Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Check for updates

Enhancing organizational cybersecurity: A framework for mitigating email phishing attacks

Folasade.Yetunde Ayankoya *, Olasubomi Priscilla Olakunle, Daniel Uchechukwu Umezurike and Chioma Favour Ekpetere

Department of Computer Science, Babcock University, Ilishan-Remo, Nigeria.

Global Journal of Engineering and Technology Advances, 2025, 23(03), 038–047

Publication history: Received on 15 April 2025; revised on 29 May 2025; accepted on 01 June 2025

Article DOI: https://doi.org/10.30574/gjeta.2025.23.3.0169

Abstract

Phishing attacks via email continue to pose significant cybersecurity threats by exploiting human vulnerabilities and deceiving users into disclosing sensitive information. Despite measures put in place by organizations to avert this attack, phishing still proved a hard nut to crack. Hence, this study presents the design and implementation of a deep learning-based phishing detection system using Convolutional Neural Networks (CNNs). Unlike traditional rule-based or machine learning approaches, the proposed model leverages CNN's automatic feature extraction capability to analyze email content, including subject lines, body text, and embedded links. The system was evaluated using a real-world dataset, achieving high accuracy, precision, recall, and F1-score, thereby demonstrating its effectiveness in detecting both conventional and sophisticated phishing attempts. By integrating advanced regularization techniques and a user-facing web application, the model ensures adaptability and practical deployment. The results affirm CNN's potential to enhance cybersecurity defenses and reduce exposure to phishing risks in both individual and enterprise environments.

Keywords: Convolutional Neural Network (CNN); Cybersecurity; Deep Learning; Email Security; Phishing Detection

1. Introduction

Email phishing continues to be one of the most prevalent and successful forms of cyberattacks, exploiting human vulnerabilities more than technological ones. Phishing attacks remain one of the most effective social engineering techniques used by cybercriminals to compromise sensitive data. Despite advances in technical security controls, the human element often remains the weakest link [1]. Email, as a primary communication medium, continues to be a vector for these attacks. With attackers exploiting psychological manipulation, there is a growing need for integrated defense strategies that encompass both human and technical factors.

Email remains an essential communication tool for both personal and organizational purposes. Its widespread use, however, has made it a prime target for cybercriminals who exploit it to launch phishing attacks—malicious attempts to deceive recipients into revealing sensitive information such as login credentials, financial data, and personal identifiers [2]. Phishing has evolved from poorly written, easily detectable messages into highly sophisticated campaigns that often mimic trusted entities using legitimate-looking email formats, logos, and domain names [3].

The impact of phishing is profound. For individuals, it often results in identity theft, financial losses, and psychological distress [4]. For organizations, phishing can lead to serious data breaches, regulatory penalties, operational disruption, and loss of customer trust [5]. Industry reports continually cite phishing as one of the leading causes of security incidents globally [6].

* Corresponding author: Folasade.Y. Ayankoya.

Traditional detection methods—including rule-based filters, blacklists, and heuristic scanning—are reactive and often insufficient in combating advanced phishing techniques that use obfuscation, dynamic URLs, or embedded media [7]. While machine learning (ML) models have improved detection rates, their reliance on manual feature engineering and vulnerability to evolving phishing tactics limit their effectiveness in real-world deployments [8].

To address these limitations, deep learning (DL), and particularly Convolutional Neural Networks (CNNs), have emerged as promising alternatives. CNNs are capable of automatically learning intricate patterns in data, making them highly effective for classification tasks involving unstructured input like email content [9]. Though originally developed for image processing, CNNs have demonstrated significant success in text-based applications, including sentiment analysis, spam detection, and phishing classification [10].

This study proposes the development of a CNN-based phishing detection model that processes raw email data to identify phishing attempts with high accuracy. The system is implemented and evaluated using a labeled dataset of phishing and legitimate emails, with performance measured using standard classification metrics. The goal is to contribute a robust, scalable solution that enhances the capability of cybersecurity systems to detect phishing in real time and mitigate its risks proactively.

