International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 13 Issue 6 ǁ June 2025 ǁ PP. 01-11

Comparative Study of Gated Recurrent Units and CNN for Phishing URL Attack Classification

Umejuru Daniel[1], Eke Bartholomew[2], and Fubara Egbono[3]
[1,2,3]Department of Computer Science, University of Port Harcourt, Choba, Nigeria

ABSTRACT
It is illegal to gather sensitive information from internet users using social engineering techniques combined with technology. The current systems lacked a web crawler component to retrieve URL background information of phishing attacks and a notification component to alert users via speech. The goal is to create an enhanced Phishing attack detection and notification system utilizing Convolutional Neural Network and Gated recurrent unit network. It was created to offer users with auditory notifications while also obtaining background information about URL addresses related with phishing attempts using a web crawler. This study outlined a novel method capable of identifying malicious phishing URLs, focusing on features primarily obtained from the phishing and real URL addresses. A temporal tokenizer was generated and used for URL text processing which scanned, recognized characters, symbols and redundant tokens. This made it easier to separate specific features from the URL address and return as a list while identifying directories, keyword arguments, and extensions. The model kernel, weights, and bias values were tuned with a penalty term which increased model detection accuracy. We also included a web crawler program that collected details such as internet content, URL links, and design codes from websites, increasing the resilience of the suggested system. The experimental findings show that CNN performed better having 99.60% accuracy and RNN-GRU yielded a recommendable prediction accuracy of 91.0% respectively.

Keywords: *Gated-recurrent network, CNN, web crawler, www, feature extraction*

Date of Submission: 24-05-2025 Date of acceptance: 04-06-2025

I. INTRODUCTION

A phishing attack is a criminal activity that uses deceptive behavior and technical trickery patterns to obtain unlawful access to client-confidential data from individuals or algorithms for learning (Manohar et al.,2020). Phishing attacks are spam emails presented as real with a subject or message designed to deceive recipients into providing critical information (Maniraya et al.,2019). The importance of data privacy, protection, and prevention from phishing efforts cannot be emphasized. The intruder decides to duplicate and then selects unfortunate folks whose data must be taken. The attackers establish fake platforms that look to be authentic in order to attract victims into providing vital information (Maurya and Jain 2020). Pastor-Galindo et al.,(2020) created a framework to describe the many stages of a cyber-attack. There are various deep learning algorithms for phishing site identification, including encoders and decoders, deep belief networks, convolutional neural networks, recurrent neural networks, boltzman machines, and others (Basit et al., 2020). Machine learning is one of the most prevalent methods for detecting phishing sites (Sindhu et al., 2020). Phishing URLs and accompanying webpages are symbolized by a collection of widely employed properties, including URL data, website layout, and JavaScript capabilities. Rashid et al.,(2020) recommended the conventional approach of irregular decision trees and emphasizes that forest land of trees traversing together with inclined hyper-planes which can definitely increase accuracy. An overall concept that the problems of a classifier may only increase to a measure of accuracy before over fitting occurs, resulted from the perception of a more complicated classifier becoming significantly more accurate. Phishing attacks are classified into four types: misleading phishing, spear phishing, whaling, and pharming (Safi and Singh 2023). Basit et al. (2021) presented four separate phishing attack categories, including communication methods, target devices, attack strategies, and countermeasures. The most prevalent kind of phishing attack involves deceptive phishing, which pretends to be an actual network or webpage and sends the user text messages (or emails) that appear to be authentic (Javed et al.,2020). The URLs that are malicious in these text messages (or emails) would prompt the recipient to make a click on the URL. The perpetrators have created a phishing website that intends to gather all of the user's username and password and other sensitive information and deliver it to them. The spear phishing scheme is similar to the deceptive phishing kind, which focuses on just one user. The fraudsters seek to deceive someone into handing over confidential information. A personalized message or email is sent to the user with the goal of deceiving them. The email is personalized to include the majority of the user's details, such as the user name, place of employment, designation, and so on.(Jain et al.,2020). The most often used medium for spear phishing is a social media site like LinkedIn, where they can easily find out. The whale attack occurs when phishers target people in position of power, such as CEOs. Prior to the attack, the culprit would spend a significant amount of time studying the target. The intruder sends an email message to the victim in an attempt to trick them into disclosing confidential information. Whaling is regarded as a very risky attack because executive group members have access to the organization's most sensitive information (Kumar 2020). Pharming is a type of phishing which does not call for a specific individual as the target. The attacker can do harm to a large number of users without being directly targeted. There are two strategies for carrying out pharming attacks: (a). It requires emailing the target codes, which update every local host files on the machine. The host files would convert the URLs into numerical strings, which the system would use to access websites. Even if the target person enters a genuine URL, it may lead them to a harmful website. (b) Another pharming attack approach is DNS cache poisoning, which alters the website's domain name system tables while leaving the local host files intact. This leads the victim to be unintentionally directed to undesirable web pages. The user would believe they are visiting a trustworthy website, but due to DNS poisoning, they are actually viewing a hostile domain (Mittal et al.,2020). The simpler gating network reduced the existing system's ability to extract complex information from the proposed Phishing dataset. Existing approaches find it exceedingly difficult to determine optimal solutions when the number of trainable parameters increases and greater parameters are needed to learn for complex issues. The following is how this paper is structured: The introduction is given in Section 1, the results and a detailed discussion of the results are covered in Section 4, the model's materials and methods are introduced in Section 3, the paper's conclusion is given in Section 5, and a brief assessment of previous approaches to the topic and the gap in studying the proposed model is given in Section 2.

II. RELATED WORKS

This section addresses numerous phishing strategies, explaining how they differ and what traits they share (Zamir et al., 2019). Several methods and approaches have been researched to better understand phishing assaults and provide a defense against them. Numerous researches on the strategies used by phishers or attackers are available in this regard, but we are focusing on the ones that have proven to be the most accurate and linked to our subject matter. Fu et al. (2021) compared several distinct machine learning (ML) techniques in order to identify phishing sites. The SVM had the lowest detection accuracy, whereas the RF performed the best. Liu et al. (2020) presented a method of mining a website's connected webpage set to detect phishing websites. They investigated the interaction to the specified website in terms of text similarity, ranking relationship, link relationship, and similarities in webpage layout. Their tests yielded an accuracy rate of 91.44 percent and a false alert rate of roughly 3.40 percent. An ANN model was used by Zhu et al. (2020) to identify phishing websites. This was carried out to ascertain whether the website was phishing or not. The proposed study used 1-hidden layer level, 17-features, 17-neurons as input, and 2-synapses as output. Training and testing set were created from the total data se and the accuracy of the suggested model was 92.48 percent. Verma et al. (2020) created clusters of linked phish by using a structural evaluation technique that compared local subdomain files. The model demonstrated exceptional performance, with the testing set achieving an average score of 70% in several reviews. Phishing websites are grouped together according to copied brands using a non-binary categorization scheme.

