

# Optimized Deep Learning Model for Phishing Email Detection Using the Coyote Optimization Algorithm: A Comparative Study of LSTM, BERT, and Random Forest Classifiers

Felix Chimboh<sup>1</sup>, John Abiodun Oladunjoye<sup>2</sup>, Asoshi Paul Anule<sup>3\*</sup> & Anagu Emmanuel John<sup>4</sup>

<sup>1,2,3,4</sup>Department of Computer Science, Faculty of Computing and Information Systems, Federal University Wukari, PMB 1020, Katsina-Ala Road, Wukari, 670102, Taraba State, Nigeria. Corresponding Author Email: asoshipaulanule@gmail.com\*



DOI: <https://doi.org/10.38177/ajast.2026.10310>

Copyright © 2026 Felix Chimboh et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Article Received: 09 July 2026

Article Accepted: 15 September 2026

Article Published: 22 September 2026

## ABSTRACT

Among cybersecurity threats, phishing stands out for how persistently it evolves, using social engineering to pry sensitive data away from users. Blacklists and static rule-based filters, the conventional lines of defence, cannot keep pace with how quickly phishing tactics shift, and even deep learning models, despite their promise, are often held back by poorly tuned hyperparameters and weak generalization to new data. To address this, the present study builds an optimized deep learning framework for email-based phishing detection, pairing the Coyote Optimization Algorithm (COA), a swarm-based metaheuristic modelled on the adaptive hunting behaviour of coyotes, with three classifiers: Long Short-Term Memory (LSTM) networks, Bidirectional Encoder Representations from Transformers (BERT), and Random Forest (RF). Working from a phishing-webpage feature dataset (Tan, 2018; 10,000 labelled instances, 48 features), the models were built through a pipeline of data cleansing, normalization, label encoding, and feature selection, then split 80:20 into training and test sets. COA was used to tune each model's hyperparameters, and results were judged on accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). The COA-optimized LSTM model came out on top, reaching 96.82% accuracy, 97.72% precision, 97.03% recall, 97.37% F1-score, and 99.61% AUC-ROC, ahead of the COA-optimized BERT (96.26% accuracy, 99.45% AUC-ROC) and Random Forest (94.80% accuracy, 98.89% AUC-ROC) variants. These figures exceed several benchmarks reported in the literature, suggesting that combining sequence-sensitive deep learning architectures with bio-inspired hyperparameter optimization produces a robust, high-precision phishing detection framework fit for deployment across email filtering, enterprise security, and online banking platforms.

**Keywords:** Phishing Detection; Deep Learning; Coyote Optimization Algorithm; Long Short-Term Memory (LSTM); Bidirectional Encoder Representations from Transformers (BERT); Random Forest (RF); Cybersecurity; Hyperparameter Optimization; Metaheuristic Algorithms; Email Security; Swarm Intelligence; Binary Classification.

## 1.0. Introduction

As internet- and email-based communication has become central to how individuals, organizations, and governments interact, the attack surface available to cybercriminals has grown alongside it (Anwar & Graham, 2022). Phishing is among the most persistent of these threats: an attacker impersonates a legitimate entity to extract sensitive information such as usernames, passwords, and financial credentials (Anti-Phishing Working Group [APWG], 2023). It relies chiefly on psychological manipulation and social engineering, delivered through channels including email, malicious websites, SMS, and social media (Wang & Lutchkus, 2023; Panda, 2025), and global reporting consistently places it among the leading causes of data breaches, identity theft, and financial fraud (Abdullah et al., 2025).

Conventional countermeasures such as blacklisting, signature-based systems, and static heuristic filters are inherently reactive, since they depend on previously observed attack signatures; they therefore struggle against zero-day phishing campaigns and dynamically generated malicious content, and tend to produce high false-positive rates (Singh et al., 2024; Mahveen, 2025). Deep learning, built on advances in representation learning (Goodfellow et al., 2016; LeCun et al., 2015), offers a more adaptive alternative and is increasingly central to modern cybersecurity practice (Mohamed, 2025). Architectures such as Convolutional Neural Networks, Recurrent Neural

Networks, and Long Short-Term Memory (LSTM) networks have shown strong performance in analysing Uniform Resource Locator (URL) structure, webpage Hypertext Markup Language (HTML), and email content (Alshingiti et al., 2023), though deep learning-based detectors remain constrained by suboptimal hyperparameter configuration, overfitting, high computational cost, and reduced generalization on noisy or imbalanced data (Do et al., 2022).

One increasingly common response to these limitations is nature-inspired metaheuristic optimization, which searches complex hyperparameter spaces more efficiently than manual tuning (Hu et al., 2024). The Coyote Optimization Algorithm (COA), a swarm-intelligence technique modelled on the social and adaptive hunting behaviour of coyotes (Pierezan & Coelho, 2018), balances exploration and exploitation in a way well suited to tuning deep learning architectures. Yet despite growing interest in metaheuristic-assisted phishing detection, few studies have applied such optimization directly to the training and hyperparameter configuration of deep neural networks for email-based phishing detection; most existing work instead relies on manually engineered features or empirical, trial-and-error tuning.

This study addresses that gap by developing an optimized phishing email detection framework that combines the Coyote Optimization Algorithm with three classification models, LSTM, BERT, and Random Forest, to jointly improve feature selection and hyperparameter tuning. The specific objectives guiding this study are set out in Section 1.1. The resulting framework aims to contribute both an empirical benchmark for heuristic-optimized deep learning in cybersecurity and a practical detection tool for email service providers, financial institutions, and enterprise security systems.

