Progresif: Jurnal Ilmiah Komputer
https://ojs.stmik-banjarbaru.ac.id/index.php/progresif/index
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0877

Phishing URL Detection Using Hybrid CNN-BiLSTM Character-Level Deep Learning

DOI: http://dx.doi.org/10.35889/progresif.v22i3.3753

Alrafiful Rahman[1*], Pratiwi Rachmadi[2], Farah Kaylila[3], Sae Khatami[4], Maria Stefani[5]
Data Science, Perbanas Institute, Jakarta, Indonesia
*e-mail *Corresponding:* alrafiful.rahman@perbanas.id

Abstract

Phishing attacks conducted through fraudulent URLs continue to become a significant cybersecurity challenge, demanding detection methods that are both rapid and reliable. This research introduces a hybrid deep learning model that integrates Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) architectures for automated phishing URL detection. The CNN component is utilized to identify lexical and character-level patterns within URLs, whereas BiLSTM captures contextual dependencies from both forward and backward sequence directions. The proposed model was trained using 450,176 labeled URL samples with character-level tokenization, minimizing the need for manual feature extraction. Experimental evaluations demonstrated excellent performance, achieving an accuracy of 99.78%, while precision and recall consistently remained above 99% according to confusion matrix analysis. Furthermore, the trained model was implemented into a Python-based check_link() function that integrates deep learning prediction scores with rule-based analysis to categorize URLs into safe, suspicious, or dangerous classes. Nevertheless, inference testing revealed that the model is sensitive to incomplete URL formats, particularly legitimate domains lacking the www prefix. In general, the CNN-BiLSTM approach proved highly effective for large-scale phishing URL classification, although incorporating additional external features may help reduce false positive predictions.

Keywords: *Phising; URL; Deep learning; Convolutional Neural Network; Bidirectional Long Short-Term Memory.*

1. Introduction

The rapid growth of digital services has made URLs one of the most frequently used access points for online communication, financial transactions, e-commerce, education, and public services, because internet connectivity has expanded access to information, services, communication, and transactions globally [1]. Along with this growth, phishing attacks have become a serious cybersecurity threat because attackers often use fraudulent URLs to imitate legitimate websites and deceive users into submitting sensitive data such as usernames, passwords, personal identities, and financial information [2]. Phishing URL detection is therefore an important research topic because the effectiveness of early URL identification can reduce the risk of credential theft, financial loss, malware distribution, and unauthorized access to digital systems [3]. As phishing techniques continue to evolve through deceptive domain names, shortened links, typosquatting, homograph attacks, and manipulated URL structures, detection systems must be able to adapt to increasingly sophisticated URL spoofing, obfuscation, and structural manipulation patterns [4]. Therefore, phishing URL detection systems must recognize malicious patterns quickly and accurately by extracting URL features, learning lexical and contextual URL characteristics, and classifying URLs into legitimate or malicious categories in real time [5].

In the context of this research, the object of study is URL text data that contains structural and lexical characteristics indicating whether a link is benign or malicious [6]. The main problem is that phishing URLs are highly dynamic and can be easily modified by attackers

to avoid conventional detection mechanism [7]. Blacklist-based systems are limited because they generally depend on previously recorded malicious URLs and are less effective against newly generated or zero-day phishing links [8]. In addition, traditional machine learning approaches often require handcrafted feature extraction, such as URL length, domain age, number of dots, number of special characters, or keyword-based indicators [9]. This dependency on manual features can reduce adaptability when attackers change URL patterns [10]. Therefore, a detection approach that can automatically learn character-level patterns from raw URL sequences is needed to improve robustness and reduce dependence on manual feature engineering [11].

