, indicating whether the website is phishing or legitimate.

3.4 Assessment Criteria

Evaluating your model is the most important part of the research [30]. It is important to use standardized evaluation measures. For the evaluation of classifier performance, we used several performance evaluation measures. These measures are accuracy, precision, recall, F-measure, and MCC [30–33].

Accuracy is the greatest natural performance measurement and, in overall observations, this is a relation of suitably predicted or classified observations.

Overall, how often is the classifier correct? It is the point that how much the classification is correct and can be calculated as

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

Here TN is the percent of the true-negative category, the number of true-positive categories is TP, the percent of false-negative classifications is FN, and the percent of the false-positive category is FP.

Precision is the relation of all-out estimated positive explanations or observations with fruitfully predicted positive observations. When it predicts yes, how often is it correct? It is the count of positive predictions divided by the total count of positive class values projected.

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

Recall is the relation of accurately predicted positive observations to all observations in a class yes. It is measured as the ratio of the TP model with a high option to the total count of positive models.

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

F-measurement is the one-sided average of Recall and Precision with these outlines, both false positives and false negative scores can be deliberate. Impulsively it is not as just like precision, but F1-measurement is normally more supportive, mostly if you have an uneven class circulation. The F-measure is a measure of a test's accuracy.

$$\text{F1 Score} = 2 \times \frac{(\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \tag{4}$$

Matthew's Correlation Coefficient, also called MCC for short, was introduced by Brian Mathews in 1975. MCC is a statistical tool used to evaluate a model. Its function is to estimate or measure the difference between the predicted values and the actual values and this is equivalent to chi-square statistics for a 2 × 2 emergency table.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}} \tag{5}$$

4 Results and Discussion

This section is divided into two phases. 1st phase represents the results achieved based on the Kaggle dataset while 2nd phase represents the results achieved using the Mendeley dataset. These are detailed in the subsequent.

4.1 Result on Kaggle Dataset

Fig. 2 shows the accuracy ratings for the three models that were used: ANN, CNN, and LSTM. The maximum accuracy, 95%, is attained by ANN, demonstrating its superior ability in correctly predicting or categorizing data. This can be ascribed to ANNs' capacity to train on big datasets, handle a broad variety of input characteristics, and discover complicated patterns and connections within the data. Although the accuracy of CNN and LSTM is somewhat lower, at 93% and 92%, respectively, they each have their advantages in terms of identifying spatial patterns and managing sequential data. Overall, the unique job and data characteristics determine which approach is better.

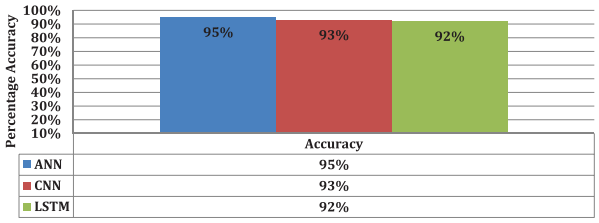

Figure 2: Accuracy analysis on Kaggle dataset

The accuracy, recall, and F-measure scores for the models used to categorize phishing websites are shown in Table 11. Overall, ANN performs best across all three criteria, with accuracy, recall, and F-measure all coming in at 0.95. This shows that ANN is successful in avoiding false positives and false negatives while properly recognizing phishing websites. An F-measure of 0.93 is achieved by CNN, which comes closely behind with a slightly lower precision of 0.92 but a high recall of 0.95. The F-measure for LSTM is 0.93 since it has a high accuracy of 0.95 but a poor recall of 0.90. Although all three methods are effective, ANN is a viable contender for accurate phishing website categorization because of its better precision and balanced recall.

