JAIMT
JOURNAL OF ARTIFICIAL INTELLIGENCE AND MODERN TECHNOLOGY
A Peer Reviewed (Refereed) International Journal
ISSN:1595-6261 VOLUME 6, ISSUE 1 (JANUARY) 2026

TEXT TO IMAGE CONVERSION FOR PHISHING ATTACK CLASSIFICATION USING CNN AND URL KEY FEATURES

UMEJURU DANIEL
Department of Computer Science,
University of Port Harcourt, Choba, Nigeria
E-mail: daniel_umejuru@uniport.edu.ng
https://orcid.org/0009-0007-9843-2248
D.O.I: 10.5281/zenodo.18270371

ARTICLE INFORMATION

Received: 2nd December, 2025
Accepted: 26th December, 2025
Published: 15th January, 2026

KEYWORDS: URL, Classification, Address, Extension, CNN, Image, Features

JOURNAL URL:
https://ijois.com/index.php/jobpef

PUBLISHER: Empirical Studies and Communication (A Research Center)
Website: www.cescd.com.ng

ABSTRACT

Phishing attacks remain an evolving challenge to web user's world all over. These attacks which emanates from phishing URLs are many and very problematic to internet users globally. The curiosity by well-meaning security professionals has led to further research, and thus propels the development of newer models to solve the lingering challenge in the Cyber domain. The updated classification models are been developed in other to curb the gaps of experiencing phishing attacks. This study aims at strategically using text to image conversion for phishing attack classification by using the URLs key features which are address and extension. CNN was also used as deep learning algorithm in other to increase models detection accuracy, reduce time complexity and also address misclassification issues as well as poor prediction accuracy. This was done in other to increase the resilience of the suggested model as well as enhancing classification prediction. The performance evaluation metrics employed for the proposed model are accuracy, precision, recall, F1 score, confusion matrix and AUC-ROC. This study outlined a novel method capable of identifying phishing attacks using features primarily obtained from the phishing and real URL addresses. A temporal tokenizer was also 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.

1.0 Introduction

Phishing attack is a criminal activity which 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.

2.0 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 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.

2.1.1 Convolutional Neural Network (CNN) as a tool for Phishing 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.

3.0 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 Phishing 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.

(a). 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. 2015-2023 phishing dataset for ML task was used. The dataset was downloaded from the kaggle site "https://www.kaggle.com/datasets/shashwatwork/phishing-dataset-for-machine-learning" and experiment contained 10,000 items. It comprises of 5000 phishing and 5000 legitimate attacks with 48 features (id, NumDots, subdomainlevel, Pathlevel and etc). 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.

(i). 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. A batch normalization technique was used to scale all of the values in the proposed dataset to lie between 0 and 1 in the target set using a standard scaler. Scaler.fit_transform() was used for both training and testing datasets in this process. We defined a text variable and passed the word cloud text as a parameter to the generate() method, which joined all the features in the phishing dataset to transform text data into an image set. A tokenizer was created to pass the URL strings to list of characters. The pre-processing stage comprises of training data split and feature extraction.

Phishing and legitimate data: Each website in the data set includes HTML code, user info, URL, and all files included in the web page. A Phishing and legitimate sample comprises of 10,000 sites, of which 5,000 constitute phishing attempts and the remaining 5,000 are legitimate. Preprocessing is the process of transforming input data so that the model can understand it. Key words were extracted by translating URL data into visual representation using a wordcloud class library. The following code was executed to translate URL data and store the result as an image in the word_load2 variable for CNN training and visualization. The URL's key words were extracted from it.

```
# Key word with larger fonts has the highest number of frequencies(Keywords)
plt.figure(figsize=(15, 10))
phishing_url =" ".join(i for i in df22.URLS)
word_load2=WordCloud(collocations = False, background_color = 'black',
                     width=1600, height=1000, colormap='Paired').generate(phishing_url)
# saving the image
word_load2.to_file('got2.png')
# Display the generated Word Cloud
plt.imshow(word_load2, interpolation='bilinear')
plt.axis("off")
# plt.tight_layout(pad=0)
plt.title('Key Features in URL extension ', fontsize=25)
plt.show()
```

Wordcloud Code for Text Conversion into Image Format

(ii). Training and Testing data split: Three primary subsets were created from data split: the training set, which is employed to train the algorithm; the validation set, which is used to monitor the parameter settings and prevent over-fitting; and the testing set, which is used to

assess the effectiveness of the model on newly collected data. This is taking an 80% random sample of the rows (you can change this) and adding them to our set of training data without replacing them. We also included the balance of twenty percent (20%) of the sample in our test set. Numerical data were converted into image sets for the purpose of training CNN, which performs best with image datasets.

