Volume 10, Issue 7, July – 2025 International Journal of Innovative Science and Research Technology
ISSN No: 2456-2165 https://doi.org/10.38124/ijisrt/25jul1306

A Comparative Study of Malicious URL Detection Model: CNN vs. Logistic Regression and Gated Recurrent Units

Dr. Umejuru Danie[1]; Dr. Ugbari Augustine[2]

[1]Department of Computer Science, University of Port Harcourt, Choba, Nigeria
[2]Department of Information Technology, University of Port Harcourt, Choba, Nigeria

Publication Date: 2025/08/18

Abstract: Malicious URLs pose a serious threat on the world wide web to users all over the world. The challenges emanating from URLs which are malicious are many and very worrisome to internet users globally. This has informed, and thus propels the development of newer models to solve the lingering challenge in the Cyber Security space. These newer notification and detection models are been developed in other to mitigate the gaps and also curb the challenges caused by unknowingly using or clicking further using a malicious URL. This study aims at developing a novel malicious URL detection and notification model by using CNN and further incorporating CNN with penalty term on kernel, weight and bias in other to increase models detection accuracy, reduce time complexity and also address misclassification issues as well as poor prediction accuracy. The CNN with penalty term is being used against Logistic Regression (LR) and Recurrent Gated Units (RNN-GRU) which increased the resilience of the suggested model as well as enhancing classification prediction. The diagnostic tools employed for the proposed model are accuracy, confusion matrix, precision, recall, F1 score, and AUC-ROC. This study outlined a novel method capable of identifying malicious URLs using 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 also identifying directories, keyword arguments, and extensions. Summary of the experimental results shows that the proposed CNN with penalty term (98.2%) fared better than LR and RNN-GRU approaches which yielded a recommendable prediction accuracy of 89.85% and 91.5% respectively.

***Keywords**: CNN, Logistic Regression, Gated-Recurrent Network, URL, Penalty Term.*

How to Cite: Dr. Umejuru Danie; Dr. Ugbari Augustine (2025). A Comparative Study of Malicious URL Detection Model: CNN vs. Logistic Regression and Gated Recurrent Units. *International Journal of Innovative Science and Research Technology*, 10(7), 3599-3611. https://doi.org/10.38124/ijisrt/25jul1306

I. INTRODUCTION

A malicious 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 (AL-Otaibi et al., 2020) The attacks are URL links presented as real with a subject or message designed to deceive recipients into providing critical information after which users are been redirected or even defrauded right away (Baig et al., 2021). 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 (Javed et al.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 (Dželila and Kevrić 2020). (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 by just a click on the malicious URL. 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 arranged: 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.

➢ *Deep Learning (DL) for Malicious URL 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 malicious 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.

- *Convolutional Neural Network (CNN) as a Tool for Malicious Attacks*

There are numerous CNN variations, such as VGG16, AlexNet, ResNet, and others. The structures like CNN with 1D, 2D and 3D kernels have all been studied in great detail; numerous adjustments to the loss functions (specific emphasis) and adaptations to new CNN models, such as U-Nets and ResNets, have enhanced efficiency. However, when it comes to accurately detecting malicious activities with other than images, CNN designs are not as effective as they may be, mostly due to their variability in terms of place of residence, shape, size, picture intensity, and texture. Even though these methods relied on the self-extraction of capabilities, it was discovered that a number of them produced superior results when further details were included, such as the inclusion of atlas coordinates to demonstrate dependency. The CNN designs are determined by the kernel size, which has an impact on system performance. Greater parameterization results from larger kernels and smaller kernels cause specific regions to be missed. Therefore, some sort of compromise is necessary. The majority of the datasets under consideration appear to be highly unbalanced, which causes over-fitting. This might be avoided by combining data fusion with precision or recall based on an objective function. (Duffner et al. 2021). (Liu et al., 2023) used algorithms based on computer vision to detect URLs that were phishing attempts. This is due to URLs that are malicious frequently which features of a real URL intended to impersonate legitimate websites and fool visitors into clicking on them. CNN training necessitates the use of a large number of phishing and legal URL addresses to assist the model in learning patterns and features. CNNs are regarded the best tool for identifying phishing URLs because they are capable of learning important features from basic input data. This implies that as opposed to manually creating attributes based on expertise in the field, the network learns and extracts the most relevant features for the challenge. CNNs are extremely scalable and can handle enormous amounts of data to increase its precision by training on robust computer clusters.

- *Logistic Regression (LR)*

LR, also known as the logistic or logit model, is a tool for examining the relationships underlying categorical variables that are dependent and a large number of independent factors. It also determines the likelihood that an event will occur by matching data to a logistic curve. The S-shaped curve of the logistics regression model rises initially linearly and then exponentially. The LR method can be used to estimate the likelihood of a specific outcome that depends on individual factors by fitting a regression curve with the formula $y=f(x)$. LR can be trained to fit a curve to illustrate the relationship between both independent and dependent variables whenever the output is a binary variable and the dependent variable has a numerical value.

- *Recurrent Neural Networks as a Tool for URL Address Detection*

Recurrent neural networks are one of the neural network architecture paradigms that had generated achievements in this discipline (RNNs). RNNs are a type of neural network that uses sequential input to generate sequential output by sharing parameters between time steps. Natural language processing, sensitively and appropriately, and speech recognition have all seen breakthroughs thanks to RNNs. RNNs have proven beneficial in various sequence and series-based learning applications, but its application to raw time series prediction remains underutilized. (Yuan *et al.* 2020) The R-NN with KNN was adopted to improve high level image features detection in comparison with the original object. The R-NN model worked superbly with the KNN dataset using polynomial feature injection.

III. METHODOLOGY

A. Methodology

The methodology majors on the tools and methods needed to detect malicious URL domains. This section addresses the feasibility of the proposed technique for detecting malicious URLs, as well as the CNN classifier's evaluation results on the test set. The object oriented analysis and methodology was been used to ensure model is effective, efficient, reliable, reusable and fast as well and was basically structured into object analysis, design, implementation and testing.

➢ *URL Data Collections:*

The data collecting task is important to the historic study of machine learning industry. Our dataset was obtained from kaggle and consists of 10000 records with 5000 trustworthy and 5000 malicious URLs respectively and 49 features.

➢ *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.

➢ *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.

➢ *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 vector is incorporated into the new hidden state. A candidate vector 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.