2.1 Deep learning (DL) for phishing attack detection

The latest developments in DL approaches claim that when it comes to categorizing phishing websites, they outperform conventional ML systems. The choice of various learning parameters, however, has a significant impact on the outcomes of using deep neural networks. There are several deep learning (DL) methods that can be employed, including deep neural networks, (2) feed-forward deep neural networks, recurrent neural networks, convolutional neural networks, limited Boltzmann machines, deep belief networks, and deep auto-encoders for phishing attack detection (Ferrag et al., 2020). The neurons are given a collection of input data, and certain weights are allocated to determine if the traffic is authentic or phishing-related. According to Benavides et al. (2020), who describe the DL algorithms used in each arrangement, Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) are the most frequently. Shie(2020) adopted numerous approaches and discussed various tactics for accurately identifying phishing attempts. Due to high accuracy and robustness, feature extraction-based DL techniques perform well. Models for categorization also show strong performance. Maurya and Jain (2020) presented a phishing prevention architecture that relies on using a phishing identification model reliant on DL, at the ISP's level in order to ensure security at all levels rather than merely average execution. This approach places a temporary security barrier between different workers and end clients at ISPs. The effectiveness of implementing this framework rests in the ability to make certain that lots of clients will be safeguarded from an individual phishing attack with just one blocking goal. End users are given secure help irrespective of their framework without extremely efficient processing machines, and ISPs are the only ones who must perform computation overhead of phishing detection models. Li et al.(2020) innovative methodology involved sending a URL as input and extracting HTML-related elements from it. A sort of stacking meta-learning model was created in merging classifiers following feature extraction. The experiment made use of a variety of datasets, among which of which was a collection of 2000 web pages including 100 genuine and 1,000 phishing attacks from Phish-tank. The second dataset consisted of 49,947 web pages overall, 30,873 of which were real cases, and 19074 instances of phishing. The capabilities of ANNs can be integrated to create a stacking-based model to achieve greater accuracy. The stacked model proved beneficial and produced excellent accuracy when using many classifiers together. Different researchers employed a variety of machine learning classifiers, but they were not as accurate or effective as they could have been. Researchers have previously made use of two separate machine learning datasets that are freely available via Phish-tank and the UCI ML repository. Some of the previous studies compared the functionality of a small number of ML classification algorithms without using feature reduction methods.

Barlow et al.,(2020) Adopted ANN in conjunction with binary classification. The ANN achieved a detection accuracy of 95.69%. The experiment used a set of 4,000 data points, which limits the predictions and reduced the detection accuracy of the model. Liu et al.,(2020) used different data mining techniques in order to detect phishing websites. Their tests yielded an accuracy rate of 91.44 percent and a false alert rate of roughly 3.40 percent. Most of the phishing websites were wrongly classified as authentic. Chiew et al. (2019) employed RF combining different decision trees as an ensemble learner to detect phishing websites. The RF model was employed to identify phishing websites. The results showed that HEFS could identify phishing features up to 94.6 percent of the time. The model could not generalize well on testing set. Al-Sarem et al. (2021**)** presented two innovative ML-based models which includes: AdaBoost and LightGBM. The suggested models, which had the best and most suitable features for spotting phishing websites, performed incredibly well. The model could not extract pertinent information as needed to improve model performance. Hong (2020) combined six different ML algorithms, including RF, CART, LR, SVM, and ANN in order to categorize phishing emails. Over 92% of the phishing emails were correctly predicted by the classifiers under study. The model had performance and resulted in type I and type II error. Whittaker et al. (2010), presented numerous novel capabilities and evaluated the strategy using widely accessible ML algorithms. The recommended model performed exceptionally well since with the best and most appropriate features for identifying phishing websites. The algorithm trained could not learn and extract content-driven or proprietary features. Afroz and Greenstadt (2020**)** proposed a Fuzzy logic was combined with a hashing technique in a white listed approach. The model attained a noteworthy accuracy rate of almost 96%. There was no generality to new phishing attacks because of human interaction. Gowtham et al. (2021) introduced the Automated Individual Whitelist (AIW) technique to monitor user visits to well-known innocuous websites. This approach is quite successful in thwarting dynamic phishing and pharming attacks. The researchers themselves acknowledged that their method was not the best. It removes the need for a pre-established login password during data exchanges between the client and server. Huang et al. (2020), suggested methods include two procedures, one for registering and the other for login, with four parties taking part. Phishing websites and XSS attacks housed on hijacked domains are not detectable. AlEroud and Karabatis (2020) used a GAN in order to categorize the URLs and get beyond the blacklist-based scam detectors. The model had above-average detection accuracy. **Limitations**: (a). There was no available phishing prevention tool for detecting rogue URLs within an enterprise (b). Model could not be used to safeguard user login credentials. Jain and Gupta (2021) combined NB and SVM approaches to efficiently recognize fraudulent URL addresses. The accuracy of URL detection increased to a higher percentage. (a). Model struggled to learn from phishing patterns (b). Model was unable to retain their prior findings in memory. Chiew et al. (2019) combined their knowledge of machine learning with a hybrid ensemble selection strategy to effectively identify phishing sites. They were able to distinguish between phishing and genuine websites with an accuracy of more than 80% and a low false positive rate. (a). had problem of picking up phishing patterns (b). Lost track of its previous discoveries Almseidin et al, (2019) put together ML proto-types for detecting phishing attacks. RF was the most effective classifier based on its 98.11% accuracy rate in identifying phishing attacks. Model had prolonged training time.

III. Methodology

The methodology focuses on the tools and methods required to detect phishing URL domains. This section addresses the feasibility of the proposed technique for detecting phishing URLs, as well as the CNN classifier's evaluation results on the test set. The components listed below are addressed and assessed in order to implement the proposed method.

