IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Deep learning for email threat intelligence: Feature importance and model performance in phishing detection

Kittipol Wisaeng[1*], Benchalak Muangmeesri[2]

[1]Technology and Business Information System Unit, Mahasarakham Business School, Mahasarakham University, Mahasarakham 44150, Thailand.
[2]Engineering Management, Suan Sunandha Rajabhat University, 1 U-Thong nok Road, Dusit, Bangkok 10300, Thailand.

Corresponding author: Kittipol Wisaeng (*Email: Kittipol.w@acc.msu.ac.th*)

Abstract

Phishing emails remain among the most widespread and insidious vectors for cyberattacks, leveraging social engineering tactics and psychological manipulation to deceive recipients and compromise information systems. This study examines the effectiveness of advanced deep learning architectures, including Feedforward Neural Networks (FNN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and Bidirectional Encoder Representations from Transformers (BERT), in the automated classification of phishing emails versus legitimate ones. Utilizing a large, feature-enriched dataset that integrates linguistic, structural, and behavioral attributes, the models were evaluated using comprehensive performance metrics, including Accuracy, Precision, Recall, F1 Score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Experimental findings indicate that BERT achieves superior performance, attaining a classification accuracy of 97% and the lowest error rates, which can be attributed to its ability to model deep semantic and contextual nuances within email content. Feature importance analysis further demonstrates the significance of attributes such as hyperlink count, urgency-inducing keywords, and orthographic anomalies as discriminative indicators of phishing attempts. These results affirm the critical value of deep contextual representation in combating sophisticated phishing strategies. The study positions BERT as a state-of-the-art benchmark for high-fidelity email threat detection. Future research will focus on adapting cross-lingual models, enhancing adversarial resilience, and integrating real-time detection capabilities into enterprise-level security infrastructures.

Keywords: BERT, Cybersecurity, Deep Learning, Feature Engineering, Phishing Email Detection.

DOI: 10.53894/ijirss.v8i5.9302
Funding: This research project was financially supported by Mahasarakham Business School, Mahasarakham University, Thailand.
History: Received: 25 June 2025 / **Revised:** 25 July 2025 / **Accepted:** 29 July 2025 / **Published:** 15 August 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Institutional Review Board Statement: The Ethical Committee of the Taif University, Saudi Arabia has granted approval for this study on 17 November 2023 (Ref. No. 85216).
Acknowledgments: This research project was financially supported by Mahasarakham Business School, Mahasarakham University, Thailand. Special thanks are extended to the maintainers of the Enron Email Dataset and the publicly available Phishing Email Dataset for providing the foundational data that enabled this study. Lastly, the authors appreciate the insightful feedback and support from colleagues and reviewers, whose constructive input significantly improved the quality and clarity of this work.
Publisher: Innovative Research Publishing