Several previous studies have attempted to address phishing URL detection using machine learning and deep learning approaches. Investigated phishing URL detection using machine learning methods by classifying URLs based on extracted features [12]. Conducted a systematic literature review on phishing website detection techniques and showed that various machine learning and deep learning models have been applied to improve phishing detection accuracy [7]. Proposed PhishNot, a cloud-based machine learning approach for detecting phishing URLs, while developed PhiUSIIL, a phishing URL detection framework based on similarity index and incremental learning [13]. Introduced a character-level convolutional neural network model that learns phishing-related patterns directly from URL strings [14]. Meanwhile, explored a hybrid AI-based phishing website detection approach using LSTM, CNN, and Random Forest ensemble learning, and proposed a deep learning-based phishing detection system [15]. Although these studies have shown promising results, several gaps remain, particularly the need for a model that combines local character pattern extraction, sequential context learning, threshold optimization, and interpretable link-checking output in a single detection framework.

To address these remaining gaps, this study proposes a hybrid deep learning framework that integrates Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) using character-level URL representation [16]. The proposed concept is based on the assumption that phishing indicators are not only reflected in individual URL features, but also in character patterns, lexical structures, and sequential relationships within the URL text [17]. CNN is employed to extract local character patterns such as suspicious symbols, repeated characters, deceptive subdomains, and phishing-related keywords [18], while BiLSTM is used to capture contextual dependencies from both forward and backward URL sequences. This combination is expected to improve detection robustness compared with approaches that rely only on handcrafted features or single deep learning architectures. The state of the art of this research lies in the integration of character-level CNN-BiLSTM, threshold optimization using the Precision-Recall curve, and a Python-based check link() function that combines deep learning prediction scores with rule-based analysis [19]. Therefore, the novelty of this study is not limited to producing high classification accuracy, but also to developing an interpretable phishing URL detection mechanism that classifies links into safe, suspicious, or dangerous categories while providing understandable reasons for the prediction [20].

2. Metodologi

This research was carried out through a series of systematic stages as illustrated in Figure 1. The research flow starts from URL dataset collection, followed by pre-processing, dataset splitting, and model development, model training, threshold optimization, model evaluation, and the production of a Python-based phishing link detection function. This study is limited only to the implementation of the final link checking function in Python and is not continued to web development, mobile application development, or backend API development [21].

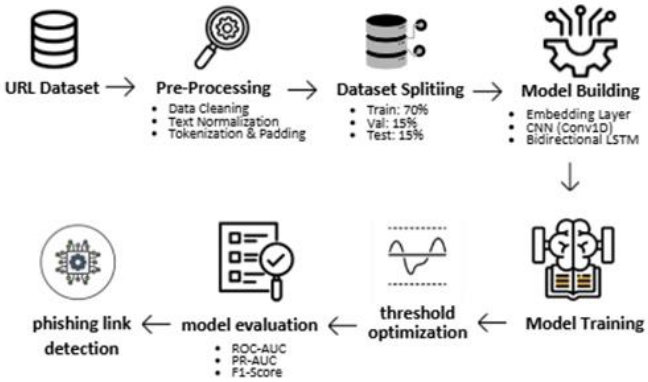

Figure 1. Pipeline of the Proposed CNN-BiLSTM Phishing URL Detection Framework

2.1. Data Collecting and Pre-processing

The dataset used was obtained from a public repository containing a collection of labeled URLs, consisting of the columns url, label, and result. The result column uses binary representation, where value 0 indicates a benign (safe) URL and value 1 indicates a malicious (dangerous) URL. The dataset contains a total of 450,176 URL entries, with a distribution of 345,738 benign URLs and 104,438 malicious URLs. Class imbalance in the dataset was addressed through class weighting during the training stage [22].

Table 1. Dataset Class Imbalance

Label	Meaning	Count	Percentage
0	Benign/Safe	345.738	76,8%
1	Malicious/dangerous	104.438	23,2%

The data preprocessing stage involved several cleaning procedures, including the removal of duplicate URLs, elimination of missing entries in the url and result columns, normalization of URL text into lowercase format, and validation to ensure that labels contained only binary values of 0 and 1. Because URL strings contain distinctive character-level patterns that are highly relevant for phishing detection—such as the presence of the @ symbol, excessive hyphens, and deceptive subdomains imitating popular brand names—this study employed a character-level tokenization approach. To avoid data leakage and maintain model generalization, the tokenizer was fitted exclusively on the training dataset. The tokenization process produced a vocabulary consisting of 142 unique characters, and all URL sequences were standardized through padding to a fixed maximum length of 200 characters (MAX_LEN = 200).

