PhishGuard: A Convolutional Neural Network-Based Model for Detecting Phishing URLs with Explainability Analysis

Preprint · April 2024

DOI: 10.13140/RG.2.2.36677.26081

6 authors, including:

Md. Robiul Islam
Uttara University
4 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Md Mahamodul Islam
University of Oslo
6 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Mst Suraiya Afrin
Khulna University of Engineering and Technology
2 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Nujhat Tabassum
North South University
11 PUBLICATIONS **16** CITATIONS
SEE PROFILE

PhishGuard: A Convolutional Neural Network-Based Model for Detecting Phishing URLs with Explainability Analysis

Md Robiul Islam
Dept. of Computer Science and Engineering
Uttara University
Dhaka, Bangladesh
robiul.cse.uu@gmail.com

Md Mahamodul Islam
Dept. of Informatics
University of Oslo
Oslo, Norway
mdmahamodul1998@gmail.com

Mst. Suraiya Afrin
Dept. of Building Engineering and Construction Management
Khulna University of Engineering and Technology (KUET)
Khulna, Bangladesh
abontyefu@gmail.com

Anika Antara
Dept. of Electrical and Electronics Engineering
BRAC University
Dhaka, Bangladesh
anikaantara1000@gmail.com

Nujhat Tabassum
Dept. of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
nujhat1tabassum2@gmail.com

Al Amin
Dept. of Computer Science
American International University Bangladesh (AIUB)
Dhaka, Bangladesh
alaminbhuyan321@gmail.com

***Abstract*—Cybersecurity is one of the global issues because of the extensive dependence on cyber systems of individuals, industries, and organizations. Among the cyber attacks, phishing is increasing tremendously and affecting the global economy. Therefore, this phenomenon highlights the vital need for enhancing user awareness and robust support at both individual and organizational levels. Phishing URL identification is the best way to address the problem. Various machine learning and deep learning methods have been proposed to automate the detection of phishing URLs. However, these approaches often need more convincing accuracy and rely on datasets consisting of limited samples. Furthermore, these black-box intelligent models' decision to detect Suspicious URLs needs proper explanation to understand the features affecting the output. To address the issues, we propose a *1D Convolutional Neural Network (CNN)* and trained the model with extensive features and a substantial amount of data. The proposed model outperforms existing works by attaining an accuracy of 99.85%. Additionally, our explainability analysis highlights certain features that significantly contribute to identifying the phishing URL.**

***Index Terms*—cybersecurity, phishing detection, URL, deep learning, CNN, explainability.**

I. Introduction

The increasing digitization of our daily lives reflects the rapid evolution of global networks and communication tools. Cyberattacks gain in this anonymous playground, leaving everyone vulnerable to data breaches, malware infections, and sophisticated scams. Phishing scams are designed to exploit human vulnerabilities, making it inherently difficult to completely prevent users from falling for them, regardless of their caution or experience [1]. Skilled phishers recognize that personality plays a role in online behaviour and capitalize on that to deceive even the most experienced users who might otherwise be alert [2]. Ordinary online users are the new battleground for cybercrime. These targeted attacks steal precious personal data and money, draining billions from individuals every year [3]. Exploiting these deceitful URLs, phishers aim to harvest sensitive and personal data from victims, including financial details, personal information, usernames, passwords, and more [4]. If users mistakenly access such fraudulent sites, assuming they are legitimate, they may unwittingly provide their sensitive information without suspicion. This is because the deceptive web page appears identical to the original one. In a correlated investigation concerning user encounters with phishing attacks [5], computer users fall for phishing due to a limited understanding of URLs, the lack of knowledge of identifying trustworthy websites, haste or accidental clicks, difficulty in assessing the true nature of a link and the challenge of spotting phishing scams.

In this paper, we focus on creating an intelligent real-time phishing web page detection system. For the establishment of a robust model, we have created a large volume of training data and extracted features. Finally, we have proposed an explainable 1D Convolutional Neural Network (CNN). Our main contributions to the research work are as follows -

- We proposed a fine-tuned 1D Convolutional Neural Network (CNN) based phishing detection model, which outperforms other benchmark models.
- We have done extensive analysis from different perspectives to evaluate the performance of the model. Creating a large dataset and extracting numerous features are the key contributions that positively impact the model's

978-1-5386-5541-2/18/$31.00 ©2024 IEEE

performance.