1. Introduction

The pervasive use of email as a communication medium has rendered it a primary vector for cyberattacks, with phishing emerging as one of the most persistent, deceptive, and costly threats to organizational security. Often serving as the initial stage of more sophisticated intrusions, such as Advanced Persistent Threats (APTs), ransomware, and exploit kits, phishing campaigns have evolved in complexity, frequency, and precision, far outpacing the capabilities of conventional filtering mechanisms [1]. In 2020, an alarming 75% of organizations reported phishing-related incidents, with 96% of these attacks delivered via email. The year also witnessed a staggering 64% increase in email-based threats compared to 2019, with phishing implicated in over 90% of successful data breaches, resulting in average financial losses of $3.86 million per breach. The COVID-19 pandemic further intensified this threat landscape, fueling a 667% surge in phishing emails, as attackers capitalized on global anxiety by impersonating trusted institutions such as health agencies and governments [2]. Compounding the challenge, cyber adversaries are increasingly adopting machine learning (ML) to craft highly personalized and evasive phishing content, enabling them to bypass legacy detection systems with greater success [3]. These alarming trends highlight the urgent need for intelligent, adaptive, and context-aware email threat detection solutions. Over the past decade, the research community has responded with a growing body of literature focused on applying machine learning techniques to phishing email detection. However, multiple recent reviews and empirical studies [4-6] have identified critical limitations within existing approaches. These include an over-reliance on balanced or artificially curated datasets that fail to reflect real-world class imbalance, a limited exploration of deep learning architectures, suboptimal use of performance metrics, and outdated datasets that lack relevance to contemporary attack patterns. Such limitations undermine the real-world applicability and generalizability of many proposed solutions. Current ML-based detection systems typically focus on either structural features, including email headers, domain reputation, and embedded URLs, or textual features extracted from the email body, often utilizing Natural Language Processing (NLP) techniques for content analysis. However, the lack of integration between these complementary feature types has constrained model accuracy and adaptability. Furthermore, while ensemble learning methods have proven effective in related domains such as phishing website identification [7] and financial fraud detection [8], their deployment in phishing email classification remains relatively underexplored. The escalating sophistication of phishing tactics drives this study, the limitations of current detection frameworks, and the absence of unified deep learning models that incorporate both structural and textual information. In response, we present a comprehensive evaluation of advanced deep learning models, Feedforward Neural Networks (FNN), Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and Bidirectional Encoder Representations from Transformers (BERT) for phishing email detection. By integrating diverse features and leveraging cutting-edge neural architectures, this work aims to enhance the robustness, accuracy, and interpretability of email threat detection systems. Our contributions address existing gaps and pave the way for the deployment of scalable, real-time, and intelligent phishing defense mechanisms.

2. Related Work

Phishing email detection has garnered significant attention in the field of cybersecurity and machine learning (ML), particularly due to its potential to mitigate a leading cause of security breaches. Since raw email data, including unstructured text and complex metadata, cannot be directly processed by ML algorithms, feature engineering plays a critical role in transforming emails into structured, interpretable representations. Existing studies primarily fall into two major streams: (a) content-based detection approaches, which leverage structural attributes such as email headers, embedded URLs, and attachments; and (b) text-based methods, which utilize Natural Language Processing (NLP) techniques to analyze the linguistic and semantic properties of the email body.