2.2. Dataset Splitting and Model Development

The dataset was stratified into 70% training data, 15% validation data, and 15% testing data, so that the proportion of benign and malicious classes remained balanced in each subset. The data split resulted in 315,123 training data, 67,526 validation data, and 67,527 test data. The developed model architecture is a sequential deep learning model that combines an Embedding Layer, 1D CNN, Bidirectional LSTM, GlobalMaxPooling1D, Dense Layer, Dropout, and sigmoid output [23].

Table 2. Dataset Splitting

Stage	Number of data	Benign	Malicious
Train	315.123	242.016	73.107
Validation	67.526	51.861	15.665
Test	67.527	51.861	15.666

The Embedding Layer converts character indices into 128-dimensional vectors. The 1D CNN layer uses 128 filters, a kernel size of 5, and ReLU activation to extract local features from the URL character sequence. MaxPooling1D with a pool size of 4 is used to reduce feature dimensions, followed by Dropout 0.25. Next, Bidirectional LSTM with 64 units processes the sequence from the forward and backward directions to capture the temporal context relationships of URLs more comprehensively. The model was compiled using the Adam optimizer, Binary Crossentropy loss, and the metrics Accuracy, Precision, Recall, and AUC. To address class imbalance, class weights of 0.651 for the benign class and 2.155 for the malicious class were applied. The total number of trainable parameters in this model is 207,361 [24].

Table 3. Main Formulas Used In The Model

Model	Formula
Convolution (CNN)	$h\,(i) = \ ReLU\,(W_{cnn} \ * \ E\,[i{:}\,i + k - 1] \ + \ b)$
BiLSTM	$\overrightarrow{h_t}, \overleftarrow{h_t}$ = LSTM$\big(h(i)\big)$ $h_t \ = [\overrightarrow{h_t}; \overleftarrow{h_t}]$

2.3. Threshold Tuning, Evaluation, and Final Link Checking

Threshold tuning was performed on the validation set using the Precision-Recall curve to select the threshold that maximizes the F1-score. Final evaluation was conducted on an independent test set using ROC-AUC, PR-AUC, F1-score, Precision, Recall, and Confusion Matrix. After evaluation, the Python function check_link() was developed to combine the deep learning score and rule-based analysis. The final score is calculated with a weight of 70% from the machine learning score and 30% from the rule score. URLs are classified as safe for scores below 45, suspicious for scores between 45 and 74, and dangerous for scores of at least 75. To provide a clearer interpretation of the model performance, the evaluation metric criteria adopted in this study are summarized in Table 4 [25].

Table 4. Benchmark Criteria For Classification Model Performance

Metric	Poor	Fair	Good	Excellent
Accuracy	< 70%	70–80%	80–90%	> 90%
Precision	< 0.70	0.70–0.80	0.80–0.90	> 0.90
Recall	< 0.70	0.70–0.80	0.80–0.90	> 0.90
F1-Score	< 0.70	0.70–0.80	0.80–0.90	> 0.90
ROC-AUC	< 0.60	0.60–0.75	0.75–0.90	> 0.90
PR-AUC	< 0.60	0.60–0.75	0.75–0.90	> 0.90

3. Results

3.1. Threshold Tuning

This study performed threshold tuning on the validation set to find the best probability cutoff between Precision and Recall. The movement of the threshold value from 0.0 to 1.0 triggered opposite responses between the Precision and Recall curves. The equilibrium point was searched using the F1-Score metric as the harmonic mean of both. Through the optimization process, the Best Threshold was found to be 0.5725. At figure 2, the model produced maximum performance on the validation data with precision of 0.9974, recall of 0.9949, and the highest F1-score of 0.9961.

