International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

An Adaptive Multi-Modal Deep Learning Approach for Real-Time Phishing URL Detection

Sachin Singh[1], Deepak Agrawal[2]

[1,2] Dept. Of Computer Science & Engg, Shri Ram Institute of Science &Technology, Jabalpur (M.P.)

Abstract - In today's rapidly evolving digital environment, phishing attacks have emerged as a major cybersecurity threat, targeting individuals, organizations, and online services through deceptive websites designed to steal sensitive information such as login credentials, financial details, and personal data. Traditional phishing detection approaches, including blacklists, signature-based methods, and rule-based systems, often struggle to identify newly generated, obfuscated, and previously unseen phishing websites. This research proposes a Multi-Model Deep Learning framework for phishing URL detection using URL, domain, and web content intelligence, with the objective of improving detection accuracy, generalization capability, and robustness against evolving phishing attacks.

The research contributes a comprehensive and scalable approach to phishing detection by combining multi-source website intelligence with multi-model deep learning. The proposed framework addresses limitations of conventional URL-only detection techniques and provides a foundation for developing intelligent, adaptive, and real-time phishing detection systems capable of responding to continuously evolving cyber threats.

Key Words: Phishing Detection, Phishing URLs, Multi-Model Deep Learning, URL Intelligence, Domain Intelligence, Web Content Intelligence, Cyber security, Deep Learning, Machine Learning,

I. INTRODUCTION

The dependence on online services on our diverse aspects is increased. Life, frequency and complexity of cell creatures aimed at personal data grow up. Fishing is an ordinary and serious threat when an attacker those who deceive people who disclose confidential information such as password and bank details [1]. Many phishing attacks based on web pages imitate security it is difficult to detect certificates such as true websites and online remains. The distribution is so serious that there are about 4.7 billion people Attack attempts were registered in 2022 [2] the numbers are constantly increasing. Recognition and understanding of methods Protection of the threat is still

not enough to the public, making phishing one of the most effective and destructive types of attack today.

Phishing attacks are not technically complicated, and batches require some effort. But in general, it is very effective. The attacker is very difficult to identify phishing sites by making a phishing website well with the emergence of legal sites that want to pretend to be. They affect pension recipients who can damage their personality and accounts, which lead to stroke as well as the potential crisis of trust in online services. . A study conducted by ENISA [2] shows that phishing attacks are one of the most common cyber necents that can be applied to European average companies. In a report on threats to cyber security, CISCO suggests that fishing accounts for about 90%of data violations in 2020 [3]. In addition, 86%of the organization attempted to connect to the FIG site. Indeed, as discussed in [4], people are in principle victims of phishing attacks due to insufficient interest in the victims of phishing attacks, especially the website and lack of proper education. According to a report from the phishing trend [5], the number of total phishing webs observed in the first quarter of 2022 exceeds millions of. Another interesting result reported by Appwg indicates the most sectors for attackers. Financial services, including banks, are especially easy to have phishing.

In the face of more and more complex phishing attacks, there are a few mechanisms such as blacklists and hub rules inefficient. Creating a trained ML model trained in many phishing data Related functions, existing and phishing attack development can be detected more precisely. But the biggest problem is Effective choice of specific characteristics for a concentrated detection model. Otherwise, detection with flaws it allows theft of important confidential information or starts a wrong warning. Therefore, the most relevant definition the brilliant features of the ML algorithm are important.

Our work aims to provide a broad and comprehensive review of the state of the art in the area of phishing website detection by focusing on the most relevant solutions proposed for the same. We subdivide these solutions in three main categories according to their target, namely:

- List-based.

• Similarity-based.

• Machine learning based.

The Feature Selection method can be widely classified into three categories.

i) Filter method using tests such as chi-quadrates and mutual information by measuring statistical significance, we evaluate the relevance of signs;

ii) Lapping method to select and apply multiple sub - sets of functions Model performance metrics such as accuracy or F1 evaluation to ultimately choose 2 magazine and

iii) Meta -Hebree optimization method using algorithms such as gene algorithms or particles by minimizing or minimizing certain target functions.

For each category we describe the suggested detection methods, and the datasets considered for their assessment. In addition, we discuss the main strengths and weaknesses of these approaches and identify the most important research gaps that need to be filled.