(iii). 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.

(b). 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 text processing/classification. The CNN is an appropriate technique to analyze a stream of data with high accuracy.

CNN model was designed 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 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.

(i). 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. The CNN convolutional layer passes result to the next layer once applying the convolution operation in the input. The first layer is designed to obtain 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.

(ii). Pooling layer: The pooling layer takes the convoluted map and further reduces the spatial diemnsion(height and width). This is done 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.

(iii). 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.

Table 3.1: Data Login Table

Login_Table

Field_Name	Data_Type	Type description
User_Name	Varchar(20)	The login username
Password	Varchar(20)	The login password of user

Table 3.2: Data Tables for URL Address

Field_Name	Data_Type	Width	Type description
URL_ID	Integer	5	For URL id
URL_Lemgth	Integer	5	The length of URL address
NumDots	Integer	5	Total number of dot(.) in the URL
NumDash	Integer	5	Number of slashes/dash in URL

4.0 RESULT AND DISCUSSIONS

The obtained results and holistic discussions of the proposed model are displayed and discussed using suitable visualization tools. The concept and its implementation are being refined to produce more accurate and reliable results. AI/ML performance and evaluation tools were used to present and explain experiment results, such as confusion matrix, ROC curves, charts and tables.

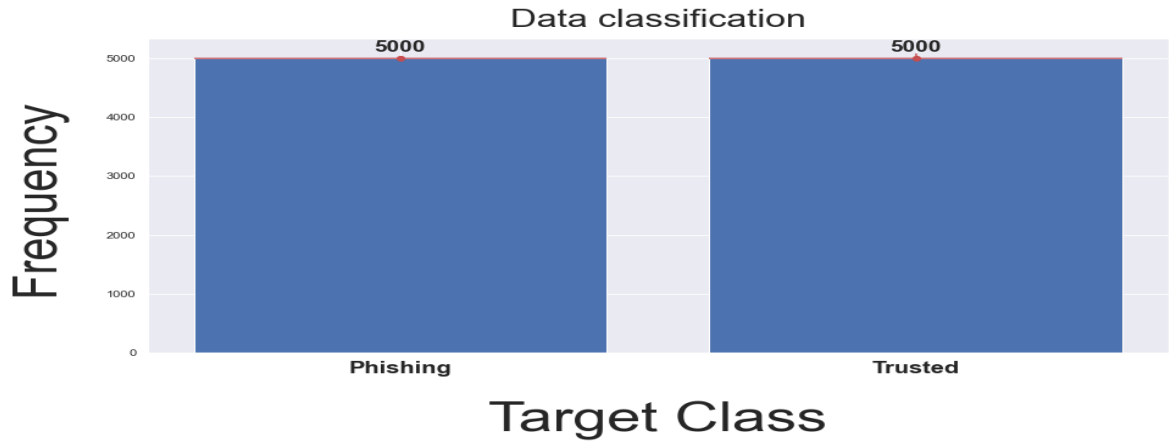

Figure 4.1: Target Classification

Fig 4.1 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.

