Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 260 (2025) 1089–1097

Procedia Computer Science

www.elsevier.com/locate/procedia

Seventh International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R-2024)

Identifying Malicious URLs Using Deep Learning based VGG16 Architecture with Transfer Learning Method

Ajay Indian[a], Gaurav Meena[a]*, Krishna Kumar Mohbey[a], Siddharth Singh Kushwaha[a]

[a]*Department of Computer Science, Central University of Rajasthan, 305817, India*

Abstract

Recently, a sharp rise in cybersecurity attacks, including ransomware, phishing, malware injection, etc., has been seen on many websites worldwide. Hence, numerous commercial institutions, e-commerce companies, and individuals suffered substantial monetary losses. Experts in cyber security need help in this situation because new varieties of attacks are emerging daily. Deep learning-based algorithms have outperformed compared to other traditional machine learning algorithms in a variety of applications. However, classifying the Uniform Resource Locators (URLs) into malicious and non-malicious URLs is complex using feature extraction-based traditional machine learning algorithms. Therefore, there is some scope for further improvement. This article suggests a VGG16-based transfer learning approach to develop a model to detect malicious URLs. The malicious URL dataset, comprising 651,191 URLs, where 428,103 are benign or safe, 96,457 defacements, 94,111 phishing, and 32,520 malware, is used to develop the suggested model. The efficacy of the suggested model is measured using the loss, accuracy, precision, recall, and f1-score, and the suggested model attained efficacy of 0.152, 95.15%, 95.27, 95.03%, and 95%, respectively. The efficacy of the suggested model is also compared with the other existing models, and it is observed that the proposed model outperformed.

© 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the Seventh International Conference on Recent Trends in Image Processing and Pattern Recognition.

Keywords: Transfer learning, VGG16, Deep Learning, Malicious URL Classification, Cybersecurity;

* Corresponding author.
E-mail address: gaurav.meena@curaj.ac.in

1877-0509 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the Seventh International Conference on Recent Trends in Image Processing and Pattern Recognition.
10.1016/j.procs.2025.03.294

1. Introduction

The Uniform Resource Locator (URL) is a coherent structure used to uniquely identify and access the World Wide Web (WWW). Typically, a valid URL consists of three elements: i) Protocol (tells which protocol to use, such as HTTP, HTTPS, etc.), ii) Host name (another name for the resource name). It includes the domain name or IP address of the location of the actual resource. iii) Path (describes the specific path that leads to the resource). According to Figure 1, for a given URL, the protocol/scheme is HTTP; the primary domain name is example.com. Another element of the URL is the top-level domain, which indicates the kind of website, such as commercial (.com), educational (.edu), organization (.org), etc.

Fig. 1. Basic Elements of a URL

Malicious URLs are altered or compromised URLs used to attack the internet. Typically, a malicious URL or website includes various malware, Trojans, and unsolicited material in the form of phishing, drive-by downloads, and spam. The most popular means for disseminating malicious URLs include email and social networking platforms like Facebook, Twitter, WhatsApp, Orkut, etc. [1][2][3]. The hazardous website's primary goal is to defraud users or rob their personal or financial information. During the COVID-19 disease, significant growth in cybercrime events has been seen. Hence, numerous economic institutions, e-commerce industries, and individuals faced substantial monetary losses [4]. The delisting method is the foundation for most commercial products on the market [5]. This technique makes use of a dataset with a list of malicious URLs. Using scanning and crowdsourcing techniques, the anti-virus company updates the delisting regularly. Using domain knowledge to extract lexical characteristics from URLs is now the most often used technique, followed by a machine learning-based solution. Support Vector Machines (SVMs) are the most commonly used algorithm for machine learning models, while Bag-of-words (BoWs) are the widely utilized feature extraction technique [8]. Though delisting methods can be replaced by a machine learning-based solution, these methods come with certain drawback which leads to the motivation to pursue this study.

- Conventional machine learning solutions depend on handcrafted feature engineering, whereas traditional URL representation approaches fail to acquire links and sequential patterns between the characters in the URL.
- This is a challenging job that necessitates deep expertise in the field of cyber security. It does not generalize the test data and cannot perform the extraction of hidden features.
- Additionally, many unique words put a memory limit on the machine-learning model while it is being trained.

