# Phishing Website Detection Using Ensemble Algorithm Convolutional Neural Network and Bidirectional LSTM

**Idi Sumardi[1], Diash Firdaus[2]**
[1] Teknik Informatika, STMK JABAR, Bandung, Indonesia
[2] Informatics, institut Teknologi Nasional, Bandung, Indonesia
Email: Idis@stmikjabar.ac.id[1] , diash@itenas.ac.id[2]

*Received 11 Juli 2024 | Revised 9 september 2024 | Accepted 22 Oktober 2024*

## ABSTRACT

*This study focuses on phishing website detection by leveraging an ensemble of Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) models. Phishing is a type of cyber attack where attackers disguise themselves as legitimate entities to trick individuals into providing sensitive information such as usernames, passwords, and credit card details. Given the escalating threat of phishing attacks and the limitations of traditional detection methods, this research explores the potential of machine learning techniques to enhance detection accuracy and robustness. By integrating CNN and BiLSTM models within an ensemble framework, the study demonstrates improved performance in identifying phishing websites through real-time analysis. The ensemble model benefits from the strengths of both CNN and BiLSTM architectures. CNNs are effective in feature extraction from input data, and capturing spatial hierarchies, while BiLSTMs excel at understanding sequential dependencies. The model achieves an accuracy of 89.5%, with training and validation accuracies converging to high values, and exhibits a consistent decrease in both training and validation losses, indicating robust performance without overfitting.*

## 1. INTRODUCTION

As technological advancements progress at a remarkable pace, the digital environment has become crucial to our everyday activities. This transformation offers numerous advantages; however, it also poses significant challenges, particularly in the realm of cybersecurity. Among the various sophisticated cyber threats that have emerged, phishing attacks are notably prevalent.[1]. In the contemporary digital era, phishing websites pose a substantial threat. These sites are crafted to trick users into revealing sensitive information including passwords, credit card details, and other personal data. The growing complexity of phishing schemes, along with the continual emergence of new phishing sites, highlights the urgent necessity for robust detection mechanisms.[2][3].

Detecting Attacks such as phishing websites, DDoS [4], worms, etc is inherently challenging due to the dynamic and evolving nature of these threats. Traditional detection methods mainly for Phishing, which often rely on blacklists and heuristic-based approaches, struggle to keep pace with the rapid emergence of new phishing techniques. Phishing websites frequently employ obfuscation tactics, such as URL manipulation and content cloaking, to evade these conventional detection systems.

Machine learning (ML) has emerged as a powerful tool in the fight against phishing. By leveraging vast datasets and sophisticated algorithms, ML models can identify patterns and anomalies indicative of phishing activity. However, single-model approaches can be limited by their specific biases and weaknesses, necessitating the exploration of more robust, ensemble-based methods. Ensemble algorithms enhance detection accuracy by combining the strengths of multiple models. This approach mitigates the limitations of individual models, leading to improved performance in identifying phishing websites. Ensemble methods, such as bagging, boosting, and stacking, have shown promise in various domains, making them a suitable candidate for phishing detection[5][6].

Convolutional Neural Networks (CNNs) can be effectively used to extract features from web URLs by transforming the URLs into a suitable numerical representation. CNNs excel at capturing local patterns and hierarchical structures, which can be useful for identifying characteristic patterns in URLs associated with phishing attempts. By treating the URL as a sequence of characters or tokens, CNNs can learn to identify suspicious patterns or components, such as unusual subdomains, path structures, or query parameters. This approach leverages the CNN's ability to perform spatial analysis to detect deceptive URL structures that are often indicative of phishing activities. Bidirectional Long Short-Term Memory (BiLSTM) networks are a type of recurrent neural network (RNN) that processes data in both forward and backward directions. This bidirectional processing capability allows BiLSTMs to capture contextual information more effectively, making them well-suited for analyzing sequential data, such as URLs and textual content of websites. Integrating CNN and BiLSTM models into an ensemble framework leverages the strengths of both architectures. While CNNs excel in extracting spatial features, BiLSTMs are adept at understanding sequential dependencies. This complementary combination can provide a comprehensive analysis of phishing websites, improving detection accuracy and robustness[7].