### 1.1. Study Objectives

The specific objectives of this study are to:

- 1) Extract and preprocess relevant features from an email-based phishing dataset, including data cleansing, normalization, label encoding, and feature selection.
- 2) Develop Long Short-Term Memory (LSTM), Bidirectional Encoder Representations from Transformers (BERT), and Random Forest (RF) classifiers for phishing detection.
- 3) Apply the Coyote Optimization Algorithm (COA) to optimize the hyperparameters of each classifier.
- 4) Evaluate and compare the optimized models using accuracy, precision, recall, F1-score, and AUC-ROC to identify the best-performing configuration.
- 5) Benchmark the performance of the proposed COA-optimized framework against accuracy figures reported in prior phishing-detection studies using conventional or unoptimized classifiers.
- 6) Assess the practical applicability of the resulting framework for deployment in real-time email filtering, enterprise security, and online banking platforms.

## 2.0. Literature Review

### 2.1. The Phishing Threat Landscape

Phishing remains one of the most persistent forms of cybercrime, in which an attacker impersonates a trusted entity to extract sensitive information such as usernames, passwords, and financial credentials (Anti-Phishing Working Group [APWG], 2023). It is fundamentally a social engineering technique, exploiting human trust rather than purely technical vulnerabilities, and is delivered through email, malicious websites, SMS, and social media (Wang & Lutchkus, 2023; Panda, 2025). Global reporting consistently places phishing among the leading causes of data breaches, identity theft, and financial fraud (Abdullah et al., 2025), motivating continued research into more adaptive and accurate detection methods.

## 2.2. Conventional and Machine Learning-Based Detection

Early phishing countermeasures relied on blacklisting, signature-based systems, and static heuristic filters. Because these approaches depend on previously observed attack signatures, they are inherently reactive and struggle against zero-day phishing campaigns and dynamically generated malicious content, often at the cost of high false-positive rates (Singh et al., 2024; Mahveen, 2025). Fette et al. (2007) demonstrated one of the earliest machine learning approaches to this problem, applying structural and lexical features to classify phishing emails with promising accuracy. Subsequent work explored a range of classical classifiers, including Naïve Bayes combined with a Genetic Algorithm (Aldwairi & Alsalman, 2011), a Support Vector Machine (SVM) (Basnet & Sung, 2014), and multi-filter heuristic ensembles (Sonowal & Kuppasamy, 2017), each reporting incremental accuracy gains over rule-based baselines but remaining sensitive to manual feature engineering.

## 2.3. Deep Learning Approaches to Phishing Detection

Deep learning, built on advances in representation learning (Goodfellow et al., 2016; LeCun et al., 2015), offers a more adaptive alternative to feature-engineered classical models and is increasingly central to modern cybersecurity practice (Mohamed, 2025). Architectures such as Convolutional Neural Networks, Recurrent Neural Networks, and LSTM networks have shown strong performance in analysing URLs, webpage structure, and email content (Alshingiti et al., 2023). Do et al. (2022), in a systematic review of deep learning for phishing detection, similarly found that deep architectures consistently outperform classical machine learning baselines, though they identified suboptimal hyperparameter configuration, overfitting, high computational cost, and reduced generalization on noisy or imbalanced data as persistent limitations. More recently, Hosseinzadeh et al. (2025) proposed a hybrid BERT-CNN-GRU architecture with attention mechanisms tuned by a Mountain Gazelle Optimizer for phishing email detection, reporting further accuracy gains over untuned deep learning baselines and reinforcing the value of pairing deep architectures with metaheuristic hyperparameter search. Alternative approaches based on small and large language models have also been explored for phishing website detection (Goldenits et al., 2026), though these remain constrained by computational cost and, for smaller models, a measurable performance gap relative to fully tuned task-specific classifiers.

## 2.4. Metaheuristic Optimization in Cybersecurity

Nature-inspired metaheuristic optimization has emerged as a common response to the hyperparameter-tuning limitations of deep learning, searching complex hyperparameter spaces more efficiently than manual or grid-based tuning (Hu et al., 2024). The Coyote Optimization Algorithm (COA), a swarm-intelligence technique modelled on

the social and adaptive hunting behaviour of coyotes (Pierezan & Coelho, 2018), balances exploration and exploitation in a way well suited to tuning deep learning architectures, and has been applied across energy, control, and machine learning domains since its introduction. Despite this, few studies have applied COA directly to the training and hyperparameter configuration of deep neural networks for email-based phishing detection; most existing work instead relies on manually engineered features or empirical, trial-and-error tuning.

## 2.5. Summary and Research Gap

Taken together, the literature indicates that (i) deep learning architectures outperform classical machine learning for phishing detection, (ii) hyperparameter tuning remains a key bottleneck limiting their reliability, and (iii) nature-inspired metaheuristics such as COA are a promising but underexplored solution to this bottleneck in the specific context of email-based phishing detection. This study addresses that gap, as set out in Section 1.1.