1.1 Working model of malicious web-page attack

Most typical attacks are in the form of malicious WebPages, phishing, and spreading of web viruses [6]. In malicious web attacks, attackers use camouflage technologies to push malicious web content to the front of users and lure them to enter malicious spam sites. Phishing refers to an attack in which an attacker sends users to a fake web link or sends them a highly deceptive E-mail to lure them into clicking on a link. Consequently, users' private information such as the password is leaked. In addition, to form a web virus, attackers use scripting languages to write malicious code and insert the virus into a browser bug. When users enter a web site, the virus is immediately woken up. Malicious programs are used to add, delete, and change the files of the local computer and even to close multiple functions of the system or format the disk. The attack's intention is achieved by exploiting security holes in the user's browser [7]. These attack behaviors seriously threaten users' information security. The prevention of accessing malicious WebPages is achieved mainly by identifying them, using static or dynamic methods. Static detection includes two methods: malicious link detection and static analysis based on web content. The former is achieved primarily by detecting phishing and Trojan links. The later attempts to detect webpages 'source codes based on the features of malicious codes. In general, static detection methods use an analysis tool to analyze the static features and functional modules of malicious codes. Dynamic detection is based mainly on discriminating interactive behaviors, where the state of the browser's interaction with the web server is monitored during a user's visit. If the state is abnormal, the webpage is identified as malicious. Dynamic detection is frequently applied in sandboxes or honey pots. The main principle to monitor the interaction behavior and record the attack method of the attacker while protecting the local computer.

Multi-webpage's detection process in the dynamic analysis model can be viewed as a Markov decision process (MDP), and thus machine learning (ML) such as decision-tree can achieve the optimal solutions via train-and-test if the decision-tree is deep enough. As a ML model for the detection of malicious WebPages, the both MDP and decision-tree ML technologies can accurately classify WebPages without analyzing errors and adjusting weight parameters. Compared with the decision-tree, the combing MDP with decision tree called as a Markov detection tree can represent a series state of webpage's based on the relation of URL among webpage's and thus provides more automatic decision for each webpage detection by employing the forward and backward searches of MDP. Therefore, we propose a detection approach based on MDP and decision-tree to improve the accuracy and efficiency during the process of classifying webpages.

1.3 Motivation

Phishing attacks are among the most common and dangerous forms of cybercrime today. Cybercriminals disguise malicious websites and emails to trick users into revealing sensitive information such as passwords, credit card numbers, and personal data. With the rise of e-commerce, online banking, and social media, millions of users are vulnerable to phishing scams every day.

Traditional methods of detecting phishing rely heavily on blacklists, which are often outdated and fail to catch newly created phishing URLs. Hence, there is a critical need for intelligent, real-time detection systems that can analyse the structure and behaviour of a URL to predict its legitimacy — even if it has never been seen before.

This paper aims to build a real-time phishing URL detection system using machine learning techniques that analyse various features of a URL (such as length, special characters, domain reputation, presence of suspicious keywords, etc.). The goal is to empower users

with instant alerts about potentially harmful links, helping prevent data breaches, financial losses, and identity theft.

II. LITERATURE SURVEY

2.1 Machine Learning Methods for Cyber Security

These techniques offer both generalizability and robustness, making them highly resistant to real-world attacks [8]. The work in [9] proposed a machine learning approach for malicious URL detection by combining linear and non-linear space transformation approaches. The authors employed Singular Value Decomposition (SVD), Distance Metric Learning - Nystrom techniques (DML-NYS) algorithms by using a dataset of 331,622 with 62 classes for training. The features are gathered and utilized to evaluate the classification using the RF and Gradient Boosting Decision Tree (GBDT) machine learning methods. The findings demonstrate the suggested method's performance by reaching a superior accuracy of 96.4%. Reference [10] suggests a machine learning-based method for identifying malicious URLs. A dataset of 470,000 URLs was used for training, and it had an accuracy of 92.174% [11].

Reference [12] suggested a machine learning approach for developing and evaluating real-time malware detection for URLs. In the other study [13], the authors proposed a methodology to detect malicious URLs and the types of attacks based on multiclass classification. They utilized classification algorithms like One-Vs-All (OVA) L1-reg L2-loss SVM (OVA SVM), One-Vs-One (OVO) L1-reg L2-loss SVM (OVO SVM), and Multi-Class Confidence Weighted Learning (MC-CW). The dataset, which contains 49935 URLs, was collected from the Alexa top sites, PhishTank, MalwareDomainList, and jwSpamSpy. From a total of 117 features, they extracted 65 lexical, 34 content-based, and 18 host-based attributes. They have achieved the highest accuracy of 99.86% in the detection of malicious URLs using a binary setting and an average accuracy of 98.44% using CW.