Current research in phishing detection has explored various ML and deep learning techniques, with notable success in specific contexts [2]. However, the integration of CNNs and BiLSTMs within an ensemble framework remains relatively underexplored. This approach has the potential to advance the state-of-the-art, offering a more holistic solution to phishing detection. For phishing detection systems to be effective, they must operate in real-time, providing immediate protection to users. The latency associated with traditional methods can render them ineffective against fast-moving phishing campaigns. Ensemble algorithms incorporating CNN and BiLSTM models can be optimized for real-time performance, ensuring timely detection and mitigation of phishing threats.

The integration of ensemble algorithms with CNN and BiLSTM networks represents a promising advancement in phishing website detection. By harnessing the strengths of these deep learning models, researchers and practitioners can develop more accurate and resilient detection systems. Future research should focus on optimizing these ensemble frameworks, exploring their deployment in real-world environments, and continuously adapting to the evolving landscape of phishing threats.

## 2. METHODOLOGY

### 2.1 Research Method

Phishing detection is an essential endeavor within the cybersecurity domain, aimed at pinpointing malicious attempts that seek to procure sensitive information via deceptive methods. The outlined flowchart presents a structured approach to both develop and assess a phishing detection model leveraging machine learning techniques. The process encompasses several key steps: data preprocessing, feature extraction, model training, and evaluation. By adhering to this methodical approach, the goal is to improve both the accuracy and reliability of phishing detection systems. Figure 1 illustrates this research methodology for phishing detection using a CNN-BiLSTM architecture.

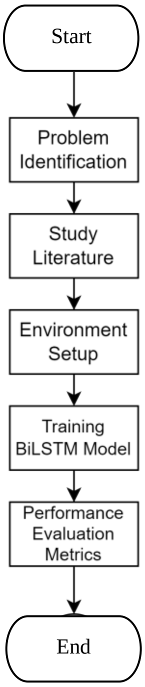


**Figure 1. Research Method Phishing Detection**

1. **Problem Identification**
   In this phase, the particular problem or research question to be addressed is pinpointed. This entails identifying and articulating the primary issues or challenges that the project seeks to resolve or explore.
2. **Study Literature**
   This phase involves conducting a thorough review of the existing literature related to the identified problem. The objective is to compile relevant information, theories, and previous research findings that can establish a foundational understanding of the issue and inform the project's trajectory.
3. **Environment Setup**
   In this step, the requisite environment for executing the project is established. This involves setting up tools, software, hardware, and any other resources essential for conducting the research or development activities.

**Table 1. Environment Setup**

| No | Name | Version |
|---|---|---|
| 1 | Operating System | Windows 11 |
| 2 | Programming Language | Python 11 |
| 3 | Supporting Tools | Library Tensor |
| | | Library Pandas |
| | | Library Numpy |
| | | Library Sckit-learn |

| | | |
|---|---|---|
| 4 | Hardware | CPU AMD Ryzen 7 5800<br>RAM 16 GB |

4. **Trainig**
   This study presents a comprehensive methodology for developing a phishing detection model using a CNN-BiLSTM architecture. The process begins with the input of a phishing dataset, which is then encoded numerically using a label encoder. Stopwords are removed to refine the data, followed by transforming it with a TF-IDF vectorizer for feature extraction. The dataset is split into training and testing sets to evaluate model performance. The CNN-BiLSTM model, a combination of convolutional and bidirectional long short-term memory networks, is employed to capture both spatial and temporal features. Finally, the model undergoes evaluation through various metrics to ensure its effectiveness in detecting phishing attempts. This approach aims to establish a robust and reliable phishing detection system.
5. **Performance Evaluation Metrics**
   The trained model will be evaluated using the testing subset, with its performance measured through evaluation metrics such as accuracy and loss.