- To comprehend the internal mechanisms of our proposed model, we have analyzed its explainability. It will help users to be cautious about the substantial features contributing to phishing URLs.

The rest of the paper is organized as follows: Section II refers to related works about phishing detection. In the next Section III, we discussed the dataset creation and pre-processing. Furthermore, in Section IV, we have discussed the methodology of our proposed model. Next, we have discussed the analysis of the result in Section V. In Section VI, we conclude our work.

II. Related Work

Numerous researchers have investigated the outcomes associated with phishing websites. Our method incorporates fundamental principles derived from previous research findings. Previous URL-based phishing detection initiatives significantly influenced our current approach.

Aljofey et al. [6] proposed a method to identify phishing based on a specific publication's URL; the researchers employed multiple algorithms. They analyzed diverse data points using different deep-learning models and layered architectures to compare the outputs. The initial method examines multiple URL characteristics, furthermore, assesses the website's credibility by analyzing its affiliations and administration, and the third evaluates its visual presentation. Yao et al. [7] suggested a revolutionary strategy for identifying phishing websites, relying solely on URL analysis, which has been hailed for its remarkable accuracy and effectiveness. One strategy involves initiating by eliminating superficial URL property. However, in [[8], [7]], the researchers create reliable and precise complex characteristics of URLs, leveraging basic features to assess URL authenticity. Aggregating outcomes from all segments determine the overall effectiveness of this technology. Korkmaz et al. [9] introduced a technique to detect phishing webpages using Hypertext Markup Language (HTML), Tag Distribution Language (HTDL), and URL properties. They developed compact HTDL and URL properties, enabling the creation of HTDL string-embedding functions without external dependencies, facilitating integration into a legitimate recognition application. The effectiveness of the method was confirmed through testing on a substantial dataset comprising more than 30,000 HTDL and URL attributes. The authors claim their system reached 98.24% accuracy, with a 3.99% True Positive rate and a 1.74% False Negative rate [9]. To identify a brand-new phishing scheme targeting vulnerabilities. Stokes et al. [10] proposed a exceptional strategy for intelligent phishing detection considering the website url properties. The researchers utilized resource identification principles and sequencing strategies consistent with their framework's usual practices. As per the researcher's findings, the proposed method effectively identifies phishing attempts and zero-day attacks, with a True Positive rate of 95.38%. Earlier research used how website text is organized to make tools for spotting phishing.

Yerima et al. [11] explored the effectiveness of the long short-term memory (LSTM) classifier, contrasting it with a Random Forest (RF) classifier-based approach through a scientific investigation of web addresses involving fourteen criteria. Initially, researchers constructed an LSTM model that treats a hyperlink as a textual sequence, discerning its legitimacy. Despite its feature creation not necessitating specialized expertise, users found that the LSTM algorithm surpasses on average, the accuracy rates of the RF classifier. Even if there's no preparation or arrangement of features, their approach acquire 97.4% accuracy [11]. Their study primarily focused on the textual features of webpages, neglecting to incorporate additional elements like frame characteristics and website images, which could potentially improve the model's effectiveness. Selamat et al. [12] employed two sets of URL data to detect phishing, employing logistic regression alongside CNN and CNN-LSTM models. These datasets were compiled from diverse sources, including malware domains, lists of malware domains, and phishing domains sourced from OpenPhish and PhishTank. The dataset contains over 70,000 URLs for training purposes and over 60,000 URLs for testing. They utilized this dataset to train models for detecting phishing URLs using both CNN-LSTM and CNN. The LSTM method was chosen because it considers the raw web address as input data. In their experiment, the CNN- LSTM architecture surpassed the alternative model, achieving an accuracy rate of approximately 97% in categorizing URLs [13]. On the contrary, the proposed method relies solely on text-based features and could benefit from augmenting with additional attributes and fine-tuning variables to enhance accuracy. Hence, the limitations observed in earlier research studies formed the basis for our introduced IPDS. Janet et al. [14] utilized deep learning techniques, and they aimed to identify phishing activities. Employing classification methods like Support Vector Machine (SVM), ANN, and RF, they found that the RF algorithm demonstrated strong performance. Rishi Kotak [15] identified phishing attacks utilizing different deep learning techniques. Finally, Rabab et al. [16] tested DL models for spotting phishing attempts, and the results showed that random forests performed exceptionally well.