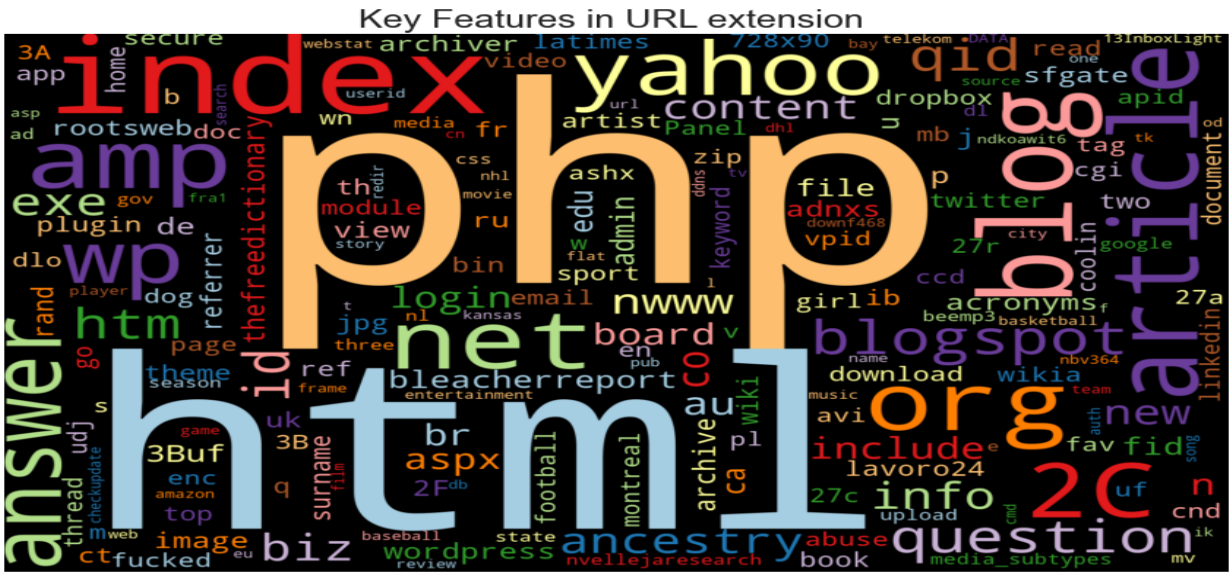

Figure 4.2: Key Features in URL Extension

Figure 4.2 depicts a wordcloud 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.

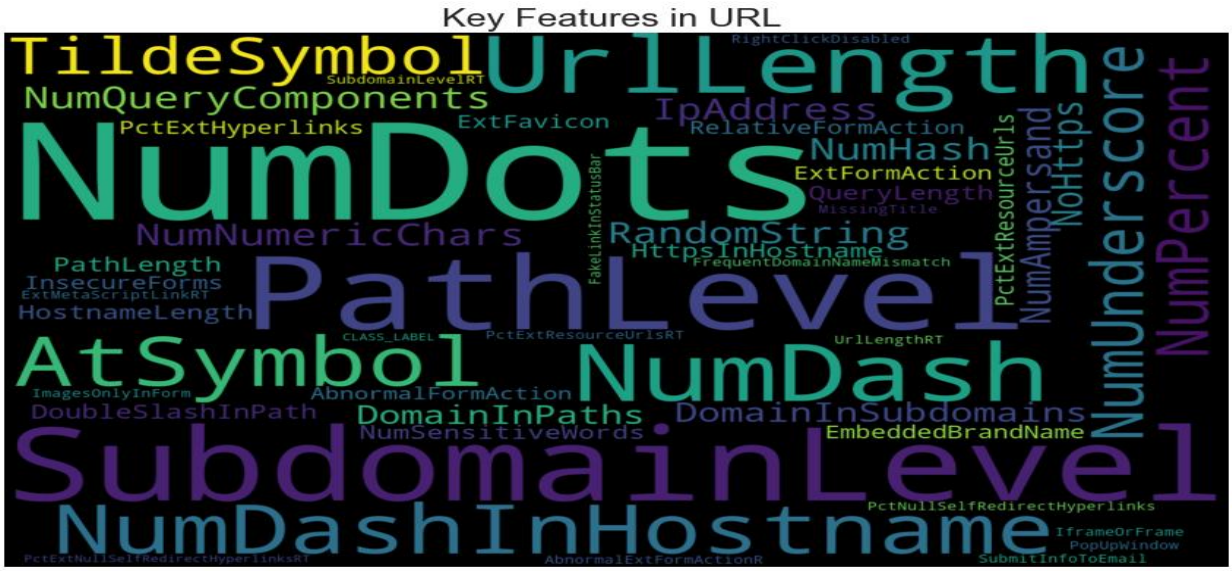

Figure 4.3: Key Features in URL Address

Figure 4.3 shows a wordcloud 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 data sets 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, subdomainlevel, and so on.

