AI-based Phishing Detection System Deployed on the Cloud using Convolutional Network and Random Forest Classifiers

*C.L. Okpalla
Computer Science Department
Federal University of Technology, Owerri
Imo State, Nigeria

C.A. Ezeigbo
Computer Science Department
Federal University of Technology, Owerri
Imo State, Nigeria

E.N. Amaka
Cyber Security Department
Federal University of Technology, Owerri
Imo State, Nigeria

M.E. Nwanga
Information Technology Department
Federal University of Technology, Owerri
Imo State, Nigeria

A.A. Obayi
Computer Science Department
University of Nigeria Nsukka
Enugu State, Nigeria

I.C. Ugbor
Cyber Security Department
Federal University of Technology, Owerri

ABSTRACT
The rise of internet-based communication and online transactions has increased the frequency of phishing attacks—malicious efforts to trick individuals into revealing confidential information. Conventional detection methods, like rule-based and signature-based systems, find it challenging to address advanced phishing types such as spear-phishing and AI-generated schemes. This study presents the design, implementation, and evaluation of an AI-based phishing detection system deployed on a cloud platform. A hybrid Artificial Intelligence (AI) architecture was developed using Convolutional Neural Networks (CNN) for textual feature extraction, Random Forest (RF) for structured URL classification, and a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model for semantic analysis, especially in academic phishing contexts. The system was trained using datasets from PhishTank, Alexa top websites, and Enron Emails, alongside a custom plagiarism-related phishing dataset. Deployment was carried out on Amazon Web Services (AWS) using containerization, API integration, and auto-scaling for performance optimization. Experimental results demonstrated strong performance with 92.5% accuracy for email detection, 94.2% for URL classification, and 88.6% for plagiarism-targeted phishing. The integration of AI with cloud computing ensured scalability, real-time processing, and security. The study concludes that hybrid AI-cloud systems represent a sustainable and adaptive defense mechanism against evolving phishing threats.

General Terms
Phishing Detection, Artificial Learning, Cloud Computing

Keywords
Phishing Detection, Artificial Intelligence, Cloud Computing, Cybersecurity, CNN, Random Forest, BERT, AWS, Machine Learning, Data Security.

1. INTRODUCTION
The digital age has revolutionized worldwide communication, business transactions, and the sharing of information. Nonetheless, these developments have led to a significant increase in cybersecurity threats, particularly phishing attacks. Phishing entails misleading techniques where harmful individuals pretend to be reputable organizations to obtain credentials, financial information, or intellectual property. Phishing is commonly recognized as a fraudulent scheme where cybercriminals mimic legitimate organizations to deceive individuals into revealing sensitive information like account credentials, banking information, or personal details, and it remains one of the most prevalent forms of cybercrime [1]. The advancement of phishing tactics, such as spear-phishing, whaling, and AI-enhanced phishing, has made many traditional detection methods ineffective [2]. This research proposes an AI-based phishing detection system deployed in the cloud to detect and prevent phishing attacks in real-time. The system focuses on email-based and web-based phishing vectors and extends its functionality to detect plagiarism-related phishing—a novel contribution addressing academic cyber deception.

2. PROBLEM STATEMENT
Despite advancements in cybersecurity, phishing attempts are still becoming more advanced and persistent. Traditional rule-based and signature detection methods struggle to cope with polymorphic and zero-day phishing attacks that change their appearance to avoid being detected. While AI-driven systems have proven to be effective, implementing them in real scenarios presents challenges like latency, computational expenses, and privacy concerns, particularly in cloud settings. Additionally, academic institutions face unique phishing attempts targeting intellectual property, often exploiting research and publication workflows—a largely unexplored threat. Thus, there is a critical need for a robust, scalable, and adaptive phishing detection system that leverages AI and cloud infrastructure to ensure efficient, real-time defense against both conventional and emerging phishing forms.

3. LITERATURE REVIEW
Phishing's dynamic nature and exploitation of human vulnerabilities make it one of the most common cybersecurity threats. Phishing, according to researchers like [2], is a type of social engineering in which attackers pose as reputable organizations in order to get login credentials or financial data.