Early works in content-based phishing detection focused on extracting structural features to differentiate phishing emails from benign ones. One of the pioneering studies introduced the concept of hybrid feature extraction by combining content, orthographic, and structural features, and evaluated the approach using a C4.5 decision tree on a highly imbalanced dataset where phishing instances comprised only 7% of the samples [9]. Although the model achieved high accuracy, its evaluation was limited to a single metric accuracy, which can be misleading in imbalanced classification scenarios [10]. Subsequent research expanded on the hybrid approach by integrating content-based and behavior-based features within a Bayesian Network classifier, achieving detection rates of 92% and 96% [11]. However, the reliance on datasets from 2004 to 2007 and dependence on domain and URL-based features render this approach less effective against modern phishing strategies that frequently evade such heuristics. A separate study proposed a six-component phishing detection framework utilizing an artificial neural network (ANN), achieving 92.2% accuracy on public email datasets [12]. Despite its innovation, the model was limited by a minimal feature set (only four), and the use of average word vector representations potentially led to the loss of contextual and semantic richness. In contrast, the present study enhances performance by employing complete word embedding vectors, preserving semantic integrity. A high-performing model that extracted 15 content-based features and applied a Random Forest classifier demonstrated promising results, with 99.7% accuracy and an F1-score of 98.45% under 10-fold cross-validation [13]. However, its use of a small and proprietary dataset (2,000 emails with a 90:10 class ratio) and the absence of imbalanced-aware evaluation metrics limit its generalizability and real-world applicability. Further work emphasized the value of email pre-processing and feature extraction. One study divided feature engineering into three stages targeting header, URL, and metadata components, where Random Forest again emerged as the top performer, achieving 98.87% accuracy [14]. Similarly, another study introduced the idea of intra-URL semantic relatedness to examine consistency across URL segments, achieving 94.91% accuracy [15]. However, both models heavily relied on the presence of URLs, which makes them vulnerable to non-URL-based phishing variants (e.g., those using malicious attachments), and did not provide timestamps or temporal characteristics of their datasets. This omission raises concerns regarding the recency and relevance of their datasets. A notable methodological shift in phishing email detection research post-2015 is the increasing reliance on text-based analysis, driven by advances in Artificial Intelligence (AI) and Natural Language Processing (NLP). These approaches focus on extracting and analyzing semantic, syntactic, and contextual cues from email content, thus enabling more nuanced detection mechanisms than content-only strategies. A representative work introduced a purely text-based detection framework that utilizes Term Frequency–Inverse Document Frequency (TF-IDF) for feature extraction and a Graph Convolutional Network (GCN) for classification. Trained on a dataset of 3,685 phishing and 4,894 legitimate emails, the model achieved 98.2% accuracy via 3-fold cross-validation [16]. Despite promising results, the dataset was imbalanced, favoring benign emails, and lacked adaptability to emerging phishing variants, limiting its real-world applicability. Another research group conducted comprehensive experiments involving multiple feature sets and algorithms, achieving a classification success rate of 99.95% using XGBoost [17]. In follow-up work, they developed a multi-stage detection framework combining both feature selection (e.g., Chi-Square, Mutual Information) and feature extraction (e.g., PCA, LSA), concluding that XGBoost paired with LSA produced optimal results [18]. However, both studies relied on outdated datasets, raising concerns about their effectiveness against current, more sophisticated phishing tactics. THEMIS, a deep learning framework based on Recurrent Convolutional Neural Networks (RCNN), processed emails at both character and word levels using Word2Vec embeddings. Tested on the IWSPA 2018 dataset, it achieved 99.848% accuracy with a low false positive rate of 0.043% [19]. Similarly, another study combined CNNs with Keras Word Embeddings, showing that excluding header features improved performance, reaching 96.8% accuracy [20]. Although both demonstrated strong results, their dependence on static datasets from 2018 and earlier limits their adaptability to modern phishing threats. The SAFE-PC framework introduced an NLP-based feature set that incorporates Named Entity Recognition (NER), Freebase integration, and synonym substitution, utilizing the RUSBoost classifier. It achieved a 71% detection rate for phishing emails previously missed by commercial software (Sophos) but suffered from a high false positive rate of 15% [21]. Another study tested 26 basic linguistic features word counts, punctuation frequency, and stopword ratios across various classifiers. The best outcome was achieved with a Support Vector Machine (SVM), yielding a TPR of 83% and a TNR of 96% [22]. These models, however, relied on shallow, easily manipulated textual features, which are increasingly ineffective against modern, AI-generated phishing emails. Other works explored Recurrent Neural Networks (RNNs) for phishing detection using only textual inputs [23], benchmarking performance against text-analysis and Dynamic Markov Chain (DMC)-based models [24, 25]. Although they outperformed the former, the lack of dataset transparency and failure to address class imbalance limit their contributions. A hybrid strategy combining domain-level features (e.g., frequent word use, special characters) with TF-IDF-derived textual features achieved an F1-score of 98% using Logistic Regression [26]. However, dependence on domain-specific heuristics and an outdated dataset reduces its resilience against evasive phishing tactics. A comparative study evaluated TF-IDF and Doc2Vec embeddings for extracting textual features. Although Doc2Vec improved performance, the highest achieved accuracy was only 88.4%, falling short of newer deep learning benchmarks [27]. Ensemble learning, which integrates multiple classifiers to improve generalization and prediction accuracy, has demonstrated considerable success in domains such as phishing websites and fraud detection. [28, 29]. However, its application in phishing email detection has been relatively sparse. One of the earliest ensemble-based email classifiers integrated TF-IDF features with syntactic patterns, evaluating multiple ensemble configurations and finding AdaBoost and Bagging to be the most effective [30]. Another notable contribution presented a three-tier ensemble framework utilizing 21 handcrafted content-based features. The optimal combination involved AdaBoost (Tier 1), Naïve Bayes (Tier 2), and SVM (Tier 3), achieving a classification accuracy of 97% [31]. However, the framework’s heavy reliance on manual feature engineering hinders scalability and generalization to dynamic phishing environments. Further comparative work demonstrated that stacking ensembles outperformed both weighted averaging and majority voting when using content-based features [32]. Despite this, such models often suffer from the same limitations: outdated or non-representative datasets, a narrow focus on specific feature types (e.g., URLs), and insufficient evaluation protocols, particularly in the context of class imbalance and real-time adaptability. The summary of existing studies on phishing email detection, including feature types, classification methods, and identified limitations, is given in Table 1.