### 2.2 Training BiLSTM

Figure 2. shows a flowchart illustrates the methodology for developing a phishing detection model utilizing a CNN-BiLSTM architecture.

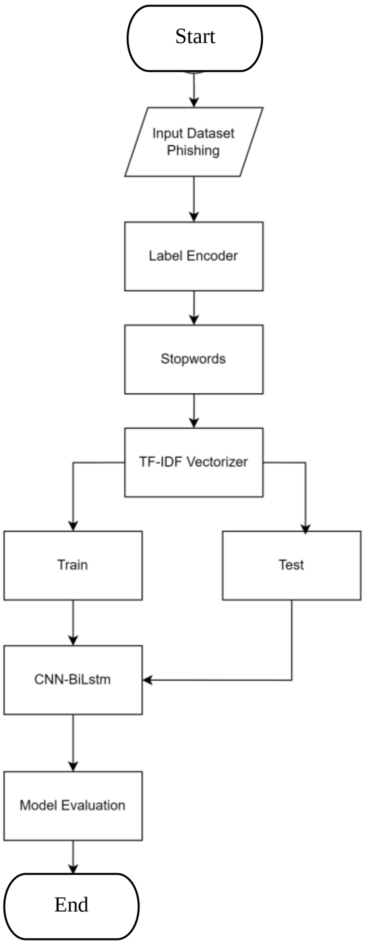


**Figure 2. Developing Method Phishing Detection**

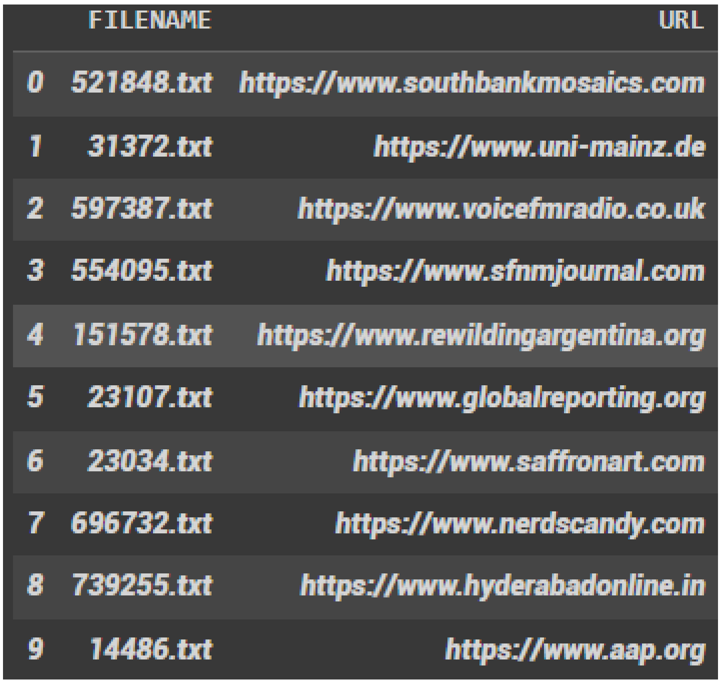
1. **Input Dataset Phishing**

   This step involves collecting and inputting the dataset containing emails or texts classified as phishing or non-phishing. This dataset was developed in 2024 [8]. Table 1: Comparison between Phishing and Non-Phishing Websites in the Dataset. In the dataset, the label for phishing is 0 and the label for non-phishing is 1, which is crucial for correctly interpreting the results and understanding the model's performance.