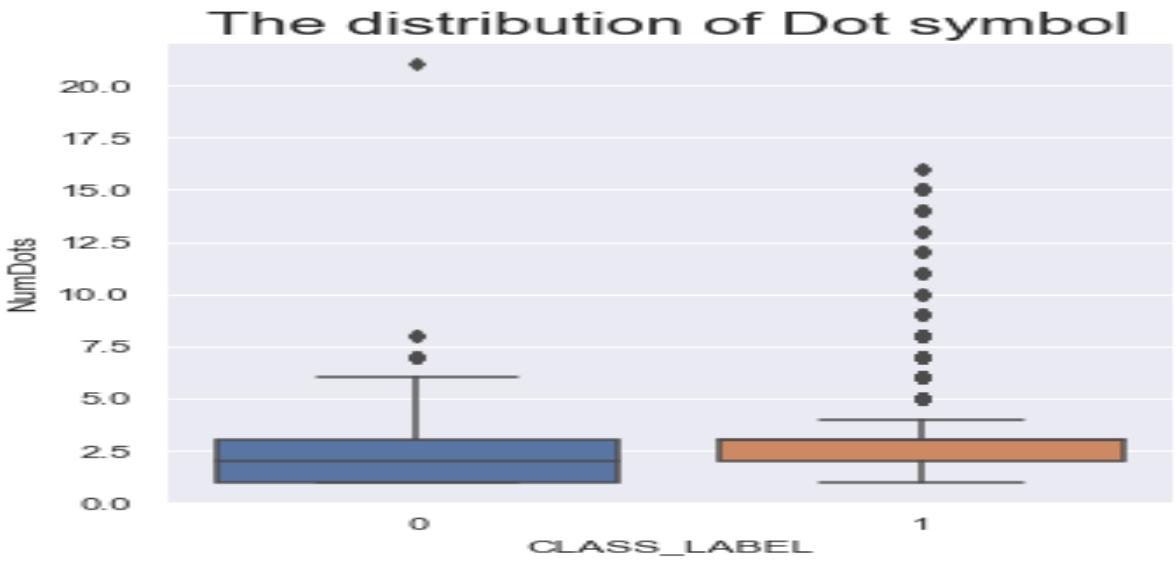

Figure 4.4: Distribution of Dot in URL Address

Figure 4.4 displays the distribution of the Dot (.) symbol, which is more common and widely distributed in phishing sites than in non-phishing sites. The dot symbols appear more frequently in phishing sites (1) than trusted URL addresses (1). The phishing bar plot starts at 2.6 and ends a few steps away from 17.5, with multiple dots occurring within the range of data, as opposed to non-phishing, which had only three dots in a scattered order.

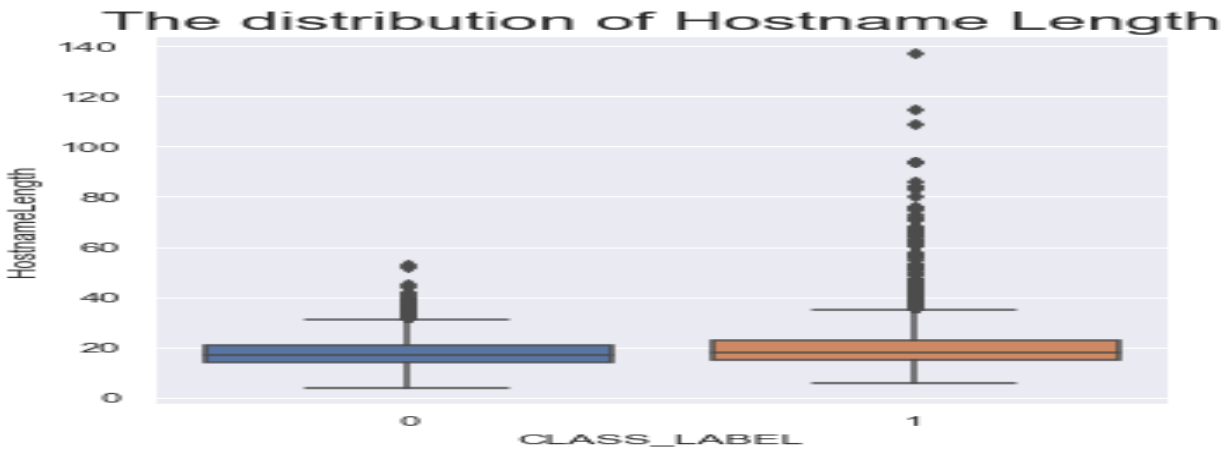

Figure 4.5: Distribution of Hostname length in URL Address

\Figure 4.5 shows the hostname length of URL address for trusted and phishing sites of the proposed system dataset. In the dataset, phishing sites exhibited a longer spread of hostnames represented by "1"as compared to trusted sites represented by "0".The trusted measured less than 60, while the phishing recorded several unusual patterns above 60 but below 120.