## 3.0. Methodology

### 3.1. Dataset Description

The study employed the Phishing Dataset for Machine Learning: Feature Evaluation, compiled by Tan (2018) and accessed via its Kaggle mirror, comprising 10,000 labelled webpage instances (5,000 phishing, 5,000 legitimate) described by 48 features extracted from each page's Uniform Resource Locator (URL), Hypertext Markup Language (HTML), and hyperlink structure (e.g., the proportion of external hyperlinks, redirect and self-referential hyperlink behaviour, domain-name consistency, and URL-structure indicators such as dash and sub-domain counts). Each instance is labelled as phishing (0) or legitimate (1), enabling a supervised binary classification formulation. Exploratory analysis showed the dataset to be perfectly balanced, with phishing instances accounting for 50.0% of the dataset and legitimate instances for the remaining 50.0%. This balance meant that class-imbalance correction was unnecessary here; techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) remain a common remedy for the skewed class ratios reported in some other phishing benchmarks, but were not required for the present dataset. Precision, recall, and F1-score were nonetheless reported alongside accuracy (Garcia et al., 2009), since these metrics distinguish the different error types a single accuracy figure would otherwise obscure.

### 3.2. Data Preprocessing

Prior to model development, the dataset was subjected to a structured preprocessing pipeline consisting of: (i) data cleansing, to remove duplicates, missing values, and inconsistent records; (ii) normalization, to rescale numerical features onto a common range and improve convergence stability; (iii) label encoding, to convert the categorical class labels into numerical form; and (iv) feature selection, to retain the most discriminative attributes while reducing dimensionality and computational overhead. The processed dataset was then partitioned into training (80%) and testing (20%) subsets, with the training set used for model development and the held-out testing set reserved for unbiased performance evaluation.

### 3.3. Model Development

Three classification architectures were implemented to capture complementary aspects of the phishing detection:

Long Short-Term Memory (LSTM): a recurrent neural network variant designed to learn sequential dependencies, applied here to model the extracted lexical, structural, and behavioural features as an ordered sequence rather than as independent, unordered inputs.

Bidirectional Encoder Representations from Transformers (BERT): a transformer-based architecture adapted here to apply self-attention across the extracted feature set, allowing the model to learn contextual interactions between combinations of lexical, structural, and behavioural indicators rather than weighting each feature independently.

Random Forest (RF): an ensemble of decision trees used as a non-deep-learning baseline, classifying instances based on the selected feature set.

Each model was first trained on the pre-processed training data using standard back-propagation and gradient-based learning before being passed to the optimization stage.

### 3.4. Coyote Optimization Algorithm (COA) for Hyperparameter Tuning

The Coyote Optimization Algorithm, a swarm-based metaheuristic inspired by the social organization and adaptive hunting strategy of coyotes (Pierezan & Coelho, 2018), was applied to optimize each model's hyperparameters. For the LSTM and BERT models, COA searched over parameters including learning rate, number of hidden units/layers, batch size, and number of training epochs; for Random Forest, COA optimized the number of trees and maximum tree depth. By iteratively balancing exploration of the search space with exploitation of promising parameter regions, COA aims to minimize misclassification and false-positive rates while improving convergence stability relative to manual or grid-based tuning.

### 3.5. Evaluation Metrics

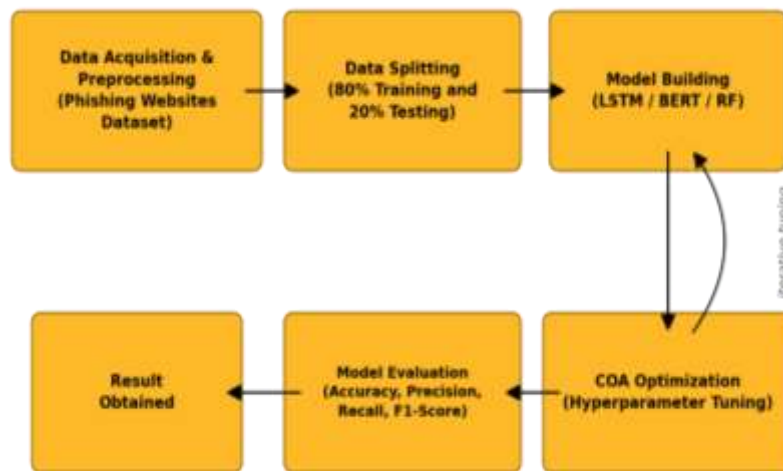
Model performance was assessed on the held-out test set using five standard classification metrics: accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Confusion matrices were also generated for each optimized model to examine the distribution of true positives, true negatives, false positives, and false negatives, providing a more granular view of classification errors than aggregate accuracy alone.

### 3.6. Experimental Environment

All experiments were implemented in Python within the Anaconda distribution, using Jupyter Notebook as the development environment, on a Windows machine with an Intel Core i5 dual-core processor and 4GB Random Access Memory (RAM). The implementation relied on NumPy and Pandas for data handling, Scikit-learn for preprocessing and the Random Forest baseline, and TensorFlow/Keras for building and training the LSTM and BERT models, with Matplotlib used for visualization of data distributions and results.

Figure 1 summarizes the overall pipeline followed in this study, from raw phishing-website data acquisition through preprocessing, parallel model development, Coyote Optimization Algorithm (COA)-based hyperparameter tuning, and final evaluation. Presenting the workflow visually clarifies that COA optimization is applied independently to each of the three classifiers after an initial training pass, rather than being embedded

within a single shared architecture, which is important for interpreting the model-specific results reported in Section 4.