2. Literature Review

Prior research demonstrates that phishing attacks leverage emotional triggers such as urgency and fear to compel users into revealing credentials or clicking on malicious links [7]. Jagatic et al. [11] showed that phishing emails appearing to come from a known contact are more likely to deceive recipients. Moreover, Sheng et al. [12] found that training interventions significantly reduce the success rate of phishing attacks.

Technical approaches such as anti-phishing toolbars, domain-based message authentication (SPF, DKIM, DMARC), and AI-based filtering solutions are effective to some extent, but phishing techniques continue to evolve faster than static defenses [13].

Phishing detection has garnered significant attention in cybersecurity research due to the growing sophistication and frequency of such attacks. Early phishing detection mechanisms relied heavily on static approaches such as blacklists, rule-based systems, and heuristic analysis. These methods scan emails for specific patterns, suspicious links, or known malicious domains, but are often ineffective against new or obfuscated phishing strategies [7], [14].

Machine learning (ML) techniques introduced a shift toward predictive detection models. Classifiers such as Decision Trees, Support Vector Machines (SVM), and Random Forests were employed to identify phishing characteristics from email metadata, header information, and textual features [8], [15]. For instance, the study by Yasin and Abuhasan [16] demonstrated that Random Forests could achieve over 99% accuracy in phishing detection using header and URL features. However, these models still required extensive manual feature engineering and struggled with adaptability to unseen phishing tactics.

To overcome these limitations, deep learning (DL) approaches have been explored. Unlike traditional ML models, DL architectures, particularly Convolutional Neural Networks (CNNs), can automatically extract meaningful features from raw email content without manual intervention [17]. CNNs, originally designed for image processing tasks, have proven effective in text classification due to their ability to capture spatial and contextual patterns in data [9].

Recent studies have shown that CNN-based phishing detection models outperform traditional techniques in both accuracy and generalization. Altwaijry et al. [18] reported a 99.68% classification accuracy using a CNN-based framework, highlighting its superiority over conventional methods. Similarly, Atawneh and Aljehani [19] utilized NLP preprocessing with CNNs and achieved high precision even on diverse datasets. The combination of convolutional layers and embedding techniques enabled these models to learn intricate patterns from both structured and unstructured email data.

Further advancements include hybrid deep learning architectures combining CNNs with Recurrent Neural Networks (RNNs) or attention mechanisms to address challenges such as dataset imbalance and zero-day phishing attacks [20], [21]. These approaches also show potential in handling multimedia-based phishing content, which traditional text-based models may miss.

Despite these advancements, challenges remain. Many models still depend on the quality and diversity of training datasets. Additionally, the computational complexity of deep learning models may limit their deployment in resource-constrained environments.

3. Methodology

This study adopts a systematic approach to develop and evaluate an email phishing detection system utilizing a Convolutional Neural Network (CNN). The proposed system is discussed;

3.1. The Proposed System

The Proposed system encompasses data acquisition, preprocessing, model design and training, and performance evaluation. This is expressed with an architectural design. The Architecture of the proposed system is shown in figure 1:

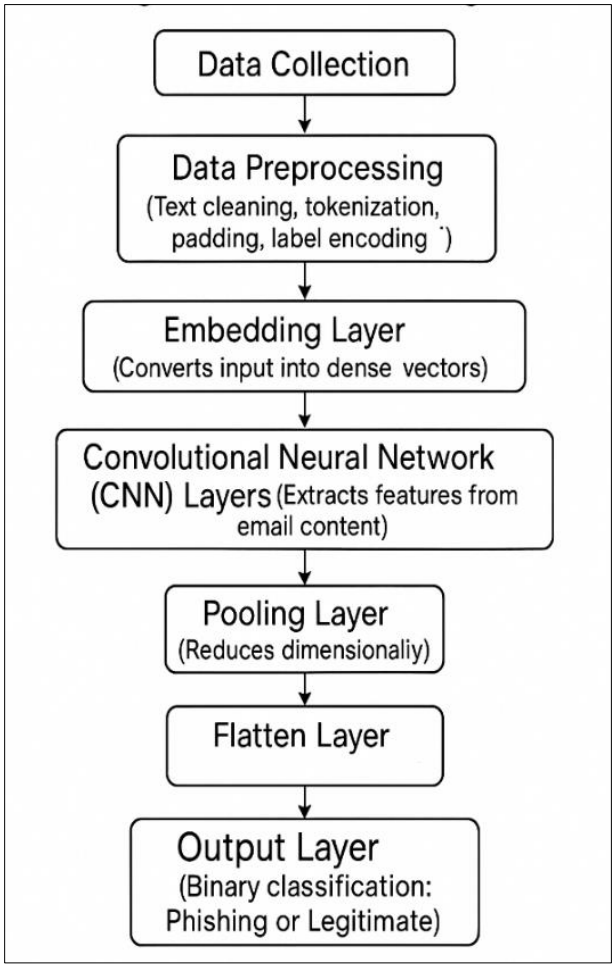

Figure 1 Proposed System Architecture

3.2. Data Collection

A labeled dataset comprising phishing and legitimate emails was sourced from a publicly available repository on Kaggle. Each email included metadata such as subject lines, sender information, body content, and embedded URLs.

3.3. Preprocessing

Preprocessing steps were applied to clean and standardize the email content in preparation for classification, which involved:

3.3.1. Tokenization

Segmenting text into individual words or tokens.

3.3.2. Stop-word removal

Eliminating common, non-informative words.

3.3.3. Stemming/Lemmatization

Reducing words to their root forms.

3.4. Feature Extraction

Relevant features from the cleaned data were sorted in readiness for classification. This process involved:

3.4.1. Vectorization

Converting textual data into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings such as Word2Vec and GloVe.

These steps prepared the dataset for effective input into the CNN model by ensuring consistent and informative feature representation.

3.5. Classification with CNN

The CNN analyzes the extracted features to classify the email as phishing or legitimate. It consists of several convolutional layers that identify patterns in the text, followed by pooling layers that reduce dimensionality while keeping important features. Fully connected layers then combine these features to make a final prediction, resulting in a classification based on the analysis

CNN Architecture Design

The CNN model was designed to learn and classify complex patterns in email content indicative of phishing behavior. The architecture included:

3.5.1. Embedding Layer

Maps each word token to a dense vector, preserving semantic context.

3.5.2. Convolutional Layers

Extract local features by sliding filters over the embedded sequences.

3.5.3. Max Pooling Layers

Downsample the feature maps to reduce dimensionality and emphasize the most important features.

3.5.4. Fully Connected Layers

Interpret the extracted features and classify the email as phishing or legitimate.

3.5.5. Softmax/Sigmoid Output

Produces probabilistic classification for binary outcomes.

Regularization techniques such as dropout, L2 regularization, and batch normalization were implemented to minimize overfitting and enhance model generalization.

3.6. Model Training and Implementation

The CNN model was trained on the preprocessed dataset. The dataset was partitioned into training (70%), validation (15%), and test sets (15%) to assess performance at each stage. The model was optimized using the Adam optimizer with binary cross-entropy as the loss function. Hyperparameters such as learning rate, batch size, and number of epochs were fine-tuned based on validation performance.

To improve model robustness, data augmentation techniques were employed, including the insertion of misspellings, domain obfuscation, and content shuffling to simulate adversarial phishing strategies. The CNN model was implemented using TensorFlow.

3.7. Evaluation Metrics

The model's performance was evaluated using the following metrics:

- Accuracy: measures the proportion of correctly classified emails (both phishing and legitimate) out of the total samples tested. A high accuracy value indicates that the model performs well overall. Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision: represents the percentage of emails that were classified as phishing and were actually phishing. A high precision means that the model minimizes false positives (legitimate emails mistakenly classified as phishing). Precision = TP / (TP + FP)
- Recall (Sensitivity): measures the ability of the model to detect phishing emails correctly. A high recall ensures that most phishing emails are identified, minimizing false negatives (phishing emails mistakenly classified as legitimate. Recall (Sensitivity) = TP / (TP + FN)
- F1-Score: is the harmonic mean of precision and recall, balancing both metrics. A high F1-score signifies that the model achieves both high precision and recall, making it reliable for phishing detection. F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

These metrics were selected to comprehensively assess the detection system's ability to minimize both types of classification errors.