Table 11: Precision, Recall, and F-measure of employed classifiers on Kaggle dataset

Technique	Precision	Recall	F-measure
ANN	0.95	0.95	0.95
CNN	0.92	0.95	0.93
LSTM	0.95	0.90	0.93

According to Fig. 3's MCC scores, ANN has the greatest MCC score of 91%, demonstrating both its outstanding predictive capability and capacity for handling unbalanced datasets. Although significantly less successful than ANN in categorizing the data, CNN and LSTM both have MCC scores of 85%. True positives, true negatives, false positives, and false negatives are all considered by MCC when assessing the performance of the classifier. Given the classification challenge, the higher MCC score of ANN shows that it is excellent at identifying underlying patterns and producing precise predictions, making it a dependable strategy.

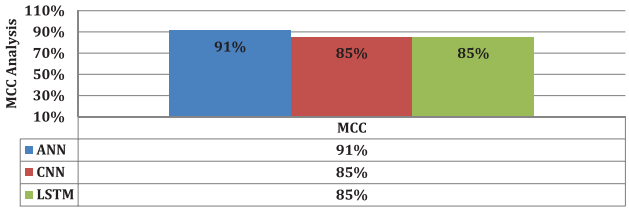

Figure 3: MCC of employed classifiers on Kaggle dataset

4.2 Result on Mendeley Dataset

The accuracy ratings of the models that were used to categorize phishing websites using the Mendeley dataset are shown in Fig. 4. The greatest accuracy score, 98%, is attained by CNN, demonstrating its skill in correctly identifying phishing websites. With a 97% accuracy rate, ANN comes in second place, proving that it is also successful in classifying phishing websites. A somewhat lower accuracy of 96% is attained with LSTM. These excellent accuracy results for all three strategies show that they can successfully distinguish between trustworthy and fraudulent websites. To stop possible cyber risks and ensure online security, it is essential to classify phishing websites accurately. As a result, both CNN and ANN demonstrate potential as solid methods for classifying phishing websites.

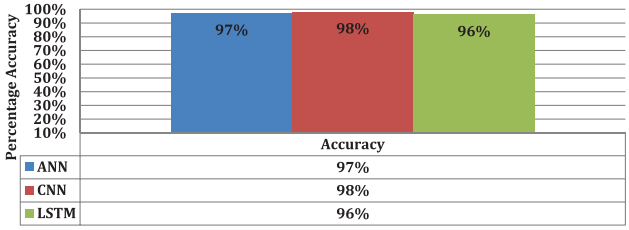

Figure 4: Accuracy analysis on Mendeley dataset

Table 12 lists the accuracy, recall, and F-measure scores for the models that were used. High accuracy, recall, and F-measure scores of 0.97 and 0.98 are achieved by both ANN and CNN, proving their efficacy in properly identifying phishing websites. With accuracy, recall, and F-measure scores of 0.96, LSTM performs somewhat worse. These results show that all three approaches work well in properly classifying phishing websites, with CNN performing the best. High precision values show a low probability of false positives, preventing legitimate websites from being mistakenly classified as phishing sites. Due to the strong recall values, it may be assumed that a sizable percentage of real phishing websites have been accurately recognized. The F-measure combines precision and recall to provide a fair assessment of performance and highlights CNN's impressive performance in this

situation. Overall, CNN outperforms ANN in terms of accuracy, recall, and F-measure scores, and both are effective methods for correctly categorizing phishing websites.

Table 12: Precision, Recall, and F-measure of employed classifiers on Mendeley dataset

Technique	Precision	Recall	F-measure
ANN	0.97	0.97	0.97
CNN	0.98	0.98	0.98
LSTM	0.96	0.97	0.96

The MCC scores for each model used to categorize phishing websites are shown in Fig. 5. The best MCC score, 96%, was attained by CNN, demonstrating both its outstanding predictive capability and its capacity to handle unbalanced datasets. With an MCC score of 94%, ANN comes in second place, proving how well it can identify phishing websites. A slightly lower MCC score of 93% is attained via LSTM. The higher MCC scores for CNN and ANN imply that these methods are more trustworthy for classifying phishing websites because they are better able to identify underlying patterns and anticipate outcomes.