Reviewing the literature on Phishing Detection in Modern Security focusing on URLs exposed various potential research opening. One such gap involves the absence of standardization in identifying phishing URLs, complicating result comparisons for researchers and the implementation of robust phishing spotting systems by organizations. Another gap pertains to limited research on real-world data; most studies rely on synthetic or artificially generated datasets, potentially overlooking the complexity of actual phishing attacks. Furthermore, many studies solely concentrate on the technical aspects of phishing detection, disregarding user interaction with phishing URLs. It's important to understand user behavior and decision-making processes to create successful phishing detection strategies. Lastly, there needs to be more focus on emerging threats; studies tend to target known phishing attack types without taking into account new methods of attack. Our proposed

model aims to bridge these gaps by using a 1D CNN-based approach to precisely classify legitimate websites from phishing ones. Its effectiveness has been assessed using the PhishTank dataset. We have also analyzed the explainability of the model.

III. Dataset creation and pre-processing

A well-crafted dataset, which may be the secondary or primary data, fuels any Machine Learning (ML) model. We must care about the data; it should be representative and free of null, garbage, and missing values. Otherwise, we can't think of having a good ML model. If we input garbage data, it will produce a garbage model [17]. So, collecting or creating a training dataset is our model's first and critical stage. We will vividly discuss the entire process in the following subsections.

A. Dataset collection

Throughout our work, we used **PhishTank** dataset [18]. The dataset consists of URLs of websites that people report as scams. This dataset has information about these sites, like their *web addresses, when they were reported, whether they're still active or not, etc*. There are almost 38,000 instances and 8 features. We can easily get it through an API or as a CSV file format. This data helps researchers make better tools to fight against online scams. It gets updated daily, so using the newest version is important.

The legitimate URLs are obtained from the *open datasets of the University of New Brunswick*. This dataset consists of 35000 authentic URLs [19]. For demonstration purposes, we have shown some sample data in Table I. All of the URLs of the dataset are legitimate web addresses. We used these two datasets to create our training dataset.

TABLE I: Legitimate Dataset Sample Data

Index	URLs
1	https://lastpass.com/signup2.php?ac=1&from_uri...
2	http://persianblog.ir/wEPDwUKMTgwNjcyNjU0MA9kF...
3	https://twitter.com/share?text=%D0%9D%D0%B0+%D...
4	https://asana.com/guide/videos/%22//fast.wisti...

B. Data Preprocessing and Feature extraction

This is the most essential part of our model, and we put a lot of effort into preprocessing the data and extracting the features. From both of the datasets (phishing and non-phishing), we only select the features *'URLs' and 'Labels'*. To extract the features, we used different custom functions to create each function based on specific criteria. Those functions extract the features based on different criteria like *IP Address*,*Abnormal URL*, and so on. We used regular expressions and some string functions to extract the features from the URLs. We set 21 different criteria to have 21 features. Finally, we merge two datasets to create our final training dataset.

IV. Methodology

This section describes our proposed CNN-based phishing URL detection method. The suggested method consists of a few phases that work jointly to accomplish high accuracy in recognizing fraudulent websites. Figure 1 shows the overall approach of our work. The initial step involves dataset preparation and preprocessing, which is clearly described in the previous section III. The last step is to develop a 1D CNN model to detect the phishing URLs. Though CNN is a well-known deep learning-based image analysis model, we have customized the model for URL detection. Lastly, we also outline the attributes of the various layers within the CNN network, which allows us to make sense of the trained model's inner working principles to identify the important features for detecting phishing URLs.

