Phishing URL Attack Detection using Logistic Regression and Convolutional Neural Network

Umejuru Daniel
Department of Computer Science, University of Port Harcourt, Port Harcourt, Nigeria

Eke Bartholomew
Department of Computer Science, University of Port Harcourt, Port Harcourt, Nigeria

Fubara Egbono
Department of Computer Science, University of Port Harcourt, Port Harcourt, Nigeria

ABSTRACT
Phishing is a problem that is quickly spreading worldwide and costs internet users billions of dollars each year. It is illegal to gather sensitive information from internet users using social engineering techniques combined with technology. The overall performance is unreliable, inefficient, and requiring improvement in terms of prediction accuracy, time complexity, misclassification error and robustness. Phishing strategies can be recognized in a variety of communication channels, including email, instant chats, pop-up messages, and web pages itself. Over time, existing methods and approaches have been unable to detect all connected dangers and provide an all-encompassing solution. Although it is widely believed that a successful phishing attack entails developing a website that looks exactly like the target site in order to trick the internet user, this idea has not been incorporated into existing approaches to assess the dangers and thoroughly analyze the gaps. In this study, the aim is to create an enhanced Phishing attack detection system utilizing logistic regression and Convolutional Neural Network (CNN). This study outlined a novel method capable of identifying malicious phishing URLs with an emphasis on using features primarily obtained from the phishing and real URL addresses. The model kernel, weights, and bias values were tuned with a penalty term which increased model detection accuracy. The experimental findings show that CNN performed better incorporating penalty term which recorded a detection accuracy of 98.20% and LR yielded a recommendable prediction accuracy of 89.85%.

Keywords
Logistic regression, CNN,phishing URL attack, feature extraction, penalty term

1. INTRODUCTION
Phishing is a method that uses deception to gain unauthorized access to client-confidential data from users or learning models. Phishing comprises spam mails disguised as genuine with a subject matter or message, meant to trick the victims into disclosing sensitive information. In deceptive phishing, email alerts from credit card companies, security departments, banks, suppliers, online payment processors, or IT managers are utilized to take advantage of the uninformed open. The notification requests that the person receiving it urgently enter or update their personal data [1]. The necessity for data privacy, protection, and prevention against phishing attempts cannot be overstated. Over the years, technological innovation has led to a significant increase in data through social networks, IoT devices, and online transactions. Phishing is one of the oldest techniques still in use today, despite the fact that cybercriminals are constantly coming up with novel methods to gain entry to networks, applications and data without authorization. Learning algorithms are susceptible to these phishing attacks, and hackers employ them to trick them in order to compromise ML detection power [2]. Phishing is a type of cybercrime that is expanding rapidly, and when people respond to messages or submit sensitive information to hackers, their data is put at risk.

There are different types of phishing attacks, including deceptive phishing, spear phishing, whaling, and pharming [3]. The four different phishing attack types that [4] proposed comprise communication channels, target devices, attack strategies, and countermeasures.

(a). deceptive phishing: The most frequent type of phishing attack is deceptive phishing, which impersonates an actual platform or webpage and sends text messages (or emails) to the user that look to be authentic [5]. The malicious links in these text messages (or emails) would instruct the recipient to click on the URL. The phishing website that the attackers have set up will gather all of the user's login credentials and other sensitive information and send it to the attacker. For instance, the lower case "a" in the email address usercreditcard@amazon.com might be eliminated. Consequently, usercreditcard@mazon.com is used to deceive the user and obtain sensitive information.

(b). spear phishing: The spear phishing attack is comparable to the deceptive phishing kind that only targets one user. The scammers aim to trick someone into giving them private information. A tailored message or email is delivered to the user with the intention of misleading them. The email is personalized to include most of the user's details, such as user name, workplace, designation, and so on [6]. The most frequently used platform for spear phishing is the social media site like LinkedIn where it is simple for them to find out a user's occupation.