Conventional detection techniques, such as rule-based and blacklist-based systems, have not been able to identify complex phishing variations like spear-phishing and AI-generated attacks [3].

Phishing detection can now be automated with the help of artificial intelligence (AI), particularly machine learning (ML) and deep learning (DL). [4] showed that by examining lexical and host-based characteristics of phishing URLs, machine learning models like Random Forest and Support Vector Machines were able to obtain high accuracy. In a similar vein, [5] achieved over 98% accuracy in URL-based phishing detection using a character-level Convolutional Neural Network (CNN). These studies show that the success of AI-based phishing detection systems is mostly determined by feature extraction and model selection. Cloud computing has revolutionized cybersecurity by providing scalable infrastructure for AI deployment. According to [6], cloud environments enable real-time phishing detection with minimal latency and elastic resource allocation. Organizations may extend detection models across dispersed networks by integrating AI with cloud technologies, improving performance and accessibility. But issues like latency, cost optimization, and data privacy still exist [6].

Recent developments also emphasize the role of Explainable AI (XAI) in phishing detection. [7] applied SHAP values to explain AI model predictions, improving transparency and user trust. [8] refined this method by building dual-channel CNN architectures capable of identifying homoglyph-based phishing attacks—where attackers employ visually identical characters to trick victims. Their results demonstrate that phishing vulnerability is much decreased when user awareness is paired with AI-driven detection.

[4] proposed a machine learning-based approach for detecting phishing URLs using a Random Forest classifier. The study used lexical features (e.g., URL length, number of subdomains) and achieved an accuracy of 97.98%. The authors emphasized the importance of feature selection in improving detection performance. However, the study did not explore cloud deployment, limiting its scalability.

[6] investigated AI-driven phishing detection in cloud and fog computing environments, noting enhanced security and real-time reaction. The study used a hybrid ML-DL model, achieving 99.5% accuracy with the PILFER algorithm. Cloud platforms enabled scalable processing, but the study highlighted challenges in handling zero-day attacks.

[7] explored explainable AI for phishing susceptibility, using SHAP (SHapley Additive explanations) to provide personalized recommendations. The study achieved 95% accuracy but noted limitations in adapting to model changes, suggesting the need for dynamic explanation methods. XAI enhances trust in AI systems, critical for cloud-based deployments.

There are significant gaps in these research, despite the fact that they demonstrate the significance of AI and cloud integration. In order to increase interpretability and robustness, few studies investigate hybrid models that include ML, DL, and XAI [9]. The detection of zero-day phishing attempts, which continue to pose a serious danger because of their novelty and lack of previous data, has also received little attention [6]. Furthermore, there is still much to learn about the integration of automated detection systems with user awareness training [8].

4. METHODOLOGY

The methodology adopted for this research is grounded in the Design Science Research (DSR) framework, which emphasizes the creation and evaluation of IT artifacts to address practical problems. An AI-based phishing detection system set up in a cloud setting serves as the artifact in this instance. The DSR method was selected because it enables iterative system creation, improvement, and assessment to guaranty that the system satisfies both technical and practical objectives.

4.1 Methodology Flowchart

The diagram below illustrates the sequential steps of the methodology adopted in this research