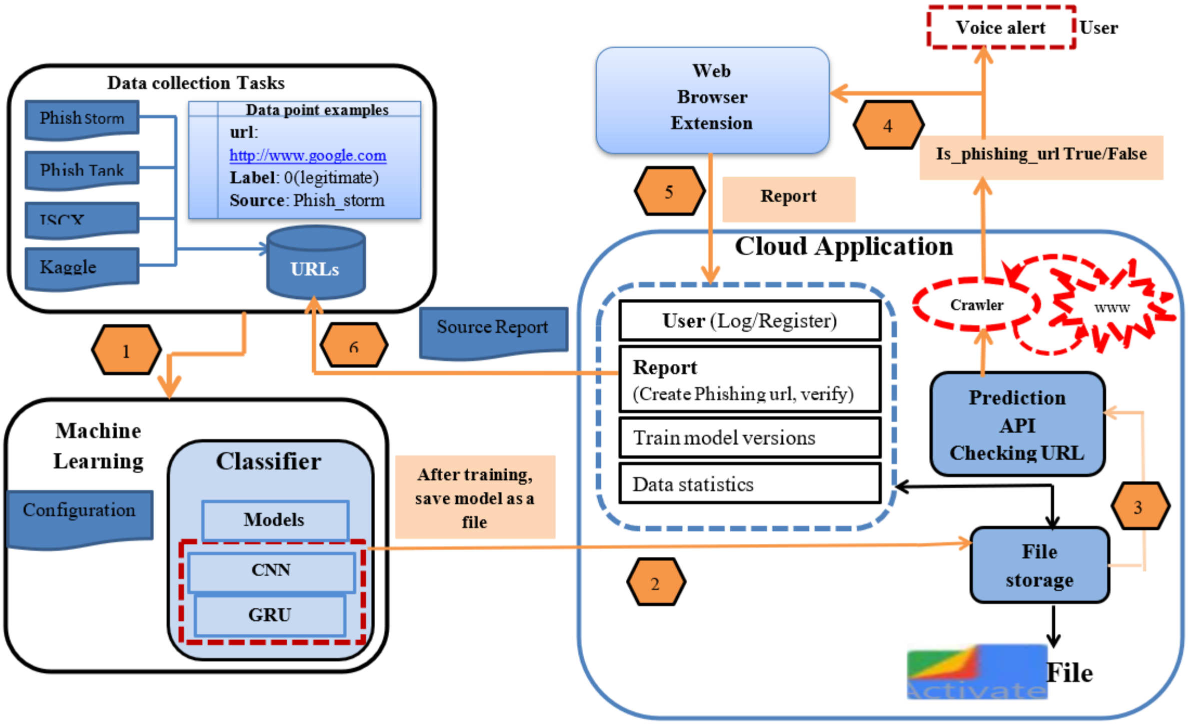

Figure 3.1: Proposed System Architecture

(a). Phishing data collections: The data collecting task is crucial to the biblical study of machine learning industry. It consists of a phishing storm with 96018 trustworthy and 48,009 phishing URLs. The Phising Storm dataset comprised 96018 trusted and 48,009 phishing sites, whereas ISCX-URL had 96018 trusted and 9965 phishing sites.

(b). **Preprocessing** is contained in the collection step to filter attribute values or database entities. The data preparation phase was utilized in order to find and eliminate false and missing values from the suggested system dataset. The numerical fields are preprocessed into an appropriate format prior to training. The training dataset was padded out and the replaced missing value function was used during the preprocessing phase before creating the model.

(c). Feature Extraction: The feature selection process was adopted to determine the correlation between variable or attribute pairs based on the level of correlation using a score value. The higher the score value, the higher the correlation between attribute pairs. This was used in order to prioritize the features that have the greatest influence on model predictions.

(d). Gated recurrent unit(GRU) network: The GRU networks functions similarly to the LSTM but with fewer hyper-parameters, making it quicker to train and more efficient in terms of computation. The cell that stores information state is replaced using a candidate activation vector, which is often updated via two gates known as the reset and update gates. The reset mechanism determines how much of the previous hidden state to completely remove, whereas the update gate specifies how much of the candidate activation vetor is incorporated into the new hidden state. A candidate vetor is configured and utilized to update concealed states at each iteration or training cycle. The logic underlying GRU is reset gate® and update gate(z) derived using current input x and the prior hidden state (h_t-1).

R_t = sigmoid(w_r * [h_t-1, x_t]) 1

X_t = sigmoid(w_z*[h_t-1, x_t]) 2

Where w_r and w_z are network-specific weight metrics learned during training period. The candidate activation function or vector can be generated using current input x and modify the prior hidden state in reset gate.

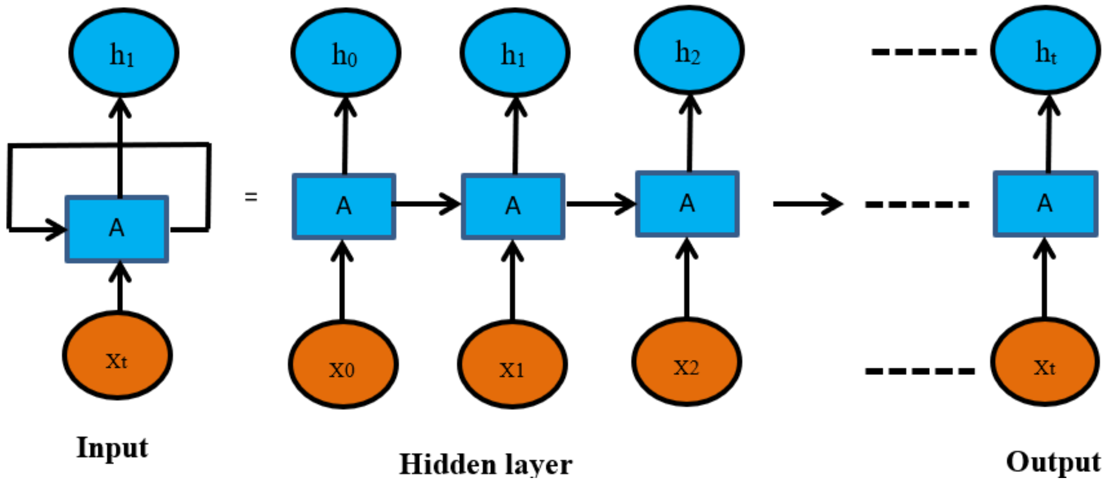

Figure 3.2: The GRU Network Structure (**Source**: Sandhya & Harish, 2021)