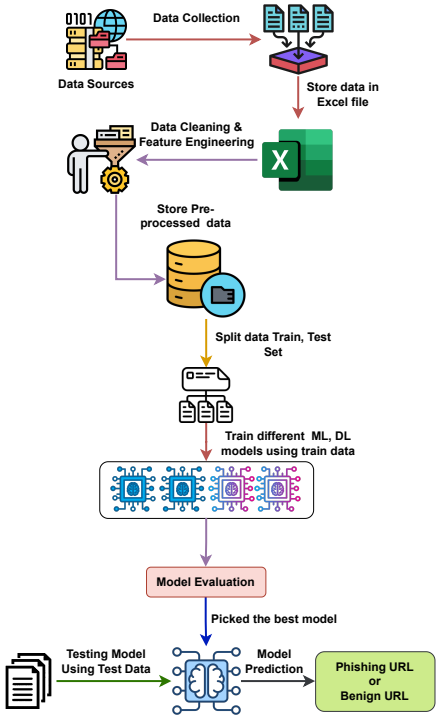


Fig. 1: Proposed Methodology Workflow

A. Proposed CNN Model

After using a couple of machine learning algorithms, we have seen that the deep learning algorithm performs far better than the machine learning algorithm. We have chosen CNN as it produces the best accuracy. We have fine-tuned the CNN model according to the following manner. Table II presents a summary of the proposed Convolutional Neural Network (CNN) model, detailing the constituent layers of the model alongside the corresponding count of parameters associated with each layer.

V. Evaluation and Explainability Analysis

The evaluation matrix, also known as performance metrics, is an essential aspect of machine learning (ML) and deep

learning (DL). It refers to the set of metrics used to measure the effectiveness of an ML and DL model. These metrics help to determine how well a model can make accurate predictions or classifications on unseen data. Several evaluation metrics such as ***accuracy***, ***precision***, ***recall***, and ***F1-score*** used to evaluate our machine and deep learning model [20].
We trained multiple machine learning models and selected the best model based on the performance. For the purpose of execution, we utilized a personal computer for this task. The details of the computer's specifications are summarized in Table III. In the following subsections, we have discussed all of our experimental results sequentially.

A. Accuracy and Loss for Training and Testing Phase

Figures 2a illustrated our proposed CNN model training and testing accuracy per epoch. This graph makes it quite evident that, in comparison to the testing accuracy score, the training accuracy score improved steadily. Testing accuracy was somewhat inconsistent over the first 25 epochs, later, it became consistent, and the model was successfully trained. Conversely, 2b shows the proposed CNN model's training and testing loss score per epoch. While the training loss went down steadily, the testing loss fluctuated in between 5 to 20 epochs, and became consistent after 20 epochs. So, the model works well both for the training and testing phases.

B. Explainability Analysis

SHAP is a useful method to understand how machine learning and deep learning models make predictions. It helps us to guess the importance of each feature in a prediction, explaining why the model decided on a certain outcome. In this research, we used the *SHAP DeepExplainer* method to figure out the SHAP values for our CNN model. Since our dataset is extensive, we took 5000 samples to calculate these values.

1) SHAP Global Interpretation: The summary plot shows the most essential features and the magnitude of their impact on the model. Figure 3 provides a hierarchical summary of the average effect on model output magnitude.

The size of the SHAP values shows how much a feature affects the prediction. If a feature has bigger SHAP values (either positive or negative), it has a stronger effect on predicting phishing URLs. Notably, the feature "*URL Length*" has the largest average impact on the model's classification, whereas "*Count WWW*" contributes the least. Furthermore, "*Hostname length*," is credited with having a major impact on phishing URL detection, contributing to the second-highest mean SHAP value. On the other hand, "*Count question mark*" makes the second-smallest contribution to the classification. "*Count @*," "*Suspicious words*," "*Count HTTP*," "*No of Embedded*," "*Abnormal URL*," "*Contain Ip Address*" and "*Is Google Index*" play least significant role to the classification of phishing URL detection.

2) SHAP Local Interpretation: Figure 4, shows the decision plot makes it possible to observe the amplitude of each change, taken by a sample for the values of the displayed features. In figure 4, each line shows what the model predicts for a specific example. The horizontal line in the middle represents the average prediction of the model. The features of the

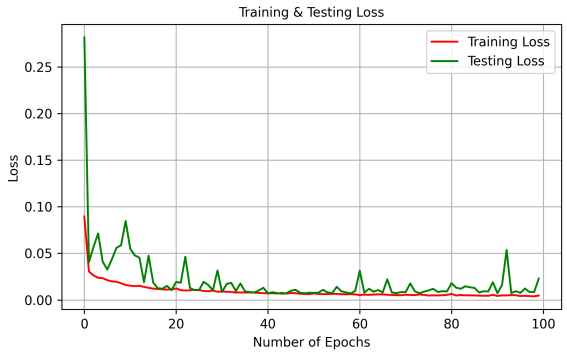
TABLE II: Summary of the proposed CNN model

Model: Sequential		
Layer (type)	**Output Shape**	**Param**
Conv1D	(None, 19, 32)	128
MaxPooling1D	(None, 9, 32)	0
Conv1D	(None, 7, 64)	6208
GlobalAveragePooling1D	(None, 64)	0
Dense	(None, 64)	4160
Batch_Normalization	(None, 64)	256
Dense_1	(None, 1)	65
Total Params: 10817 Trainable Params: 10689 Non-trainable Params: 128		

TABLE III: System Configuration.