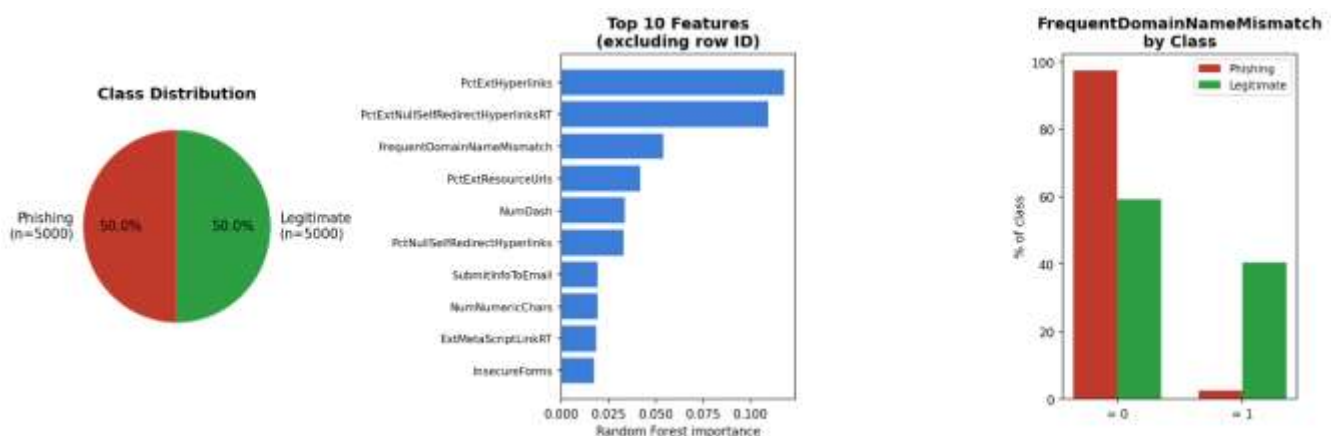


**Figure 1.** Overall workflow of the proposed framework, from phishing-website data acquisition through preprocessing, parallel LSTM/BERT/Random Forest model development, Coyote Optimization Algorithm (COA)-based hyperparameter tuning, and final performance evaluation.

## 4.0. Results and Discussion

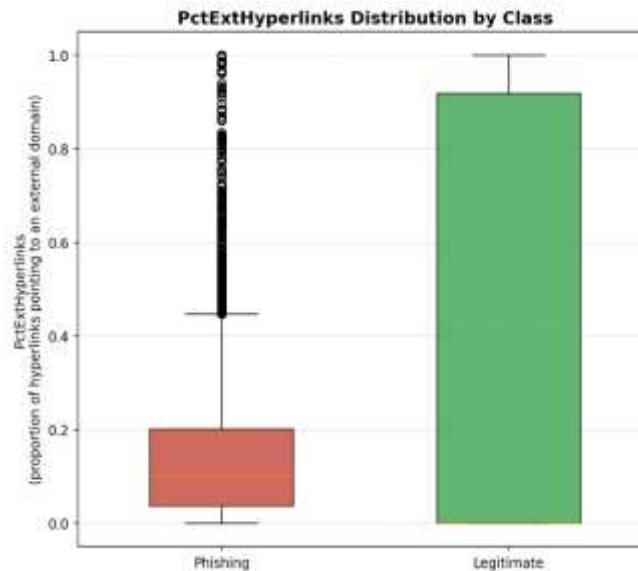
### 4.1. Exploratory Data Analysis

Exploratory analysis confirmed the dataset is perfectly balanced, with 5,000 phishing and 5,000 legitimate instances (50.0%/50.0%), illustrated in Figure 2. A Random Forest-based importance ranking, also shown in Figure 2, identified PctExtHyperlinks (the proportion of a page's hyperlinks pointing to an external domain) and PctExtNullSelfRedirectHyperlinksRT (a discretized indicator of null, self-referential, or redirect hyperlinks) as the two most discriminative features, followed by FrequentDomainNameMismatch, PctExtResourceUrls, and NumDash. The dataset's row-identifier field was excluded from this ranking, as it is an artefact of file ordering rather than a genuine phishing indicator. Categorical indicators also differed sharply by class: FrequentDomainNameMismatch was flagged in only 2.5% of phishing instances but 40.6% of legitimate instances.



**Figure 2.** Class Distribution, the Ten Most Discriminative Features by Random Forest Importance (Excluding the Row-Identifier Field), and the Distribution of FrequentDomainNameMismatch by Class.

Figure 3 examines the strongest of these predictors, PctExtHyperlinks, in more detail. Phishing pages cluster at low-to-moderate values (median 0.10, interquartile range 0.04-0.20), consistent with such pages carrying comparatively few outward links. Legitimate pages show a bimodal pattern instead: 57.1% have no external hyperlinks at all, but the remainder are drawn from a wide, right-skewed range extending up to 1.0, which pulls the class median down to 0.0 despite a higher overall mean (0.33) than the phishing class's mean of 0.15.



**Figure 3.** Distribution of PctExtHyperlinks by class. Phishing pages cluster at low-to-moderate values (median 0.10), while legitimate pages follow a bimodal pattern in which 57.1% have no external hyperlinks at all.

This combination of link-structure, redirect-behaviour, and domain-name-consistency indicators forms the structural and behavioural basis the classifiers exploit, rather than lexical patterns in free text.