(c). whaling: The whaling attack type happens when phishers seek people in positions of power, such as the CEO. Prior to the attack, the perpetrator would spend a great amount of time analyzing the target. The attacker sends an email message to target in order to manipulate them into providing confidential information. Whaling is considered as a very dangerous attack since the people in executive bands have access to the organization's most confidential information. The intruder sends these individuals an email message to trick the recipient into divulging private information. Whaling is regarded as a very hazardous attack because executive bands have the ability to access the most sensitive information about the company,

(d). Pharming: Pharming is a subset of phishing that does not require a specific person to be the target. Without needing to be specifically targeted, the attacker can harm a huge number of users.

There are two techniques to carryout pharming attack: (a). It entails emailing the target codes that change every local host file on the system. The host files would change the URLs into number strings that the system would utilize to access websites. Even though the target user enters a legitimate URL, this may link them to a malicious website. (b). Another pharming attack technique is called DNS poisoning, which modifies the website's domain name system tables but leaves the local host files unharmed. This causes a target to be unknowingly diverted to inappropriate web pages. The user would think they are visiting a reliable website, but due to poisoning of the DNS, they would actually be visiting a hostile domain [7]. Deep neural network training is computationally expensive and difficult to optimize because of the large parameter size. Existing solutions fail to detect risks when hackers insert characters into URL addresses that security filters misinterpret as a remark but browsers interpret as a genuine or legitimate web domain. Thus, the malicious URL addresses successfully overcame security; but, when victims clicked on the malicious link inside, they got directed to a phony landing page.

The following is how the paper is set up: The paper is organized as follows: Section 1 provides an introduction; Section 2 offers a brief evaluation of prior approaches related to the topic and the gap in studying the proposed model; Section 3 introduces the model's materials and methods; Section 4 covers the results and in-depth discussion of the results; and Section 5 provides the paper's conclusion.

2. RELATED WORKS

Several techniques and strategies have been investigated to comprehend phishing attacks and offer a defense against them. [8] compared seven distinct machine learning techniques in order to identify phishing sites. The SVM had the lowest detection accuracy, whereas the RF performed the best. [9] 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 [10] 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. [11] presented two innovative machine learning-based decision-making or detection approaches and used the best appropriate features for identifying phishing websites. [12] used ANN in combination with binary classification. The ANN had a detection accuracy of 95.69%. They used 4,000 data points in their experiment, which limited the model's predictions and detection accuracy. [13] used several data mining approaches to detect phishing websites. Their testing revealed an accuracy rate of 91.44 percent and a false alert rate of around 3.40 percent. The majority of the phishing websites were incorrectly labeled as real. [14] employed RF to detect phishing websites by combining different decision trees as an ensemble learner. The results showed that HEFS could identify phishing features up to 94.6 percent of the time, but the model did not generalize well to the testing set. [15] used six different machine learning algorithms, including RF, CART, LR, SVM, and ANN, to categorize phishing emails. Over 92% of the phishing emails were properly predicted by the classifiers under consideration. The model had performance and produced type I and type II errors. [16] demonstrated various innovative capabilities and evaluated the technique using commonly available machine learning algorithms. The recommended model did incredibly well because it had the greatest and most relevant capabilities for detecting phishing websites. The trained algorithm was unable to learn and extract features based on content or confidential information. [17] used multiple ML classification models to group keyword-based features from message contents. The model achieved 98% classification accuracy. Outliers and noise had an impact on the model performance. [18] combined fuzzy logic with a hashing technique in a white-listing strategy. The model achieved an impressive accuracy rate of nearly 96%. Because of human contact, modern phishing attacks lacked universality. [19] proposed the Automated Individual Whitelist (AIW) technique for monitoring user visits to well-known, harmless websites. This strategy is highly effective at preventing dynamic phishing and pharming assaults. The researchers themselves admitted that their procedure was not ideal. It eliminates the requirement for a predetermined login password during data exchanges between the client and server. [20] proposed two procedures, one for registration and the other for login, with four parties involved. Phishing websites and XSS attacks hosted on hijacked domains are undetectable. [21] designed a game-based study to explore traits, phishing vulnerabilities, and the effectiveness of several anti-phishing instructional tools. According to their findings, educational resources reduced users' inclination to supply sensitive information on phishing websites by 40%.