In this paper, detecting malicious URLs as a multi-class problem is solved by categorizing the raw URLs into four different classes, for example, benign or safe, phishing, malware, or defacement. Deep learning-based algorithms have outperformed compared to other traditional machine learning algorithms in a variety of applications. However, classifying the URLs into malicious and non-malicious URLs is complex using feature extraction-based traditional machine learning algorithms [5]. A model using a transfer learning approach based on VGG-16, where the model implicitly performs feature extraction from URLs, is proposed for this classification. Transfer learning is employing a model already trained to solve a new problem. These days, it is highly popular in deep learning because of its ability to train deep neural networks using a small volume of data. It is especially helpful in data analysis because most real-world cases only have a few labeled data points to develop such complex models [7]. The proposed work utilized the Malicious URLs dataset, which primarily makes the following contributions:

- We have tried to make improvements in the performance of existing systems.
- A URL classification model based on the VGG-16 standard is proposed.
- The results of the proposed model are also compared with other state of art.

The suggested model is analyzed with other models that were developed using the same datasets [10][11][12] in this study, which allowed us to perform a comparative study. In the subsequent sections, the outline of the rest of the paper is given. A brief review of research on malicious URL detection is presented in Section 2, and the suggested methodology is elaborated in Section 3. Section 4 thoroughly discusses the dataset, experimental setup, and experimental outcomes, and section 5 presents the study's conclusion and gives expected future work.

2. Related work

Detecting malicious URLs is crucial in ensuring cybersecurity and protecting users from online threats. Numerous research papers have addressed this issue and proposed various techniques and approaches. Using malicious URLs is one of the most prevalent techniques for spreading malware and ransomware, initiating phishing attacks, sending spam, and defacing websites. Malicious URLs may also be used to distribute spam. Attackers may distribute harmful URLs via email or advertisements or embed them on legal websites. When people do not pay attention and click on fraudulent URLs, they open themselves up to attack. In this paper, a hybrid strategy with machine learning and deep learning to identify fraudulent URLs is suggested. The suggested technique is trained and evaluated on a huge database with 2.5 million malicious and benign URLs [13]. Vanhoenshoven, Frank, et al. [14] proposed a machine learning-based approach for identifying malicious URLs. They used a feature extraction method based on n-grams. They employed different classification techniques, such as decision trees, random forests, and support vector machines (SVM), to categorize URLs as malicious or benign.

According to the research conducted by Chidimma et al. [15], HTMLPhish is a platform powered by deep learning and depends on data-driven, end-to-end automated phishing webpage categorization. A dataset of HTML contents was supplied in a real-life distribution containing more than 50,000 HTML pages. This method could analyze context characteristics from HTML pages without considering labor-intensive human feature engineering. The findings demonstrated that HTML Phish achieved more than 93% accuracy, indicating a successful outcome. The vulnerability of internet users to cyber-attacks and security weaknesses led to the development of algorithms based on artificial intelligence, which was accomplished by utilizing machine learning and deep learning methods [16]. Using a CNN equipped with n-gram characteristics, the authors intended to build a system capable of detecting phishing to protect against cyberattacks. The exception is a novel deep learning-based model [17] that can determine whether a URL is a phishing link. This is a distinct method compared to other methods, which extract information from the URL into two levels—the character and word levels—and rely far less on handcrafted generated characteristics.

An efficient phishing detection system was employed by Yerima et al. [18] was a one-dimensional CNN-based model. This model takes advantage of CNN for its capacity to differentiate between authentic and phishing websites. According to the authors, the model was used to analyze a website dataset with 4898 phishing websites and 6157 legal websites, respectively. In addition, the model achieved a phishing detection rate of 98.2% and an F1-score of 0.976 correspondingly. Xuan et al. [19] did another research in which they suggested an ML-based strategy for detecting malicious URLs. They employed 54 lexical, network, and content-based features to categorize URLs. The accuracy of the proposed random forest (RF) method was 96.28 percent. However, Subasi et al. [20] employed RF in an intelligent system to identify phishing websites with several ML algorithms, and they attained a greater accuracy of 97.36%. In their study, the dataset was taken from the UCI-ML repository [21]. Thirty lexical, network, and content-based characteristics were mined to categorize the URLs properly. By using a labeled capsule neural network (CapsNet) and an independently recurrent neural network (IndRNN), Yuan et al. [22] suggested a parallel neural joint model approach to examine and identify harmful URLs. The dataset was gathered via Alexa [23] and PhishTank [24]. Furthermore, Yu [25] suggested a fusion model for identifying phishing websites that incorporated the benefits of deep belief networks (DBNs) with SVM.