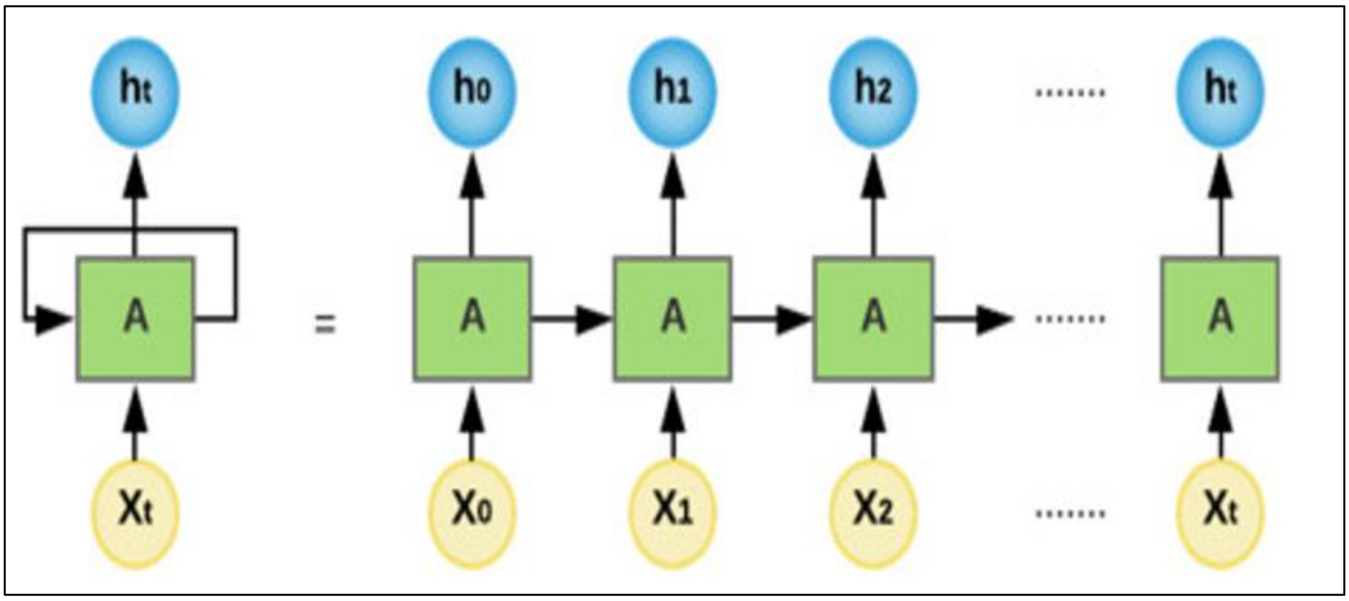

Fig 1 the GRU Network Structure (Source: Sandhya & Harish, 2021)

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 Xt, for t runs of state and produces a special input (Xo) when t=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.

➢ *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.

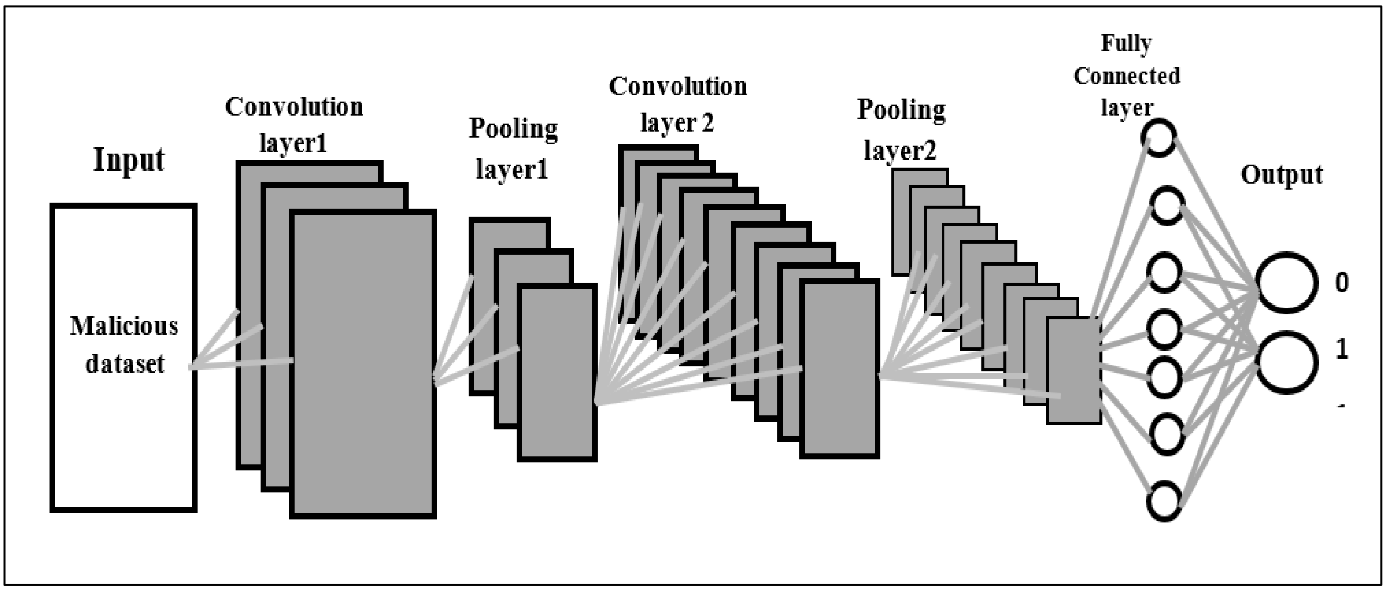

Fig 2 Design of the CNN Model

The CNN structure uses a convolutional technique to identify and differentiate between the numerous features on phishing dataset for analysis. The network has multiple pairs of spooling or convolution-based layers, and the layers are completely linked with the output features from the previous layer. The goal of the CNN design is to overcome model complexity and improve detection accuracy. The proposed CNN design is made up of three important layers such as the convolutional layers, pooling and fully connected layers.

- *Convolutional Layer:*
  The convolutional layer is the first layer used to extract features from input phishing set and its mathematical operations of convolution are carried out between the input data and a filter of a particular size.

- *Pooling Layer:*
  The purpose behind the CNN's convolutional layer, which is complemented by a pooling layer, is to cut down on the size of the convoluted feature mapping in order to lower the computational cost.

- *The Fully Connected Layer:*
  the fully linked layer houses neurons as well as the CNN algorithm weights and biases. Additionally, this particular layer is positioned before of the output layer, which makes up the next-to-last layer.

B. Implementation of Our Approach

➢ *Adding Penalty to the CNN Kernel:*

A term for L1-penalty was included to the CNN layer's bias vector, which proves helpful at times although the bias usually has less effect on the model's complexity. It had an object with the value of the coefficient of the penalty term, or l, as a parameter. We implemented weight reduction to the CNN layers and introduced a penalty term equal to 0.01 times the square root of the norm of the weights. It produced a basic CNN neural network with two hidden layers when applied to convolutional or dense layers.

➢ *Adding Penalty to Activated Weight:*

study introduced a penalty term in other to make the CNN results to be smaller (or more like 0); the model converges more quickly and with more accuracy because it gives the user control over the layer's output. The CNN framework is discouraged from having large weights by this penalty term, which is determined by the value of the weights.

➢ *Adding Penalty Term to CNN Bias Weights:*

A penalty term was also added to the layer's bias weights. The learning CNN algorithm's bias weights are modified to promote the use of small weights by the network. We modified the loss computation utilized in the network optimization process to take the weight sizes into account. This is done to reduce model weights in the optimization process.

IV. RESULT AND DISCUSSIONS

The findings of the CNN, LR, and RNN-GRU 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 ROC curves, charts, tables and comparative graphs.