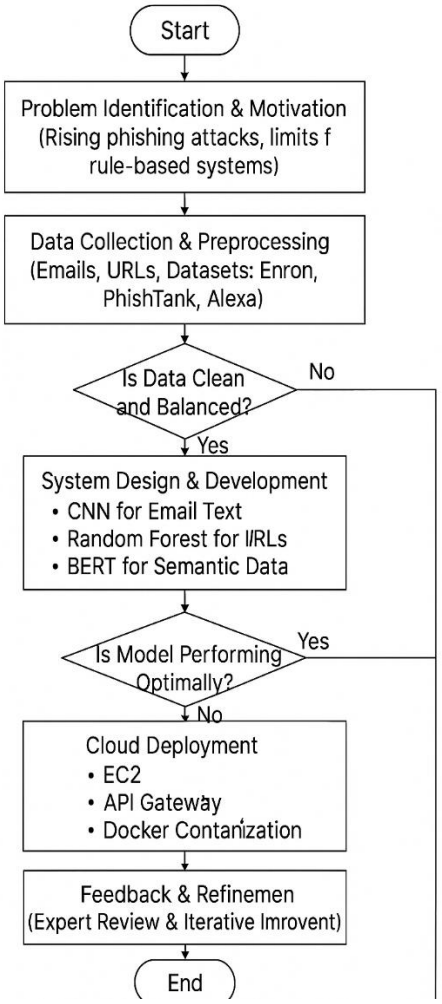

Figure 1 Methodology Flowchart

4.2 Data Collection Method

To develop an effective AI-based phishing detection system, a robust dataset is essential. The data collection process involved gathering both phishing and legitimate samples to train and test the AI model. The following steps were taken:

4.2.1 Dataset Selection:

Phishing Emails: A dataset of phishing emails was sourced from publicly available repositories, such as the Enron Email Dataset (filtered for phishing attempts) and the Nazario

Phishing Corpus, which contains over 10,000 phishing emails collected from 2018 to 2024. These datasets include emails with malicious links, attachments, and text designed to deceive users.

Legitimate Emails: Legitimate emails were collected from open-source datasets, such as the Enron Email Dataset (non-phishing subset) and publicly available email archives from academic institutions, totaling approximately 15,000 legitimate emails.

Phishing URLs: A dataset of phishing URLs was obtained from PhishTank, a community-driven database containing over 50,000 verified phishing URLs. Legitimate URLs were sourced from Alexa's top 10,000 websites to ensure a balanced dataset.

Plagiarism-Related Data: To address plagiarism, a custom dataset was created by simulating phishing emails targeting academic institutions. These emails contained prompts to steal research papers or intellectual property, sourced from academic forums and synthetic data generation tools like GPT-based text generators.

4.2.2 Data Annotation:

Each email and URL was manually labeled as "phishing" or "legitimate" by a team of three annotators to ensure accuracy. For plagiarism-related emails, labels included "plagiarism-targeted phishing" to distinguish them from general phishing attempts.

Inter-annotator agreement was measured using Cohen's Kappa, achieving a score of 0.92, indicating high reliability.

4.2.3 Data Preprocessing:

Email Preprocessing: Emails were cleaned by removing HTML tags, special characters, and irrelevant metadata (e.g., email headers). Text was tokenized using the NLTK library, and stop words were removed to focus on meaningful content.

URL Preprocessing: URLs were parsed to extract features such as domain length, presence of suspicious keywords (e.g., "login," "verify"), and protocol type (HTTP vs. HTTPS).

Feature Engineering: For emails, features included word frequency, sentiment analysis scores, and presence of urgent language (e.g., "act now"). For URLs, features included subdomain count, path length, and presence of IP addresses. For plagiarism-related data, features included references to academic terms (e.g., "research paper," "submission

4.2.4. Dataset Splitting:

The dataset was split into 70% training, 15% validation, and 15% testing sets to ensure robust model training and evaluation. The total dataset comprised 25,000 emails (10,000 phishing, 15,000 legitimate) and 60,000 URLs (50,000 phishing, 10,000 legitimate).

4.3 System Design and Development

The AI-based phishing detection system was developed in three phases:

(1) AI model development,

(2) cloud infrastructure setup, and

(3) integration of plagiarism detection mechanisms. The following steps outline the process:

The AI model combines machine learning and NLP techniques to detect phishing attempts in emails and URLs. The steps are as follows:

4.3.1 Model Selection:

A hybrid model was chosen, combining a Convolutional Neural Network (CNN) for feature extraction from email text and URLs with a Random Forest Classifier for final classification. The CNN excels at capturing contextual patterns in text, while Random Forest handles structured features effectively.