A stacking model was used in another research conducted by Zamir and colleagues [26], which presented a framework for identifying phishing websites using the dataset from Kaggle [27]. They collected 32 characteristics from the material, the lexicon, and the network. The classifiers with the most significant scores were used to develop two different stacking models: Stacking 1 (radio frequency (RF) + neural network (NN) + bagging classifier (BC)) and Stacking 2 (K-nearest neighbor (KNN) + RF + BC). Both models were stacked. Using the Stacking 1 model (RF + NN + BC), the best level of accuracy that could be attained was 97.4%. Alkhudair et al. [28] used a different strategy when implementing a malicious URL detection technique using four machine-learning methods. They employed 20 lexical, content-based, and network-based characteristics and collected their dataset from the Kaggle [29] and Urcuqui et al. [30] datasets. The RF method produced the best results, with an accuracy of 95%. On the other hand, Deebanchakkarawarthi et al. [31] used ML to reduce reliance on a database, boost productivity, and

identify dangerous URLs with superior results (accuracy of 97 percent). Selvaganapathy et al. [32] suggested a technique based on a stacked restricted Boltzmann machine to select the features with DNN to identify and classify malicious URLs. MalwareDomainList, the Spambase Datasets from the UCI-ML Repository [33], the Phishing Dataset from the UCI-ML Repository [34], and Alexa [23] were used to collect the data. There were 98 characteristics taken out of the data. DBN was the most accurate method (up to 75% accuracy).

Web services have rapidly become an essential component of modern living. Sadly, these services are under attack from many different fronts. These dangers include phishing, viruses that are spread over email, programs known as Trojan horses, denial of service (DoS) attacks, etc. One of the attacks, a distributed DoS attack, is a particularly robust and active assault on the internet [35]. Similar heuristic methods for detecting phishing sites hosted on hacked servers were developed by Rao et al. [36]. They chose to use ten content-based features, six lexical features, and one network-based feature. The accuracy of the twin SVM (TWSVM) was found to be the highest at 98.05%. It is not unusual for websites and mobile apps to have a careless defect that negatively influences the privacy and security of users' data. Attacks that exploit database vulnerabilities are becoming increasingly widespread and damaging [37]. A model that offers a strong method to protect the privacy of the data and the classifier was suggested by the authors [38]. The effectiveness of the suggested model is calculated by running trials using a Naive Bayes classifier over a variety of data sets. The study [39] presents a unique model that uses machine learning, differential privacy, and k-anonymization to accomplish classification tasks.

3. Proposed methodology

The proposed methodology adopted to pursue this research to detect malicious URLs is presented in Figure 2. The proposed methodology starts with collecting the URL dataset from the source, then loading the dataset and applying the Unicode encoding on the dataset as per the requirement of the suggested classification model, discussed in the preceding section. After applying the encoding, an encoded URL must be reshaped into a 24x24 matrix by applying zero padding. Once the URL dataset is encoded using 24x24 matrices, it is partitioned into the train (80%) and tests (20%) samples using the random-split technique. According to the URL classification problem requirements, the suggested approach alters the classification layers of the VGG-16, VGG-19, and AlexNet standard architectures. The softmax activation function is utilized in the classification layer. The suggested VGG -16 architecture is displayed in Figure 3. The experiment's data set was collected from a dataset of malicious URLs. Along with the classification accuracy, other performance measuring criteria, for example, precision, recall, and the F1-score, were employed to analyze the outcomes as the dataset employed for the present study is imbalanced.