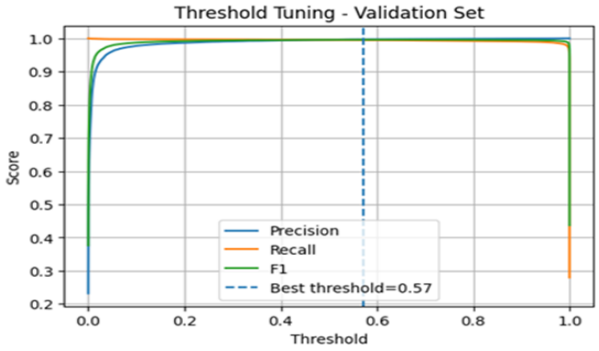

Figure 2. Threshold tuning on the validation set

The decision to shift the threshold from the standard 0.5 to 0.5725 proved effective. This increase tightened the model's tolerance limit for filtering score uncertainty in the middle probability range, without sacrificing the model's ability to capture phishing links. This data-driven approach ensures that the detection system built has a very minimal False Positive error rate when implemented as a link checking function.

3.2. Evaluation Matrix Analysis and Confusion Matrix Data Test

To measure the real performance of the model in real-world scenarios, final testing was conducted using 67,527 independent test data. Based on the classification report and Confusion Matrix, the results were 51,803 True Negative (TN) safe URLs correctly predicted, 15,578 True Positive (TP) phishing URLs successfully detected, 58 False Positive (FP) safe URLs incorrectly detected as phishing, and 88 False Negative (FN) phishing URLs that escaped the model's detection system.

Table 5. Main Formulas Used In The Model

Metric	Excellent
ROC AUC	0,9997
PR AUC	0,9995
F1-Score	0,9953
Recall malicious	0,9944
Precision malicious	0,9963
Accuracy	99,78%

The malicious class recorded a Precision of 0.9963, which means that of all links suspected by the model as phishing, 99.63% were proven valid. On the other hand, the Recall value of 0.9944 proves that the model was able to capture 99.44% of the total attack threats in the dataset. Overall, the model recorded Accuracy of 99.78%, F1-Score of 0.9953, ROC AUC of 0.9997, and PR AUC of 0.9995. The success in suppressing False Positives to only 58 cases indicates that the model is reliable enough for the Python-based link detection prototype stage.

3.3. URL Detection Implementation in Python

After training and evaluation, the CNN-BiLSTM model was implemented only in Python through the check_link() function, which combines deep learning prediction with rule-based analysis to detect suspicious keywords, shortlinks, the @symbol, IP-based hosts, and brand impersonation using official domain matching.

Final score formula for the final link checking function [15]:

$$Final\ Score = (0.7 * ML\ Score) + (0.3 * Rule\ Score) \quad (1)$$

$$Status: safe\ if\ score < 45;$$

$$suspicious\ if\ 45 \leq score < 75;$$

$$dangerous\ if\ score \geq 75.$$

Table 6. Final Link Checking

Sample URL	Final Score	Status	Main Reason
https://www.tokopedia.com	1,27	SAFE	Official TOKOPEDIA domain
https://shopee-hadiah-gratis-login.com	83,83	DANGEROUS	Words login/gift/free and not Shopee official domain
https://bca-update-akun-verifikasi.com	84,02	DANGEROUS	Impersonates BCA and is not an official domain
https://paypal-secure-account-update.xyz	84,39	DANGEROUS	Impersonates PayPal and is not an official domain
https://bit.ly/claim-hadiah-gratis	76,12	DANGEROUS	Shortlink and the words gift/free/claim
wa.me/6281234567890	70,00	SUSPICIOUS	High ML score without strong rule support

The test results prove that the integration of the hybrid CNN-BiLSTM model into the Python function has worked very well and is adaptive in classifying threat levels informatively for users. URLs from official domains such as Tokopedia are classified as safe, while URLs containing patterns such as hadiah, login, verification, account update, or shortlink are classified as dangerous. However, the result on wa.me shows that the model can still assign a high score to certain domains even though rule-based analysis does not find strong indicators, so whitelist mechanisms and external domain reputation are still needed.