Component	**Specification**
OS	Windows 11 (64-bit)
Processor	AMD Ryzen 5 5600G with Radeon Graphics
RAM	16GB
Storage	HP SSD EX900 500GB

(a)



(b)

Fig. 2: (a) Training and Testing accuracy per epoch, (b) Training and Testing loss per epoch.

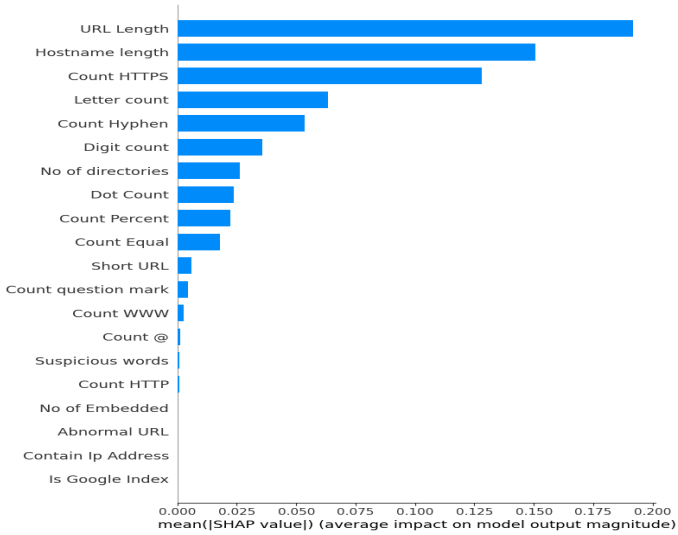


Fig. 3: Hierarchy of features with respect to their role in classification and Average impact on model output magnitude.

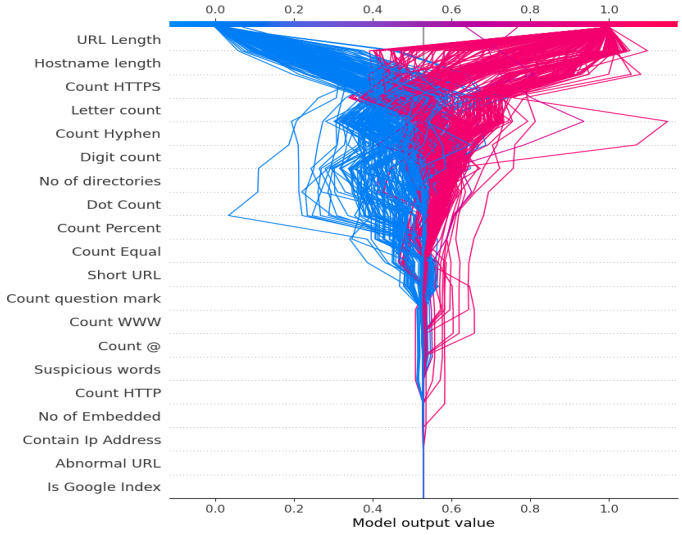


Fig. 4: SHAP value decision plot.

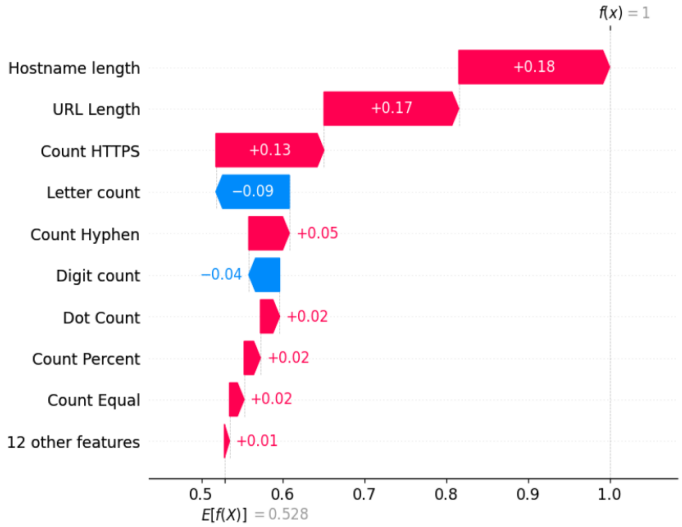


Fig. 5: SHAP value waterfall plot.

model are shown on the vertical axis. The colored lines start at the bottom and reach the middle line, representing the model's prediction for each example. The color of the line depends on the predicted value. As you move up the graph, we see how each feature adds to or subtracts from the model's prediction, showing the contribution of each feature to the overall prediction.