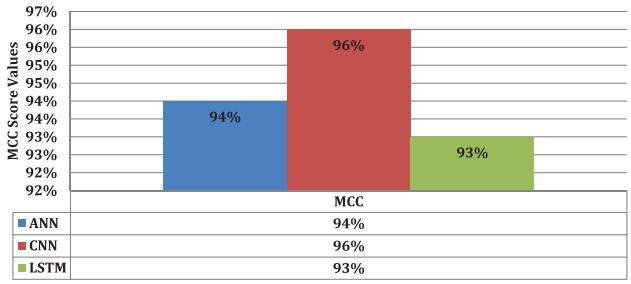

Figure 5: MCC of employed classifiers on Mendeley dataset

This research focuses on the performance of NN algorithms on two datasets that are from the Kaggle and Mendeley repositories. There is heterogeneity in the results achieved using both datasets, due to the difference in attributes and records saved in each dataset. On the Kaggle dataset, ANN shows better performance as compared to the rest of the employed algorithms. The accuracy gap between ANN and the used approaches is seen in Fig. 6 when comparing ANN with the Kaggle dataset. On the Mendeley dataset, CNN shows better performance as compared to the rest of the employed algorithms. Fig. 7 contains the difference in accuracy between CNN and employed techniques on the Mendeley dataset.

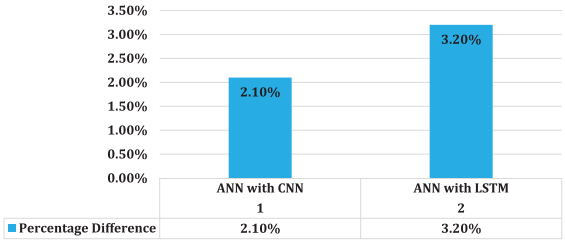

Figure 6: Accuracy difference between ANN and other employed classifiers

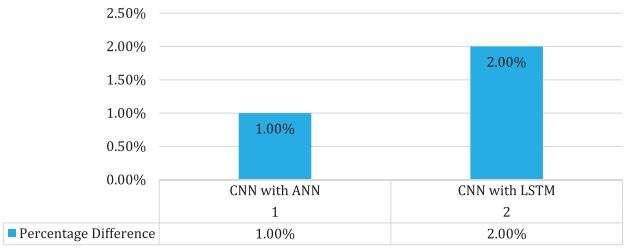

Figure 7: Accuracy difference between CNN and other employed classifiers

The research study greatly adds to cybersecurity by combating phishing assaults using ML approaches. It thoroughly assesses the performance of modern ML models ANN, CNN, and LSTM in categorizing phishing websites, employing real-world datasets and a variety of assessment measures. While it highlights ML's superiority over traditional methods and its potential to improve cybersecurity against evolving phishing tactics, limitations such as a narrow focus on specific models, a lack of discussion on challenges such as adversarial attacks, and a lack of exploration into hyperparameters and ensemble methods may limit its applicability. Addressing these issues and recommending future study areas would increase the overall effect.

This empirical analysis serves as a valuable resource for practitioners seeking insights into the effectiveness of these models for phishing detection. By providing a comprehensive comparison of established methods, the research aids in determining their strengths and limitations in a specific application context. Such a detailed evaluation, even without revolutionary techniques, contributes to the refinement of practical approaches in the field and supports the ongoing efforts to enhance cybersecurity measures.

4.3 Threats to Validity

This section discusses the threats that may accrue.

4.3.1 Internal Validity

This study's analysis is based on accepted assessment metrics that have been used in several other studies in the past. These benchmarks are used to assess and gauge the efficacy and performance of applied procedures. Therefore, the accuracy may be reduced by the introduction of additional assessment criteria as a substitute. Additionally, the currently used approaches can be changed with newer ones.