**Table 2. Comparison Phishing and non-phishing**

| Phising | Non-Phishing |
|---|---|
| 100945 | 134850 |

Figure 3. show Example of the Top 10 Data Points in the Dataset with Non-Phishing Labels



| | FILENAME | URL |
|---|---|---|
| 0 | 521848.txt | https://www.southbankmosaics.com |
| 1 | 31372.txt | https://www.uni-mainz.de |
| 2 | 597387.txt | https://www.voicefmradio.co.uk |
| 3 | 554095.txt | https://www.sfnmjournal.com |
| 4 | 151578.txt | https://www.rewildingargentina.org |
| 5 | 23107.txt | https://www.globalreporting.org |
| 6 | 23034.txt | https://www.saffronart.com |
| 7 | 696732.txt | https://www.nerdscandy.com |
| 8 | 739255.txt | https://www.hyderabadonline.in |
| 9 | 14486.txt | https://www.aap.org |

**Figure 3. Example of Non-Phishing website**

2. **Label Encoder**

   At this stage, the labels or categories of the phishing dataset (e.g., "phishing" and "non-phishing") are converted into numerical form using label encoding techniques. This is crucial for the machine learning algorithm to process the data [9].

3. **Stopwords**

   Stopwords (common words that frequently appear, such as "and", "or", "in") are removed from the text to reduce noise and improve feature quality. Removing stopwords helps focus on more significant words for analysis [10].

4. **TF-IDF Vectorizer**

   This process converts the text into numerical representation using the TF-IDF (Term Frequency-Inverse Document Frequency) technique. TF-IDF measures the importance of a word in a specific document relative to the entire corpus, highlighting more significant words for phishing classification.

5. **Train Test split**
   The dataset is split into training and testing subsets with 80% for training and 20% for testing. The training subset is used to train the phishing detection model. The model learns from the training data and adjusts parameters to minimize errors [11].
6. **CNN-BiLSTM**
   The phishing detection model is developed using a combined architecture of Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM). CNN is used for feature extraction, while BiLSTM captures long-term dependencies in textual data [3]. Figure 4. show the architecture of CNN - BiLSTM from [12]

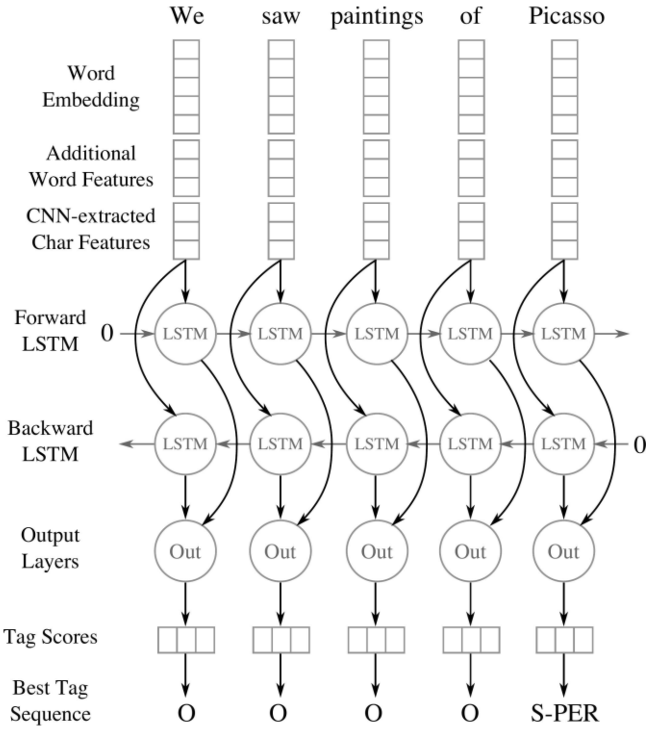


**Figure 4. CNN-BiLSTM Architecture [12]**

7. **Detection Phishing Model**
   The trained phishing detection model is used to predict whether a given text or email is phishing or non-phishing based on the training data.
8. **Model Evaluation**
   The trained model is evaluated using the testing subset. The model's performance is measured using evaluation metrics such as accuracy and loss [13].

## 3. RESULTS AND DISCUSSION

Figure 3 presents the training and validation accuracy of our phishing detection model over a series of epochs. The accuracy graph is a crucial indicator of the model’s performance and its ability to generalize to unseen data. The horizontal axis represents the number of epochs. An epoch is defined as one complete pass through the entire training dataset.