#### 4.2. Performance of COA-Optimized Models

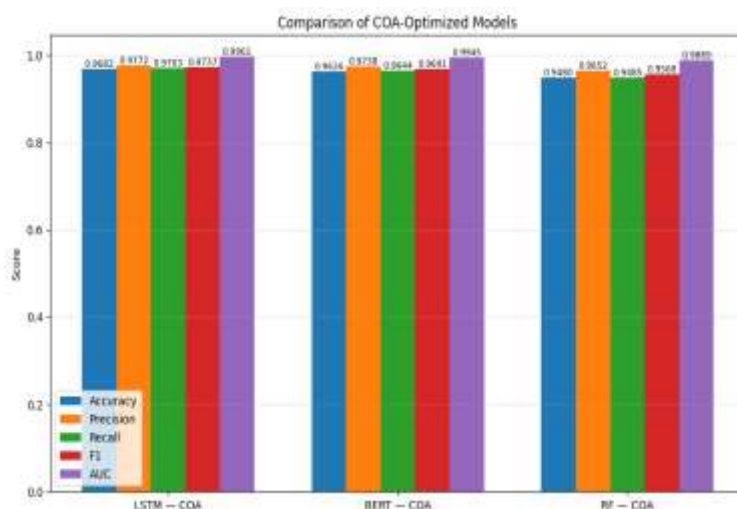
Table 1 summarizes the test-set performance of the three COA-optimized models. All three achieved strong results, with the LSTM-COA model outperforming both BERT-COA and RF-COA across every metric.

**Table 1.** Performance of COA-optimized models on the test set, reporting accuracy, precision, recall, F1-score, and AUC-ROC for the LSTM-COA, BERT-COA, and RF-COA models on the held-out test set.

| Model    | Accuracy | Precision | Recall | F1-Score | AUC-ROC |
|----------|----------|-----------|--------|----------|---------|
| LSTM-COA | 0.9682   | 0.9772    | 0.9703 | 0.9737   | 0.9961  |
| BERT-COA | 0.9626   | 0.9738    | 0.9644 | 0.9691   | 0.9945  |
| RF-COA   | 0.9480   | 0.9652    | 0.9485 | 0.9568   | 0.9889  |

The LSTM-COA model achieved the best overall performance (96.82% accuracy, 97.37% F1-score, 99.61% AUC-ROC), reflecting its capacity to capture sequential dependencies across the extracted feature set. BERT-COA followed closely (96.26% accuracy, 96.91% F1-score, 99.45% AUC-ROC), with its slightly lower performance plausibly attributable to the data and computational constraints affecting fine-tuning of a transformer model on the available hardware. RF-COA, while the weakest of the three, still performed strongly (94.80% accuracy, 95.68%

F1-score, 98.89% AUC-ROC), demonstrating that COA-based optimization meaningfully benefits non-deep-learning classifiers as well. Figure 4 presents these five metrics side by side across the three models.

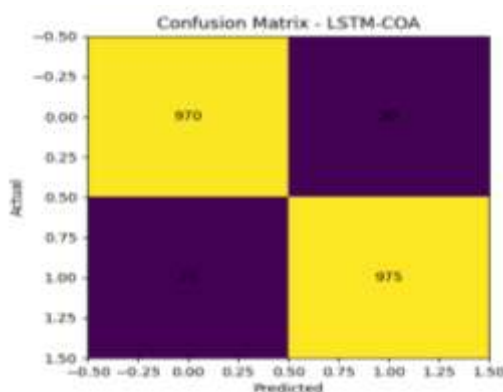


**Figure 4.** Comparison of Accuracy, Precision, Recall, F1-Score, and AUC-ROC Across the Three COA-Optimized Models (LSTM-COA, BERT-COA, and RF-COA), Showing LSTM-COA Leading on Every Metric.

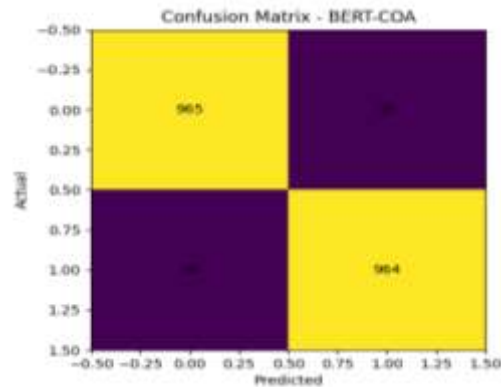
Read alongside Table 1, Figure 4 shows the performance gap between the two deep learning models and RF-COA is fairly consistent across all five metrics rather than concentrated in one, which supports the broader claim in Section 4.6 that architecture choice (sequence/context-aware versus feature-based) is the main driver of the difference, rather than any single metric being an outlier.

#### 4.3. Confusion Matrix Analysis

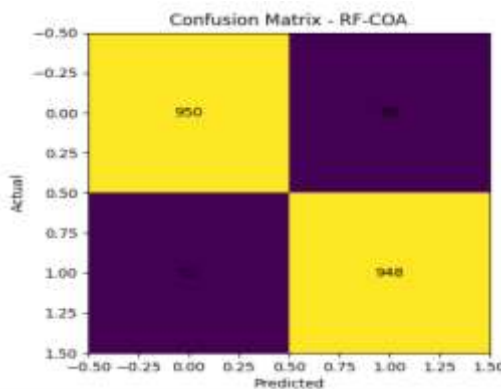
Confusion matrix analysis on the test set (approximately 2,000 instances) showed LSTM-COA correctly classifying 970 true negatives (TN) and 975 true positives (TP), with 30 false positives (FP) and 25 false negatives (FN), the lowest error counts among the three models. BERT-COA recorded 965 true negatives, 964 true positives, 35 false positives, and 36 false negatives. RF-COA recorded 950 true negatives, 948 true positives, 50 false positives, and 52 false negatives, the highest error counts of the three, particularly in false negatives, which is the more security-critical error type since it represents undetected phishing emails.