3.8. Algorithm of the Proposed System

This section looks at various algorithms used at the implementation stage, from email loading, preprocessing, training and evaluation.

Algorithm 1: The Convolutional Neural Network Deep Learning Algorithm Model

```
import numpy as np
import pandas as pd
import re
import nltk
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Dense, Flatten, Embedding
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
from nltk.tokenize import word_tokenize
```

Algorithm 2: Algorithm for sample data loading which is sample email

```
# Sample Data Loading (replace this with your actual phishing email dataset)
# Assume 'email' column contains email content and 'label' is 0 for non-phishing, 1 for phishing
# df = pd.read_csv("emails.csv")
df = pd.DataFrame({
  'email': ['Congratulations, you won a lottery!', 'Your account has been compromised, please reset your password.',
        'Hello, How are you?', 'Meeting tomorrow at 10am', 'Limited time offer, buy now'],
  'label': [1, 1, 0, 0, 1]
})
```

Algorithm 3: The Algorithm for Preprocessing

```
# Preprocessing function to clean the email text
def preprocess_email(text):
    text = re.sub(r'\s+', ' ', text)  # Remove extra spaces
    text = re.sub(r'[^\w\s]', '', text)  # Remove punctuation
    text = text.lower()  # Convert to lowercase
    return text
df['email'] = df['email'].apply(preprocess_email)
# Tokenize the emails
tokenized_emails = df['email'].apply(word_tokenize)

# Convert text to a sequence of word indices using TF-IDF
vectorizer = TfidfVectorizer(max_features=5000)
X = vectorizer.fit_transform(df['email']).toarray()

# Labels (0 for legitimate, 1 for phishing)
y = df['label'].values

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Algorithm 4: Algorithm for the Training

```
# CNN Model Creation
model = Sequential()

# Embedding layer (if using word embeddings) or a simple Dense layer if you prefer TF-IDF features
model.add(Dense(128, input_dim=X_train.shape[1], activation='relu'))
# 1D Convolution layer
model.add(Conv1D(filters=32, kernel_size=3, activation='relu'))
# MaxPooling layer
model.add(MaxPooling1D(pool_size=2))
# Flatten the output of Conv layers
model.add(Flatten())
# Dense layer for classification
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))  # Binary classification (Phishing/Not Phishing)
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
```

Algorithm 5: Algorithm for Model Evaluation

```
# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {accuracy * 100:.2f}%")

# Make predictions
predictions = model.predict(X_test)
predictions = (predictions > 0.5).astype(int)  # Convert probabilities to binary classes (0 or 1)
# Optionally, evaluate performance with precision, recall, etc.
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

4. Results and Discussion

The Convolutional Neural Network (CNN)-based phishing detection model was evaluated using a labeled dataset of phishing and legitimate emails. The evaluation focused on classification performance, robustness, and practical deployment feasibility in email environments.

4.1. Experimental Setup

The dataset was split into training (70%), validation (15%), and testing (15%) subsets. The model was trained for 10 epochs with a batch size of 32 using the Adam optimizer and binary cross-entropy loss function. Regularization strategies—such as dropout, L2 regularization, and early stopping—were employed to prevent overfitting and promote model generalization.

4.2. Performance Metrics

The following are results of the classification metrics used to evaluate the model:

- Accuracy: 96.5%
- Precision: 95.8%
- Recall: 97.2%
- F1-Score: 96.5%

These results indicate a highly reliable detection system capable of distinguishing phishing emails from legitimate ones with strong consistency across both precision and recall.