3.4 Discussion

The results of this study indicate that the proposed hybrid CNN-BiLSTM model is highly effective for phishing URL detection. The model achieved an accuracy of 99.78%, F1-score of 0.9953, ROC-AUC of 0.9997, and PR-AUC of 0.9995 on the independent test dataset. These results show that the model was able to distinguish benign and malicious URLs with a very low error rate. The confusion matrix also confirms that the model successfully detected most phishing URLs, with 15,578 True Positive cases and only 88 False Negative cases. This finding directly answers the main objective of the study, which is to develop a fast and reliable phishing URL detection model based on character-level deep learning. The threshold optimization at 0.5725 also contributed to a better balance between precision and recall, which is important in phishing detection because both false positives and false negatives can create practical security problems. A false positive may reduce user trust in the detection system, while a false negative may allow dangerous links to reach users.

From the perspective of the scientific concepts underlying this research, the obtained results are consistent with the theoretical strengths of CNN and BiLSTM architectures. CNN is effective in extracting local character patterns from URL sequences, such as unusual symbols, repeated characters, suspicious tokens, deceptive subdomains, and phishing-related keywords. Meanwhile, BiLSTM strengthens the model by learning contextual dependencies from both forward and backward directions in the URL sequence. This is important because phishing indicators are not always located in one fixed position within a URL; they may appear in the domain, subdomain, path, query string, or a combination of several URL components. Therefore, the combination of CNN and BiLSTM enables the model to capture both local lexical features and long-range sequential dependencies. This supports the concept that character-level representation is suitable for phishing URL detection because it allows the model to learn directly from raw URL structures without relying heavily on handcrafted feature engineering [26], [27] .

Compared with previous studies, the findings of this research support the growing evidence that deep learning-based approaches can improve phishing URL detection performance. Previous research on transformer-based URL detection showed that deep learning models can learn complex URL representations and improve phishing detection under strict false positive rate conditions [26]. Another transformer-based study also demonstrated that sequence-based models can outperform conventional detection models in malicious URL classification [27]. In addition, PyraTrans emphasized the importance of combining character-level representation with multi-scale contextual learning to improve malicious URL detection robustness, particularly under class imbalance and adversarial scenarios [28]. The results of the present study are in line with those findings because the proposed CNN-BiLSTM model also learns URL patterns at the character level and produces high classification performance. However, this study provides a different contribution by combining character-level CNN-BiLSTM with threshold tuning and a Python-based link-checking function that produces practical classification categories, namely safe, suspicious, and dangerous.

The contribution of this research to the development of knowledge in cybersecurity lies in demonstrating that phishing URL detection can be improved through the integration of local pattern extraction, sequential context learning, and interpretable rule-based scoring. Many phishing detection studies focus primarily on classification accuracy, while this study extends the model output into a practical link-checking mechanism through the check_link() function. This function combines deep learning prediction scores with rule-based indicators, such as suspicious keywords, shortlinks, the @symbol, IP-based hosts, and brand impersonation patterns. This approach contributes to the development of a more usable phishing early warning mechanism because users are not only given a prediction label, but also receive an explanation of why a URL is considered safe, suspicious, or dangerous. This direction is consistent with recent phishing detection research that emphasizes the importance of explainability and practical usability in real-world detection systems [29].

Despite the strong performance, several limitations should be discussed. First, the implementation is still limited to a Python-based function and has not yet been deployed as a web application, mobile application, browser extension, or backend API. Therefore, its performance in real-time operational environments has not been fully tested. Second, the dataset used in this study contains only binary labels, namely benign and malicious, so the model cannot yet identify more specific phishing categories such as banking phishing, e-commerce scams, fake giveaways, malware APK distribution, delivery scams, or credential harvesting pages. Third, manual testing showed that the model is still sensitive to certain URL structures, such as legitimate domains without the “www” prefix and short domains such as wa.me. This indicates that the model still needs additional external validation mechanisms, such as domain whitelisting, domain reputation checking, WHOIS information, and domain-age features. Recent studies also show that phishing detection models may remain vulnerable to adversarial manipulation, temporal drift, and dataset-related bias, which means that high test accuracy alone is not sufficient to guarantee robustness in real-world deployment [30], [31].