3. METHODOLOGY

The methodology describes the tools and procedures used to detect Phishing URL addresses. This section discusses the possibility of the proposed strategy for detecting phishing URLs, as well as the CNN classifier's evaluation findings on the test set. The following components are discussed and analyzed in order to implement the recommended strategy.

3.1 Data Collection

This is the most important in the development of ML because algorithms are developed to learn the logic from data in order to generalize well on feature set. We used 2015-2023 phishing dataset downloaded from the kaggle site "https://www.kaggle.com/datasets/shashwatwork/phishing-dataset-for-machine-learning". The phishing text data was converted into png image format using the word_cloud.to_file() function and this involved preprocessing and feature extraction methods.

3.1.1 Preprocessing

This component also helps to improve the quality of the data and provides an instance of how to analyze datasets gathered for the detection of phishing activities. The preprocessing transforms input into a comprehensible format, enabling the model to function successfully during training and testing cycles. The pre-processing stage comprises of training data split and feature extraction.

3.1.2 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.

3.2 Machine Learning Module

A convolutional neural network (CNN) is an example of machine learning, specifically a deep learning technique used largely for analyzing images and textprocessing/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.

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 that 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.

3.2.1 Convolutional Layer

This 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. The CNN convolutional layer passes result to the next layer once applying the convolution operation in the input. The first layer is designed to obtaining features from the phishing dataset comprises the convolutional layer, which undertakes mathematical operations of convolution between the input data and an appropriate size filter. We employed a pair of distinct convolutional layers, spooling layers, maximum pooling at the spooling layer, L2 regularization with multiple weight functions, ReLU and sigmoid activation functions. The RELU and sigmoid activation functions are used to deliver superior converging performance, resulting in shorter running times in order to solve the decreasing gradient issue in CNN reverse propagation, When the convolution operation has been applied to the input, the CNN convolutional layer transmits the results to the following layer.

3.2.2 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. This reduces the interaction among layers and allows for distinct tasks on every component map.

3.2.3 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.

3.3 Building CNN

3.3.1 Model Definition

We constructed a sequential model and added the appropriate number of input features one at a time using the TensorFlow library. The training data size of 32 input units was set for the input_dim argument using the RELU activation function. We constructed a 1-dimensional convolution with a batch normalizing layer for the second layer, a kernel, bais and activity regularization, and a ReLU function with 32 units in the first layer using a dropout layer in order to penalize the model for learning from lesser network weights.

3.3.2 Model Compilation

We used the stochastic gradient descendant algorithm (Adam's) optimizer to search through various weights as an additional property before training our model, and we utilized the loss function to evaluate the set of weights. The learning rate was set to 0.00001 beforehand. The binary cross entropy was used as the bias and loss for binary classification defined in keras.

3.3.3 Fit Model

We trained or fitted our model using training data by invoking the fit() function. The training lasted for 200 epoch and each iteration is split into batches. The Epoch give passes to training dataset rows while batch is the sample considered within each iteration before the weights are updated. The process allowed to run for a number of epoch with a batch size and the network configuration was chosen through trial and error. The model trained to learn mapping rows of input to output classification. The CNN then trained model to understand correlation and learn dependencies between the independent and target variables from training dataset.

3.3.4 Evaluate

The CNN was evaluated using testing dataset with evaluate() function to pass same input and output used to train model. The function returns loss, validation loss, validation accuracy and accuracy computed using: accuracy=model.evalute(X,y) in Python where “X” is input and “y” as target.

3.3.5 Make Predictions

The predictions fall between probability of 0 for legitimate set and 1 for phishing set since we are using the sigmoid activation function and converted to a crisp binary prediction for classification task. The model gives higher preference to those with smaller weights than larger weight because a penalty term was added to restrict model.

The detection system is a module that keeps track of phishing attacks or sites containing malicious content. This is the predicted target values after training stage using testing (unseen) data by the CNN model for performance evaluation. It flags-up report of Phishing for malicious content and trusted for non-phishing sites.