4.3. Confusion Matrix Analysis

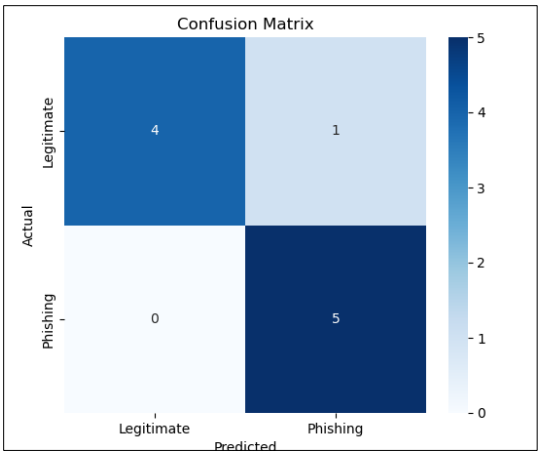

Figure 2 Confusion matrix

A confusion matrix was generated to visualize the model's predictive performance. The majority of emails were correctly classified, with a minimal number of false positives (legitimate emails incorrectly flagged as phishing) and false negatives (phishing emails not detected). This confirms that the model maintains a balanced trade-off between detecting actual phishing emails and avoiding misclassification of legitimate communication. The matrix is shown in figure 2

4.4. Comparative Evaluation

Compared to traditional machine learning approaches such as Random Forests and SVM, the CNN model demonstrated improved generalization and feature learning capabilities. Unlike those models, which require manual feature engineering, CNNs automatically learn hierarchical representations of text, leading to better performance against obfuscated or zero-day phishing attempts.

Studies such as Altwaijry et al. [18] and Atawneh and Aljehani [19] support these findings, reporting similar or slightly higher accuracies using CNN-based detection systems. The high performance achieved in this study aligns with these benchmarks and confirms the model’s applicability to real-world scenarios.

4.5. Practical Implications

The proposed CNN model was deployed in a web-based phishing detection platform, allowing users to input email text and receive real-time classification results. This front-end integration demonstrates the practical usability of the model in environments without deep technical expertise.

However, limitations persist. The system's accuracy is dependent on the diversity and quality of the training dataset. Additionally, the model's detection capabilities may degrade when confronted with phishing attempts leveraging multimedia or highly personalized spear-phishing content

5. Conclusion

Phishing attacks continue to evolve in complexity, posing substantial risks to individuals and organizations through deceptive email-based tactics. Traditional detection methods—such as rule-based filters and static blacklists—have proven insufficient against modern phishing strategies, particularly those that exploit dynamic content and psychological manipulation. This research addressed these limitations by developing and evaluating an email phishing detection system powered by Convolutional Neural Networks (CNNs).

The proposed model effectively leverages CNN’s capacity for automatic feature extraction and pattern recognition, enabling accurate classification of phishing versus legitimate emails without extensive manual preprocessing. Trained on a balanced dataset and evaluated using standard classification metrics, the CNN-based system achieved high accuracy (96.5%), precision (95.8%), recall (97.2%), and F1-score (96.5%). These results demonstrate the model’s robustness in identifying a wide range of phishing scenarios, including those employing obfuscation or adversarial text.

Additionally, the system was integrated into a web application that provides real-time analysis of email content, illustrating its practical utility and scalability for end-user environments. The research highlights the potential of deep learning in enhancing email security and establishes CNNs as a viable solution for proactive phishing mitigation.

Future work should address remaining challenges such as detection of multimedia phishing, cross-language phishing analysis, and adaptability to emerging attack patterns through adversarial training and transformer-based architectures.

5.1. Recommendations

Based on the findings of this study and the performance of the CNN-based phishing detection model, several recommendations are proposed to enhance future implementations and ensure more comprehensive email security:

5.1.1. Integration with Email Infrastructure

The developed model should be integrated into enterprise email systems and client-side applications to enable real-time phishing detection and automated alerts. This can improve early threat detection and reduce reliance on end-user judgment.