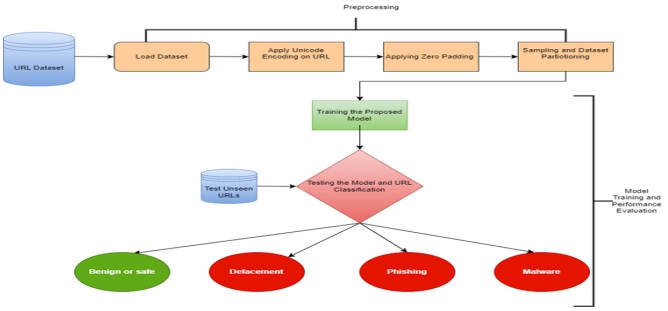

Fig. 2 Methodology Adopted to Develop Suggested Model for Detecting Malicious URLs

4. Experimental results and discussion

4.1 Dataset Description and Preprocessing

Various URL formats must be tested to evaluate the effectiveness of VGG-16, VGG-19, and AlexNet-based Transfer Learning models and standard machine learning classifiers. An extensive dataset of 651,191 URLs made public on Kaggle is utilized in this study to create a deep learning-based model to identify harmful URLs so that one can prevent them in advance before they infect computers or spread via the internet. A complete statistical analysis

of the dataset is provided in Table 1 [6]. CNNs are generally used in the vision domain. However, the obvious question is how such neural network models can be applied to classify whether a URL has malicious intentions. It is a fascinating topic because images are simply matrices with cell values ranging from 0-255, representing shades of black and white. These cell values are then fed into a neural network model, which produces relevant information about the image required for the classification. The neural network does not understand words but behaves like a function that understands mathematical objects such as numbers, vectors, and matrices. Therefore, data preprocessing will include the Unicode encoding of each character in the URL. Using Unicode encoding, the proposed neural network extracts the features to differentiate malicious and non-malicious URLs.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 24, 24, 32)	320
conv2d_1 (Conv2D)	(None, 24, 24, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 12, 12, 32)	0
conv2d_2 (Conv2D)	(None, 12, 12, 64)	18496
conv2d_3 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_4 (Conv2D)	(None, 6, 6, 128)	73856
conv2d_5 (Conv2D)	(None, 6, 6, 128)	147584
conv2d_6 (Conv2D)	(None, 6, 6, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 128)	0
conv2d_7 (Conv2D)	(None, 3, 3, 216)	27864
conv2d_8 (Conv2D)	(None, 3, 3, 216)	46872
conv2d_9 (Conv2D)	(None, 3, 3, 216)	46872
max_pooling2d_3 (MaxPooling2D)	(None, 1, 1, 216)	0
conv2d_10 (Conv2D)	(None, 1, 1, 216)	46872
conv2d_11 (Conv2D)	(None, 1, 1, 216)	46872
conv2d_12 (Conv2D)	(None, 1, 1, 216)	46872
max_pooling2d_4 (MaxPooling2D)	(None, 1, 1, 216)	0
flatten (Flatten)	(None, 216)	0
dense (Dense)	(None, 576)	124992
dense_1 (Dense)	(None, 576)	332352
dense_2 (Dense)	(None, 4)	2308
Total params: 1,155,892		
Trainable params: 1,555,892		
Non-trainable params: 0		

Fig. 3. Suggested Transfer Learning Model based on VGG-16 Architecture.

Table 1. Statistical Breakdown of the Malicious URLs Dataset

S. No.	Types of URL	No. of URLs	Percentage of URLs
1	Benign or safe	428103	65.74%
2	Defacement	96457	14.81%
3	Phishing	94111	14.45%
4	Malware	32520	4.99%
Total No. of URLs		**651191**	

This is superior to other encoding options like character embedding, word embedding, and one-hot encoding. Unicode encoding of each character is a better option for being very efficient in storage. It can also support encoding any URL with characters from multiple non-English languages. Hence, Unicode encoding of characters can uniquely encode any characters present in the URL regardless of its language. When a character is typically embedded, it is often encoded into a vector, which is more expensive than simply embedding it in its unique Unicode numerical representation. The proposed study employs a Unicode numerical representation of each character in a URL. Encoded URLs are formatted into a 24x24 matrix with padding with zeros if required for input to the VGG16 neural network to construct the malicious URL detection model. Figure 4 shows the example of a URL represented using Unicode encoding in 24x24 shapes. Random split is the most prevalent technique for partitioning data into training and testing datasets. The dataset is partitioned into train (80%) and test (20%) samples using the random-split technique employed in the suggested study [9].