The activation Relu and Sgmoid functions are used to initiate nonlinearity into the network with w representing weights between neurons and b is the bias term of the network setting. The Sigmoid and Relu computes the output values as an activation function shown in equation 1 and 2 and h depicts CNN hidden layers.

$$h_t = f_w(h_{t-1}, x_t) \quad (1)$$

$$h_t = ReLu(Whx\ \ x_t + W_{hh} h_{t-1} + b_h) \quad (2)$$

$$Y_t = Sigmoid(W_{yh} h_t + b_y) \quad (3)$$

Where b is the bias, y is the predicted, and t is computation time.

3.4 Implementation of Our Approach

3.4.1 Kernel regularizer argument

This is used in CNN Dense or Con1D layers invoked from the keras.layers model. This will use the regularizer object (l1, l2 or l1_l2) from the keras.regularizers module, where l is the penalty term and is set to 0.01 times the CNN weight's square norm.The kernel, bias weights and activities are all being regularized. The purpose of the activity regularizer is to give the user control over the output of the layers by reducing or bringing them closer to zero, which improves accuracy and speed up convergence.

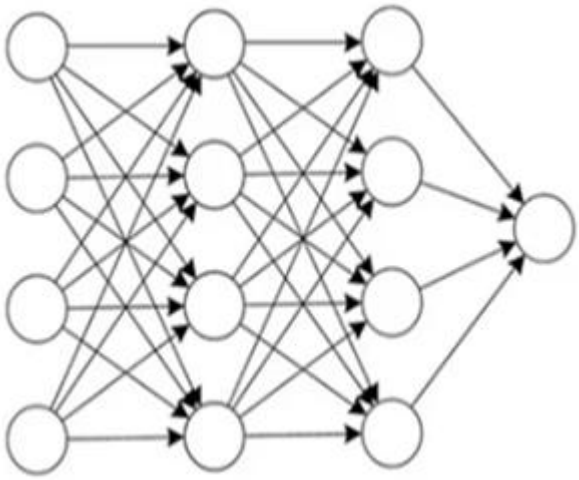

Figure 3.1: Standard CNN

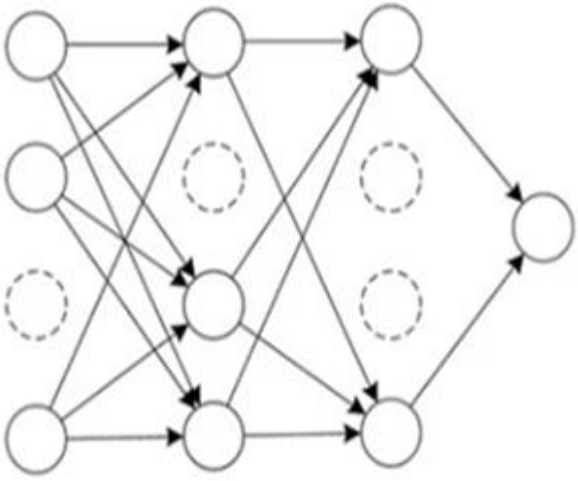

Figure 3.2:Network after weight regularization with dropout

3.4.2 Add penalty to activated weight

We are introducing 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. The above concept was built into CNN's input and output layers in all cases.

3.4.3 Add 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.

3.4.4 Add 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.

4. RESULTS

The suggested model's findings are presented and explained using the appropriate CNN classification technique. The theory and its implementation are being improved to give more accurate and reliable outcomes. We used well-known AI/ML tools like wordcloud, ROC curve and tables to display and explain experiment results.

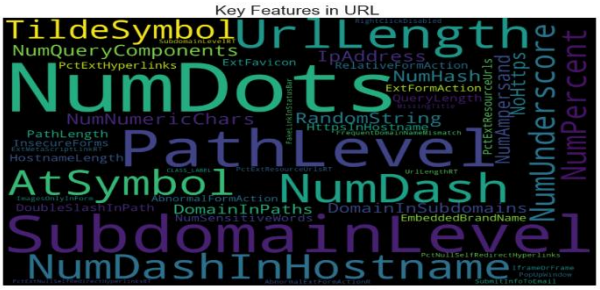

Figure 4.1: Key features in URL address

Figure 4.1 shows a word cloud representation extracting of URL patterns, and major key features from the URL address data set. This method of visualizing data facilitates the communication of complicated datasets to a larger audience and promotes data-driven decision-making. The dataset shows that the most commonly recurring features include larger fonts such as path_level, atsymbol, subdomain level, and so on.