The vertical axis represents the accuracy percentage is 89.5%. Accuracy is a metric that quantifies the proportion of correct predictions made by the model. The red line in the graph signifies the training accuracy. This metric reflects the model’s performance on the training dataset. The training accuracy demonstrates a steady increase from around 87.0% at the initial epoch to about 89.5% at the final epoch, indicating continuous learning and improvement in the model's ability to predict the training data correctly. The validation accuracy, represented by the blue line, is a measure of the model's performance on the validation dataset, which is a separate dataset not used during training. This accuracy starts at around 87.9%, experiences some fluctuations, and then stabilises at around 88.8% in the final epochs. These fluctuations in validation accuracy are common, as they reflect how the model handles a variety of data that has not been seen in the training set.

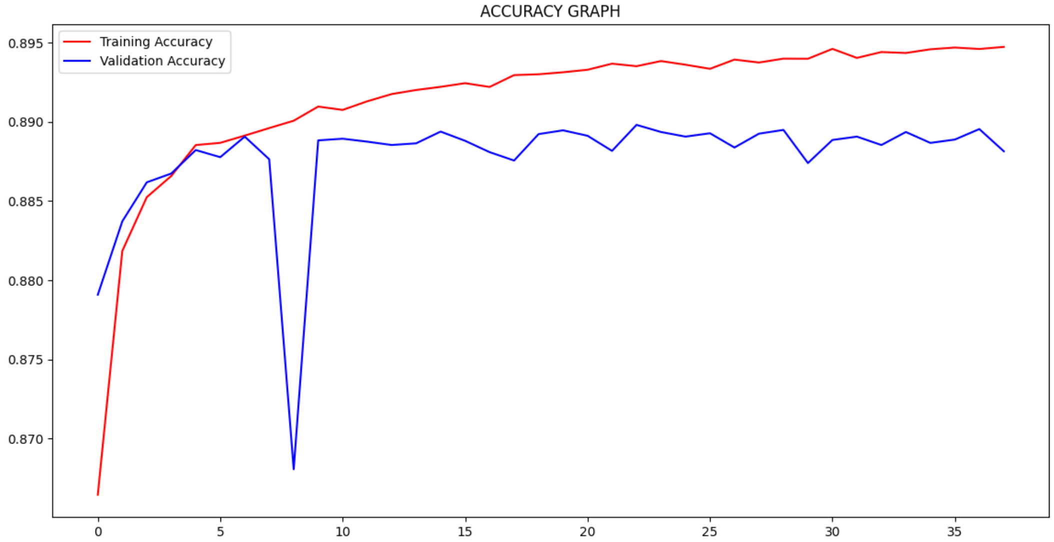


**Figure 5. Accuracy Graph**

Figure 4 presents the loss graph, which tracks the training and validation loss values over a series of epochs during the training process of our phishing detection model. Loss is a crucial metric in machine learning that is used to assess how accurate the model predictions are compared to the actual results. In the graph, the horizontal axis represents the number of epochs, which ranges from 0 to about 35, where each epoch represents one full iteration of the entire training dataset. The vertical axis shows the loss values, which range from 0.27 to 0.33. Lower loss values indicate better model performance, as the predictions are closer to the true labels. The red line on the graph reflects the training loss, which measures the error on the training dataset. The training loss starts from around 0.33 and decreases consistently throughout the epochs, reaching around 0.27. This downward trend indicates that the model is learning and improving its ability to predict the training data. The blue line represents the validation loss, which measures the error on the validation dataset. The validation loss also starts from around 0.33 but shows larger fluctuations compared to the training loss. Overall, the validation loss decreases but still shows variability, especially in the early epochs, before stabilising at around 0.29 in the later epochs.