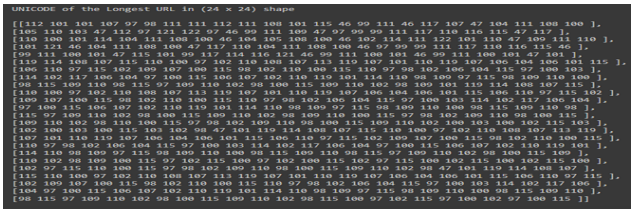

Fig. 4. A URL Encoded in 24x24 shape using Unicode Encoding.

4.2 Experimental Setup and Experimental Outcomes

All experiments were conducted using Google Colab setup with Anaconda, Python 3.10.11, Jupyter 3.0.14, and TensorFlow 2, a cloud-based service that implicitly allows writing and executing code without setting up a local environment for experimentation. It provides 12 GB RAM with access to GPU and TPU computing resources required for training the model faster. VGG16, VGG19, and AlexNet standard architectures based on deep learning are the primary concern of this study. The performance of this architecture is the best reported in the literature [10][11][12][13][14][15]. In transfer learning, pre-trained weights are applied during the model's development.

As part of the transfer learning process, the final classification layer of VGG16, VGG19, and AlexNet was altered to meet the requirements for Benign or safe, Defacement, Phishing, and Malware classifications. The classification layer employs a flattened and a dropout layer (only for AlexNet). Initially, the learning rate of 0.001 and 10 epochs were taken. After several tests, a dropout parameter with 0.2 (for AlexNet), a single flattened layer accompanied by a dense layer, and softmax activation were set. Initially, AlexNet, VGG-16, and VGG-19 models were trained with 5 to 10 epochs; on testing, VGG-16 outperformed VGG-19 and AlexNet. Thus, the results of VGG-16-based models are analyzed and evaluated against the other models. Consequently, the train, test, and validation datasets utilized the identical input shape 24x24 and batch size 100. The model that has the lowest loss in validation was saved. Figure 5 and Figure 6 present the accuracy and loss graphs for the proposed VGG-16-based model for the malicious URL datasets, respectively.

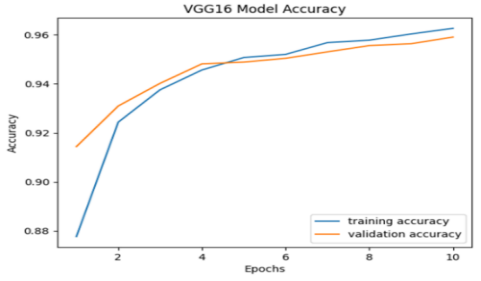

Fig. 5. Accuracy graph for Suggested VGG16-based Model

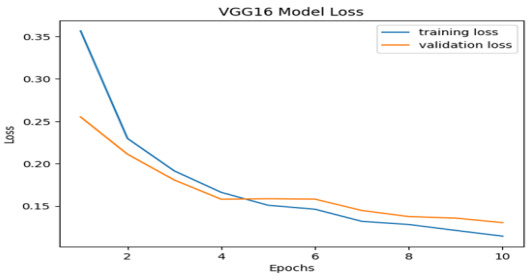

Fig.6. Loss graph for Suggested VGG16-based Model.

Table 2. Assessment of Performances among Different Suggested Transfer Learning-based Models.

Proposed CNN Architecture	No. of Epochs	Training/ Testing	Loss	Accuracy	Recall	Precision
AlexNet	5	Training	0.2011	0.9308	0.9281	0.9336
	10	Testing	0.2163	0.9260	0.9232	0.9287
VGG16	5	Training	0.14	0.9516	0.9502	0.9530
	10	**Testing**	**0.1919**	**0.9429**	**0.9420**	**0.9443**
VGG19	5	Training	0.1501	0.9511	0.9492	0.9532
	10	Testing	0.1529	0.9494	0.9481	0.9506

Experimenting showed that the proposed model with VGG-16 architecture provides the best accuracy compared to AlexNet and VGG-19 architectures regarding loss, accuracy, precision, recall, and F1-measure without requiring equalizations, which the researchers generally adopt. The outcomes are presented in Table 2 and Figure 7. The efficacy of the suggested model based on VGG-16 was observed as the loss was 0.1311, accuracy was 0.9587, precision was 0.9591, and recall was 0.9583. Following the literature review findings, a comparative analysis was

done with another state-of-the-art method for classifying the malicious URLs with the proposed model of malicious URL classification, as presented in Table 3.