Seize Malicious URL [14], proposed a novel approach to identifying harmful websites by leveraging a diverse set of machines learning techniques, including RF, Decision Trees (DT), k-Nearest Neighbors (k-NN), NB, and SVM. This approach involves the prediction of website classes, followed by the application of a threshold to refine the results. It then amalgamates the decisions based on associated class probabilities and utilizes the label with the highest-class probability to arrive at a comprehensive determination regarding unlabelled websites. Although these techniques have proven effective, their widespread implementation in industry and in real time is yet to come. Their main weakness lies in their complete dependence on data. These methods often struggle due to the challenge of creating a comprehensive and generalized dataset. Malicious URL patterns and tactics continually change, making it difficult to keep datasets up to date [15], [16]. Another significant weakness is the presence of bias in the training dataset. If the training data is biased towards certain types of malicious URLs or if it lacks diversity, the model may not perform well in detecting less common or evolving threats. Studies [17], [18] have demonstrated that methods constructed using a high accuracy machine learning method using a training dataset (such as Kaggle with over 400,000 websites) may not be effective when applied to another dataset.

A further limitation involves selecting and extracting relevant features from URL data. Inadequate feature selection and extraction can lead to suboptimal model performance, as important information may be overlooked, or irrelevant features may introduce noise into the model. The last fundamental limitation lies in the delicate trade-off between overfitting and under fitting. Overfitting occurs when models become excessively specialized in recognizing known attack patterns present in the training data. While these models may accurately detect known threats, they often struggle with novel attack methods, failing to generalize effectively.

2.2 Deep Learning Methods for Cyber Security

Deep learning is a subset of a larger family of machine learning approaches based on artificial neural networks and representation learning. In particular, it seeks to learn relevant features directly from a dataset and perform classification and clustering utilizing these features [19]. Deep learning eliminates the feature selection procedure of machine learning methods, which increases system performance and prevents the loss caused by the selection of incompatible features. A deep learning network is utilized to systematically extract features from a dataset of URLs, which is then used to identify harmful URLs. In the final step, the trained network can then return a float outcome (between 0 and 1) indicating whether the input URL is malicious or benign. The authors in [20] focused on a deep learning neural network detection approach for detecting harmful URLs. The researchers conducted two separate datasets

that utilized Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) methods, and a combination of them [21]. It employs a convolutional Gated-Recurrent-Unit (GRU) neural network based on characters as text classification parameters, yielding an accuracy rate of more than 99.6% and making it ideal for high precision classification purposes. In the other research, authors suggested URLNet, CNN-based deep neural network, to learn a nonlinear URL embedding for malicious URL detection directly from the URL. This approach allows the method to capture several types of semantic information that were not possible with the existing methods. The bag-of-words approach was presented, which is a form of lexical feature, and jointly optimized the network using character and word CNNs.

2.5 Related Work

The paper [22] presented a Factorization Machine (FM), a form of deep learning algorithm for identifying malicious URLs. It means a Temporal Convolution Network (TCN) is employed to learn the long-distance dependence between URL characters. Precise Phishing Detection with Recurrent Convolutional Neural Networks (PDRCNN) method presented in [23], suggests a rapid approach for detecting malicious URLs that relies solely on lexical features. The PDRCNN achieves a detection accuracy of 97% on a dataset with 245,385 valid URLs.

Many machine learning problems have been overcome, but there are still several major issues remaining. Massive volumes of URLs needed to be used for training to create a suitable detection method with acceptable levels of accuracy for deep learning. This problem becomes much worse when new URLs are available, and the method need to retrain [24]. Due to a lack of knowledge of rules developed by machines, which prevents upgrading and optimizing the rules by the developers [25]. Moreover, the detection method's reliability and level of accuracy are entirely dependent on the quality of the dataset [26]. Lastly, an issue of note is the feature selection contradiction, with most of the research still involving manual classification of features.