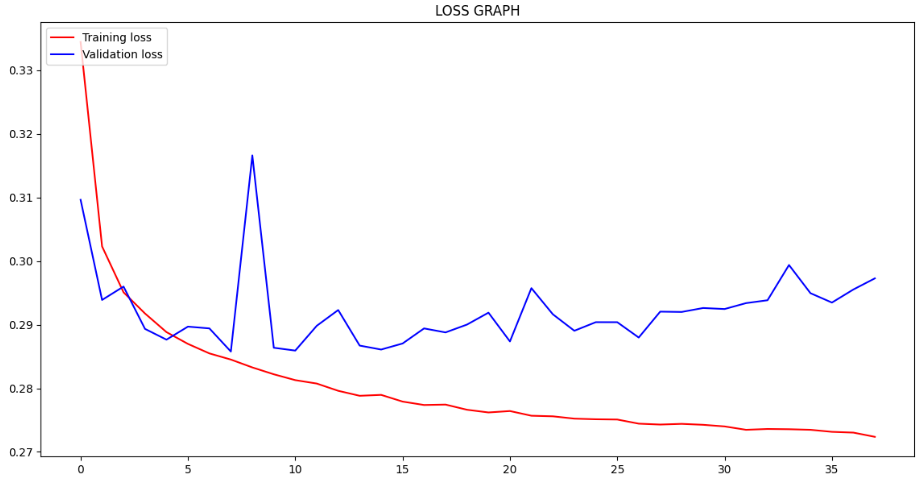


**Figure 6  Loss Graph**

Figure 5 displays the confusion matrix for the phishing detection model, providing a detailed breakdown of the model’s performance in terms of true positives, true negatives, false positives, and false negatives. The confusion matrix is a critical tool in evaluating the classification accuracy and the model's ability to distinguish between classes.

1. **True Negatives (TN)**: The model accurately identified 15,236 instances as phishing, demonstrating its effectiveness in recognizing phishing cases.
2. **False Positives (FP)**: The model incorrectly classified 4,953 phishing instances as non-phishing, indicating a higher rate of false alarms which might be due to overly cautious predictions.
3. **False Negatives (FN)**: The model misclassified 247 non-phishing instances as phishing, which is critical as it represents missed detections of potentially harmful phishing attempts.
4. **True Positives (TP)**: The model successfully identified 26,723 instances as non-phishing, highlighting its strong capability in detecting actual non-phishing cases.