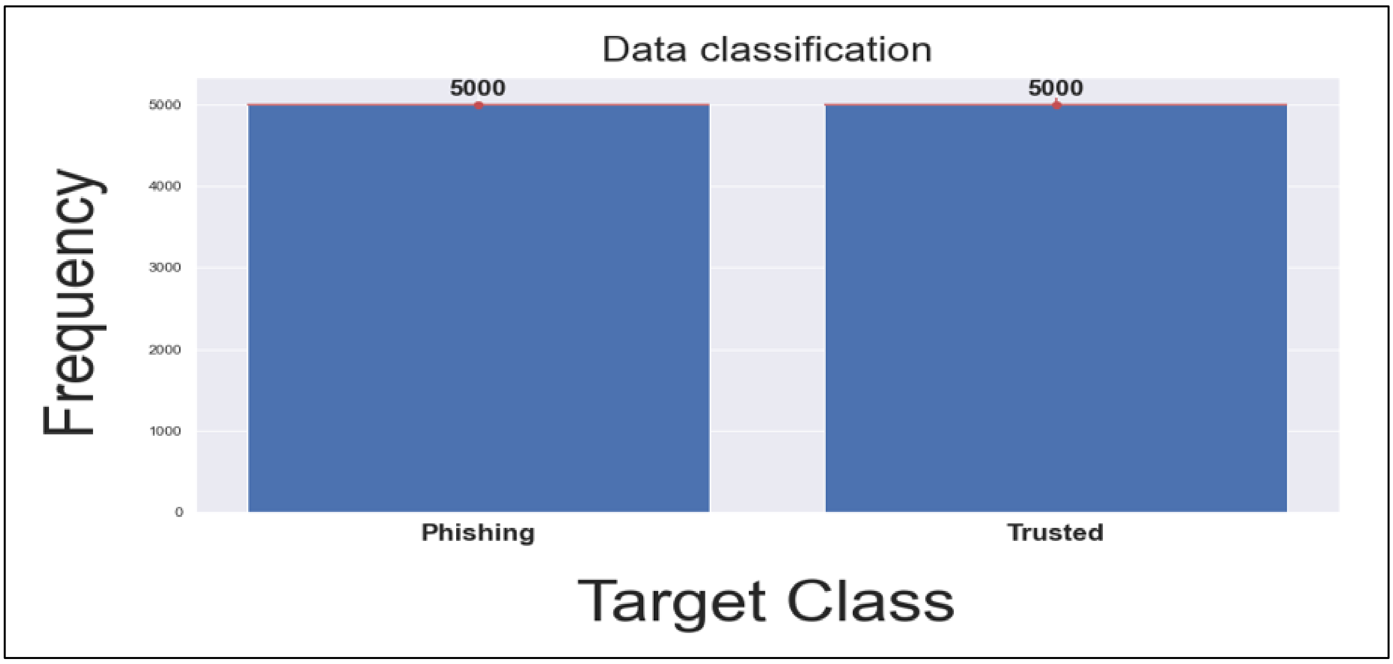

Fig 3 Target Classification

Fig 3 depicts the target classification of dataset used for training and testing of model, which resulted in a total of 10,000 records. A balance is established between malicious and trusted URL addresses to avoid bias in model predictions. The dataset contains 5,000 set for phishing and trusted URL addresses respectively, as shown at the top of each bar with labels.

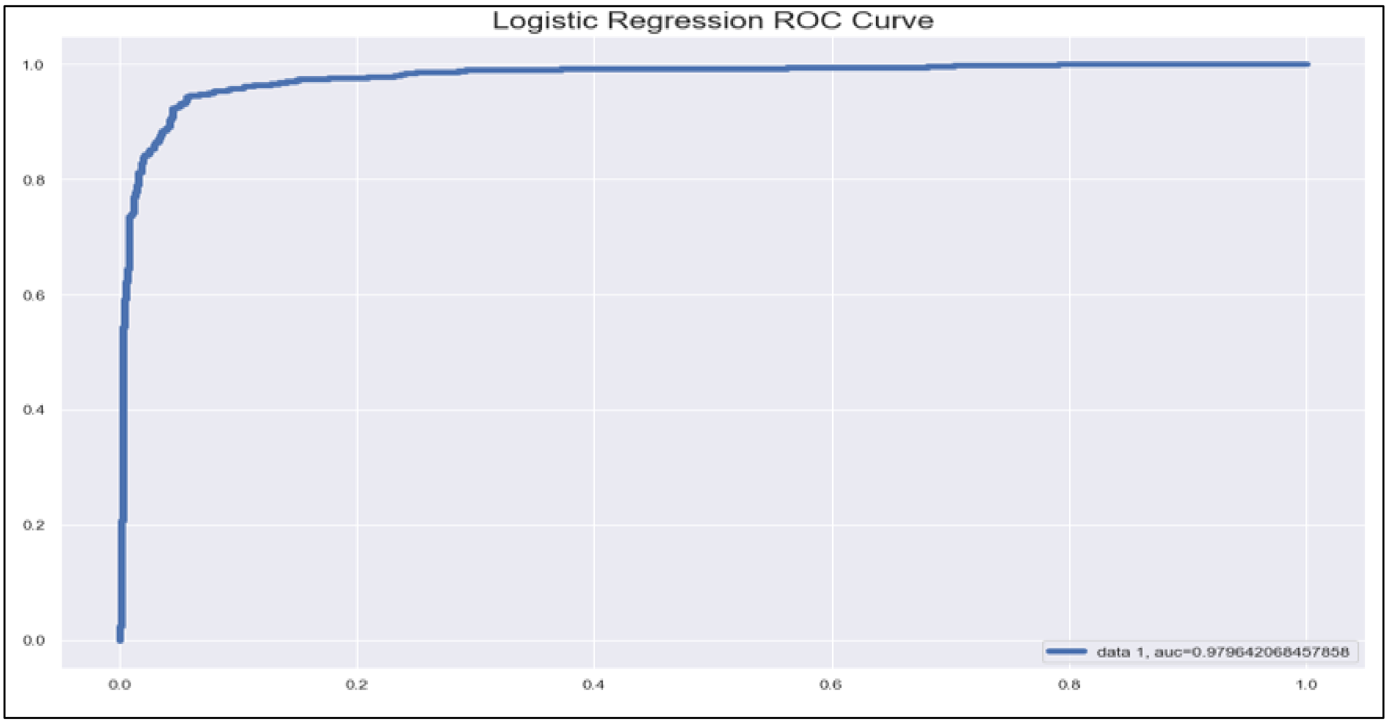

Fig 4 Logistic Regression Roc Curve

Fig 4 depicts the receiver operating characteristic (ROC) curve, which is an important diagnostic tool for evaluating a logistic regression model and determining its performance. The ROC value recorded 0.979642 rate impacted by overrfitting instances demonstrating the true positive rate (TPR) and false positive rates at every feasible threshold. The ROC value offered 0.979, indicating that the LR algorithm has a 97.9% possibility of distinguishing between both positive and negative classifications.