4.3.2 External Validity

On two datasets obtained from the Kaggle and Mendeley repositories, we conducted the experimental analysis. If we use the predicted approaches on other datasets or swap out these datasets for another, it might compromise the validity of the conclusions while also lowering their accuracy. Similarly, the methodologies used might not be able to produce improved forecast results when applied to some other datasets.

4.3.3 Construct Validity

Based on numerous assessment standards, several ML approaches are compared against one another on distinct datasets. The variety of approaches used in this study is distinguished by their

advancements over the other strategies used by researchers throughout the previous decades. The danger, though, is that if we add more new methods, it is likely that they will use up all of the anticipated techniques. Additionally, we may improve accuracy by splitting the dataset into training and testing data or by altering the number of folds used for experiment validation. Additionally, new assessment criteria can produce results that are superior to the ones that are now achieved.

4.4 Ablation Study

The goal of this study was to deconstruct and evaluate the research's essential aspects to better understand their relevance and usefulness in improving the overall performance of the detection system. The ablation research looked at the following elements:

1. **Feature Engineering Techniques:** The feature engineering approaches used on the datasets were the first thing that was scrutinized. It was able to analyze their separate contributions to the model's performance by methodically deleting certain characteristics or feature categories. This research enabled the discovery of crucial characteristics that have a major impact on the model's predictive performance.
2. **Machine Learning Models:** During this stage of the ablation study, the relative effects of each machine learning model—Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM)—were examined. To ensure that each model accurately classified phishing websites, it was evaluated on an individual basis. This made it possible to comprehend the advantages and disadvantages of each model in more detail.
3. **Dataset Sources:** The impact of dataset sources was investigated in the study, with a focus on datasets obtained from Mendeley and Kaggle libraries. It was feasible to evaluate the variability in model performance caused by variations in dataset features, such as size and composition, by assessing each model separately on each dataset. This investigation emphasized how the dataset affects the model's capacity for generalization and its performance on actual data.
4. **Performance Metrics:** Individual performance indicators were evaluated, such as accuracy, precision, recall, F-measure, and Matthews Correlation Coefficient (MCC). The study shed light on the relative significance of accuracy, recall, and other assessment criteria in the context of phishing detection by breaking down and analyzing each statistic separately. This made it possible to comprehend the trade-offs between various performance characteristics more deeply.
5. **Cross-Validation Techniques:** The ablation study took into account the function of cross-validation methods as well, namely the 10-fold cross-validation approach that was employed in the investigation. The effect of cross-validation on model resilience and generalizability was investigated by running experiments with cross-validation modifications such as leave-one-out or stratified sampling.

The ablation study's findings provide insight into the relative significance of these elements in the context of phishing detection. Model performance was shown to be significantly impacted by feature engineering, the machine learning model selected, and the source of the dataset. The study also showed that determining the performance measures and cross-validation methods were critical in determining how successful the models were. These results provide insightful information for phishing detection research in the future, directing the creation of stronger and more efficient cybersecurity solutions.

5 Conclusion

In this study, we tackled the crucial problem of phishing assaults and put forth an ML strategy for categorizing phishing websites. We set out to create a strong and effective solution to go along with the more conventional blacklist-based approaches by utilizing ML techniques. Comprehensive tests were performed on two datasets that were taken from the Kaggle and Mendeley repositories to evaluate the proposed technique. Applying well-known ML models including ANN, CNN, and LSTM, we evaluated their performance using a range of measures including accuracy, precision, recall, F-measure, and MCC. The outcomes showed how ML-based phishing detection outperformed the conventional blacklist method for phishing detection. The maximum accuracy, precision, recall, and F-measure were attained by the ANN model, demonstrating its superiority in recognizing phishing websites. Furthermore, the MCC scores indicated a strong correlation between the predicted and actual classifications. Our study adds to the body of knowledge by demonstrating how ML algorithms may be used to counter phishing assaults. The results imply that ML-based strategies can improve phishing detection systems and offer a more proactive defense against phishing tactics that are constantly developing. Despite the encouraging outcomes, there are still issues to be resolved. The performance of the algorithm might be enhanced yet more by using more representative and varied datasets. Furthermore, the investigation of ensemble methodologies and feature engineering approaches may result in improved robustness and accuracy.