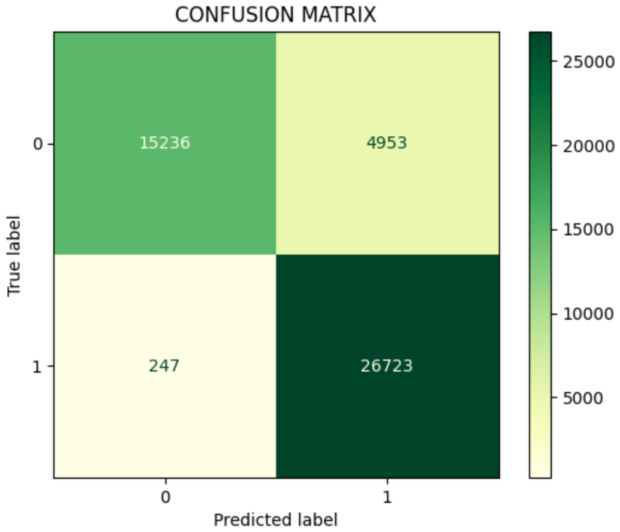


**Figure 7 Confusion Matrix**

## 4. CONCLUSION

In conclusion, this study presents a comprehensive approach to developing and evaluating a phishing detection model using machine learning techniques. The process involved multiple steps, including data preprocessing, feature extraction using TF-IDF vectorization, and model training with a CNN-BiLSTM architecture. The accuracy graph Figure 5 indicates that the model is effectively learning and generalizing well to unseen data, with training and validation accuracies is 88.8%. The loss graph Figure 6 further supports these findings, showing a consistent decrease in both training and validation losses, which suggests that the model is not overfitting and maintains a robust performance. The confusion matrix Figure 7 provides a detailed breakdown of the model's performance in terms of classification accuracy with the number of True Negative 15236 data, False Positive 4953, False Negative 247 and TP 26723. The model demonstrates high accuracy and low loss indicating its strong capability in detecting phishing attempts while minimizing false positives and false negatives. Overall, the results affirm the efficacy of the proposed phishing detection model. Future work could focus on further fine-tuning the model and validating it on additional datasets to enhance its robustness and reliability in practical applications. This study contributes to the ongoing efforts in cybersecurity to develop effective tools for combating phishing attacks, thereby enhancing the security and trustworthiness of online communications.

## ACKNOWLEDGMENTS

We gratefully acknowledge the support and contributions from STMIK Jabar and ITENAS, which were instrumental in the successful completion of this research. The resources, guidance, and collaborative efforts provided by both institutions played a crucial role in facilitating this study. We extend our sincere thanks to the faculty, staff, and students who contributed their time and expertise. Without their support, this research would not have been possible.

## REFERENCES

[1] K. Joshi *et al.*, "Machine-Learning Techniques for Predicting Phishing Attacks in Blockchain Networks : A Comparative Study," 2023.

[2] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A Systematic Review on Deep-Learning-Based Phishing Email Detection," *Electron.*, vol. 12, no. 21, pp. 1–26, 2023, doi: 10.3390/electronics12214545.

[3] N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models," *Sensors*, vol. 24, no. 7, pp. 1–19, 2024, doi: 10.3390/s24072077.

[4] D. Firdaus, R. Munadi, and Y. Purwanto, "DDoS Attack Detection in Software Defined Network using Ensemble K-means++ and Random Forest," *2020 3rd Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2020*, pp. 164–169, 2020, doi: 10.1109/ISRITI51436.2020.9315521.

[5] L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," *Mach. Learn. Knowl. Extr.*, vol. 3, no. 3, pp. 672–694, 2021, doi: 10.3390/make3030034.

[6] B. Ramadhan, D. Firdaus, and A. R. Rafi, "Teknik SMOTE Sebagai Solusi Imbalance Class dalam Model Deteksi Intrusi DDoS dengan Metode PCA-Random Forest," *J. MIND (Multimedia Artif. Intell. Netw. Database)*, vol. 8, no. 1, pp. 52–64, 2023.

[7] T. P. Nguyen, C. T. Yeh, M. Y. Cho, C. L. Chang, and M. J. Chen, "Convolutional neural network bidirectional long short-term memory to online classify the distribution insulator leakage currents," *Electr. Power Syst. Res.*, vol. 208, p. 107923, Jul. 2022, doi: 10.1016/J.EPSR.2022.107923.

[8] A. Prasad and S. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," *Comput. Secur.*, vol. 136, p. 103545, Jan. 2024, doi: 10.1016/J.COSE.2023.103545.

[9] C. Herdian, A. Kamila, and I. G. Agung Musa Budidarma, "Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear

Regresi," *Technol. J. Ilm.*, vol. 15, no. 1, p. 93, 2024, doi: 10.31602/tji.v15i1.13457.

[10] S. Sarica and J. Luo, "Stopwords in technical language processing," *PLoS One*, vol. 16, no. 8 August, pp. 1–13, 2021, doi: 10.1371/journal.pone.0254937.

[11] M. Ilman Aqilaa, D. Firdaus, and N. Naofal, "Identifikasi Serangan Lowrate Distributed Denial Of Services Dalam Jaringan Dengan Menggunakan Algoritma Adaboost," *Simpatik J. Sist. Inf. dan Inform.*, vol. 3, no. 1, pp. 34–41, 2023, doi: 10.31294/simpatik.v3i1.1829.

[12] J. P. C. Chiu and E. Nichols, "Named Entity Recognition with Bidirectional LSTM-CNNs," no. 2003, 2014.

[13] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023, doi: 10.3390/s23135941.