In this study [27], we introduce an innovative framework for malicious URL detection based on predefined static feature classification by allocating priority coefficients and feature evaluation methods. Detection accuracy of 98.95% and a precision value of 98.60%.In papers [28] [29] [30], we compare machine learning and deep learning techniques to present a method capable of detecting phishing websites through URL analysis. In work [31], we propose PhiKitA; a novel dataset that contains phishing kits and phishing websites generated using these kits. In paper [32], we propose a feature-free method for detecting phishing websites using the Normalized Compression Distance (NCD), a parameter-free similarity measure which computes the similarity of two websites by. We use the Furthest Point First algorithm to perform phishing prototype extractions, to select instances that are representative of a cluster of phishing webpages. In paper [33], we propose a deep learning-based framework for detecting phishing websites.

III. PROPOSED WORK

The architecture of the proposed model is shown below in figure below:

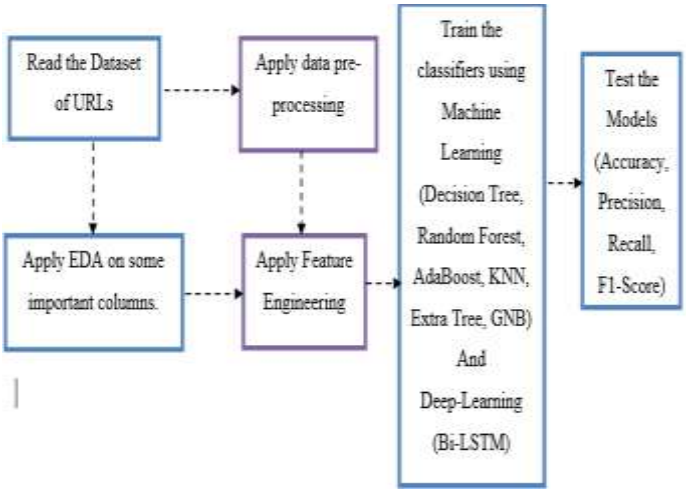


Figure 4.1: Proposed model steps.

Attackers are using AI and other techniques to lure unsuspecting individuals into their information-stealing schemes. Organizations, in turn, must use the capabilities of artificial intelligence and machine learning to detect malicious domains faster than humans alone can. Machine learning finds more patterns of malicious behavior across every threat category, and it does it faster. Protective solutions using machine learning and Deep Learning set organizations on the path to greater security.

The architecture of proposed model using machine learning and deep learning is shown below for malicious URLs detection.

3.1 Data Pre-processing

Following steps are applied for data pre-processing and feature engineering.

- Remove the WWW from URL names and calculate the URL length.
- Extract the primary domain names from URL.
- Count letters, digits and Special characters from the URLs.
- Apply URLs shortening and get the shorter urls.
- Find out the abnormal urls.
- Find out the secure URLs using https.
- Get the URLs having IP addresses?
- Find and replace all NULL values.
- Get the URLs regions.

Some of the common features that malicious URLs have are mentioned below.

1. Malicious URLs don't have hyphens or symbols in their domain name.So in our model, by checking special characters and symbols we can check for this malicious URL. For example www.google.com is not same as www.google-search.com.
2. Not having https in their names.
3. Missing of Legit Contact Information.
4. Websites without this important information are more likely to be fraudulent. Also, a fictional or vague address may signify a phishing site. So the "url-region" property check will help us in detecting malicious Urls.
5. Poor Backlink profile analysis report. A backlink is a URL that leads from one website to another. A website with many backlinks is featured on many other pages, proving its trustworthiness. Getting the root domain will help in backlink analysis of URLs.
6. Counting of Dashes in URL link will help detecting malicious Urls because they have more in numbers as compared to legitimate Urls.
7. Malicious URLs generally have longest domain names. So the URL shortening services will help in detecting these features.
8. The lexical features based on the words that appear in the URLs capture the dynamic nature of the links. The static nature of the links is captured by the descriptive features, which rely on the assumption that the characteristics between legitimate and malicious links rarely varied. For instance, phishing websites sometimes utilized related symbols or letters, such as representing the lower case of letter 'L′ with the digit '1′ to mislead the target legitimate users. Thus, the websites may have certain statistical information, such as the consecutive relationship of digits and alphabets. Using this assumption, some lexical and descriptive features may be extracted from URLs and use to train classification algorithms.

Counting of letters, digits and Special characters from the URLs will help to achieve the problems mentioned above.