Future research should therefore expand the proposed model in several directions. First, additional features such as WHOIS data, DNS records, SSL certificate information, domain age, hosting information, and external reputation scores can be integrated to reduce false positive predictions and improve robustness against newly generated phishing URLs. Second, the dataset should be expanded by including Indonesian-language phishing URLs and local scam patterns, especially those distributed through messaging platforms and social media. Third, future studies should use time-aware validation and adversarial testing to evaluate whether the model remains stable when phishing tactics change over time. Fourth, the Python-based check_link() function can be developed into a browser extension, mobile application, or API service so that the model can be tested in a real user environment. Finally, future research may compare the proposed CNN-BiLSTM model with transformer-based and multimodal phishing detection approaches to evaluate whether character-level sequential learning, visual webpage analysis, and domain reputation features can complement each other in building a more adaptive phishing detection system [26], [28], [29], [30].

4. Conclusions

This study successfully developed a deep learning-based phishing URL detection model using a hybrid Convolutional Neural Network (CNN) and Bidirectional Long Short-Term

Memory (BiLSTM) architecture with character-level tokenization. The proposed model achieved excellent performance on a dataset of 450,176 URLs, with an Accuracy of 99.78%, F1-Score of 0.9953, ROC AUC of 0.9997, and PR AUC of 0.9995. Threshold optimization at 0.5725 effectively balanced precision and recall for the malicious class. In addition, the Python-based check_link() function successfully integrated deep learning predictions with rule-based analysis, enabling the system to provide not only classification labels but also understandable explanations for users.

The findings indicate that character-based deep learning approaches can serve as a strong foundation for phishing early warning systems, particularly in short-message platforms such as WhatsApp, where suspicious URLs are commonly distributed. From a practical perspective, the integration of rule-based explanations increases transparency and helps users understand why a URL is categorized as dangerous or safe. Academically, this study demonstrates that CNN is effective in extracting local URL patterns, while BiLSTM efficiently captures sequential context for large-scale URL classification.

Despite these promising results, several limitations remain. The current implementation is still limited to a Python-based environment and has not yet been deployed as a web application, mobile application, or browser extension. Furthermore, the dataset only contains binary labels (benign and malicious), preventing the system from identifying specific phishing categories such as banking phishing, e-commerce scams, fake giveaways, malware APK distribution, or delivery scams. Manual testing also revealed sensitivity to certain URL structures, including benign domains without the “www” prefix and short domains such as wa.me. Therefore, future research should incorporate domain whitelisting, community reputation systems, WHOIS and domain-age features, as well as Indonesian-language phishing datasets to improve robustness and real-world applicability.

Daftar Referensi

[1] H. Kibriya, R. Amin, S. S. Alshamrani, S. Rehman, M. Hassan, and F. S. Alsubaei, “Lightweight malicious URL detection using deep learning and large language models,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–18, 2025, doi: 10.1038/s41598-025-26653-2.

[2] M. Elsadig *et al.*, “Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction,” *Electron.*, vol. 11, no. 22, 2022, doi: 10.3390/electronics11223647.

[3] S. S. Roy, A. I. Awad, L. A. Amare, M. T. Erkihun, and M. Anas, “Multimodel Phishing URL Detection Using LSTM, Bidirectional LSTM, and GRU Models,” *Future Internet*, vol. 14, no. 11. p. 340, 2022. doi: 10.3390/fi14110340.