A fully connected layer of GRU can be represented as with the equation
σ =f(Wh+b)
3
Where σ is output of the fully GRU connected layer
The GRU neural network structure has inputs, hidden layers and output layer (1 or 0) as shown in figure 1.
The **input nodes** accept data training and testing data into the hidden nodes through weighted channels with Xj, for j runs of state and produces a special input (Xo) when j=0, whose value is set to one in producing the bias input to the hidden layers in the network. The first half is been assigned weighted(w) values nodes(X_j) going to the hidden nodes(Z_h) labelled wh_j nodes are linked to every other input(X_o) nodes associated with weights(wh_j where j = 0 given rise to wh_o) in states. wh_o is like other weights been most recently updated and the values coming from the bias(Xo) set to 1 with hidden nodes(zh):
The **output layers** are always associated with a regression or classification problem depend mainly whether there are output nodes with label. The weights passing through the hidden layers(h) unit as vh of the unit(i) like a bias at the hidden with each output unit having a bias input from the hidden node(Z_0), where the input from hidden node(Z_0) of weights together with that input trained as grouped weights.
(e). Convolutional neural network (CNN): is an example of machine learning, specifically a deep learning technique used largely for analyzing images and text processing/classification. The CNN is an appropriate technique to analyze a stream of data with high accuracy. We are designing a CNN model to assist with the difficult and time-consuming task of altering weights during each training cycle. The weights that are included in the ordering of inputs to the CNN's layers constitute the factors that cause its weights to change. The neural net weights vary at each of the layers in addition to the activating function. The activation processes change with each subsequent cycle since they serve as the data inputs for the subsequent CNN layer. The resulting shift in distribution requires each and every CNN layer to adjust to the changing data inputs, and that is the reason why the deep learning duration for training increases.

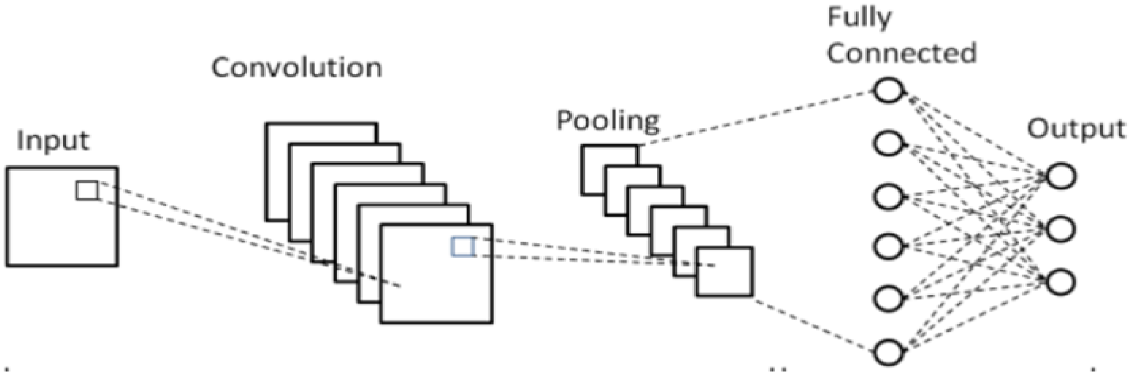

Figure 3.3: CNN Structure (**Source**: Kaur, et al.,2020)
The weighted matrix of the feed-forward neural network model shown in the above figure 3.3 becomes

$$\begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \end{pmatrix}$$ 1 h_i=WCA$_{ij}$*I$_i$

2

$$\begin{matrix} h_1 & ------\rightarrow \\ h_2 & ------\rightarrow \\ h_3 & ------\rightarrow \end{matrix} \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \end{pmatrix} \times \begin{bmatrix} X_1 \\ \\ X_2 \end{bmatrix} \qquad 3$$

The matrix product of the weighted sum becomes

$$\begin{matrix} h_1 & ------\rightarrow \\ h_2 & ------\rightarrow \\ h_3 & ------\rightarrow \end{matrix} \begin{pmatrix} W_{11} * X_1 & + & W_{12} * X_2 \\ W_{21} * X_1 & + & W_{22} * X_2 \\ W_{31} * X_1 & + & W_{32} * X_2 \end{pmatrix} \qquad 4$$

The matrix product is the product of the weighted matrix and the column of the two input(battery capacity) and battery size where $h_1, h_2,$ and h_3 are the hidden layers that stores the results of the first phase and $X_1$and X_2 are the input variables as a single column.

h_i=W$_{ij}$*B$_i$ 5

Where B$_i$ is the bias added to the result of the hidden layer

$$\begin{pmatrix} W_{11} * X_1 & + & W_{12} * X_2 \\ W_{21} * X_1 & + & W_{22} * X_2 \\ W_{31} * X_1 & + & W_{32} * X_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \qquad 6$$

h_i=σ(W$_{ij}$*I$_i$ +B$_i$) 7

Where σ is the sigmoid function

Output(O)= σ(WCA$_{ij}$*I$_i$ +B^o) 8

Given URL samples (x_1, y_1), . . . , (x_n, y_n) we are finding a vector weights called w=(w_1, w_2, . . .,w_d)€$\mathbb{R}^P$) that minimizes a loss function of the form

$$F(w)=\frac{1}{n}\sum_{j=1}^{n}\left(f\left(w;x_j\right)-y_j\right)^2 \qquad 3.16$$

Where f(w; x) is the neural network function which is the depth of d wth ReLU activation function

Web browser component: This was implemented using Jupyter notebook IDE, which is compatible with any of the browsers, was employed for ML model development. The model selects the browser to run and automatically decides whether a given URL address is phishing or not. The system immediately provides the user with initial feedback for detected phishing URLs and trusted sites. The browser will link to GUI interface to visualize and results. The web browser component gets a signal from the prediction API and links back to the GUI API, where the user can log in, review reports, and identify phishing URLs.

Cloud application: The cloud component consists of three key components: prediction API, file storage, and a user login interface for viewing reports. A user can log-in to register, verify URL addresses and view phishing statistics stored in database. The system uses a computerized procedure to validate the risks involved with these URLs that are malicious. These URL addresses are matched with the data collection component in identifying the source of data.

Voice notification alert: This component illustrates a software tool that monitors and search for suspicious or phishing attacks, flagging-up alerts of Phishing or untrusted messages when it finds such attacks in the testing.