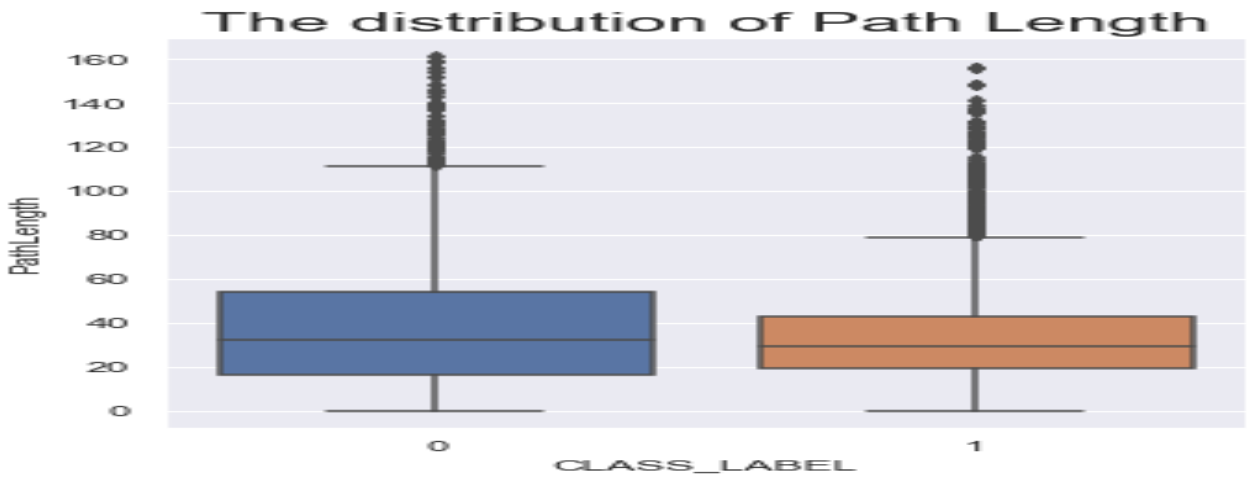

Figure 4.6: Distribution of Path length in URL Address

Figure 4.6 is the chart showing the distribution URL path length.The phishing sites denoted by "1" frequently exhibited a greater path length distribution or occurrence than trusted sites. Phishing sites had longer path lengths ranging from 80 to 160, whereas trusted sites range from 110 to 160 on the path length axis.

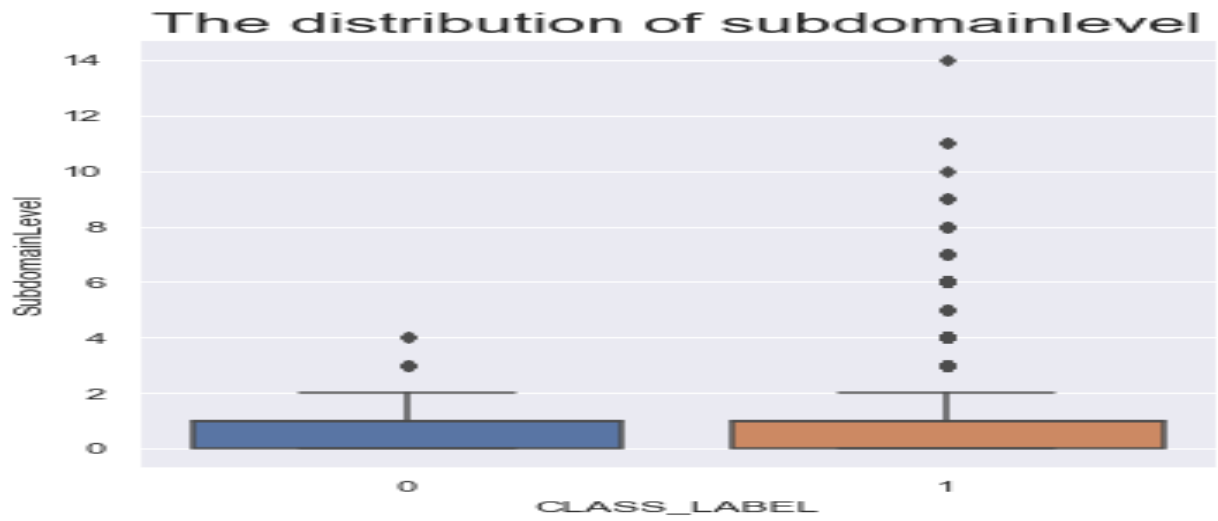

Figure 4.7: Distribution of SubdomainLevel

Figure 4.7 depicts the subdomain levels for both phishing and non-phishing URL addresses. According to the figure, phishing patterns are most commonly detected in the subdomain level of URL addresses on phishing sites as opposed to trusted sites. In the target class, the label "0" symbolizes trusted, whereas "1" denotes a phishing URL.This illustrates that attackers are growing more skilled at correlating phishing trends to the subdomain level of URL addresses, as determined by the proposed system dataset. According to the graphic, the majority of phishing URL address lengths at the subdomain level exceed four characters when compared to reputable sites.