Figure 5 shows the waterfall plot, which is a local analysis tool that uses a bar chart to display the positive and negative impacts of different features on a model's prediction. It starts with a baseline value and helps to identify the features that influence the model's predictions most. The plot shows the values of different features and their impact on the model's predictions. In figure 5, the features like *Hostname length, URL Length* and other has a positive impact (shown in red), or the features like *Letter count, Digit count* have a negative impact (shown in blue) on moving the value from what the model would expect based on the background data to the actual output for a specific prediction.

C. Comparative Analysis

It is important to compare our proposed model with other benchmark models to evaluate the acceptance of our model. We extracted the important features before jumping into the model training and it is one of our key contributions. The model has been compared both with the previous works on the same dataset [18] and other ML models with our extracted features.

1) Performance comparison with conventional machine learning algorithms: Table IV compares the performance of several machine learning models with our proposed model on a classification task. As shown in the table, our proposed model achieves the highest accuracy (99.85%) and F1-score (99.80%) among all the models. This suggests that our model is the most effective at correctly classifying instances. The MultiLayerPerceptronClassifier model also performs well, with an accuracy of 99.78% and an F1-score of 99.79%. The other models have slightly lower accuracy and F1-score scores. These results demonstrate the effectiveness of our proposed model for this classification task.

2) Performance comparison with previous works: Table V compares the accuracy and explainability of our proposed model with other phishing URL detection models. The table demonstrates that, out of all the models, our 1D Convolutional Neural Network (CNN) model have the best accuracy (99.85%). Along with the proposed model, 21 categorical and numerical features played a vital role in predicting higher accuracy. Additionally, our model only provides explainability for its predictions. This means our model can identify phishing URLs accurately and explain why it classifies them as phish-

TABLE IV: Machine learning model performance comparison with the proposed model.

Models	Accuracy	Precision	Recall	F1-Score
KNN	99.47	99.79	99.20	99.49
DT	99.67	99.71	99.65	99.68
RF	99.12	99.10	98.88	98.98
XGB	99.71	98.80	98.79	98.67
MLP	99.78	99.72	99.86	99.79
Our proposed model	**99.85**	**99.90**	**99.80**	**99.80**

ing. This is a significant advantage, as it can help users to understand how the model works and to trust its predictions. The other models listed in the table are all less accurate than ours, and none provide explainability. The Ozgur Koray Sahingoz et al. [21], their CNN model is the second most precise model in the table and has an accuracy of 98.74%, but it didn't explain their model.

TABLE V: Research comparison with others

Ref.	Algorithm Used	Features Extraction	Accuracy	Explain-ability
[22]	GCNN	Content-based	98.37%	No
[23]	LightGBM	12 features	86.00%	No
[24]	KNN	14 features	90.00%	No
[21]	CNN	URL Vectorization	98.74%	No
Our Model	**1D CNN**	**21 features**	**99.85%**	**Yes**

VI. Conclusion

In this proposed phishing URL detection model, we have worked on creating an intelligent phishing website detection technique from URLs that might assist internet users. We have proposed a 1D Convolutional Neural Network (CNN) to detect the phishing URLs. As deep learning models are data-hungry, we have collected a huge amount of secondary data and carefully extracted 21 features from the URLs. It played a vital role in the creation of a robust model. After extensive experiments and comparisons, we have established that our model outperforms the existing models. Cybersecurity is a critical issue. So, the interpretability and explainability of the model are essential. To address the issue, we have shown the explainability of the model. It will help the user to be careful about the features that significantly contribute to the phishing URLs.

VII. Acknowledgements

The authors would like to acknowledge that all authors contributed equally to this work.

References

[1] K. Greene, M. Steves, and M. Theofanos, "No phishing beyond this point," *Computer*, vol. 51, 2018.