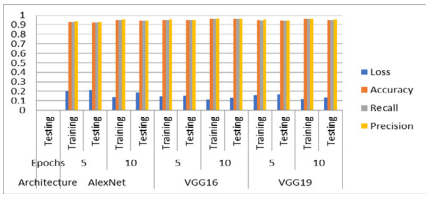

Fig. 7. Performance Assessments of Different Transfer Learning-Based Models.

Table 3. An Comparison among other states of arts for Classifying Malicious URLs and Proposed VGG16-based Transfer Learning Model

Author	Approach	Model	No. of URL Classes	Accuracy	Recall	Precision	F1-Score
Proposed Models	Transfer Learning	Alex Net	4	0.9429	0.9420	0.9443	0.94
	Model-based on	VGG16	4	0.9587	0.9583	0.9591	0.9586
	CNN	VGG19	4	0.9522	0.9516	0.9531	0.9523
U. S. D. R et al. [10]	Machine Learning	XG Boost	4	0.955	0.92	0.95	0.93
Vara Vundavalli et al. [11]	Machine Learning	Naïve Bayes'	2 (Benign and Malicious)	0.91	0.90	0.25	0.27
H. M. Junaid Khan et al. [12]	Machine Learning	Voting Classifier	2 (Benign and Malicious)	0.9537	0.88	.955	0.915

Based on the accuracy, the Proposed VGG16-based Transfer Learning Model outperformed all other models with 0.9587, followed by the XG Boost approach [10] with 0.955 and the Voting Classifier [12] with 0.9537. Moreover, in [12], URLs are only classified into two classes, whereas in our proposed approach, classification is done in four classes of URLs. As the dataset used for this study is imbalanced, accuracy is not the right metric to assess the efficacy of the models; one cannot say VGG16 is better than others. So, other metrics (Precision, Recall, and F1-score) must also be considered for unbiased analysis. Table 4 shows that VGG16 architecture outperformed other models based on Recall, Precision, and F1-Score with 0.9583, 0.9591, and 0.9586, respectively. The performance of the suggested VGG19-based Transfer Learning Model is much closer to the suggested VGG19-based model in terms of Accuracy, Recall, Precision, and F1-Score.

The experimental outcomes have shown that the suggested model works superior to the existing models regarding accuracy, precession, recall, and F1 score. In the present study, specifically, the VGG16 model is proposed, which has been already trained for image classification with high classification efficiency. Consequently, the proposed model performs better than traditional machine learning models regarding speed and accuracy. Moreover, the proposed models rely on something other than the traditional explicit feature extraction method. Instead, it uses the implicit feature extraction approach built into the proposed model. Several hyper-parameters were considered when building the transfer learning model to increase learning and reduce loss. Other hyper-parameters, for instance, learning rate, number of iterations, number of hidden layers, number of hidden nodes, and selecting an activation function, can be utilized to fine-tune the learning process. The suggested model has gone through several iterations to improve its efficacy. During the study, the standard AlexNet, VGG16, and VGG19 architectures were utilized to prepare the models, and only the classification layer was modified per the requirements of the malicious URL classification problems.

5. Conclusion and future work

The performance of existing systems for malicious URL categorization was enhanced by modifying hyperparameters of VGG16 architecture using the malicious URL dataset. Based on the VGG16 architecture, the suggested transfer learning model performed better than the other existing models. The suggested model's classification accuracy for malicious URLs is 95.87% for the malicious URLs Dataset. When comparing the

proposed model to other existing model in terms of performance and design, it is concluded that it surpasses the prevailing models. While experimenting, it is also concluded that the suggested model may be employed for real-time URL classification. A public API may be developed to utilize in future work. It is likely to retrain these models repeatedly as additional data is collected, leading to better and better model building to deal with time.