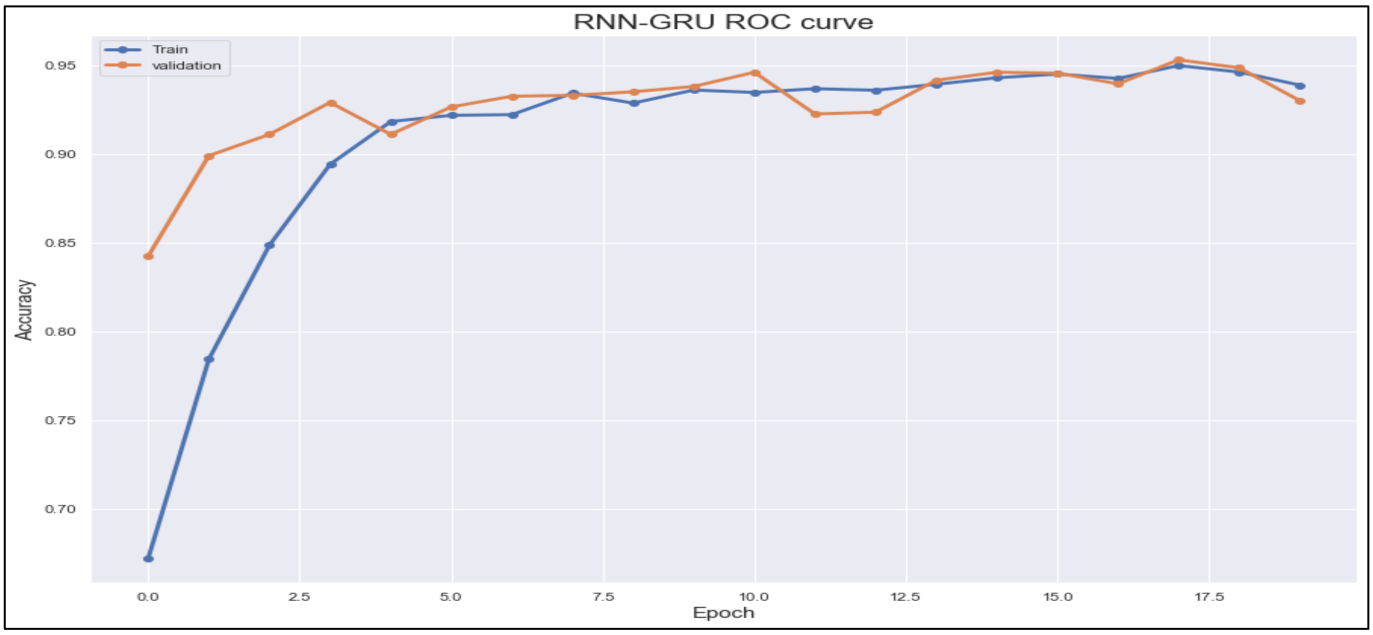

Fig 5 Training Accuracy of RNN-GRU

Fig 5 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. This figure shows an uneven movement of the training and validation curves. The RNN-GRU model struggles to learn desirable malicious attack features from the training and validation datasets. The attack accuracy grew and peaked about 17 epochs, then dropped over extended training periods.

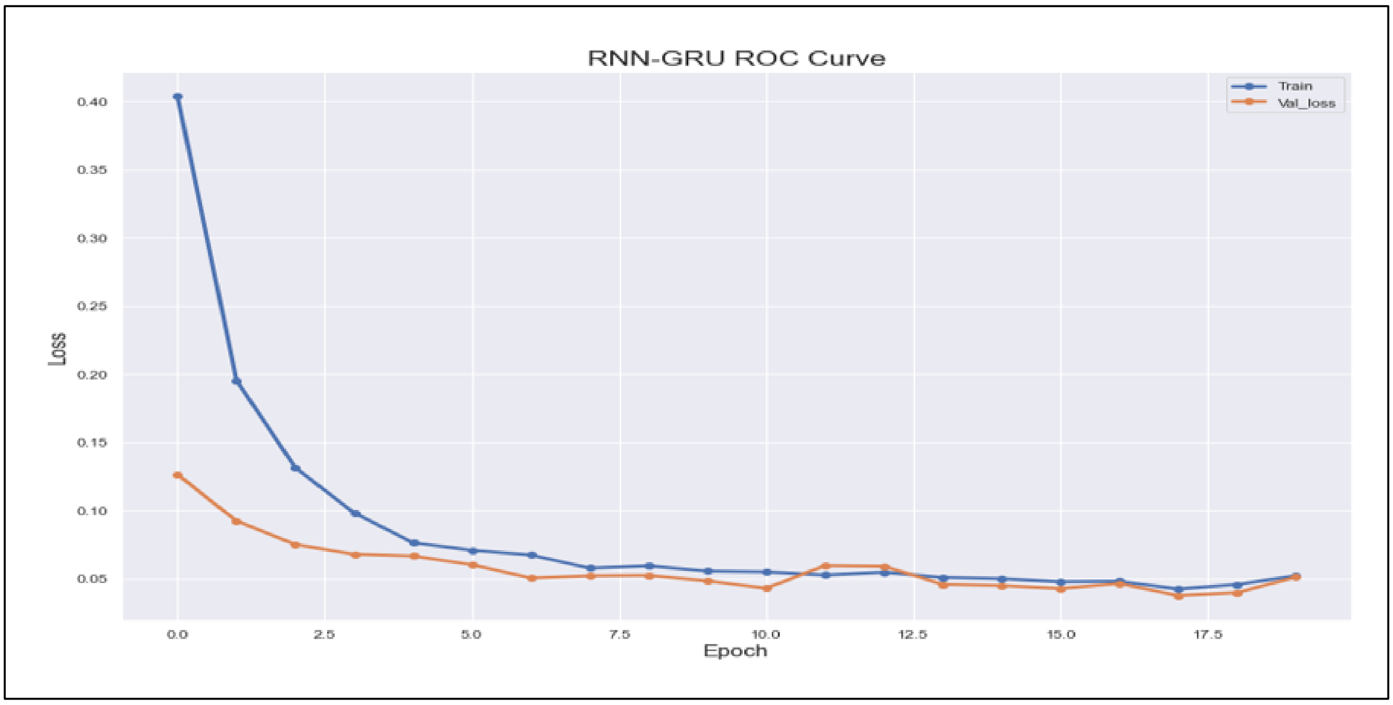

Fig 6 Validation Loss of RNN-GRU

Fig 6 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 epochs.

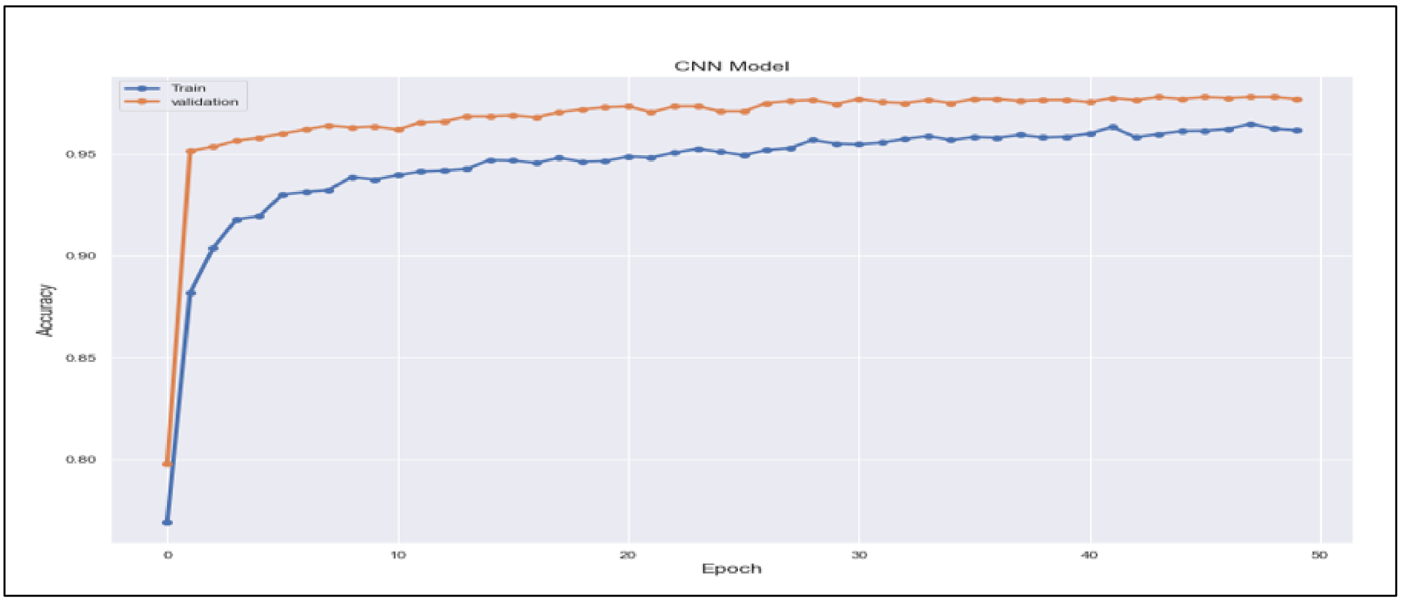

Fig 7 Training Accuracy of CNN

Fig 7 shows the CNN training accuracy and validation. 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.