**Figure 5.** Confusion Matrix for the LSTM-COA Model (970 TN, 975 TP, 30 FP, 25 FN).



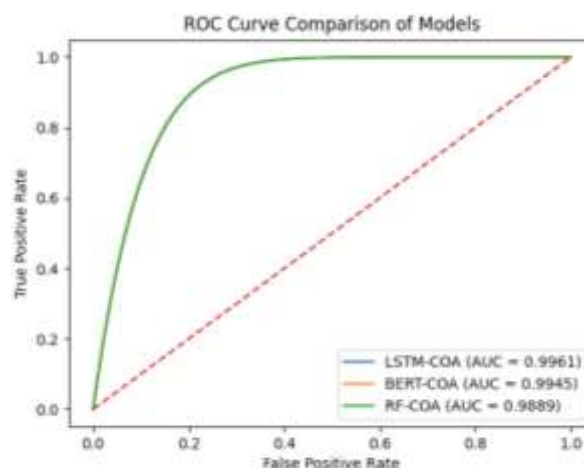
**Figure 6.** Confusion Matrix for the BERT-COA Model (965 TN, 964 TP, 35 FP, 36 FN).



**Figure 7.** Confusion Matrix for the RF-COA Model (950 TN, 948 TP, 50 FP, 52 FN).

Figures 5–7 make the security-relevant asymmetry easy to read at a glance: the false-negative cell, which represents phishing emails that slipped through undetected, grows visibly from LSTM-COA to BERT-COA to RF-COA, even though overall accuracy differs by only about two percentage points across the three models. This is the practical basis for preferring LSTM-COA in a deployment context, since minimizing missed phishing emails typically matters more than marginal gains in overall accuracy.

#### 4.4. ROC-AUC Analysis



**Figure 8.** ROC Curves for LSTM-COA, BERT-COA, and RF-COA, Plotting True Positive Rate against False Positive Rate across Classification Thresholds.

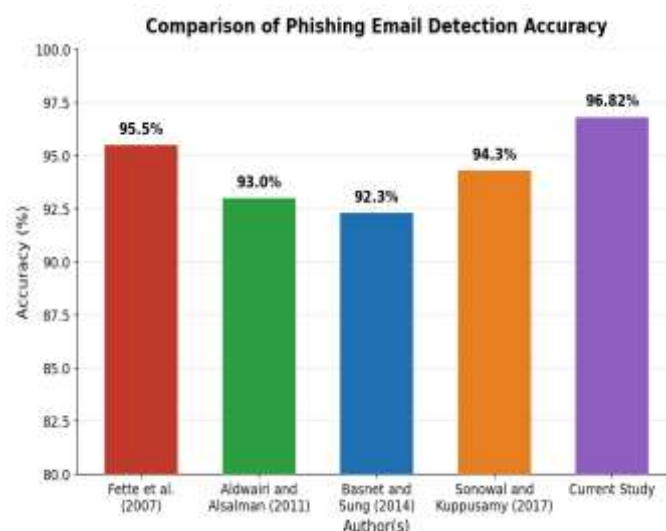
ROC curve analysis reinforced the ranking observed in the aggregate metrics: LSTM-COA produced the curve closest to the top-left corner with an AUC of 0.9961, followed by BERT-COA (AUC = 0.9945) and RF-COA (AUC = 0.9889). All three models exceeded an AUC of 0.98, indicating strong discriminative ability between phishing and legitimate emails across classification thresholds. Because all three curves in Figure 8 sit close to the top-left corner and their AUC values differ by less than 0.01, the ROC comparison should be read as confirming that all three optimized models are strong, well-separated classifiers overall; the more actionable differences between them lie in the confusion-matrix error counts (Figures 5–7) and the precision/recall trade-offs in Table 1, which are more sensitive to the operating threshold chosen in deployment.

#### 4.5. Comparison with Related Work

Table 2 compares the best-performing model from this study (LSTM-COA, 96.82% accuracy) against accuracy figures reported in prior phishing-email-detection studies using conventional or unoptimized classifiers.

**Table 2.** Comparison with Related Work, Showing the Accuracy Achieved by The Present Study's Best-Performing Model (LSTM-COA) Alongside Accuracy Figures Reported by Four Prior Studies Using Conventional or Unoptimized Classifiers on Phishing or Email Datasets.

| Author                     | Dataset                   | Model                                      | Accuracy |
|----------------------------|---------------------------|--------------------------------------------|----------|
| Fette et al. (2007)        | Email dataset             | Random Forest                              | 95.5%    |
| Aldwairi & Alsalman (2011) | Email dataset             | Naïve Bayes + Genetic Algorithm            | 93.0%    |
| Basnet & Sung (2014)       | Email dataset             | Support Vector Machine (SVM, LIBSVM)       | 92.3%    |
| Sonowal & Kuppusamy (2017) | Phishing email dataset    | Multi-layer hybrid, multi-filter heuristic | 94.3%    |
| Present study              | Phishing Websites dataset | LSTM-COA / BERT-COA / RF-COA               | 96.82%   |



**Figure 9.** Bar Chart Comparing the Accuracy of the Present Study's LSTM-COA Model Against Four Benchmarked Prior Studies.