(f). Web crawler: The crawler, often known as spider is developed in navigating the world wide web (www) in a methodical manner to collect information from webpages. We used the crawler to issue a request via HTTP to the seeded URL in order to retrieve the Hypertext Markup Language (HTML) content of a web page, which is comparable to how web browsers view a website. It analyzes HTML content, which refers to breaking it into components that form a structured format that the crawler can traverse and evaluate. The crawler recognizes and gathers hyperlinks (URLs) inside the content itself.

IV. RESULT AND DISCUSSIONS

The findings of the GRU and CNN models are displayed and discussed using suitable visualization tools. The concept and its implementation are being refined to produce more accurate and reliable results. We used widely recognized AI/ML tools to present and explain experiment results, such as wordclouds, ROC curves, charts and tables.

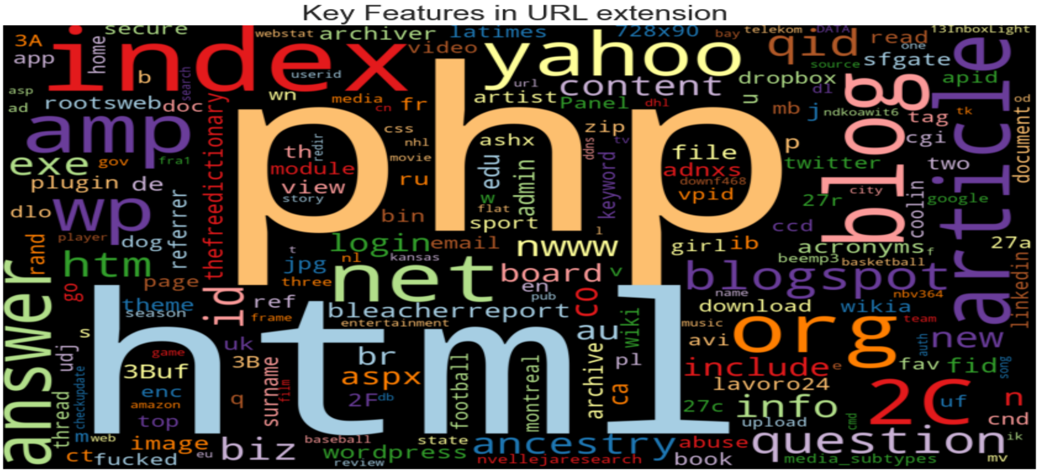

Figure 4.1: Key features in URL extension

Figure 4.1 depicts a word cloud visualization of URL address extensions, with the most frequently used words in larger and bold letters in various colors. Key word extensions such as html,.org, php, and so on appear more often in the dataset for phishing and non-phishing sites. The lighter the word's size, the less essential it is.

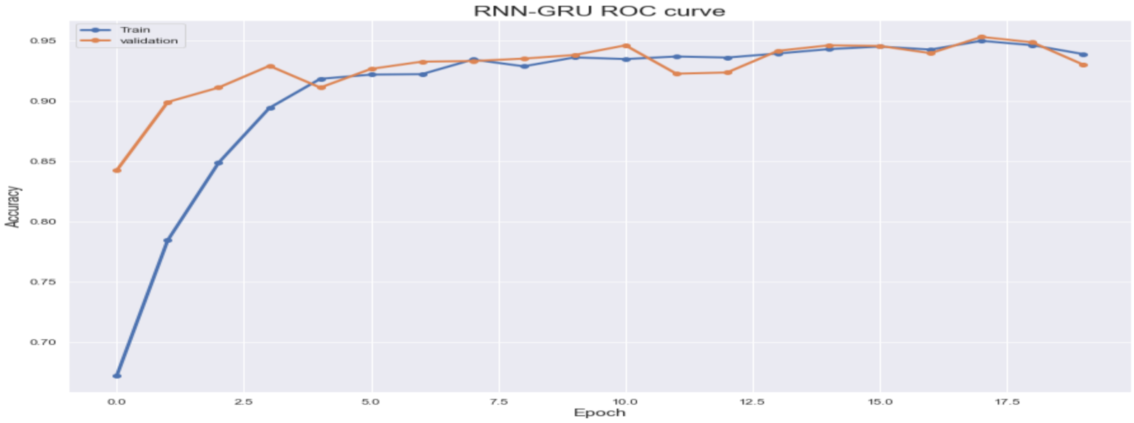

Figure 4.2: Training accuracy of GRU

Figure 4.2 depicts the training curve of GRU across multiple rounds, with higher validation at the beginning, which is difficult to model, and decreasing in some specific circumstances. Model overfitting occurs because the GRU algorithm in this scenario was trained for a total of 20 iterations, making it far too sophisticated for the data. The validation and training accuracy curves fluctuate with the training loss. Figure 4.2 shows an uneven movement of the training and validation curves. The RNN-GRU model struggles to learn desirable Phishing attack features from the training and validation datasets. The attack accuracy grew and peaked about 17 epochs, then dropped over extended training periods.

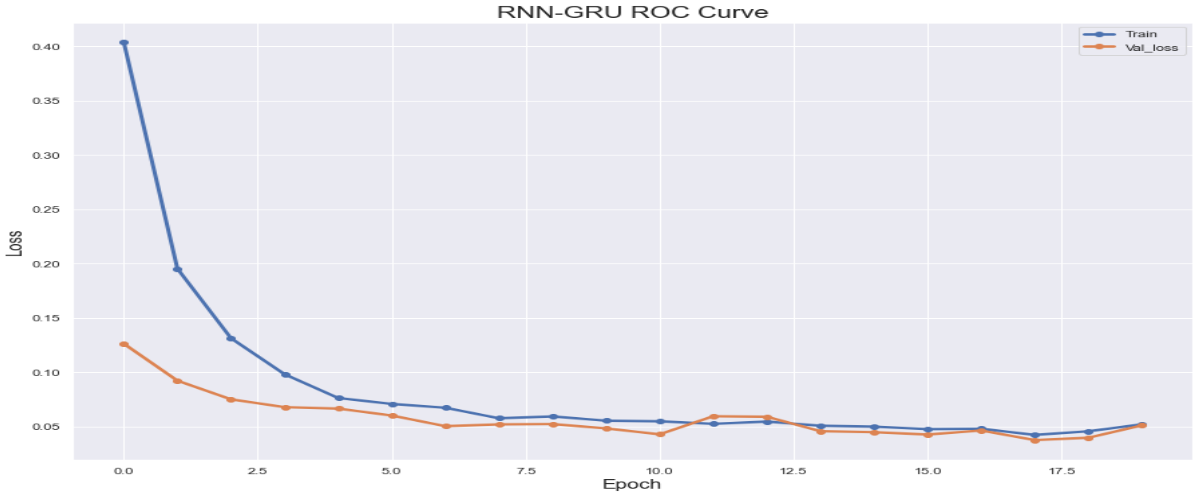

Figure 4.3: validation loss of GRU

Figure 4.3 depicts the training versus validation loss of GRU, with training outperforming better than the validation loss. The validation loss is lower, indicating that the model is converging, and training data is more difficult to model than the validation set, even if both the training and validation losses are decreasing on the plot. The model is receiving new data for both sets because the training and validation losses are clearly separated. The validation curve deviates from the training curve and overlaps at shorter training periods, resulting in losses of unclear patterns dangling around various epcohs.