3.2 Algorithm: Wide & Deep Model for Phishing URL Detection

Step 1: Data Preparation

1. **Load the preprocessed dataset** containing real-time extracted features from URLs.
2. **Split the data** into:
   - Features (X)
   - Labels (y: phishing = 1, legitimate = 0)
3. Perform a **train-test split** (e.g., 80% training, 20% testing) using stratified sampling.

Step 2: Define Model Inputs

1. Create an Input layer with the shape of the feature vector (X.shape[1]).

Step 3: Build the Deep Component

1. Pass the input through multiple **dense layers** with activation functions like ReLU.
2. This part learns complex, high-dimensional patterns in the feature space.

Step 4: Define the Wide Component

1. The **wide part uses the raw input** features directly (or feature interactions).
2. It acts like a **linear model** (e.g., logistic regression), good for memorizing feature co-occurrences.

Step 5: Combine Wide and Deep

1. Concatenate the outputs of the wide and deep parts using Concatenate().
2. This fuses both memorization and generalization capabilities.

Step 6: Output Layer

1. Add a Dense layer with a single unit and sigmoid activation to perform binary classification.

Step 7: Compile the Model

1. Use:
    - binary_crossentropy loss (for binary classification)
    - adam optimizer
    - accuracy, precision, and recall as metrics

Step 8: Train the Model

1. Fit the model to training data using model.fit() with:
    - Epochs (e.g., 25)
    - Batch size (e.g., 32)
    - validation_split (e.g., 10%)
2. Monitor training and validation accuracy, recall, and loss.

Step 9: Evaluate the Model

1. Predict on test set using model.predict().
2. Convert probabilities to binary class labels using a threshold (default 0.5 or lower like 0.3 for high recall).
3. Generate a **classification report** to evaluate performance.

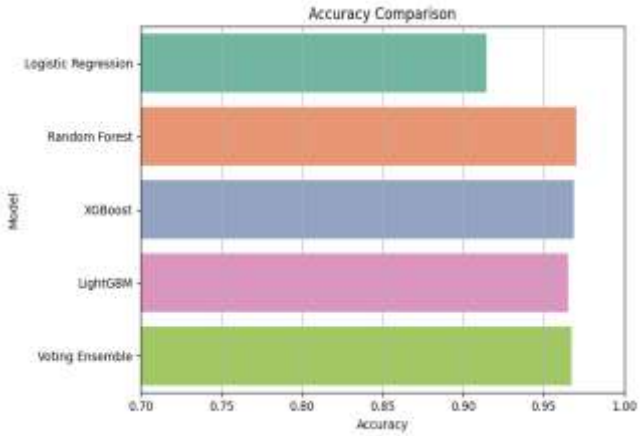
Step 10: Save and Deploy

1. Save the model to disk (model.save('wide_deep_model.h5'))
2. Use it in your real-time phishing URL detection pipeline.

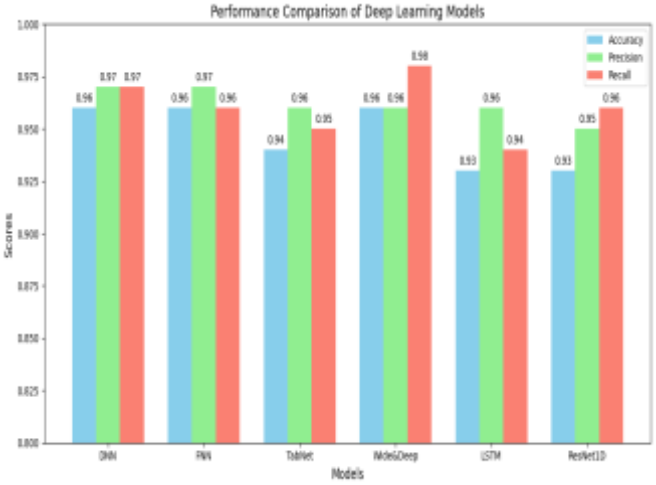
IV. RESULTS

A classification report is a comprehensive summary of

Classifiers used. The comparison of accuracies is shown below in figures.



4.1 Accuracy of existing and proposed models.



4.2 Results Comparison for Deep Learning Models

V. CONCLUSION

Machine Learning algorithms are efficient to do binary classification and to detect the malicious URLs.URL detection using a machine learning model is that it accepts the URL as user input and detects and classifies it as benign or malicious one. Model does binary classification with 99% accuracy. This model can be used in the cyber security domain and to avoid digital attacks by knowing the malicious and benign URLs in prior. Safety measures can be taken if the URL is found malicious.