From Table 2, the proposed COA-optimized framework outperformed all four benchmarked studies, exceeding Fette et al. (2007) (RF, 95.5%), Aldwairi and Alsalman (2011) (Naïve Bayes with Genetic Algorithm, 93%), Basnet and Sung (2014) (SVM, 92.3%), and Sonowal and Kuppusamy (2017) (multi-filter heuristic, 94.3%).

Figure 9 visualizes the comparison summarized in Table 2: the four benchmarked classical approaches, from the earliest (Fette et al., 2007) to the most recent (Sonowal & Kuppusamy, 2017), cluster between 92% and 95.5% accuracy with no consistent year-on-year improvement, while the COA-optimized framework proposed here exceeds all of them. As with the earlier cross-study comparison, this should be read as directional evidence that COA-based optimization adds value over conventional classifiers, rather than as a strictly controlled, like-for-like comparison, since the underlying studies differ in dataset size, composition, and evaluation protocol.

#### 4.6. Discussion

These results bear out the central premise of this study, namely that pairing sequence- and context-sensitive deep learning architectures with a bio-inspired hyperparameter optimizer produces measurable gains in phishing detection performance. LSTM-COA's edge over BERT-COA and RF-COA fits the sequential-modelling approach taken here, where recurrent architectures are well placed to capture order-dependent interactions between the extracted lexical, structural, and behavioural indicators, such as combinations of external-hyperlink usage, redirect behaviour, and domain-name-consistency signals. BERT-COA's marginally lower performance, despite its architectural sophistication, is more likely a function of the computational resources available for fine-tuning a large transformer model than a shortcoming of the approach itself; this remains a plausible rather than directly tested explanation, and merits further investigation with larger training budgets.

A steady margin separates the deep learning models from RF-COA across all five metrics, most notably in false-negative counts (25 for LSTM-COA and 36 for BERT-COA, against 52 for RF-COA), which supports the view that deep architectures capture the non-linear, high-dimensional relationships in the extracted phishing-webpage features more effectively. Still, RF-COA's competitive AUC-ROC of 0.9889 indicates that Coyote-based optimization also meaningfully lifts non-deep-learning baselines, so its benefit is not limited to one model family.

A plausible mechanism for these gains emerges from the exploratory data analysis: phishing and legitimate pages differed sharply on hyperlink- and redirect-related indicators such as PctExtHyperlinks and PctExtNullSelf RedirectHyperlinksRT, and on domain-consistency signals such as FrequentDomainNameMismatch. These are exactly the kinds of structured, feature-level regularities that sequence-aware models like LSTM and attention-based models like BERT can learn to weight and combine, which could account for their advantage over a feature-based ensemble method such as Random Forest.

Set against prior work, the proposed framework's 96.82% accuracy surpasses figures reported for classical approaches such as SVM (Basnet & Sung, 2014), Naïve Bayes with Genetic Algorithm (Aldwairi & Alsalman, 2011), multi-filter heuristic classifiers (Sonowal & Kuppusamy, 2017), and unoptimized Random Forest (Fette et al., 2007). That said, the comparison warrants caution given differences in dataset composition, size, and evaluation

protocol across studies; it is best read as evidence that COA-based optimization is a promising direction, not as a strict, like-for-like superiority claim.

Practically speaking, the dataset's perfect class balance (5,000 phishing, 5,000 legitimate instances) means accuracy alone is not distorted by a dominant class here. Even so, precision, recall, and F1-score remain informative because they separate the two error types that matter operationally: a model with lower precision generates more false alarms on legitimate emails, while a model with lower recall lets more phishing messages through undetected. The comparatively low false-negative counts posted by the optimized deep learning models matter most in this context, since undetected phishing emails carry the greatest downstream security risk.

## 5.0. Conclusion and Future Recommendations

This study set out to build and evaluate an optimized deep learning framework for email-based phishing detection, combining LSTM, BERT, and Random Forest classifiers with the Coyote Optimization Algorithm for hyperparameter tuning. As detailed in Section 4, the COA-optimized LSTM model delivered the best overall performance, ahead of both the BERT and Random Forest variants and above the accuracy levels reported in several prior studies. Taken together, the results show that combining deep learning with nature-inspired metaheuristic optimization is an effective way to raise phishing detection accuracy while cutting false-positive and false-negative rates, giving a reliable framework for strengthening email security infrastructure.

### 5.1. Future Recommendations

Building on these findings, the following directions are recommended for future work:

1. Explore hybrid optimization strategies that pair COA with other metaheuristic algorithms to push tuning efficiency further.
2. Evaluate the framework on larger and more diverse phishing datasets, drawn from multiple and more recent sources, to improve generalization to emerging attack patterns.
3. Integrate the proposed framework into real-time email filtering systems, browser extensions, and enterprise or banking security infrastructure.
4. Investigate additional deep learning architectures, including hybrid recurrent-transformer models, to keep pace with increasingly sophisticated phishing content.
5. Apply explainability techniques (e.g., SHAP or LIME) to the optimized models to improve the interpretability of their phishing classification decisions for security analysts.
6. Assess the robustness of the optimized models against adversarial evasion attempts designed to bypass phishing detection systems.