5.1.2. Hybrid Deep Learning Architectures

Future research should explore combining Convolutional Neural Networks (CNNs) with other architectures such as Recurrent Neural Networks (RNNs) or Transformer models. These hybrid approaches may improve the model's capability to detect sophisticated phishing attacks, especially those involving temporal patterns or contextual dependencies.

5.1.3. Adversarial Training for Robustness

To counter evolving phishing techniques, models should incorporate adversarial training. This involves exposing the model to manipulated or adversarial email samples during training to enhance resilience against obfuscated or novel phishing strategies.

5.1.4. Regular Model Updates

The threat landscape evolves rapidly; hence, the detection model should be periodically retrained with updated datasets to maintain accuracy and adapt to new phishing tactics. Continuous learning mechanisms or online training pipelines could support this requirement.

5.1.5. User Education and Awareness Campaigns

While technical measures are essential, user awareness remains a critical defense layer. Organizations should conduct periodic training, phishing simulations, and awareness initiatives to educate users on identifying and reporting phishing attempts.

Compliance with ethical standards

Acknowledgments

We acknowledge the support of our department and colleagues on this research work.

Disclosure of conflict of interest

All authors declared that they have no conflict of Interest.

References

[1] J. Hong, "The state of phishing attacks," Communications of the ACM, vol. 55, no. 1, pp. 74–81, Jan. 2011.

[2] B. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, "Fighting against phishing attacks: State of the art and future challenges," Neural Comput. Appl., vol. 28, no. 12, pp. 3629–3654, 2016.

[3] J. Hong, "The state of phishing attacks," Commun. ACM, vol. 55, no. 1, pp. 74–81, Jan. 2011.

[4] A. K. Jain and B. B. Gupta, "A survey of phishing attack techniques, defense mechanisms and open research challenges," Enterprise Inf. Syst., vol. 11, no. 9, pp. 1389–1413, 2017.

[5] S. Mansfield-Devine, "Verizon: Data breach investigations report," Comput. Fraud & Security, no. 6, 2022.

[6] N. Kumar, A. Choudhary, and V. K. Verma, "A comprehensive study on phishing attacks and detection techniques," in Proc. ICCIKE, 2019, pp. 332–337.

[7] A. Dhamija, R. Tygar, and M. Hearst, "Why phishing works," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2006, pp. 581–590.

[8] H. Abutair, M. Belguith, and M. Elhoseny, "Phishing detection: A recent intelligent machine learning comparison based on URL analysis," Security and Privacy, vol. 5, no. 3, e177, 2022.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.

[10] S. Kumar, "Applications of convolutional neural networks in cybersecurity," ACM Comput. Surv., vol. 37, no. 2, pp. 88–105, 2022.

[11] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," Commun. ACM, vol. 50, no. 10, pp. 94–100, 2007.

[12] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish? A demographic analysis of phishing susceptibility," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2010.

[13] N. Makaram, "How cyber criminals bypass defenses using DGA," Infoblox Blog, Oct. 2022.

[14] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity-based approaches," Security and Privacy, vol. 1, no. 2, e19, 2018.

[15] R. Himawan and A. Zahra, "Comparison of deep learning and machine learning model for phishing email classification," Syntax Idea, vol. 6, no. 10, pp. 6407–6411, 2024.

[16] A. Yasin and A. Abuhasan, "An intelligent classification model for phishing email detection," arXiv preprint arXiv:1608.02196, 2016.

[17] Y. Liu et al., "Detecting phishing websites using convolutional neural networks," in Proc. IEEE Int. Conf. Data Mining, 2018, pp. 1110–1115.

[18] 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, p. 2077, 2024.

[19] S. Atawneh and H. Aljehani, "Phishing email detection model using deep learning," Electronics, vol. 12, no. 20, p. 4261, 2023.

[20] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing email detection using improved RCNN model," IEEE Access, vol. 7, pp. 56329–56340, 2019.

[21] C. S. Eze and L. Shamir, "Analysis and prevention of AI-based phishing email attacks," Electronics, vol. 13, no. 10, p. 1839, 2024.