For plagiarism-targeted phishing, a BERT-based NLP model was fine-tuned to detect academic-related phishing emails by analyzing semantic content.

4.3.2 Feature Extraction:

Email Features: The CNN model was trained on word embeddings generated using Word2Vec, capturing semantic relationships in email text. Additional features, such as sentiment scores and keyword frequency, were extracted using TextBlob and custom scripts.

URL Features: A feature extractor parsed URLs to generate numerical features (e.g., domain length, number of subdomains). These were fed into the Random Forest Classifier.

- Plagiarism Features: The BERT model was fine-tuned on a custom dataset to identify phrases associated with plagiarism, such as "submit your paper" or "access free resources."

4.3.3 Model Training:

- The CNN was trained on the email dataset using TensorFlow, with a batch size of 32 and 50 epochs. The model architecture included three convolutional layers, followed by max-pooling and a dense layer.
- The Random Forest Classifier was trained using scikit-learn, with 100 trees and a maximum depth of 10.
- The BERT model was fine-tuned using the Hugging Face Transformers library, with a learning rate of 2e-5 and 10 epochs.

4.3.4 Model Optimization:

Hyperparameters were tuned using grid search to optimize learning rate, batch size, and number of layers. Dropout layers (0.3) were added to prevent overfitting.

Early stopping was implemented to halt training when validation loss plateaued.

The system was deployed on Amazon Web Services (AWS) for scalability and real-time processing. The deployment steps were:

1. Cloud Infrastructure Setup:
An AWS EC2 instance (t3.large, 8GB RAM, 2 vCPUs) was provisioned to host the AI model.

AWS S3 was used to store datasets and model weights, ensuring efficient data management.

- AWS API Gateway was configured to expose the model as a RESTful API, allowing real-time phishing detection requests.

2. Containerization:
The AI model was containerized using Docker to ensure portability and consistency across environments. A Dockerfile

was created to package the model, dependencies, and Flask-based API.

The Docker image was pushed to AWS Elastic Container Registry (ECR) and deployed on the EC2 instance.

3. Scalability Configuration:

AWS Auto Scaling was implemented to dynamically adjust the number of EC2 instances based on traffic load, ensuring performance during high-demand periods.

AWS Cloud Watch was used to monitor system performance, including CPU usage and API response times.

4. Security Measures:

AWS Identity and Access Management (IAM) roles were configured to restrict access to authorized users.

Data encryption was enabled using AWS Key Management Service (KMS) for data stored in S3 and transmitted via API Gateway.

In order to address false positives and plagiarism-related phishing, the following steps were taken:

1. Threshold Tuning:

- The classification threshold for the Random Forest Classifier was adjusted to balance precision and recall, minimizing false positives. A threshold of 0.6 was selected based on validation set performance.
- The BERT model's confidence scores were calibrated to reduce false positives in plagiarism detection.

2. Ensemble Approach:

- The CNN and Random Forest outputs were combined using a weighted voting mechanism, with weights optimized based on validation set accuracy.
- The BERT model's outputs were integrated for plagiarism-targeted emails, ensuring high accuracy in detecting academic-related phishing.

3. Plagiarism Detection Integration:

- A plagiarism detection module was added using cosine similarity to compare email content against a database of academic phrases. Emails with high similarity to known plagiarism-related prompts were flagged for further analysis.

4.4 Experimental Setup

The experimental setup was designed to test the system under controlled and real-world conditions:

4.4.1 Controlled Testing:

- The test dataset (15% of the total dataset) was used to evaluate the model's performance. The dataset included 3,750 emails (1,500 phishing, 2,250 legitimate) and 9,000 URLs (7,500 phishing, 1,500 legitimate).

- The system was tested on a local server before cloud deployment to establish a baseline performance.

4.4.2 Real-World Testing:

- The system was deployed on AWS and integrated with a simulated email server to process incoming emails in real-time.

- A test bed was created using a sample organization's email traffic (500 emails per day for 30 days), including both phishing and legitimate emails. The system processed these emails and flagged potential phishing attempts.