Future phishing detection research might concentrate on improving the resilience of ML models against developing and sophisticated phishing tactics. Exploration of ensemble approaches, which mix many ML models to increase overall detection accuracy, is one promising area for study. Furthermore, for real-world cybersecurity applications, understanding the impact of adversarial assaults on phishing detection models and creating countermeasures to counteract these attacks is critical. Furthermore, the development of automated systems capable of adapting and self-improving as new phishing strategies arise might enable more proactive protection against phishing attacks. Finally, including real-time data sources in the detection process, such as website traffic patterns and user behavior, might result in more dynamic and effective phishing detection systems.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Shoaib, Bilal Khan; data collection: Saifullah Jan; analysis and interpretation of results: Subhan Ullah, Aiman; draft manuscript preparation: Bilal Khan, Saifullah Jan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository. The data that support the findings of this study are openly available in Kaggle and Mendeley repositories.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Levi, "Assessing the trends, scale and nature of economic cybercrimes: Overview and issues: In cybercrimes, cybercriminals and their policing, in crime, law and social change," *Cybercriminals and Their Policing, in Crime, Law and Social Change*, vol. 67, pp. 3–20, 2017.

[2] M. Butavicius, R. Taib and S. J. Han, "Why people keep falling for phishing scams: The effects of time pressure and deception cues on the detection of phishing emails," *Computers & Security*, vol. 123, pp. 102937, 2022.

[3] V. Bhavsar, A. Kadlak and S. Sharma, "Study on phishing attacks," *International Journal of Computer Applications*, vol. 182, no. 33, pp. 27–29, 2018.

[4] R. Alabdan, "Phishing attacks survey: Types, vectors, and technical approaches," *Future Internet*, vol. 12, no. 10, pp. 1–39, 2020.

[5] B. N. K. Akarshita Shankar and R. Shetty, "A review on phishing," *A Review of Phishing Attacks*, vol. 14, no. 9, pp. 2171–2175, 2019.

[6] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran and B. S. Bindhumadhava, "Phishing website classification and detection using machine learning," in *Int. Conf. on Computer Communication and Informatics*, ICCCI, Coimbatore, India, pp. 20–25, 2020.

[7] R. M. Mohammad, F. Thabtah and L. McCluskey, "Intelligent rule-based phishing websites classification," *IET Information Security*, vol. 8, no. 3, pp. 153–160, 2014.

[8] R. M. Mohammad, F. Thabtah and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, no. 2, pp. 443–458, 2014.

[9] M. Kaytan and D. Hanbay, "Effective classification of phishing web pages based on new rules by using extreme learning machines," *Computer Science*, vol. 2, no. 1, pp. 15–36, 2017.

[10] H. Musa, A. Y. Gital, F. U. Zambuk, A. Umar, A. Y. Umar *et al.,* "A comparative analysis of phishing website detection using XGBOOST algorithm," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 5, pp. 1434–1443, 2019.

[11] M. Chatterjee and A. S. Namin, "Detecting phishing websites through deep reinforcement learning," *IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Milwaukee, WI, USA, pp. 227–232, 2019.

[12] E. Zhu, Y. Chen, C. Ye, X. Li and F. Liu, "OFS-NN: An effective phishing websites detection model based on optimal feature selection and neural network," *IEEE Access*, vol. 7, pp. 73271–73284, 2019.

[13] A. Zamir, H. Khan, T. Iqbal, N. Yousaf, F. Aslam *et al.,* "Phishing web site detection using diverse machine learning algorithms," *The Electronic Library*, vol. 38, no. 1, pp. 65–80, 2020.