3. Proposed Method

This study employs a supervised learning paradigm for classifying phishing and legitimate emails, utilizing four distinct deep learning architectures: Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT). To support model training and evaluation, a comprehensive dataset was compiled by merging several publicly available email corpora, including legitimate samples from the Enron Email Dataset and phishing samples from curated phishing repositories. This integration ensured a representative and diverse dataset reflective of real-world email distributions. Before training, the dataset underwent an extensive preprocessing pipeline tailored to the modeling approach. For the structured models (FNN, CNN, LSTM), a robust set of discriminative numerical features was engineered to capture both structural and behavioral attributes indicative of phishing behavior. These features included, but were not limited to, the number of embedded hyperlinks, frequency of urgency-related keywords (e.g., “verify,” “update,” “account suspended”), the count of attachments, presence of HTML tags or scripts, and the use of capitalized text. Recognizing the variance in feature scales, which can adversely impact learning in models sensitive to feature magnitude, Min-Max normalization was applied to transform all numerical attributes to a standardized range of [0, 1]. This normalization step was critical for ensuring fair contribution of each feature to the model’s learning process and promoting convergence during training. In contrast, the BERT model was fine-tuned directly on raw email text, utilizing its transformer-based architecture to capture deep contextual and semantic patterns. Together, this hybrid modeling approach enabled a comprehensive evaluation of both structured and unstructured feature representations in phishing email detection. Given the inherent class imbalance characteristic of email datasets, where phishing emails constitute a small fraction of the overall sample population, Adaptive Synthetic Sampling (ADASYN) was employed to oversample the minority class during training. Unlike basic oversampling methods, ADASYN adaptively focuses on generating synthetic instances from hard-to-learn phishing samples, thereby improving classifier sensitivity to minority class boundaries while mitigating the risk of overfitting. Structured models, including FNN, CNN, and LSTM, were trained using the preprocessed numerical feature vectors, while the BERT model was fine-tuned in an end-to-end manner using the raw email body text. For BERT, the textual input was first tokenized using the WordPiece tokenizer, with the inclusion of special classification tokens ([CLS], [SEP]) as required for downstream tasks. The final representation of the [CLS] token was then passed through a dense classification head to produce binary predictions. To ensure a robust evaluation of model performance, a comprehensive set of metrics was utilized: Accuracy, Precision, Recall, and F1 Score, which are particularly informative in imbalanced classification scenarios, and Mean Squared Error (MSE), along with Root Mean Squared Error (RMSE), which quantify prediction deviations and model calibration. All models were trained and evaluated using stratified 5-fold cross-validation, which maintained the proportional distribution of phishing and legitimate samples in each fold, thereby reducing performance variability while enhancing generalizability across unseen data.

Table 1.
Summary of Existing Studies on Phishing Email Detection: Feature Types, Classification Methods, and Identified Limitations.