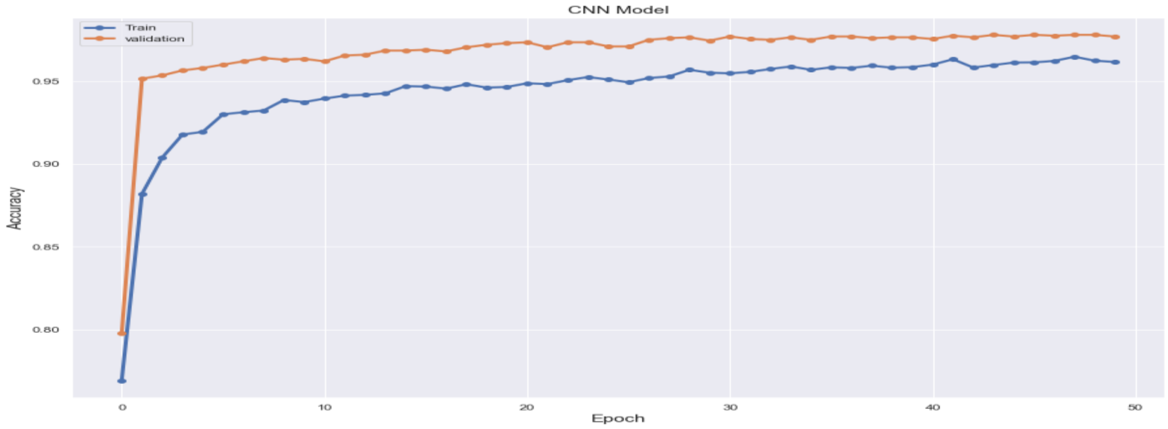

Figure 4.4: Training accuracy of CNN

The CNN training accuracy and validation loss is shown in Figure 4.4. It illustrates how the predictive algorithm fails to draw valid conclusions from the testing data. The trained model performs well on training samples, but when tested on the validation data set, its performance gradually improved, as the graph illustrates, validation loss gradually improved in fluctuating order. Model fitting happens because the artificial neural network(ANN) algorithm in this case was trained for an extended period of 50 iterations, making it purely too sophisticated for the data. Over the training loss, the validation and training accuracy curves are fluctuating. There is a wider gap between training and validation curves, indicating that the model was not able to learn feature features and generalize well on the testing set.

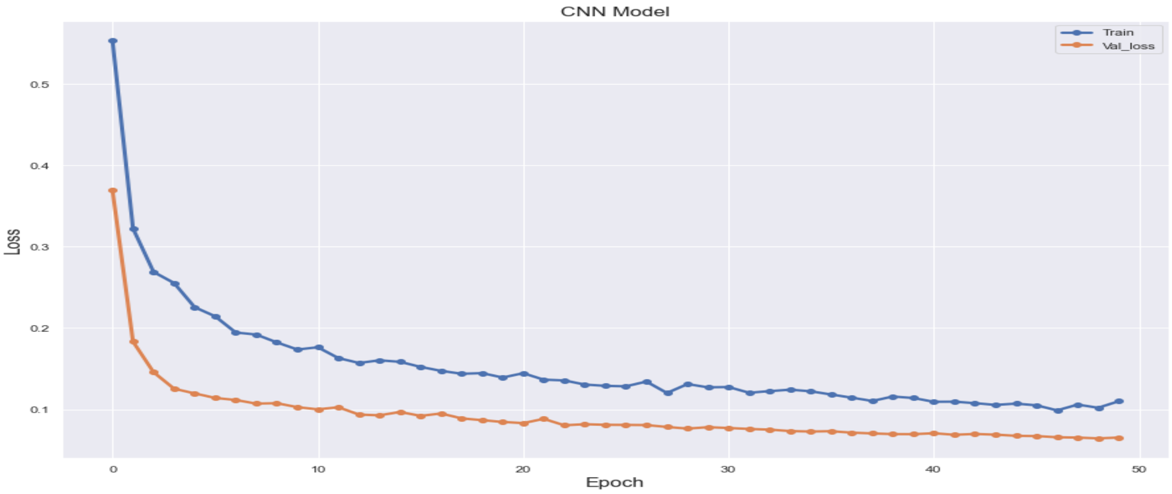

Figure 4.5: validation loss of CNN

The accuracy for training and validation loss is based on the CNN model weights' as randomly setup is shown in Figure 4.5. It provides us with further insight into how the CNN model performs over the course of the training cycle(epoch). The validation loss dropped in the same order as the training over 50 iterations. There is a correlation between the training and validation loss sets from the beginning to the end. The validation loss reduced from point zero to 10 epochs, then gradually declined with spiking patterns until 48 epochs, then it increased from point 49. Longer training intervals may not allow the model to learn additional features from the validation data. The training curve declined sharply from zero to ten, resulting in dangling patterns that rapidly diminished as training time progressed. The validation data was challenging to model in order for the model to learn as efficiently as possible. The difference between training and validation loss grows larger between 5 and 20 epochs and narrows between 20 and 50 training epochs.