[4] Y. A. Kustiawan and K. I. Ghauth, “PhishOFE: A Novel Machine Learning Framework for Real-Time Phishing URL Detection With Optimized Feature Engineering,” *IEEE Access*, vol. 13, no. August, pp. 169606–169627, 2025, doi: 10.1109/ACCESS.2025.3614126.

[5] S. Gopali, A. Siami Namin, F. Abri, and K. Jones, “The Performance of Sequential Deep Learning Models in Detecting Phishing Websites Using Contextual Features of URLs,” May 2024, pp. 1064–1066. doi: 10.1145/3605098.3636164.

[6] S. Sahoo, S. Kumar, N. Donthu, and A. K. Singh, “Artificial intelligence capabilities, open innovation, and business performance – Empirical insights from multinational B2B companies,” *Ind. Mark. Manag.*, vol. 117, pp. 28–41, 2024, doi: https://doi.org/10.1016/j.indmarman.2023.12.008.

[7] A. Safi and S. Singh, “A systematic literature review on phishing website detection techniques,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 2, pp. 590–611, 2023, doi: 10.1016/j.jksuci.2023.01.004.

[8] M. Aljabri and R. M. A. Mohammad, “Click fraud detection for online advertising using machine learning,” *Egypt. Informatics J.*, vol. 24, no. 2, pp. 341–350, 2023, doi: 10.1016/j.eij.2023.05.006.

[9] D. Dahliana, M. M. Mokhtar, and R. Peramita, “The Role of Islam in the Development of Information and Communication Technology in the Digital Age : A Systematic Literature Review,” vol. 1, no. 2024, pp. 74–82, 2025.

[10] K. M. Alshamrani, D. A. Alzahrani, Y. S. Alghamdi, L. M. Aljohani, and Z. F. Al Nufaiei, “Saudi Radiology Technologists’ Perception of Occupational Hazards from a Personal and Social Lens,” *Risk Manag. Healthc. Policy*, vol. 17, no. October, pp. 2609–2622, 2024, doi: 10.2147/RMHP.S492974.