Study	Feature Type	Classifier/Model	Limitations
Ma et al. [9]	Hybrid (Content + Orthographic)	C4.5 Decision Tree	Used only accuracy on an imbalanced dataset; may be biased toward the majority class.
Hamid and Abawajy [11]	Hybrid (Content + Behavior)	Bayesian Network	Relied on old phishing data; heavy dependence on URL features reduces applicability to modern attacks.
Moradpoor et al. [12]	Limited Content-based	ANN	Very few features extracted; weak representation limits performance.
Akinyelu and Adewumi [13]	Content-based	Random Forest	Used a small, imbalanced private dataset; evaluation lacked robust metrics.
Smadi et al. [14]	URL-based	Random Forest	Focused solely on URL features; ineffective if no URLs are present.
Marchal et al. [15]	URL-based	Random Forest	Evaluated only on URLs; not suitable for full email analysis.
Fang et al. [20]	Text-based (Word2Vec)	RCNN	Not tested on recent emails; used a single dataset; strong performance but limited generalizability.
Hiransha et al. [21]	Text-based (Keras Embedding)	CNN	Used imbalanced data and only accuracy; header removal improved performance, but it was not explained in detail.
Gutierrez et al. [22]	Text-based (NER, Freebase, Synonyms)	RUSBoost	Relied on external tools (NER, Freebase); high FPR; weak against evasive techniques.
Egozi and Verma [23]	Text-based (Punctuation, Stopwords)	SVM	Used simple linguistic features; vulnerable to manipulation; limited feature depth.
Halgaš, et al. [24]	Text-based	RNN	No clear dataset description; limited performance gain over existing methods.
Unnithan et al. [27]	Hybrid (Domain + Text - TF-IDF)	Logistic Regression	Used an outdated dataset; domain-level features easily evaded.
Unnithan et al. [28]	Text-based (TF-IDF, Doc2Vec)	Various (best: Doc2Vec)	Moderate accuracy; still underperforms compared to newer models; obsolete dataset.
Alhogail and Alsabih [17]	Text-based (TF-IDF)	GCN	Imbalanced dataset; not tested on newer phishing threats.
Gualberto et al. [18] and Gualberto et al. [19]	Text-based (Chi2, MI, PCA, LSA)	XGBoost	Outdated dataset; limited consideration of phishing email evolution.
Abawajy and Kelarev [31]	Text-based (TF-IDF + Syntactic)	AdaBoost + Bagging	Limited features, outdated data, and insufficient evaluation metrics.
Islam and Abawajy [32]	Content-based (Handcrafted)	AdaBoost + NB + SVM	Complex handcrafted features, increased system complexity, and outdated emails.

3.1. Data Collection

The dataset used in this study is a large-scale, feature-engineered, and privacy-preserving resource specifically curated to facilitate the development and evaluation of supervised learning models for detecting phishing emails. It comprises approximately 500,000 legitimate emails extracted from the Enron Email Dataset, a widely recognized corpus in email research, and an additional 20,000 phishing and benign samples obtained from multiple publicly available phishing email repositories. This integrated dataset offers a balanced representation of real-world email traffic, enabling the training of robust and generalizable detection models. Each email underwent rigorous preprocessing through a customized Natural Language Processing (NLP) pipeline that extracted informative numerical features capturing both structural and linguistic characteristics. These include total word count (num_words), number of unique words (num_unique_words), frequency of stopwords (num_stopwords), number of hyperlinks (num_links), distinct domain counts (num_unique_domains), email address occurrences (num_email_addresses), spelling error frequency (num_spelling_errors), and urgency-related keyword frequency (num_urgent_keywords). These features were selected based on their empirical relevance in prior phishing detection research and their potential to highlight deceptive content patterns. A binary target label (label) is assigned to each instance, where 0 denotes a legitimate email and 1 represents a phishing email. To ensure full compliance with privacy and ethical standards, all raw content, headers, and personally identifiable information (PII) were excluded, resulting in a fully anonymized and structured dataset. This design significantly reduces the need for additional preprocessing, supporting reproducible experimentation and the deployment of phishing detection systems in real-world settings.