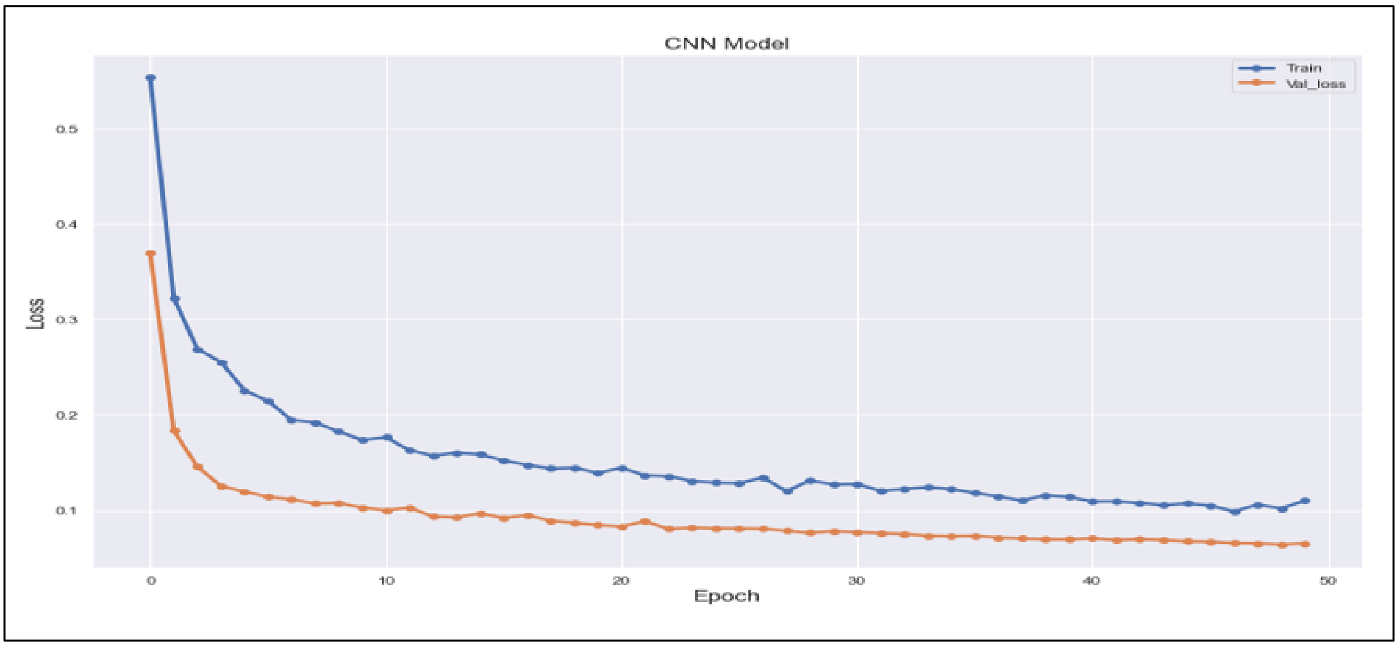

Fig 8 Validation loss of CNN

Fig 8 shows the accuracy for training and validation loss based on the CNN model weights' as randomly setup.. 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.

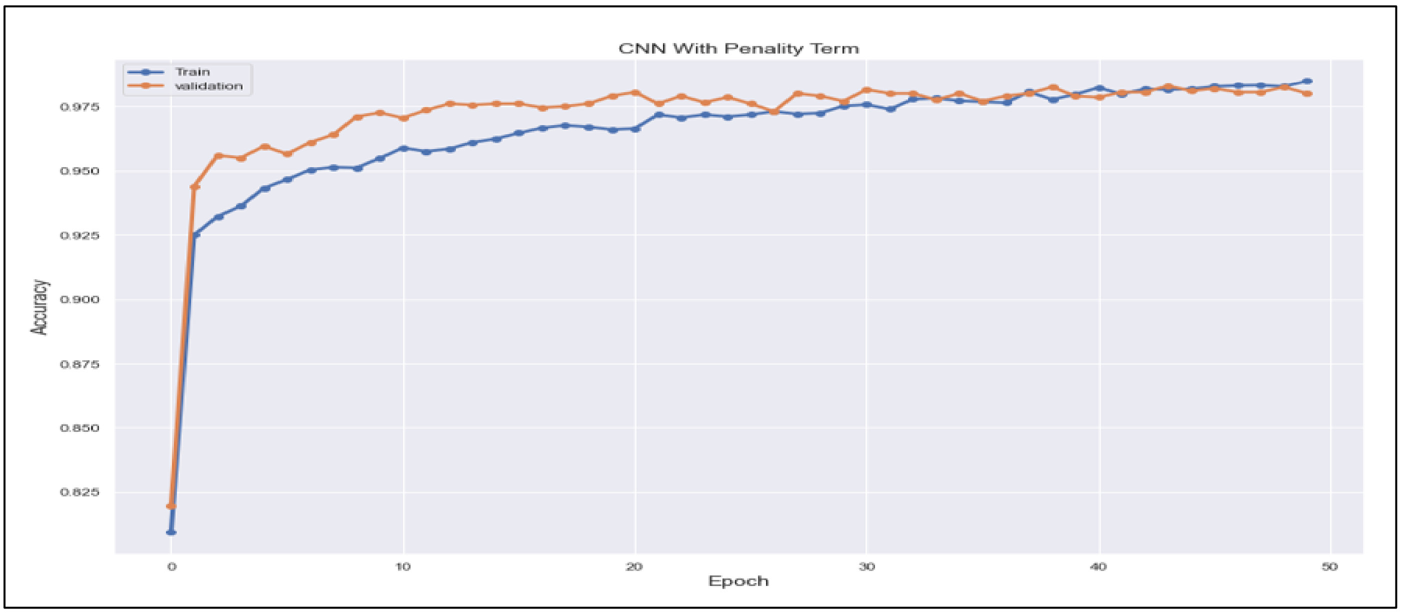

Fig 9 Training Accuracy of CNN with Penalty Term

Fig 9 shows the behavior of training and validation sets for the various training cycles in the validation accuracy as higher, demonstrating that the model outperforms its training accuracy. The validation curve is somewhat larger at the start (from 0 to 20) than the training curve and grows in tandem with the validation from 20 to 50 intervals, showing that the model did well when generalizing with the testing set. The CNN model performed better with validation set than training set for smaller iterations, although it performed better during longer or extended training periods. The disparity between training and validation increases between 0 and 20 epochs and decreases between 20 and 50 epochs. The training and validation curves overlapped between 25 and 45, with divergent patterns at 45 and higher.