References

[1] Tran, K. N., Alazab, M., & Broadhurst, R. (2013, November). Towards a feature-rich model for predicting spam emails containing malicious attachments and URLs, in 11th Australasian Data Mining Conference, Canberra.

[2] Alazab, M., & Broadhurst, R. (2016). Spam and criminal activity. Trends and issues in crime and criminal justice, (526), pp. 1–20.

[3] Alazab, M., Layton, R., Broadhurst, R., & Bouhours, B. (2013, November). Malicious spam email developments and authorship attribution. In Cybercrime and Trustworthy Computing Workshop (CTC), 2013 Fourth (pp. 58–68). IEEE.

[4] Broadhurst, R., Grabosky, P., Alazab, M., Bouhours, B., & Chon, S. (2014). An Analysis of the Nature of Groups Engaged in Cyber-Crime.

[5] Sahoo, D., Liu, C., & Hoi, S. C. (2017). Malicious URL detection using machine learning: A survey. arXiv preprint arXiv:1701.07179.

[6] Malicious URLs Dataset (2022), https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset?select=malicious_phish.csv. Accessed: Mar. 12, 2022. [Online]

[7] Agarwal, N., Sondhi, A., Chopra, K., Singh, G. (2021). Transfer Learning: Survey and Classification. Advances in Intelligent Systems and Computing, vol 1168. Springer, Singapore. https://doi.org/10.1007/978-981-15-5345-5_13

[8] Sahoo, D., Liu, C., & Hoi, S. C. (2017). Malicious URL detection using machine learning: A survey. arXiv preprint arXiv:1701.07179.

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

[10] U. S. D. R, A. Patil, and Mohana, "Malicious URL Detection and Classification Analysis using Machine Learning Models," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 470–476, doi: 10.1109/IDCIoT56793.2023.10053422.

[11] Vara Vundavalli, Farhat Barsha, Mohammad Masum, Hossain Shahriar, and Hisham Haddad, "Malicious URL Detection Using Supervised Machine Learning Techniques," 13th International Conference on SIN, November 2020, Article No.: 21, Pages 1–6.

[12] H. M. Junaid Khan, Q. Niyaz, V. K. Devabhaktuni, S. Guo and U. Shaikh, "Identifying Generic Features for Malicious URL Detection System," 2019 IEEE 10th Annual UEMCON, New York, NY, USA, 2019, pp. 0347–0352.

[13] Gogoi, Bronjon, Tasiruddin Ahmed, and Arabinda Dutta. "A Hybrid approach combining blocklists, machine learning and deep learning for detection of malicious URLs." 2022 IEEE India Council International Subsections Conference (INDISCON). IEEE, 2022.

[14] Vanhoenshoven, Frank, et al. "Detecting malicious URLs using machine learning techniques." 2016 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2016.

[15] Opara, C.; Wei, B.; Chen, Y. HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis. In Proceedings of the 2020 IJCNN, Glasgow, UK, 19–24 July 2020; pp. 1–8.

[16] Korkmaz, M.; Kocyigit, E.; Sahingoz, O.K.; Diri, B. Phishing Web Page Detection Using N-gram Features Extracted From URLs. In Proceedings of the 2021 3rd International Congress on HORA, Ankara, Turkey, 11–13 June 2021; pp. 1–6.

[17] Tajaddodianfar, F.; Stokes, J.W.; Gururajan, A. Texception: A Character/Word-Level Deep Learning Model for Phishing URL Detection. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2857–2861.

[18] Yerima, S.Y.; Alzaylaee, M.K. High Accuracy Phishing Detection Based on Convolutional Neural Networks. In Proceedings of the 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 19–21 March 2020; pp. 1–6.

[19] C. Do Xuan, H. D. Nguyen, and T. V. Nikolaevich, "Malicious URL detection based on machine learning," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 1, pp. 148–153, 2020, doi: 10.14569/ijacsa.2020.0110119.

[20] A. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," in Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA), Jun. 2018, pp. 1–5, doi: 10.1109/ICECTA.2017.8252051.