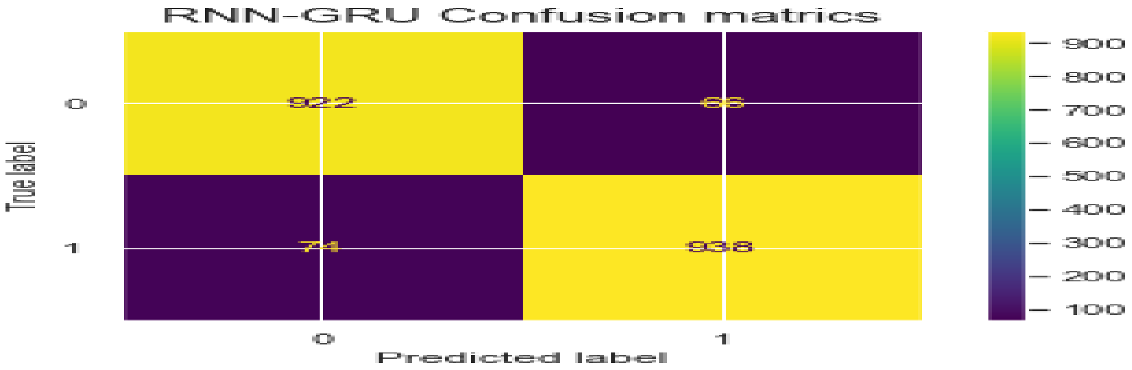

Figure 4.6: Confusion matrix of GRU(**TP**=922, **TN**=938, **FP**=71 and **FN**=66)

Figure 4.6 depicts a gated recurrent unit (GRU) network with true positive, true negative, false positive and false negative classification. The GRU recorded 922 true positives, 938 true negatives, 71 false positives, and 66 false negatives on the testing set. The overall number of correct predictions is the sum of the TP and TN classes (TP+TN=922+938=1860 occurrences). Misclassifications resulted in a sum of FP+FN=71+66 or 137. The GRU network made 66 Type-I and 71 Type-II errors in its categorization. This means that 66 false negative(FN) values and 71 false positive(FP) kinds of data are simultaneously classified as phishing and trustworthy.

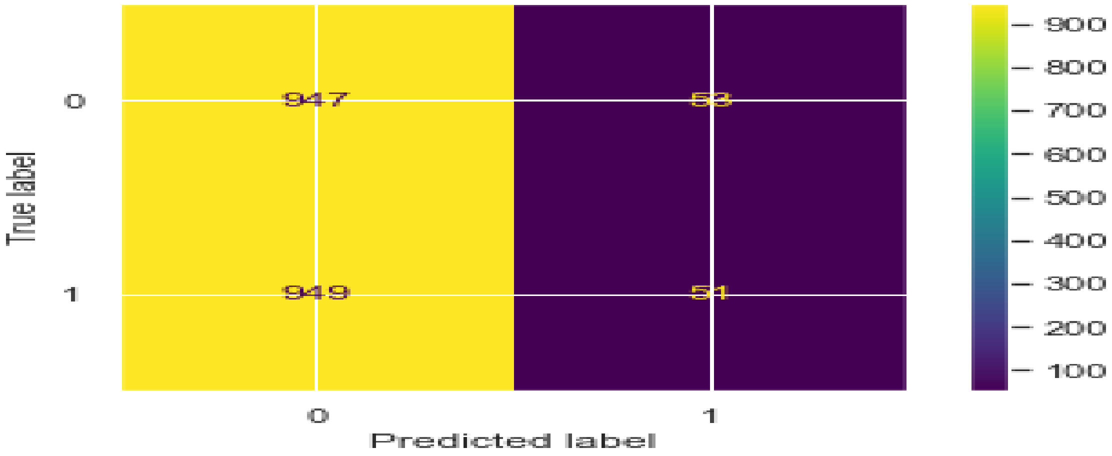

Figure 4.7: Confusion matrix of CNN(**TP**=947, **TN**=51, **FP**=949 and **FN**=53)

A 4×4 confusion matrix showing the total numbers of targeted classes are used to determine the performance of a binary classification report, as shown in Figure 4.15. This is done to compare the predicted values of the CNN model with the actual target values, which are divided into four mutually incompatible possibilities. Figure 4.7 displays the true positive and negative cases of the CNN phishing site detection and classification system. According to the data, 947 are the correct predictions and 949 classes were wrongly classified cases of trusted and Phishing sites. The confusion matric help to explain how well a classification system performs on a set of experimental data for which the true values are known is the artificial neural network confusion matrix. The overall number of correct predictions was 947 + 51 = 998, while wrong predictions yielded 949 + 53 = 1002 instances.

V. CONCLUSION

The proposed CNN achieved reliable and accurate results and GRU methods of operation has proven to be highly ineffective in practice when it comes to detecting phishing URL addresses. The majority of the issues were resolved by the new system, including alerting users and detecting phishing codes found in URL backgrounds. The aforementioned analysis led us to the conclusion that the system identified phishing sites more accurately. This is a standalone ML research serving as the supplementary output that enables future replications and/or modifications of the conducted experiment. Diagnostic tools such as ROC, AUC, confusion matrix, and others make it easy to see and assess the model's performance. It illustrates the trade-offs between the TP and FP classes. The two-dimensional ROC graph is created by plotting the FP rate on the X-axis and the TP rate on the Y-axis.

We recommend that government agencies, programmers, and machine learning engineers utilize this system if they require a system that can accurately distinguish between real and illegitimate URL addresses. This will reduce the alarming frequency of these attacks and help create anti-phishing solutions to counteract misleading URL operations.

REFERENCE

[1]. AlEroud A, Karabatis G.(2020) Bypassing detection of URL-based phishing attacks using generative adversarial deep neural networks. In: Proceedings of the Sixth International Workshop on Security and Privacy Analytics. 16, 53–60.

[2]. Almseidin, M., Abu-Zuraiq, A. M., Al-kasassbeh, M. and Alnidami, N.(2019) Phishing detection based on machine learning and feature selection methods," International Journal of Interactive Mobile Technologies, 13(12), 71–183, doi: 10.3991/ijim.v13i12.11411

[3]. Al-Sarem, M., Saeed, F., Al-Mekhlafi, Z.G., Mohammed, B.A., Al-Hadhrami, T., Alshammari, M.T., & Alshammari, T.S. (2021). An Optimized Stacking Ensemble Model for Phishing Websites Detection. Electronics, 10(11), 1285.

[4]. Afroz, A. and Greenstadt,R.(2020) PhishZoo detecting phishing websites by looking at them, in Proceedings of IEEE Fifth International Conference on Semantic Computing, 368–375.

[5]. Barlow, L., Bendiab, G., Shiaeles, S. and Savage, N.(2020) A Novel Approach to Detect Phishing Attacks using Binary Visualisation and Machine Learning," in Proceedings - 2020 IEEE World Congress on Services, SERVICES, 177–182.

[6]. Basit, A., Zafar, M., Liu, X., Javed, A.R., Jalil, Z., Kifayat, K., (2021). A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommun. Syst. 76(1), 139–154. https://doi.org/10.1007/s11235-020-00733-2.