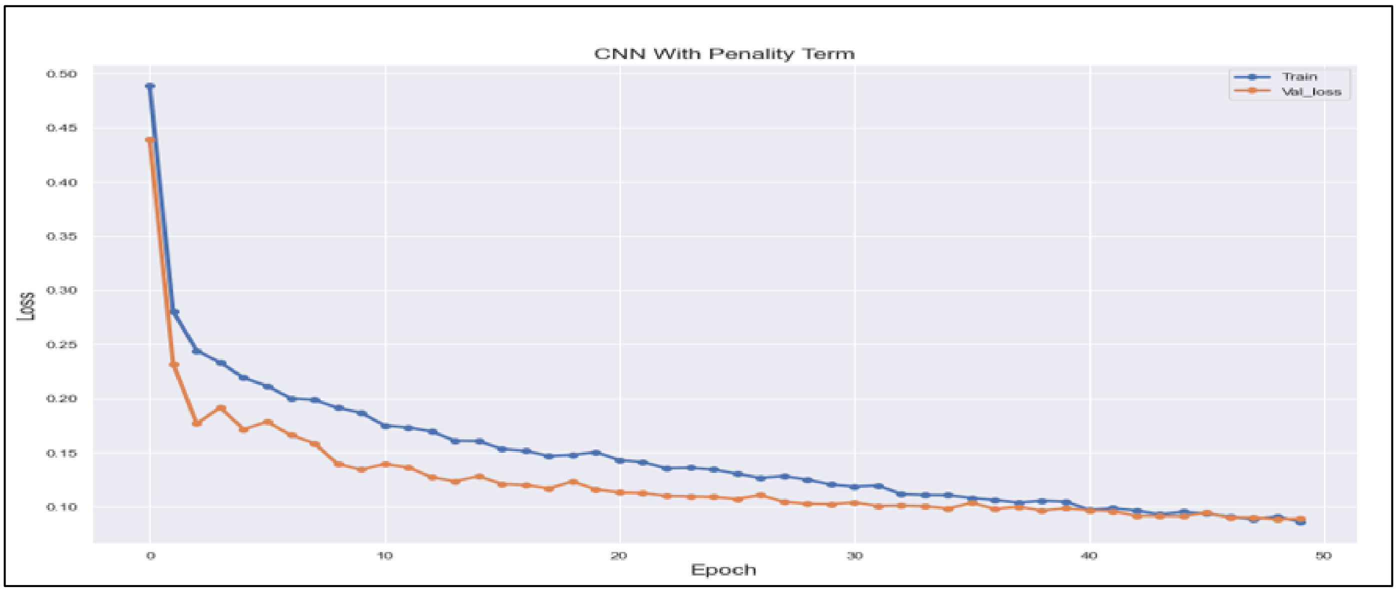

Fig 10 Validation Loss of CNN with Penalty Term

Fig 10 describes training against validation loss. The training is better, and the validation loss is lower than the training loss. The validation loss in the above instance is reduced, suggesting that the predictive model is converging as expected. The training data proves more challenging and the validation loss is somewhat lower than the training loss, even though both losses are decreasing in the plot. The CNN model with penalty term receives novel information for both sets because the training and validation losses are closely separated at the beginning and lies at the same plane from 40 epochs, The training and validation loss had a larger margin from 5 to 30 epochs, a narrower margin from 30 to 50 iterations, and a diplomatic tire from 40 to 50 in a hanging curve. The model proved capable to learn more advantageous malicious features from the validation and training sets throughout longer training periods than shorter training time.

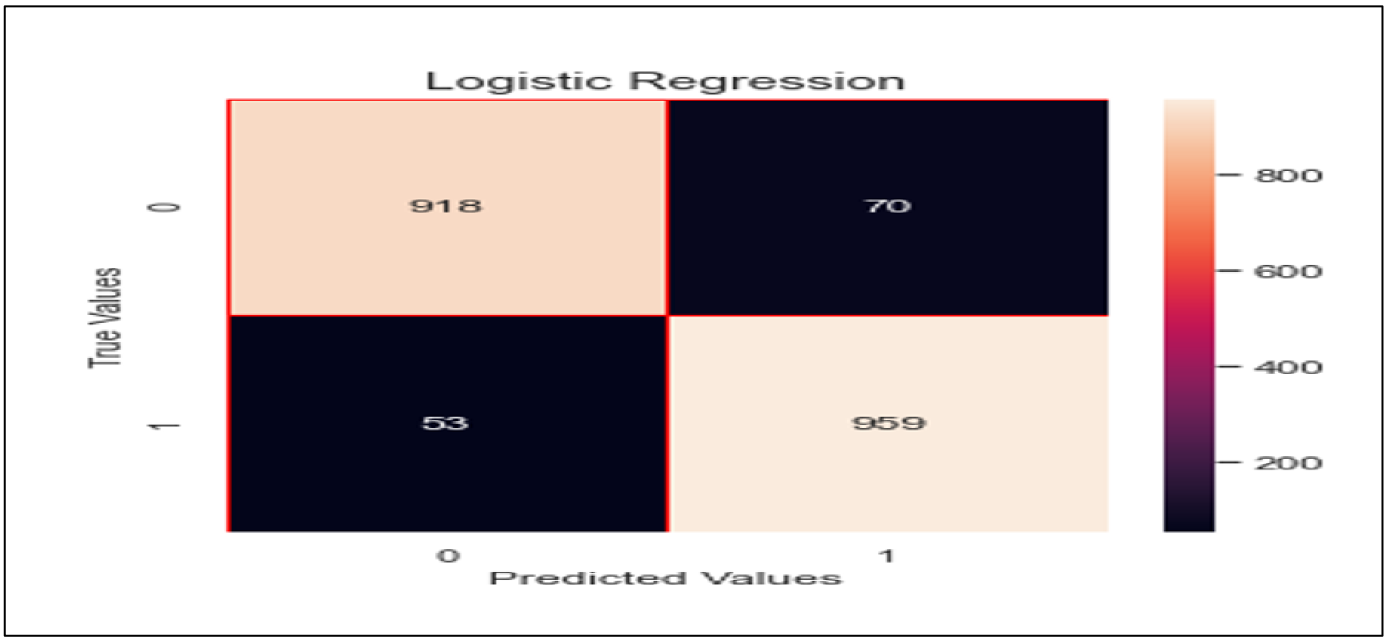

Fig 11 Confusion Matrix of LR

Fig 11 shows the confusion matrix which displays a table structure of the various prediction results of a binary-classification task. This is used to show the predicted and actual values of a classification model. Cell values above and below the main diagonal or off-diagonal elements showing the incorrectly predicted values. The total numbers of correctly predicted values are equal to the actual or true values. The greater the diagonal value, the more accurate the predicted model results. According to the confusion matrix, URL address with trusted content had 70 incorrectly predicted cases with 918 correct predictions. While URL sites with phishing content provided 53 incorrectly predicted values with 959 true positive class predictions. The overall number of correct predictions was 918 + 959 = 1877, while wrong predictions yielded 53 + 70 = 123 instances.

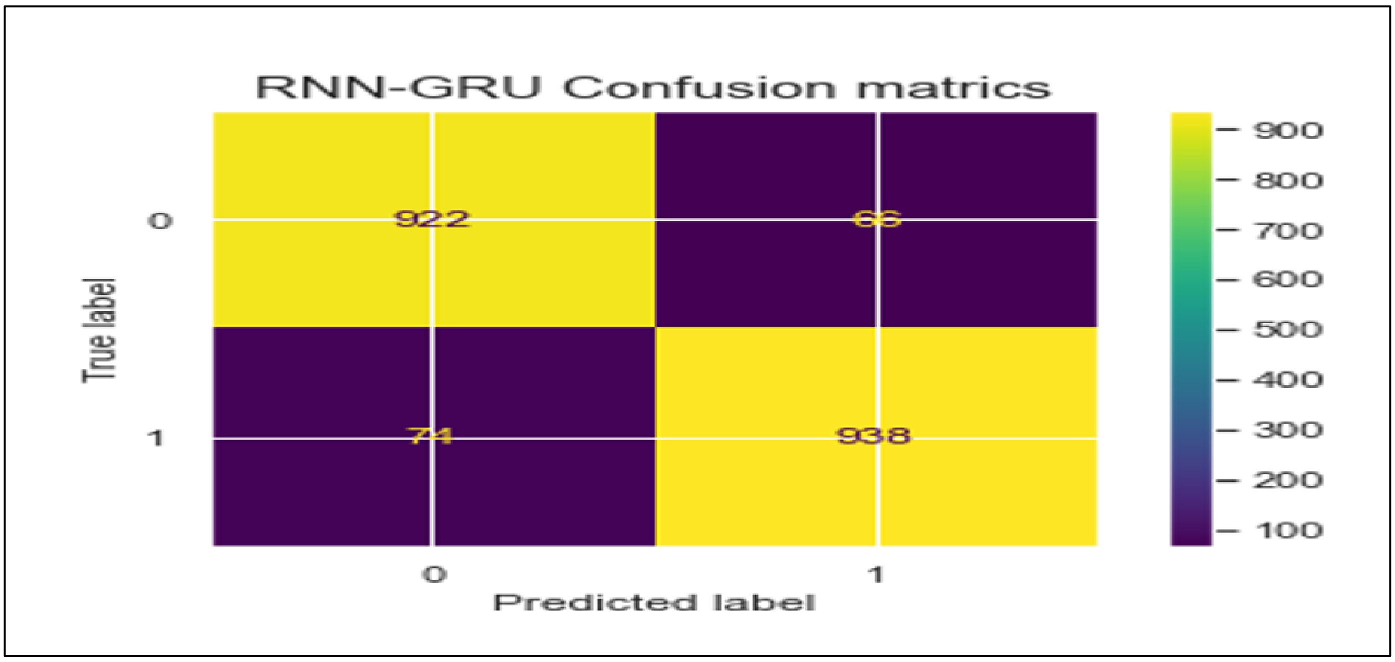

Fig 12 Confusion Matrix of GRU

Fig 12 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.