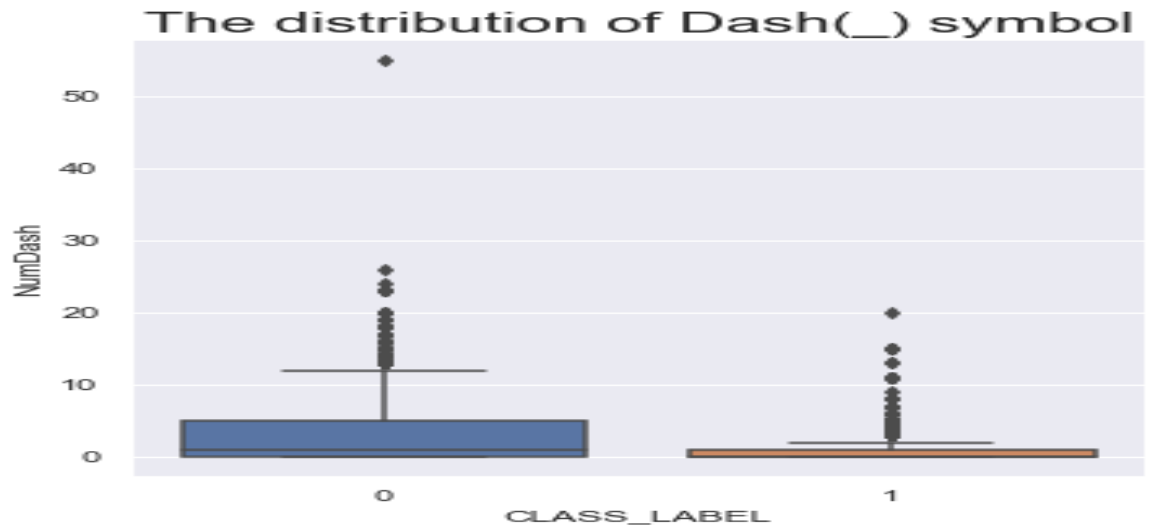

Figure 4.8: Distribution of Dash Symbol

Figure 4.8 depicts the distribution of dash sysmbol across trusted and phishing URL address. According to the graphic, the dash symbol is most typically encountered in phishing URL addresses, meaning that strategies such as appending the dash symbol to URLs to fool the user into believing it is a trusted site which are commonly used by hacker to trick users into disclosing private information.