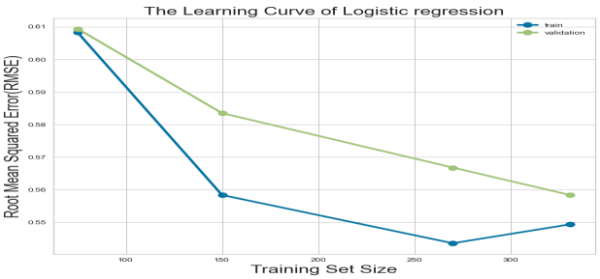

Figure 4.2: Learning Curve of LR

Figure 4.2 shows the LR model that learns from more training data and can further reduce test error. As we add additional training samples, the inaccuracy in the testing and validation curves decreases. The training and cross validation scores are very high at the beginning and decreases gradually as we increased training samples. There is still substantial improvement in the validation process of the LR model.

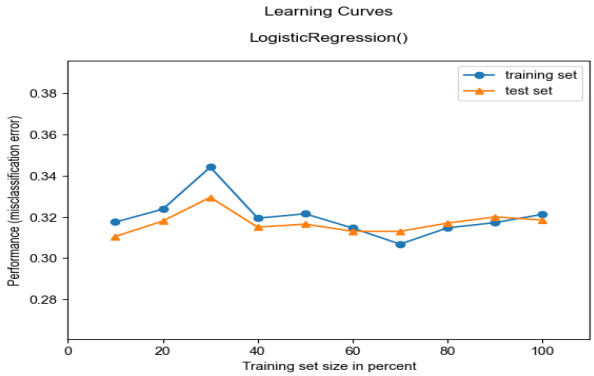

Figure 4.3: The performance plot of LR

Figure 4.3 is the LR performance plot misclassification against training instances. The training and testing scores fluctuate between 0.31 and 0.35 when more samples from the training set are added, yet our LR model reported a significant training error with a high bias issue.

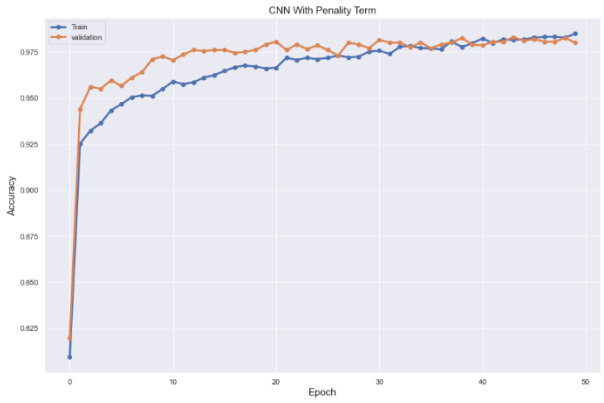

Figure 4.4: Training accuracy of CNN with penalty term

The behavior of training and validation sets for the various training cycles are shown in Figure 4.4. The validation accuracy is 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 training loss from 20 to 50 intervals, showing that the model did well when generalizing well 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 loss 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.

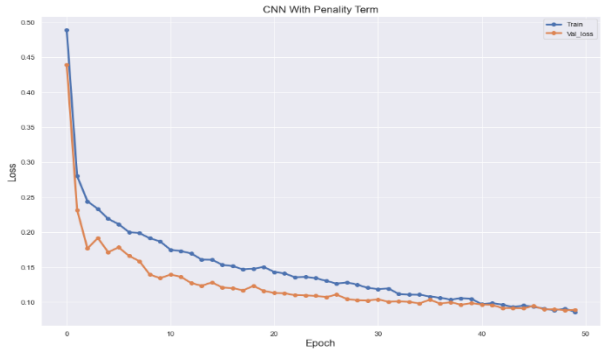

Figure 4.5: validation loss of CNN with penalty term

Figure 4.5 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 phishing features from the validation and training sets throughout longer training periods than shorter training time.

Table 4.1. Confusion matrix

Models	TP	TN	FN	FP
LR	918	959	53	70
CNN	1935	55	13	20

Table4.1 shows the confusion matrix of LR, 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.