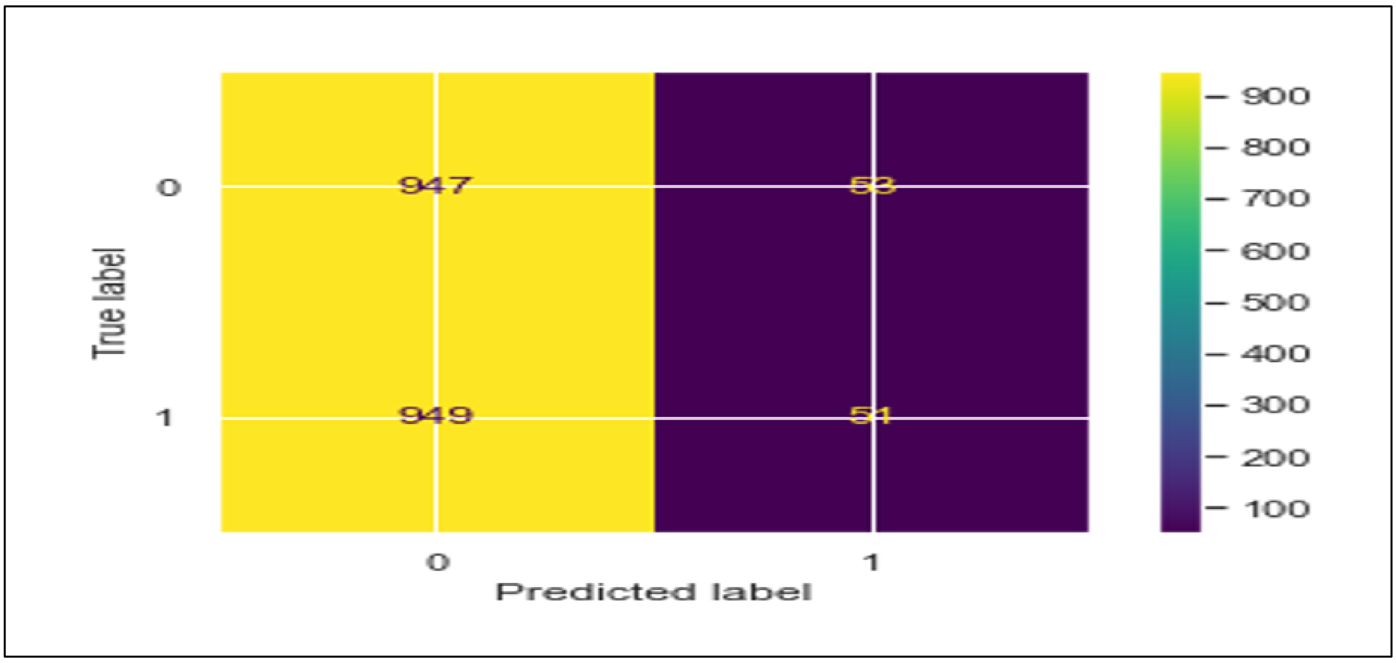

Fig 13 Confusion Matrix of CNN

Fig 13 shows the total numbers of targeted classes are used to determine the performance of a binary classification report. 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. It 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 malicious sites. The confusion matrix help to explain how well a classification system performs on a set of experimental data for which the true and false values are known. The overall number of correct predictions was 947 + 51 = 998, while wrong predictions yielded 949 + 53 = 1002 instances.

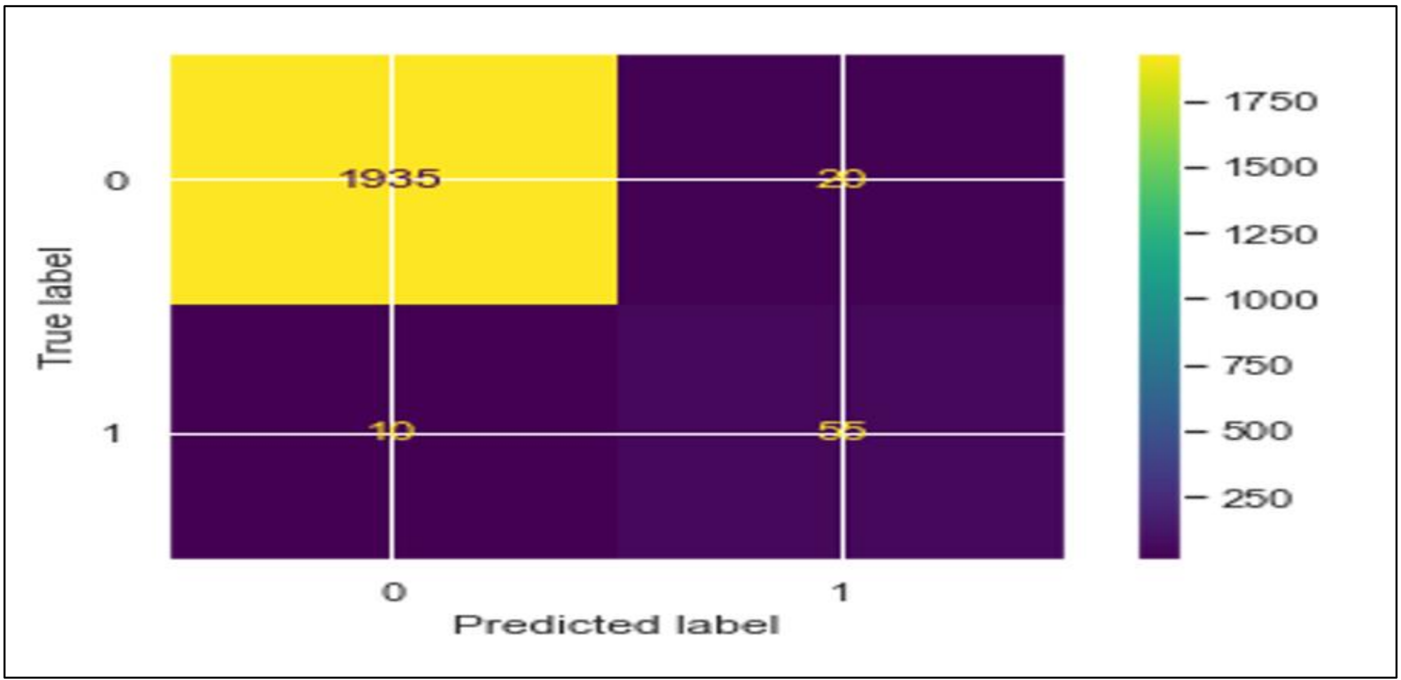

Fig 14 Confusion Matrix of CNN with Penalty Term

Fig 14 shows a confusion matrix used to evaluate the performance of CNN classification system with penalty term using the validation set. It displays the type of errors made by the classifier. The suggested model confusion matrix was created, with accurate predictions displayed at the secondary diagonal and inaccurate predictions noted above and below the main diagonal, or "off-diagonal elements," in that order using the testing dataset. overall number of correctly predicted values recorded to be TP+TN = 1935+55, or 1990 instances with malicious URL addresses; while incorrectly predicted cases yielded FT+FN = 10+20 or 30 cases.