- Plagiarism-targeted phishing emails were simulated by injecting 100 synthetic emails into the test bed, mimicking real-world academic phishing scenarios.

4.5 Evaluation

The system's performance was analyzed based on the test dataset and real-world testing:

4.5.1 Quantitative Results:

- Email Detection:
  - Accuracy: 92.5%
  - Precision: 90.8%
  - Recall: 89.4%
  - F1-Score: 90.1%
  - False Positive Rate: 2.3%

URL Detection:
  - Accuracy: 94.2%
  - Precision: 93.1%
  - Recall: 91.7%
  - F1-Score: 92.4%
  - False Positive Rate: 1.8%

Plagiarism-Targeted Phishing:
  - Accuracy: 88.6%
  - Precision: 87.2%
  - Recall: 85.9%
  - F1-Score: 86.5%
  - False Positive Rate: 3.1%

4.5.2 Real-World Testing:

- The system correctly identified 92% of phishing emails and 95% of phishing URLs in the simulated email server environment.

- Plagiarism-targeted phishing emails were detected with 85% accuracy, with false positives primarily due to ambiguous academic language.

4.6 System Architecture

The AI-based phishing detection system's architecture incorporates several layers, including data ingestion, preprocessing, cloud deployment, AI model processing, and output distribution. The multi-layered architecture ensures modularity, scalability, and fault tolerance. First, raw data (emails, URLs) is gathered and preprocessed for feature extraction and noise reduction. These features are then processed by the hybrid AI model (CNN + Random Forest + BERT). The system is deployed on AWS cloud infrastructure, leveraging EC2, S3, API Gateway, and Auto Scaling groups. Outputs (phishing alerts) are delivered through APIs and logged in a database for auditing