[14] W. Ali, "Phishing website detection based on supervised machine learning with wrapper features selection," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 9, pp. 72–78, 2017.

[15] M. Karabatak and T. Mustafa, "Performance comparison of classifiers on reduced phishing website dataset," in *2018 6th Int. Symp. on Digital Forensic and Security (ISDFS)*, Antalya, Turkey, IEEE, vol. 2018, pp. 1–5, 2018.

[16] P. Yang, G. Zhao and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, no. 2019, pp. 15196–15209, 2019.

[17] W. Hadi, F. Aburub and S. Alhawari, "A new fast associative classification algorithm for detecting phishing websites," *Applied Soft Computing*, vol. 48, no. 1, pp. 729–734, 2016.

[18] M. Aburrous, M. A. Hossain, K. Dahal and F. Thabtah, "Predicting phishing websites using classification mining techniques with experimental case studies," in *7th Int. Conf. on Information Technology: New Generations*, Washington DC, USA, pp. 176–181, 2010.

[19] Y. X. Wu, D. Wang, Y. K. Zou and Z. Y. Huang, "Improving deep learning based password guessing models using pre-processing," in *Int. Conf. on Information and Communications Security*, Tianjin, China, pp. 163–183, 2022.

[20] G. Vrbančič, I. Fister Jr. and V. Podgorelec, "Datasets for phishing websites detection," *Data in Brief*, vol. 33, no. 1, pp. 106438, 2020.

[21] C. L. Tan, "Phishing dataset for machine learning: Feature evaluation," *Mendeley Data*, vol. 1, 2018. https://doi.org/10.17632/h3cgnj8hft.1
[22] S. Wedyan and F. Wedyan, "An associative classification data mining approach for detecting phishing websites," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 4, no. 12, pp. 888–899, 2013.
[23] C. Do Xuan, H. D. Nguyen and T. V. Nikolaevich, "Malicious URL detection based on machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, pp. 148–153, 2020.
[24] C. L. Tan, K. L. Chiew, N. Musa and D. H. Abang Ibrahim, "Identifying the most effective feature category in machine learning-based phishing website detection," *International Journal of Engineering & Technology*, vol. 7, no. 4. 31, pp. 1–6, 2018.
[25] F. Al-Areqi, M. Konyar and M. Zeki, "Effectiveness evaluation of different feature extraction methods for classification of COVID-19 from computed tomography images: A high accuracy classification study," *Biomedical Signal Processing and Control*, vol. 76, pp. 103662, 2022.
[26] D. Berrar, "Cross-validation," *Encyclopedia of Bioinformatics and Computational Biology*, vol. 1, no. 3, pp. 542–545, 2018.
[27] L. Zhang, Y. Pan, X. Wu and M. J. Skibniewski, "Introduction to artificial intelligence, beginning deep learning with TensorFlow: Work with Keras," *MNIST Data Sets, and Advanced Neural Networks*, vol. 163, pp. 1–15, 2021.
[28] S. Albawi, T. A. M. Mohammed and S. Alzawi, "Layers of a convolutional neural network," in *Int. Conf. on Engineering and Technology (ICET)*, Antalya, Turkey, IEEE, pp. 1–6, 2017.
[29] H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband *et al.,* "Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting," *Water*, vol. 12, no. 5, pp. 1–18, 2020.
[30] R. Caruana and A. Niculescu-Mizil, "Data mining in metric space: An empirical analysis of supervised learning performance criteria," in *10th Int. Conf. on Knowledge Discovery and Data Mining*, Seattle, Washington, USA, pp. 69–78, 2004.
[31] H. M and S. M. N, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, 2015.
[32] L. A. Mullen, B. Panigrahi, J. Hollada, B. Panigrahi, E. T. Falomo *et al.,* "Strategies for decreasing screening mammography recall rates while maintaining performance metrics," *Academic Radiology*, vol. 24, no. 12, pp. 1556–1560, 2017.
[33] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020.