[2] S. R. Curtis, P. Rajivan, D. N. Jones, and C. Gonzalez, "Phishing attempts among the dark triad: Patterns of attack and vulnerability," *Computers in Human Behavior*, vol. 87, pp. 174–182, 2018.

[3] A. N. Shaikh, A. M. Shabut, and M. A. Hossain, "A literature review on phishing crime, prevention review and investigation of gaps," in *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, pp. 9–15, IEEE, 2016.

[4] B. B. Gupta, N. A. Arachchilage, and K. E. Psannis, "Defending against phishing attacks: taxonomy of methods, current issues and future directions," *Telecommunication Systems*, vol. 67, pp. 247–267, 2018.

[5] M. Volkamer, K. Renaud, B. Reinheimer, and A. Kunz, "User experiences of torpedo: Tooltip-powered phishing email detection," *Computers & Security*, vol. 71, pp. 100–113, 2017.

[6] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from url," *Electronics*, vol. 9, no. 9, p. 1514, 2020.

[7] W. Yao, Y. Ding, and X. Li, "Deep learning for phishing detection," in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pp. 645–650, IEEE, 2018.

[8] C. Opara, B. Wei, and Y. Chen, "Htmlphish: Enabling phishing web page detection by applying deep learning techniques on html analysis," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.

[9] M. Korkmaz, E. Kocyigit, O. K. Sahingoz, and B. Diri, "Phishing web page detection using n-gram features extracted from urls," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–6, IEEE, 2021.

[10] F. Tajaddodianfar, J. W. Stokes, and A. Gururajan, "Texception: a character/word-level deep learning model for phishing url detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2857–2861, IEEE, 2020.

[11] S. Y. Yerima and M. K. Alzaylaee, "High accuracy phishing detection based on convolutional neural networks," in *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1–6, IEEE, 2020.

[12] N. Q. Do, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita, "Phishing webpage classification via deep learning-based algorithms: An empirical study," *Applied Sciences*, vol. 11, no. 19, p. 9210, 2021.

[13] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," *Journal of Enterprise Information Management*, vol. 36, no. 3, pp. 747–766, 2023.

[14] Y. V. V. Murugesan, B. Janet, and S. Reddy, "Anti-phishing system using lstm and cnn," in *2020 IEEE International Conference for Innovation in Technology*, IEEE, 2021.

[15] H. Chapla, R. Kotak, and M. Joiser, "A machine learning approach for url based web phishing using fuzzy logic as classifier," in *2019 International Conference on Communication and Electronics Systems (ICCES)*, pp. 383–388, IEEE, 2019.

[16] R. A. A. Helmi, C. S. Ren, A. Jamal, and M. I. Abdullah, "Email anti-phishing detection application," in *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, pp. 264–267, IEEE, 2019.

[17] M. F. Kilkenny and K. M. Robinson, "Data quality:"garbage in–garbage out"," 2018.

[18] "Phishtank developer information." https://phishtank.org/developer_info.php. (Accessed on 02/09/2024).

[19] Canadian Institute for Cybersecurity — UNB, "URL 2016 — Datasets — Research — Canadian Institute for Cybersecurity — UNB." https://shorturl.at/hDIPT. (Accessed on 02/09/2024).

[20] H. Dalianis and H. Dalianis, "Evaluation metrics and evaluation," *Clinical Text Mining: secondary use of electronic patient records*, pp. 45–53, 2018.

[21] O. K. Sahingoz, E. Buber, and E. Kugu, "Dephides: Deep learning based phishing detection system," *IEEE Access*, 2024.

[22] M. Korkmaz, E. Kocyigit, O. Sahingoz, and B. Diri, "A hybrid phishing detection system using deep learning-based url and content analysis," *Elektronika ir Elektrotechnika*, vol. 28, no. 5, 2022.

[23] S. H. Ahammad, S. D. Kale, G. D. Upadhye, S. D. Pande, E. V. Babu, A. V. Dhumane, and M. D. K. J. Bahadur, "Phishing url detection using machine learning methods," *Advances in Engineering Software*, vol. 173, p. 103288, 2022.

[24] M. Mehndiratta, N. Jain, A. Malhotra, I. Gupta, and R. Narula, "Malicious url: Analysis and detection using machine learning," in *2023*

10th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1461–1465, IEEE, 2023.