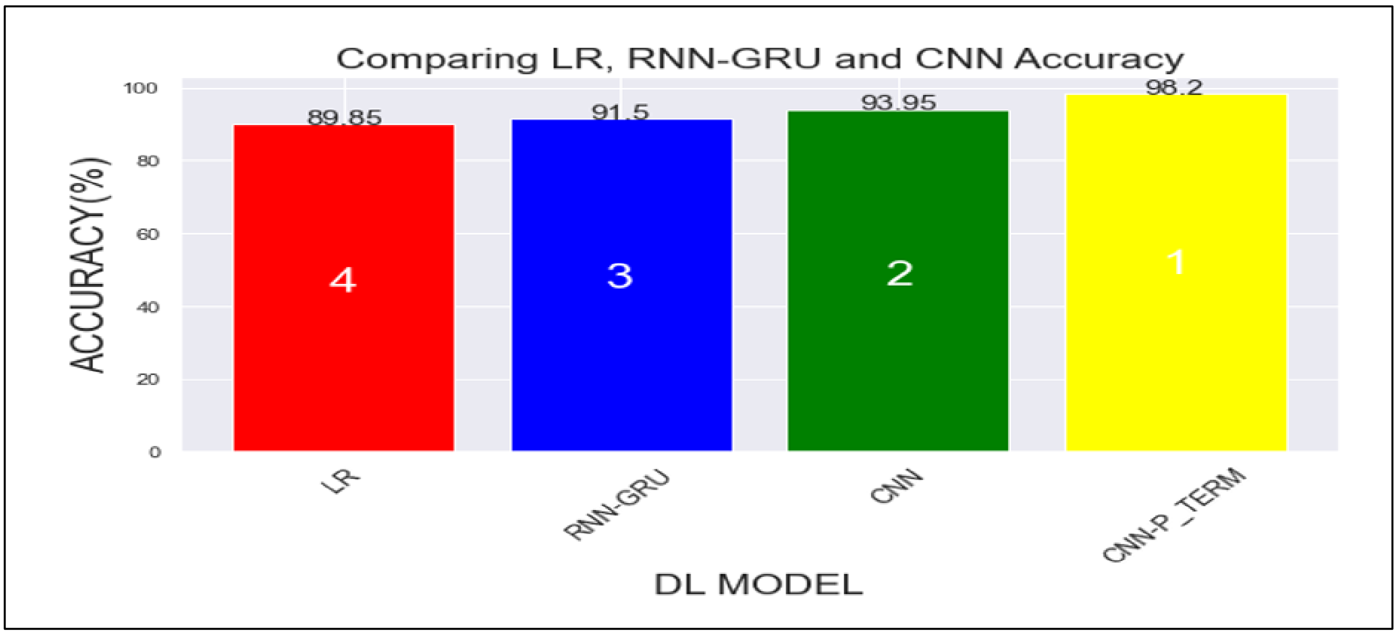

Fig 15 Comparing LR, RNN-GRU and CNN Accuracy

Fig 15 displays the performance accuracy of LR, GRU and suggested CNN with penalty term shown. The CNN with weight penalizing concept excelled and produced the best results, with an accuracy of 98.2%; CNN without penalty terms came second with 93.95%, and the added weight penalizing idea enhanced model detection accuracy while decreasing the difficulty of the model training challenge. The existing LR had the lowest prediction accuracy of 89.85%, followed by GRU which gave 91.5%. The weight penalty concept was added to the CNN optimization process in order to penalize the model for not learning from smaller or lesser weights, which produced a more accurate predictive model.

Table 1 Classification Report of Logistic Regression(LR)

	Precision	**Recall**	**F1-score**	**Support**
TRUSTED	0.51	0.49	0.50	1000
PHISHING	0.41	0.52	0.51	1000
Accuracy			0.51	2000
macro avg	0.51	0.51	0.51	2000
weighted avg	0.51	0.51	0.36	2000

Table 2 Classification report of RNN-GRU

	Precision	**Recall**	**F1-score**	**Support**
TRUSTED	0.50	0.97	0.66	1000
PHISHING	0.49	0.03	0.03	1000
Accuracy			0.58	2000
macro avg	0.50	0.50	0.36	2000
weighted avg	0.50	0.50	0.36	2000

V. CONCLUSION

The proposed CNN with penalty term achieved reliable and accurate results than LR and RNN-GRU methods of operation and has proven to be highly ineffective in practice when it comes to detecting malicious URL addresses. The major issue resolved by the new system includes time complexity and detecting malicious codes found in URL backgrounds. The aforementioned analysis led us to the conclusion that the system identified malicious 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.

REFERENCES

[1]. AL-Otaibi, A. F. and Alsuwat, E. S.(2020) A study on social engineering attacks: phishing attack, International Journal of Recent Advances in Multidisciplinary Research, 7(11), 6374-6380.

[2]. Baig, M. S. Ahmed F. and Memon, A. M.(2021) Spear-Phishing campaigns: Link Vulnerability leads to phishing attacks, Spear Phishing electronic/UAV communication-scam targeted, 2021 4th International Conference on Computing & Information Sciences (ICCIS), 1-6, doi: 10.1109/ICCIS54243.2021.9676394.

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

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

[5]. Duffner, S. Garcia, C.(2021) An online back propagation algorithm with validation error based adaptive learning rate, in: Artificial Neural Networks, Porto, Portugal, 34.

[6]. Dželila, M and Kevrić, J.(2020) Phishing Website Detection Using Machine Learning Classifiers Optimized by Feature Selection, Traitement du Signal, 37(4), 34-45

[7]. 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.

[8]. 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.

[9]. Jain, A. K., Parashar, S., Katare, P., & Sharma, I. (2020). Phishskape: A content based approach to escape phishing attacks. *Procedia Computer Science*, *171*, 1102–1109.

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

[11]. Kumar, A., Chatterjee, J. M., & Díaz, V. G. (2020). A novel hybrid approach of svm combined with nlp and probabilistic neural network for email phishing. International Journal of Electrical and Computer Engineering, 10(1), 486.

[12]. 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

[13]. 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

[14]. Liu, T., Zheng, H., Zheng, P., Bao, J., Wang, J., Liu, X. and Yang, C.(2023) An expert knowledge-empowered CNN approach for welding radiographic image recognition, Advanced Engineering Informatics, 56, 101963, ISSN 1474-0346

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

[16]. 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.

[17]. 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.

[18]. Shie, E. W. S. (2020). *Critical analysis of current research aimed at improving detection of phishing attacks*, Selected computing research papers, p. 45.

[19]. Sindhu, S., Patil, S.P., Sreevalsan, A., Rahman, F., Saritha, A.N., (2020). Phishing detection using random forest, SVM and neural network with back propagation. 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.

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

[21]. Yuan, L., Zeng, Z., Lu, Y., Ou, X. and Feng, T.(2020) A character-level bigru-attention for phishing classification. In Information and Communications Security: 21st International Conference, ICICS 2020, springer, 15–17.

[22]. 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

[23]. Zhu, E., Ju, Y., Chen, Z., Liu, F., Fang, X.,(2020). DTOF-ANN: an artificial neural network phishing

[24]. detection model based on decision tree and optimal features. Application of Soft Computing J. 95,. https://doi.org/10.1016/j.asoc.2020.106505 106505.