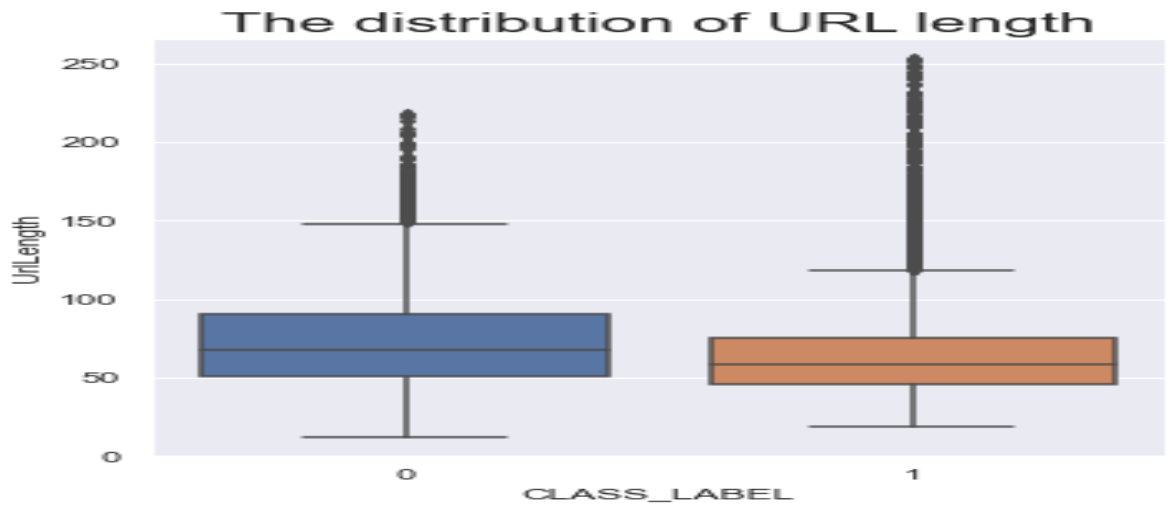

Figure 4.9: Distribution of URL length

Figure 4.9 depict the distribution of phishing and trusted sites in the target class for model training. The diagram shows that trusted URLs were shorter in length than phishing URL addresses. It depicts a distribution that is nearly equal of phishing and non-phishing URL length. Majority of URLs with shorter character lengths were rated to be trustworthy, but the majority of phishing URLs contained phishing content.

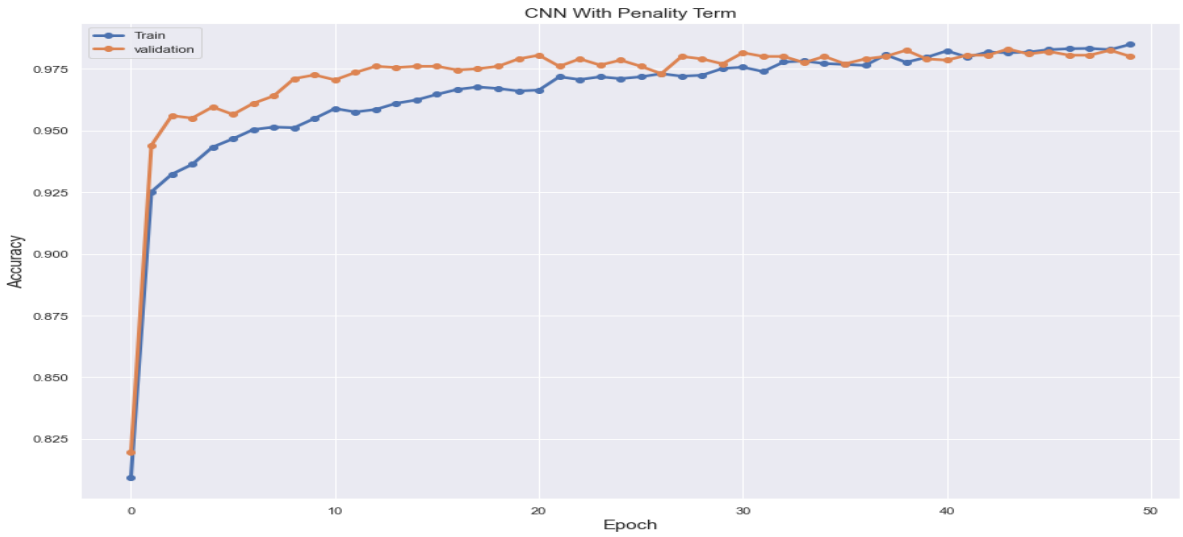

Figure 4.10: Training accuracy of CNN

Fig 4.10 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.

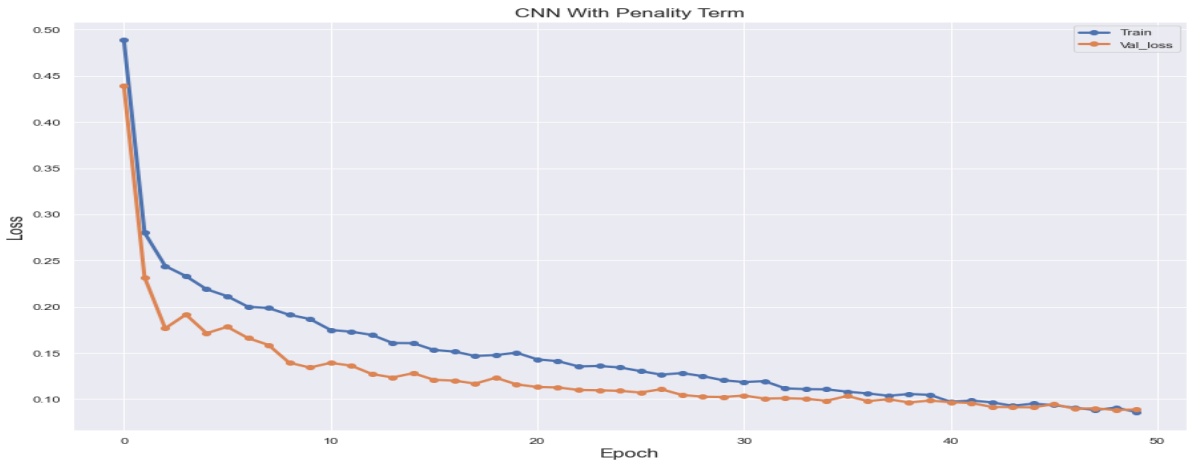

Figure 4.11: Validation loss of CNN

Fig 4.11 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.