In the future work, we would like to use different feature selection ensembles, clustering algorithms and feature engineering techniques for the hidden feature generation that helps in improving the detection accuracy of the model. Some of the future suggestions are:

- Need huge quantity of customized, structured training data.

- Needs rigorous training.
- To build ready-to-use machine learning models for detection of Malicious URLs.

VI. REFERENCES

[1] P. Zhang, A. Oest, H. Cho, Z. Sun, R. Johnson, B. Wardman, S. Sarker, A. Kapravelos, T. Bao, R. Wang, Y. Shoshitaishvili, A. Doupe, and G.-J. Ahn, ‘‘Crawl Phish: Large-scale analysis of client-side cloaking techniques in phishing,’’ in Proc. IEEE Symp. Secur. Privacy (SP), May 2021, pp. 1109–1124.

[2] ENISA. (2021). Cybersecurity for SMEs—Challenges and Recommendations. [Online]. Available: https://www.enisa.europa.eu/publications/ enisa-report-cybersecurity-for-smes.

[3] Cisco. (2021). Cyber Security Threat Trends: Phishing, Crypto Top the List. [Online]. Available: https://umbrella.cisco.com/info/2021-cybersecurity-threat-trends-phishing-crypto-top-the-list.

[4] M. Alsharnouby, F. Alaca, and S. Chiasson, ‘‘Why phishing still works: User strategies for combating phishing attacks,’’ Int. J. Hum.-Comput. Stud., vol. 82, pp. 69–82, Oct. 2015.

[5] Anti-Phishing Working Group—APWG. (2022). Phishing Activity Trends Report-1Q. [Online]. Available: https://docs.apwg.org/reports/ apwg_trends_report_q1_2022.pdf.

[6] I. Vayansky and S. Kumar, ‘‘Phishing—Challenges and solutions,’’ Comput. Fraud Secur. vol. 2018, no. 1, pp. 15–20, Jan. 2018.

[7] Y. Cohen, D. Hendler, and A. Rubin, ‘‘Detection of malicious webmail attachments based on propagation patterns,’’ Knowl.-Based Syst., vol. 141, pp. 67–79, Feb. 2018.

[8] J. Yuan, Y. Liu, and L. Yu, ‘‘A novel approach for malicious URL detection based on the joint model,’’ Secur. Commun. Netw. vol. 2021, pp. 1–12, Dec. 2021.

[9] T. Li, G. Kou, and Y. Peng, ‘‘Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods,’’ Inf. Syst., vol. 91, Jul. 2020, Art. no. 101494.

[10] C. D. Xuan, H. Dinh, and T. Victor, ‘‘Malicious URL detection based on machine learning,’’ Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 1, pp. 148–153, 2020.

[11] R. Naresh, A. Gupta, and S. Giri, ‘‘Malicious URL detection system using combined SYM and logistic regression model,’’ Int. J. Adv. Res. Eng. Technol., vol. 11, no. 4, pp. 1–7, 2020.

[12] C. Ding, ‘‘Automatic detection of malicious URLs using fine-tuned classification model,’’ in Proc. 5th Int. Conf. Inf. Sci., Comput. Technol. Transp. (ISCTT), Nov. 2020, pp. 302–320.

[13] D. R. Patil and J. B. Patil, ‘‘Feature-based malicious URL and attack type detection using multi-class classification,’’ Int. J. Inf. Secur., vol. 10, no. 2, pp. 141–162, 2018.

[14] D. K. Mondal, B. C. Singh, H. Hu, S. Biswas, Z. Alom, and M. A. Azim, ‘‘SeizeMaliciousURL: A novel learning approach to detect malicious URLs,’’ J. Inf. Secur. Appl., vol. 62, Nov. 2021, Art. no. 102967.

[15] M. Al-Janabi, E. D. Quincey, and P. Andras, ‘‘Using supervised machine learning algorithms to detect suspicious URLs in online social networks,’’ in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining, Jul. 2017, pp. 1104–1111.

[16] K. Ramesh, M. A. Bennet, J. Veerappan, and P. Renjith, ‘‘Performance metric system for malicious URL data using revised random forest algorithm,’’ in Proc. 5th Int. Conf. Comput. Methodolog. Commun. (ICCMC), Apr. 2021, pp. 1188–1191.