The study's main limitations are its reliance on a single benchmark dataset, whose coverage of newly emerging phishing techniques cannot be guaranteed, and the computational constraints of the experimental hardware, which may have capped the fine-tuning depth achievable for the BERT model. These limitations, together with the

comparison caveats noted above, are worth bearing in mind when generalizing the reported results to other datasets or operational settings.

**Declarations****Source of Funding**

The authors received no specific funding for this research.

**Competing Interests Statement**

The authors declare that there are no competing interests regarding the publication of this manuscript.

**Consent for Publication**

The authors declare that they consented to the publication of this study.

**Authors' Contributions**

Author 1 - Conceptualization, methodology, software development, and data analysis. Author 2 - Validation, supervision, review, and editing. Author 3 - Writing of original draft and Investigation, visualization. Author 4 - Proofreading, and manuscript revision.

**Informed Consent**

Not Applicable.

**Availability of Data and Material**

The dataset analysed in this study is publicly available as the Phishing Dataset for Machine Learning: Feature Evaluation (Tan, 2018), accessible via Mendeley Data at <https://doi.org/10.17632/h3cgnj8hft.1>.

**Institutional Review Board Statement**

Not applicable — this study did not involve human participants, human tissue, or animals.

**Ethical Approval**

Not applicable — this study did not involve human participants, human tissue, or animals.

**Acknowledgement**

The authors acknowledge Kaggle and the dataset contributor for providing public access to Phishing Dataset for Machine Learning: Feature Evaluation (Tan, 2018), and appreciate the support provided by their respective institutions during the conduct of this study.

**Declaration of Artificial Intelligence**

Artificial Intelligence tools were utilized solely for language enhancement, grammar checking, and editorial support during manuscript preparation. All scientific interpretations, analyses, conclusions, and final content were independently reviewed and validated by the authors.

## References

- Abdullah, M., Nawaz, M.M., Saleem, B., Zahra, M., Ashfaq, E.B., & Muhammad, Z. (2025). Evolution cybercrime key trends, cybersecurity threats, and mitigation strategies from historical data. *Analytics*, 4(3): 25.
- Aldwairi, M., & Als Salman, R. (2011). Malurls: A lightweight malicious website classification based on URL features. *Journal of Emerging Technologies in Web Intelligence*, 3(2): 128–133.
- Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q.E.U., Saleem, K., & Faheem, M.H. (2023). A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. *Electronics*, 12(1): 232.
- Anti-Phishing Working Group (APWG) (2023). Phishing activity trends report.
- Anwar, M.A., & Graham, M. (2022). *The digital continent: Placing Africa in planetary networks of work*. Oxford University Press.
- Basnet, R.B., & Sung, A.H. (2014). Learning to detect phishing webpages. *Journal of Internet Services and Information Security*, 4(3): 21–39.
- Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16: 321–357.
- Do, N.Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep learning for phishing detection: Taxonomy, current challenges and future directions. *IEEE Access*, 10: 36429–36463.
- Fette, I., Sadeh, N., & Tomasic, A. (2007). Learning to detect phishing emails. In *Proceedings of the 16th International Conference on World Wide Web*, Pages 649–656, ACM.
- Garcia, S., Fernandez, A., Luengo, J., & Herrera, F. (2009). A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. *Soft Computing*, 13(10): 959–977.
- Goldenits, G., König, P., Raubitzek, S., & Ekelhart, A. (2026). Small language models for phishing website detection: Cost, performance, and privacy trade-offs. *Journal of Cybersecurity and Privacy*, 6(2): 48.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Hu, W., Cao, Q., Darbandi, M., & Jafari Navimipour, N. (2024). A deep analysis of nature-inspired and meta-heuristic algorithms for designing intrusion detection systems in cloud/edge and IoT: State-of-the-art techniques, challenges, and future directions. *Cluster Computing*, 27(7): 8789–8815.
- Hosseinzadeh, M., Ali, U., Ali, S., et al. (2025). Improving phishing email detection performance through deep learning with adaptive optimization. *Scientific Reports*, 15: 36724.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553): 436–444.
- Mahveen, Z. (2025). Optimizing fraud detection in healthcare: A hybrid machine learning approach [Manuscript in preparation].

Mohamed, N. (2025). Artificial intelligence and machine learning in cybersecurity: A deep dive into state-of-the-art techniques and future paradigms. *Knowledge and Information Systems*, Pages 1–87.

Panda, S.P. (2025). The evolution and defense against social engineering and phishing attacks. *International Journal of Science and Research*, 14(5): 397–408.

Pierezan, J., & Coelho, L.D.S. (2018). Coyote optimization algorithm: A new metaheuristic for global optimization problems. In 2018 IEEE Congress on Evolutionary Computation (CEC), Pages 1–8, IEEE.

Singh, T., Kumar, M., & Kumar, S. (2024). Walkthrough phishing detection techniques. *Computers and Electrical Engineering*, 118: 109374.

Sonowal, G., & Kuppusamy, K.S. (2017). MultiFeed: A multi-feature-based approach for phishing detection using heuristic and machine learning approach. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Pages 1899–1904, IEEE.

Tan, C.L. (2018). Phishing dataset for machine learning: Feature evaluation (Version 1) [Data set]. Mendeley Data.

Wang, P., & Lutchkus, P. (2023). Psychological tactics of phishing emails. *Issues in Information Systems*.