[7]. Basit, A. Zafar, M., Javed, A. R. and Jalil, Z.(2020) A Novel Ensemble Machine Learning Method to Detect Phishing Attack, Telecommun. Syst, 23(4), 1-20. doi: 10.1109/INMIC50486.2020.9318210.

[8]. Benavides, E., Fuertes, W., Sanchez, S., & Sanchez, M. (2020). Classification of phishing attack solutions by employing deep learning techniques: A systematic literature review. In Developments and advances in defense and security, 51–64.

[9]. Chiew, K.L., Tan, C.L., Wong, K., Yong, K.S., & Tiong, W.K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Information Sciences, 484, 153-166.

[10]. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.

[11]. Fu, A. Y., Liu, W. & Deng, X. T.(2021). Detecting Phishing web Pages with Visual Similarity Assessment based on Earth Mover's Distance (EMD), *IEEE Transactions on Dependable and Secure Computing,* 3(4), 301-311.

[12]. Gowtham, R. Krishnamurthi, L. and Kumar, S.(2021) An efficacious method for detecting phishing webpages through target domain identification, Journal of Decision Support Systems, Elsevier Press, 1-20.

[13]. Hong, J. (2020). The state of phishing attacks. Communications of the ACM, 55(1), 74-81.

[14]. Huang, C. Ma, S. Chen, K.(2020) Using one-time passwords to prevent password phishing attacks, Journal of Network and Computer Applications, Elsevier Press, 1-10.

[15]. Jain A. K. & Gupta, B. B.(2020). A Novel Approach to Protect Against Phishing Attacks at Client-Side Using Auto-Updated White-list, EURASIP Journal on Information Security, 16(1), 9.

[16]. Jain, A. K, Yadav, S. K. and Choudhary, N.(2020) A novel approach to detect spam and smishing SMS using machine learning techniques, International Journal of E-Services and Mobile Applications (IJESMA), 12(1), 21-38.

[17]. Javed, A. R., Jalil, Z., Moqurrab, S. A., Abbas, S., and Liu, X. (2020), Ensemble adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles, *Transactions on Emerging Telecommunications Technologies*, 45.

[18]. Kaur, S., Gupta, S., Singh, S. and Gupta, I.(2020), Hurricane Damage Detection From Satellite Imagery Using Convolutional Neural Networks, International Journal of Information System Modeling and Design, 13(10), 1-3

[19]. Kumar, J., Santhanavijayan, A. Janet, B., Rajendran, B. and Bindhumadhava, B. S. (2020) Phishing website classification and detection using machine learning, International Conference on Computer Communication and Informatics(ICCCI), 45, 3-20.

[20]. Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2020). A stacking model using url and html features for phishing webpage detection. Future Generation Computer Systems, 94, 27–39

[21]. Liu, X., Fu, J.,(2020). SPWalk: Similar Property Oriented Feature Learning for Phishing Detection. IEEE Access 8, 87031–87045. https://doi.org/10.1109/ ACCESS, 2992381.

[22]. Liu, G., Qiu, B. & Wenyin, L. (2020) Automation of Phishing Target from Phishing Web-pages, International Conference on Pattern Recognition, 50, 4153-4156.

[23]. Maniraya, S., Aditya, S., swarna, D. S. and Shaab, A.(2019), Credit Card Fraud Detection using Machine Learning and Data Science, International Journal of Engineering and Technological Research, 8(9), 110-115.

[24]. Manohar, S, Bedi. A., Kumar. S, Singh. Kr. S.(2020) Fraud detection in credit card using machine learning techniques, International Research Journal of Engineering and Technology, 07(04), 1786-1791.

[25]. Maurya, S. and Jain, A. (2020). Deep learning to combat phishing, *Journal of Statistics and Management Systems*, 1–13.

[26]. Maurya, S., Saini, H. S. and Jain, A.(2019) Browser Extension based Hybrid Anti-Phishing Framework using Feature Selection, International Journal of Advanced Computer Science and Applications, 10(11), 20-30.

[27]. Mittal, M., Iwendi, C., Khan, S., and Rehman-Javed, A. (2020). Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using Levenberg–Marquardt neural network and gated recurrent unit for intrusion detection system. *Transactions on Emerging Telecommunications Technologies*, p. e3997.

[28]. Pastor-Galindo, J., Nespoli, P., Mármol, F. G. and Martínez Pérez, G. (2020) The Not Yet Exploited Goldmine of OSINT: Opportunities, Open Challenges and Future Trends, in IEEE Access, 8, 10282-10304, doi: 10.1109/ACCESS.2020.2965257.

[29]. Rashid, J., Mahmood, T., Nisar, M. W., Nazir, T.(2020) Phishing detection using machine learning technique, in: 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), 43–46.

[30]. Safi, A. and Singh, S.(2023), A systematic literature review on phishing website detection techniques, Journal of King Saud University – Computer and Information Sciences, 35(2), 591-611.

[31]. Sandhya, G. V., and Kumar, B. T.(2021), Phishing attack detection using deep neural network, International journal of advanced research in computer and communication engineering, 10(6), 633-642.

[32]. Sindhu, S., Patil, S.P., Sreevalsan, A., Rahman, F., Saritha, A.N., (2020). Phishing detection using random forest, SVM and neural network with backpropagation. In: Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics(ICSTCEE), 391–394. https://doi. org/10.1109/ICSTCEE49637.2020.9277256.

[33]. Verma, R., Shashidhar, N., & Hossain, N. (2020). Detecting Phishing Emails the Natural Language Way. In Computer Security–ESORICS, 824-841.

[34]. Whittaker, C., Ryner, B. and Nazif, M.(2020), Large-Scale Automatic Classification of Phishing Pages**.,** Conference: Proceedings of the Network and Distributed System Security Symposium, NDSS 2010, San Diego, California, USA,1- 20

[35]. Zamir, A, Hu, K., Iqbal, T., Yousaf, N., and Aslam F.(2020), Phishing web site detection using diverse machine learning algorithms, The Electronic Library, 38(1), 65–80.

[36]. Zhu, E., Ju, Y., Chen, Z., Liu, F., Fang, X.,(2020). DTOF-ANN: an artificial neural network phishing detection model based on decision tree and optimal features. Appl. Soft Comput. J. 95,. https://doi.org/10.1016/j.asoc.2020.106505 106505.