[17] B. Janet and R. J. A. Kumar, ‘‘Malicious URL detection: A comparative study,’’ in Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS), Mar. 2021, pp. 1147–1151.

[18] Y. Kumar and B. Subba, ‘‘A lightweight machine learning based security framework for detecting phishing attacks,’’ in Proc. I

[19] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, ‘‘URLNet: Learning a URL representation with deep learning for malicious URL detection,’’ 2018, arXiv: 1802.03162.

[20] T. T. T. Pham, V. N. Hoang, and T. N. Ha, ‘‘Exploring efficiency of character-level convolution neuron network and long short term memory on malicious URL detection,’’ in Proc. 7th Int. Conf. Netw., Commun. Comput., Dec. 2018, pp. 82–86.

[21] W. Yang, W. Zuo, and B. Cui, ‘‘Detecting malicious URLs via a keyword-based convolutional Gated-Recurrent-Unit neural network,’’ IEEE Access, vol. 7, pp. 29891–29900, 2019.

[22] Y. Liang, Q. Wang, K. Xiong, X. Zheng, Z. Yu, and D. Zeng, ‘‘Robust detection of malicious URLs with self-paced wide & deep learning,’’ IEEE Trans. Depend. Secure Comput., vol. 19, no. 2, pp. 717–730, Mar. 2022.

[23] W. Wang, F. Zhang, X. Luo, and S. Zhang, ‘‘PDRCNN: Precise phishing detection with recurrent convolutional neural networks,’’ Secur. Commun. Netw. vol. 2019, pp. 1–15, Oct. 2019.

[24] N. A. ALfouzan and N. C, ‘‘A systematic approach for malware URL recognition,’’ in Proc. 2nd Int. Conf. Comput. Inf. Technol. (ICCIT), Jan. 2022, pp. 325–329.

[25] K. H. Park, H. M. Song, J. D. Yoo, S.-Y. Hong, B. Cho, K. Kim, and H. K. Kim, ‘‘Unsupervised malicious domain detection with less labeling effort,’’ Comput. Secur. vol. 116, May 2022, Art. No. 102662.

[26] S. Afzal, M. Asim, A. R. Javed, M. O. Beg, and T. Baker, ‘‘URLdeepDetect: A deep learning approach for detecting malicious URLs using semantic vector models,’’ J. Netw. Syst. Manage., vol. 29, no. 3, pp. 1–27, Jul. 2021.

[27] A. S. Rafsanjani, N. BintiKamaruddin, M. Behjati, S. Aslam, A. Sarfaraz and A. Amphawan, "Enhancing Malicious URL Detection: A Novel Framework Leveraging Priority Coefficient and Feature Evaluation," in IEEE Access, vol. 12, pp. 85001-85026, 2024, doi: 10.1109/ACCESS.2024.3412331.

[28] H. Zhao, Z. Chang, W. Wang and X. Zeng, "Malicious Domain Names Detection Algorithm Based on Lexical Analysis and Feature Quantification," in IEEE Access, vol. 7, pp. 128990-128999, 2019, doi: 10.1109/Access.2019.2940554.

[29] A. S. Rafsanjani, N. B. Kamaruddin, H. M. Rusli and M. Dabbagh, "QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework," in *IEEE Access*, vol. 11, pp. 92523-92539, 2023, doi: 10.1109/ACCESS.2023.3291811.

[30] M. Sanchez-Paniagua, E. F. Fernandez, E. Alegre, W. Al-Nabki and V. Gonzalez-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs,"

in *IEEEAccess*, vol. 10, pp. 42949-42960, 2022, doi: 10.1109/Access.2022.3168681.

[31] F. Castano, E. F. Fernañdez, R. Alaiz-Rodríguez and E. Alegre, "PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification," in IEEE Access, vol. 11, pp. 40779-40789, 2023, doi: 10.1109/ACCESS.2023.3268027.

[32] R. W. Purwanto, A. Pal, A. Blair and S. Jha, "PhishSim: Aiding Phishing Website Detection With a Feature-Free Tool," in IEEE Transactions on Information Forensics and Security, vol. 17, pp. 1497-1512, 2022, doi: 10.1109/TIFS.2022.3164212.

[33] L. Tang and Q. H. Mahmoud, "A Deep Learning-Based Framework for Phishing Website Detection," in IEEE Access, vol. 10, pp. 1509-1521, 2022, doi: 10.1109/ACCESS.2021.3137636.