[11] F. Shirazi, N. Hajli, J. Sims, and F. Lemke, “The role of social factors in purchase journey in the social commerce era,” *Technol. Forecast. Soc. Change*, vol. 183, p. 121861, 2022, doi: https://doi.org/10.1016/j.techfore.2022.121861.
[12] S. K. H. Ahammad *et al.*, “Phishing URL detection using machine learning methods,” *Adv. Eng. Softw.*, vol. 173, p. 103288, 2022, doi: https://doi.org/10.1016/j.advengsoft.2022.103288.
[13] M. Alani, H. Tawfik, M. Saeed, and O. Anya, *Applications of Big Data Analytics: Trends, Issues, and Challenges*. 2018. doi: 10.1007/978-3-319-76472-6.
[14] C. Opara, Y. Chen, and B. Wei, “Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics,” *Expert Syst. Appl.*, vol. 236, no. October 2020, p. 121183, 2024, doi: 10.1016/j.eswa.2023.121183.
[15] S. Kavya and D. Sumathi, “Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection,” *Artif. Intell. Rev.*, vol. 58, no. 2, p. 50, 2024, doi: 10.1007/s10462-024-11055-z.
[16] R. Chinnasamy, M. Subramanian, S. V. Easwaramoorthy, and J. Cho, “Deep learning-driven methods for network-based intrusion detection systems: A systematic review,” *ICT Express*, vol. 11, no. 1, pp. 181–215, 2025, doi: 10.1016/j.icte.2025.01.005.
[17] T. M. Bao, K. Shashvat, N. G. Nhu, and D.-N. Le, “A Comparative Analysis of Machine Learning Algorithms for Spam and Phishing URL Classification,” *Comput. Mater. Contin.*, vol. 87, no. 2, p. 35, 2026, doi: https://doi.org/10.32604/cmc.2025.075161.
[18] Z. Zhang, Y. Yang, N. Lu, W. Shi, and Z. Liu, “Anti-APhish: A Robust and Adaptive Detection Approach Against Evolving AI-powered Phishing URLs,” *Expert Syst. Appl.*, vol. 331, no. Part C, p. 133324, 2026, doi: https://doi.org/10.1016/j.eswa.2026.133324.
[19] J. D. Bhosale, S. S. Thorat, P. V. Pancholi, and P. R. Mutkule, “Machine Learning-Based Algorithms for the Detection of Leaf Disease in Agriculture Crops,” *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. April, pp. 45–50, 2023, doi: 10.17762/ijritcc.v11i5s.6596.
[20] R. Kaur, D. Gabrijelčič, and T. Klobučar, “Artificial intelligence for cybersecurity: Literature review and future research directions,” *Inf. Fusion*, vol. 97, no. April, p. 101804, 2023, doi: 10.1016/j.inffus.2023.101804.
[21] N. Nagy *et al.*, “Phishing URLs Detection Using Sequential and Parallel ML Techniques: Comparative Analysis,” *Sensors*, vol. 23, no. 7, p. 3467, 2023, doi: 10.3390/s23073467.
[22] A. Chakraborty, M. Kumar, and N. Chaurasia, “Secure framework for IoT applications using Deep Learning in fog Computing,” *J. Inf. Secur. Appl.*, vol. 77, p. 103569, 2023, doi: https://doi.org/10.1016/j.jisa.2023.103569.
[23] A. Rahman, L. S. Istiyowati, V. Valentinus, I. Ivan, and Z. Azis, “Implementasi Data Mining Dalam Prediksi Harga Saham BBNI Dengan Pemodelan Matematika Menggunakan Metode LSTM Dengan Optimasi Adam,” *JUTECH J. Educ. Technol.*, vol. 5, no. 2, pp. 427–439, 2024, doi: 10.31932/jutech.v5i2.4137.
[24] A. Rahman, V. Paramarta, A. N. Ida, M. H. Akbar, and V. S. M. Simanjuntak, “Perbandingan Kinerja SVR dan XGBoost untuk Peramalan Emisi CO_2 Global berbasis Machine Learning,” *J. Komtika (Komputasi dan Inform.*, vol. 9, no. 1, pp. 37–44, 2025, doi: 10.31603/komtika.v9i1.13449.
[25] A. Rahman and A. Bustamam, “Deep learning with concatenate model to detect COVID-19 lung disease with CT scan images,” in *AIP Conference Proceedings*, AIP Publishing, 2022. doi: 10.1063/5.0072411.
[26] P. Maneriker, J. W. Stokes, E. G. Lazo, D. Carutasu, F. Tajaddodianfar, and A. Gururajan, “URLTran: Improving Phishing URL Detection Using Transformers,” *Proc. - IEEE Mil. Commun. Conf. MILCOM*, vol. 2021-Novem, pp. 197–204, 2021, doi: 10.1109/MILCOM52596.2021.9653028.
[27] P. Xu, “A Transformer-based Model to Detect Phishing URLs,” 2021, [Online]. Available: http://arxiv.org/abs/2109.02138
[28] R. Liu *et al.*, “PyraTrans: Attention-Enriched Pyramid Transformer for Malicious URL Detection,” pp. 1–12, 2023, [Online]. Available: http://arxiv.org/abs/2312.00508
[29] Y. Lin *et al.*, “Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages,” *Proc. 30th USENIX Secur. Symp.*, pp. 3793–3810, 2021.
[30] A. Kulkarni, V. Balachandran, D. M. Divakaran, and T. Das, “From ML to LLM: Evaluating the Robustness of Phishing Web Page Detection Models against Adversarial Attacks,”

Digit. Threat. Res. Pract., vol. 6, no. 2, pp. 1–26, 2025, doi: 10.1145/3737295.
[31] J. L. Wilk-Jakubowski, L. Pawlik, G. Wilk-Jakubowski, and A. Sikora, “Machine Learning and Neural Networks for Phishing Detection: A Systematic Review (2017–2024),” *Electron.*, vol. 14, no. 18, pp. 1–65, 2025, doi: 10.3390/electronics14183744.