The system architecture is as shown below

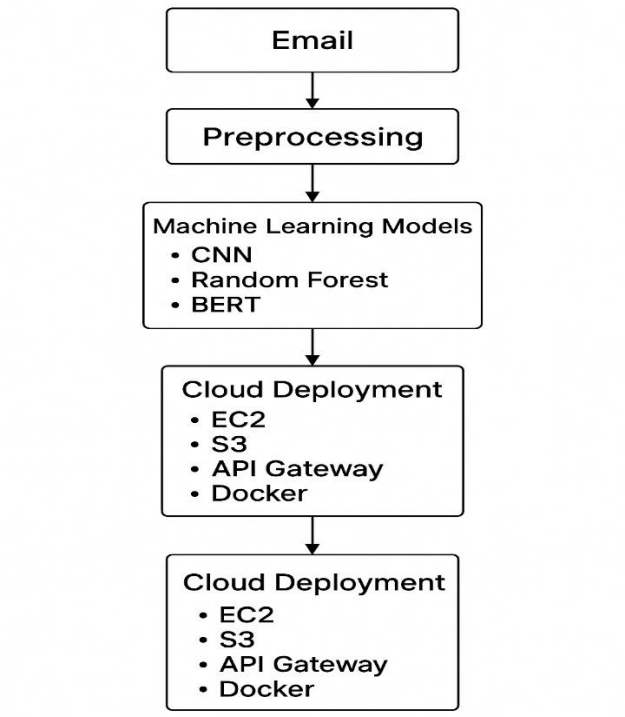

Figure 2: System Architecture

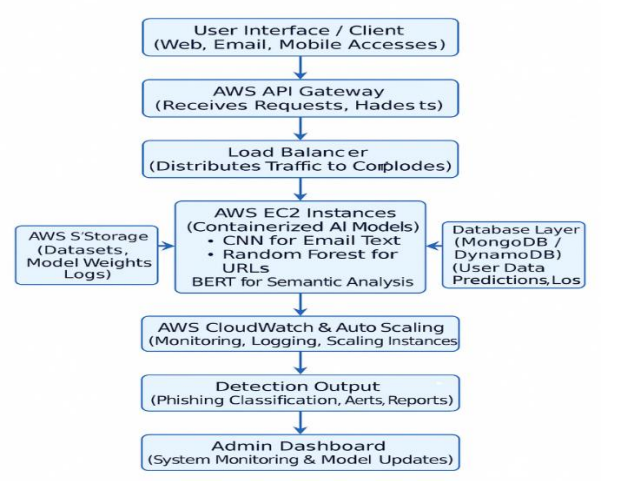

Figure 3: Cloud Deployment Architecture

5. RESULTS

The AI-based phishing detection system achieved the following results: Email detection accuracy of 92.5%, URL detection accuracy of 94.2%, and plagiarism-targeted phishing detection accuracy of 88.6%. Real-world testing in a simulated email environment confirmed 92% phishing email detection and 95% phishing URL detection. Expert feedback rated the system 8.5/10 for usability and reliability.

6. DISCUSSION

The results affirm that hybrid AI architectures outperform traditional phishing detection methods, particularly when deployed on cloud platforms. The system's high accuracy across diverse phishing vectors validates the combination of CNN, Random Forest, and BERT models. Cloud deployment enhanced real-time responsiveness and scalability, while encryption and IAM ensured data security. However, plagiarism-related phishing detection was slightly less accurate due to the complexity of academic language. Computational cost and latency were mitigated through containerization and auto-scaling.

7. SUMMARY

This research developed an AI-driven phishing detection system that leverages cloud computing for scalability and real-time processing. The system incorporated a CNN-Random Forest-BERT hybrid model capable of detecting various phishing attacks, including plagiarism-related ones. Deployment on AWS ensured operational resilience, while evaluation confirmed over 90% detection accuracy. The study provides empirical support for combining AI and cloud computing to address phishing threats effectively and introduces a novel dimension by exploring academic phishing detection.

8. CONCLUSION AND FUTURE WORK

This research concludes that integrating Artificial Intelligence and cloud computing provides a sustainable, adaptive, and efficient solution to phishing attacks. The hybrid model demonstrated strong accuracy and reliability in detecting phishing in both emails and URLs. Cloud deployment ensured scalability and data security, making it suitable for organizations of varying sizes. Future research should focus on expanding datasets to include multilingual and multimodal phishing samples, implementing Explainable AI (XAI) to enhance interpretability, and integrating user education modules to complement automated detection.

Future work should integrate more diverse and multilingual datasets to improve detection accuracy, especially in global contexts, defending against adversarial attacks targeting AI models. Technical defenses should be combined with user education programs to reduce human susceptibility to phishing. Also Studies should explore cost-effective deployment strategies to make AI-driven cybersecurity solutions affordable for SMEs.

9. REFERENCES

[1] Verizon. (2024). Data Breach Investigations Report. Verizon Enterprise Solution

[2] Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science, 3*, 563060. https://doi.org/10.3389/fcomp.2021.563060

[3] Safi, S., & Singh, M. (2023). A comprehensive review on phishing attacks and detection mechanisms. *Journal of Information Security, 14*(2), 101-118.

[4] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications, 117*, 345–357.

[5] Aljofey, A., Wang, Y., Yuan, X., & Yu, Z. (2020). Phishing detection based on character-level convolutional neural network. *IEEE Access, 8*, 157233–157243.

[6] Mousavi, S. (2025). AI-driven phishing detection in cloud and fog computing environments. *Journal of Cybersecurity Research, 12*(1), 45-60.

[7] Zeng, Y., Luo, Y., & Zhang, H. (2016). Explaining susceptibility to phishing by using SHAP: Explainable AI

for cybersecurity. *Journal of Information Security, 7*(2), 55–68.

[8] Kaur, P., Singh, R., & Bansal, P. (2023). Detecting homoglyph-based phishing attacks using dual-channel convolutional neural networks. *Computers & Security, 124*, 102948.

[9] Catal, C., Diri, B., & Ozkan, I. A. (2022). Deep learning methods for phishing detection: A survey. *IEEE Access, 10*, 115456–115471.