[21] M. Rami, M. Lee, & T. Fadi. (2015). UCI Machine Learning Repository: Phishing Websites Data Set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/phishing+websites

[22] J. Yuan, G. Chen, S. Tian, and X. Pei, "Malicious URL detection based on a parallel neural joint model," IEEE Access, vol. 9, pp. 9464–9472, 2021, doi: 10.1109/ACCESS.2021.3049625.

[23] Alexa—Top sites. Accessed: Jan. 12, 2022. [Online]. Available: https://www.alexa.com/topsites

[24] PhishTank—Join the Fight Against Phishing. Accessed: Jan. 1, 2022. [Online]. Available: https://www.phishtank.com.

[25] X. Yu, "Phishing websites detection based on a hybrid model of deep belief network and support vector machine," IOP Conf. Earth Environ. Sci., vol. 602, no. 1, Nov. 2020, Art. no. 012001, doi: 10.1088/1755- 1315/602/1/012001.

[26] A. Zamir, H. U. Khan, T. Iqbal, N. Yousaf, F. Aslam, A. Anjum, and M. Hamdani, "Phishing web site detection using diverse machine learning algorithms," Electron. Library, vol. 38, no. 1, pp. 65–80, 2020, doi: 10.1108/EL-05-2019-0118.

[27] K. Akash. (2018). Phishing Website Dataset. Kaggle. Available: https://www.kaggle.com/akashkr/phishing-website-dataset#dataset.csv/

[28] F. Alkhudair, M. Alassaf, R. U. Khan, and S. Alfarraj, "Detecting malicious URL," in Proc. Int. Conf. Comput. Inf. Technol. (ICCIT), vol. 1, Sep. 2020, pp. 97–101, doi: 10.1109/ICCIT-144147971.2020.9213792.

[29] Malicious and Benign Websites | Kaggle. Accessed: Jan. 19, 2022. Available: https://www.kaggle.com/xwolf12/malicious-andbenign-websites

[30] C. Camilo, U. López, J. O. Quintero, and A. Navarro. (2017). Machine Learning Classifiers to Detect Malicious Websites. Accessed: Jan. 19, 2022. [Online]. Available: http://ceur-ws.org

[31] G. Deebanchakkarawarthi, A. Parthan, L. Sachin, and A. Surya, "Classification of URL into malicious or benign using machine learning approach," Int. J. Adv. Res. Comput. Commun. Eng., vol. 8, no. 2, pp. 245–248, Feb. 2019, doi: 10.17148/IJARCCE.2019.8247.

[32] S. G. Selvaganapathy, M. Nivaashini, and H. P. Natarajan, "Deep belief network-based detection and categorization of malicious URLs," Inf. Secure. J., Global Perspective, vol. 27, no. 3, pp. 145–161, Apr. 2018, doi: 10.1080/19393555.2018.1456577.

[33] UCI Machine Learning Repository: Spambase Data Set. Accessed: Jan. 14, 2022. Available: https://archive.ics.uci.edu/ml/datasets/spam base

[34] Netscape. DMOZ OpenDirectoryProject. Accessed: Jan. 12, 2022. [Online]. Available: https://dmoz-odp.org/.

[35] Kumawat, Harsh, and Gaurav Meena. "Characterization, Detection, and Mitigation of Low-Rate DoS attack." Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies. 2014.

[36] R. S. Rao, A. R. Pais, and P. Anand, "A heuristic technique to detect phishing websites using TWSVM classifier," Neural Comput. Appl., vol. 33, no. 11, pp. 5733–5752, Jun. 2021, doi: 10.1007/s00521-020- 05354-z.

[37] Choudhary, Ravi Raj, Susheela Verma, and Gaurav Meena. "Detection of SQL Injection attack Using Machine Learning." 2021 IEEE International Conference TRIBES, IEEE, 2021.

[38] Gupta, R., & Singh, A. K. (2022). A differential approach for data and classification service-based privacy-preserving machine learning model in cloud environment. New Generation Computing, 40(3), 737-764.

[39] Singh, A. K., & Gupta, R. (2022). A privacy-preserving model based on differential approach for sensitive data in cloud environment. Multimedia Tools and Applications, 81(23), 33127-33150.