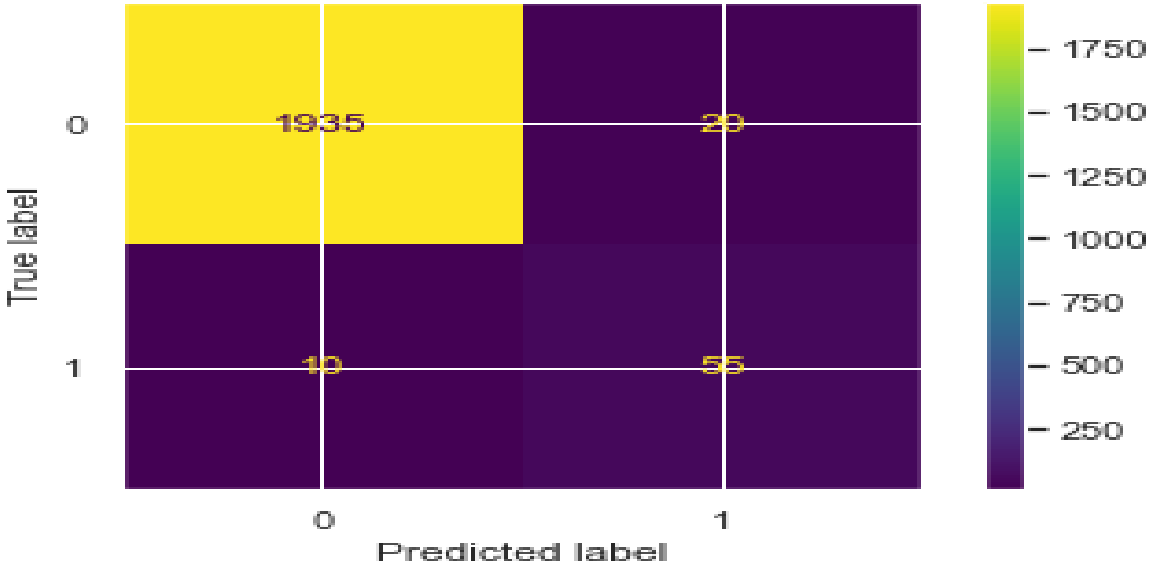

Figure 4.12: CNN Confusion Matrix

A confusion matrix shown in Figure 4.12 is used to evaluate the performance of CNN classification system 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. The overall number

of correctly predicted values recorded to be TP+TN = 1935+55, or 1990 instances with phishing URL addresses; while incorrectly predicted cases yielded FT+FN = 10+20 or 30 cases.

Table 4.1 Classification report of CNN

	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

The classification report of CNN for phishing and trusted sites are displayed in Table 4.2 along with precession, recall, and f1-score correctness. For the trusted sites, the precision accuracy score recorded 0.50, recall (0.97) and f1-score yielded 0.66. For Phishing sites; precision accuracy (0.49), recall (0.03) and f1-score gave 0.03. Detecting phishing sites has very low accuracy across multiple criteria and the accuracy was 0.85, while macro and weighted averages produced an accuracy of 0.50.

5.0 CONCLUSION

The proposed text to image classification used word cloud and CNN to achieve reliable and accurate results making it dependable for classifying phishing attacks by efficiently converting text into image format. The major issue resolved by the new system includes time complexity and detecting phishing codes found in URL backgrounds. The aforementioned analysis led us to the conclusion that the system identified phishing attacks more accurately. This is a standalone ML research serving as the supplementary output that enables future replications and/or modifications of the conducted experiment. Performance evaluation metric 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

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.

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.

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.

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

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

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.

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.

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.

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

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

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.

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.

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

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.

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

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. Application of Soft Computing J. 95,. https://doi.org/10.1016/j.asoc.2020.106505 106505.