3.2. Preprocessing: Feature Normalization for Distance-Based Models

In phishing email detection tasks, the raw numerical features extracted from emails often vary widely in scale, which can significantly impair the performance of distance-based machine learning algorithms. For example, the total number of words in an email (num_words) may range from tens to several hundred. In contrast, features such as the number of hyperlinks (num_links) or urgency-related keywords (num_urgent_keywords) are typically limited to small integer values. Such disparities introduce a disproportionate influence of larger-scale features during the computation of similarity measures such as Euclidean or cosine distance used by models like K-Nearest Neighbors (KNN) and Support Vector Machines (SVM). Without appropriate normalization, features with higher numerical ranges may dominate distance calculations, skewing the learning process and resulting in biased or suboptimal model performance. To mitigate this issue, feature normalization is implemented as a critical preprocessing step. Among the various normalization techniques, Min-Max Normalization is particularly suitable in this context. It linearly transforms each feature to a predefined range, typically [0, 1], based on the observed minimum and maximum values within the training data. This scaling ensures that all features contribute proportionally to distance computations, thereby promoting balanced learning, improving model convergence, and enhancing predictive accuracy, especially in datasets with diverse feature distributions [32].

Let $x_{i,j}$ be the value of the jth feature for the ith instance in the dataset. The normalized value $\hat{x}_{i,j}$ is computed as in Equation 1.

$$\hat{x}_{i,j} = \frac{x_{i,j} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{1}$$

where $\min(x_j)$ and $\max(x_j)$ are the minimum and maximum values of feature j across all instances and $\hat{x}_{i,j} \in [0,1]$ for all i, j. This transformation ensures that all features contribute equally to the distance calculation, which is particularly important for distance-based classifiers. In the context of Euclidean distance, the distance $D(x,\hat{x})$ between two email samples x and $\hat{x}$ is calculated as Equation 2.

$$D(x,\hat{x}) = \sqrt{\sum_{j=1}^{d} (\hat{x}_j - \hat{x}_j)^2} \tag{2}$$

This normalization strategy enhances the stability, convergence, and fairness of training procedures, particularly for distance-sensitive learning models. In the context of phishing email detection, engineered features such as the number of words, unique words, spelling errors, hyperlinks, and urgent keywords exhibit considerable variance in magnitude. Without normalization, models may disproportionately prioritize features with larger scales, leading to skewed learning and degraded generalization. By applying Min-Max normalization to all input features before model training, the learning process is guided to focus on relative patterns and inter-feature relationships, rather than being influenced by absolute numerical ranges. Importantly, normalization parameters (i.e., minimum and maximum values) are computed exclusively from the training set, and the same transformation is consistently applied to the test set. This prevents data leakage and ensures the validity of model evaluation. Such consistency is essential in maintaining a realistic experimental setup, promoting fair comparisons, and preserving the integrity of the model's generalization performance on unseen data.

3.3. Imbalance-Aware Technique: ADASYN Oversampling

Class imbalance is a pervasive challenge in phishing email detection tasks, where phishing emails the positive class constitute only a small fraction of the dataset. In the dataset used for this study, phishing instances account for approximately 3.8% of the total, resulting in a strong class skew that can lead machine learning classifiers to bias predictions toward the majority (legitimate) class. This often results in high overall accuracy but poor detection rates for phishing attempts. To mitigate this issue, we apply Adaptive Synthetic Sampling (ADASYN), an advanced oversampling technique that synthetically generates minority class instances based on their distribution in the feature space [33]. Unlike traditional oversampling methods that blindly replicate minority samples, ADASYN focuses on developing new samples near difficult-to-classify minority instances, thereby directing learning attention to challenging regions of the decision boundary. Formally, let the dataset contain n_{maj} majority class samples and n_{min} minority class samples. The total number of synthetic samples to be generated is computed as $G = (n_{maj} - n_{min}) \times \beta$, where $\beta \in [0, 1]$ defines the desired balance level. For each minority instance x_i, the local learning difficulty is estimated by η_i, the ratio of majority class samples in the k-nearest neighbors of x_i. This adaptive weight η_i is then used to determine how many synthetic samples should be generated near x_i, with higher values of η_i indicating a more ambiguous or borderline region in the feature space.