Table 4.2. 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
Weighed avg	0.51	0.51	0.36	2000

The LR classification report for trusted and phishing URL addresses are shown in Table 4.2. It includes the precession, recall, and f1-score accuracy of the existing system model. For trusted URL sites, the precision accuracy(0.51), recall(0.49) and f1-score produced 0.50. Phishing URL addresses resulted in a precision rate of 051, recall (0.52), and a f1-score score of 0.51, for an accuracy of 0.51 metrics. The model is actually biased toward predicting trusted ULR addresses.

Table 4.3. Classification report of CNN with penalty term

	Precision	Recall	F1-Score	Support
TRUSTED	0.50	0.95	0.66	1000
PHISHING	0.55	0.06	0.10	1000
Accuracy			0.51	2000
Macro avg	0.53	0.51	0.38	2000
Weighed avg	0.53	0.51	0.38	2000

The classification report of CNN with penalty term is presented in Table 4.3, with precession, recall, and f1-score classification for phishing and trusted sites. The recall ranked highest with 0.95 metrics, f1-score for trusted set to 0.66, and the precision score(0.50) trusted cases. For phishing sites; precision recorded 0.55, followed by f1-score(0.10) and recall(0.06). There is a significant improvement, as shown in the precision, recall, and f1-score values from the classification report. The macro-average shows how all categories equally contributed to the final averaged metrics, the weighted-average shows how each class appears to contribute to the average as weighted by its size, and the micro-average clearly demonstrates how all samples equitably make a contribution to the final averaged metrics.

Table 4.4. Training and testing time

Model	Testing time(s)	Training time(s)
LR	0.00099	0.07879
CNN	1	9

Table 4.4 shows the training and testing time complexities of LR and CNN for detecting phishing URL addresses. The LR required 0.00099 and 0.07879 seconds to train and test, while CNN lasted 1 second and 9 seconds, respectively, having the longest training time.

5. CONCLUSION

The proposed CNN achieved better detection accurate than logistic regression (LR) while LR had lesser training testing time complexity when it comes to detecting phishing URL addresses. The proposed system addressed most of the drawbacks, including low URL address detection accuracy and high error rates. Based on the analysis above, we draw the conclusion that the system technique was promising in terms of identifying phishing. The CNN model converged faster than the LR model, which will be useful to government organizations, programmers, and machine learning experts who desire a system capable of accurately identifying real and false URL addresses. The weight penalty term applied to CNN weights, bias, and kernels (filters) improved ML model accuracy. Diagnostic tools such as ROC plots, confusion matrix, precision, accuracy, and f1-scores make it easier to visualize model performance. It demonstrates how the TP and FP classes trade off in respect to one another, and putting the FP rate on the X-axis and the TP rate on the Y-axis forms the two-dimensional ROC graph.

6. REFERENCES

[1] Iwendi, C. Jalil, Z. Javed, A. R. Reddy, T. Kaluri, R., and Srivastava, G. J. O. (2020) Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks. IEEE Access, 8, 72650–72660. doi: 10.1109/ACCESS.2020.2988160

[2] 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.

[3] 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.

[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. Telecommun. Syst. 76(1), 139–154. https://doi.org/10.1007/s11235-020-00733-2.

[5] 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.

[6] 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.

[7] 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.

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

[10] 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.

[11] 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.

[12] 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.

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

[15] Hong J., Kim T., Liu J., Park N., Kim S. W. (2021) Phishing URL Detection with Lexical Features and Blacklisted Domains, Autonomous Secure Cyber Systems. Springer, 1-12

[16] 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

[17] Basnet, R. B., Mukkamala, S. and Sung, A. H.(2021). Detection of Phishing Attacks: A Machine Learning Approach. In Soft Computing Applications in Industry: Studies in Fuzziness and Soft Computing, 226(10), 373–383.

[18] 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.

[19] 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.

[20] 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.

[21] Sheng, S. Holbrook, M. Kumaraguru, P. and Downs, L. F. J. (2020) Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions, in Proceedings of the 28th International Conference on Human Factors in Computing Systems, USA, 2-6.