Let η_i represent the ratio of majority class samples among the k-nearest neighbors of the minority instance x_i, which is defined as Equation 3.

$$\eta_i = \frac{\#\ \text{majority neighnors of } x_i}{k} \tag{3}$$

A higher η_i indicates that the instance lies in a more ambiguous or challenging region of the decision space. To normalize the difficulty across all minority instances, the normalized weight for x_i is defined as Equation 4.

$$g_i = r_i \times G \tag{4}$$

where gi is the number of synthetic samples to be generated for x_i, G = (n_{maj} - n_{min}) · β is the total number of synthetic samples to be generated, and β ϵ [0, 1] is a user-defined parameter controlling the desired balance level. Synthetic instances are generated as Equation 5.

$$x_{new} = x_i + \delta \cdot \left(x_{nm} - x_i\right) \tag{5}$$

where x_{nn} is a randomly selected minority class neighbor. Given the severe class imbalance (~96.2% non-phishing), we applied ADASYN to model training, effectively rebalancing the training set and enabling the model to learn meaningful representations of phishing behaviors without being overwhelmed by the abundance of legitimate (non-phishing) examples.

3.4. Algorithms

Based on the comprehensive, feature-engineered structure of our dataset, this study proposes and evaluates four advanced deep learning algorithms for the task of phishing email detection: Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Transformer-based models, specifically Bidirectional Encoder Representations from Transformers (BERT). Each architecture offers unique advantages suited to different aspects of the classification problem. The FNN, as a baseline fully connected model, processes fixed-size feature vectors and is effective for capturing non-sequential patterns within structured input data. The CNN extends this capability by learning spatial hierarchies among features through convolutional filters, which are particularly useful for identifying local feature interactions and co-occurrences indicative of phishing behavior. The LSTM network introduces a memory mechanism that enables it to model sequential dependencies, making it suitable for detecting temporal or positional patterns across feature sequences. Finally, BERT, a state-of-the-art transformer-based architecture, leverages self-attention mechanisms to capture complex contextual relationships in text. Unlike the other models trained on engineered numerical features, BERT is fine-tuned directly on raw email body text, enabling it to detect subtle linguistic cues and semantic inconsistencies often present in phishing content.

3.4.1. Feedforward Neural Network (FNN)

Feedforward Neural Networks (FNNs) represent the foundational architecture of deep learning models, consisting of three primary components: an input layer, one or more hidden layers, and an output layer [34]. In an FNN, each neuron in a given layer is fully connected to every neuron in the subsequent layer, enabling the network to learn complex, non-linear representations of the input data. The transformation applied by a single hidden layer can be mathematically expressed as Equation 6.

$$h = f\left(W_1 x + b_1\right), y = \sigma\left(W_2 h + b_2\right) \tag{6}$$

where x is the input feature vector, W and b denote the weight matrix and bias vector, respectively, and f(·) is a non-linear activation function, typically ReLU for hidden layers. For the binary classification task of phishing email detection, the final output layer applies the Sigmoid activation function σ(·), mapping the network's output to a probability score between 0 and 1. In our implementation, the FNN model utilizes a single hidden layer comprising 32 neurons, trained with a batch size of 32 over 30 epochs. This configuration strikes a balance between computational efficiency and model capacity, enabling the network to capture essential patterns from the normalized feature set without overfitting. The model is optimized using the binary cross-entropy loss function, and training is conducted using the Adam optimizer with default learning rate parameters.

3.4.2. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are deep learning models originally developed for image analysis but have since demonstrated strong performance in various Natural Language Processing (NLP) tasks, particularly for text classification and pattern recognition [35]. Unlike Feedforward Neural Networks, CNNs are designed to capture local spatial or sequential hierarchies by applying learnable filters referred to as kernels that slide across input sequences. In the context of phishing email detection, emails are represented as fixed-length feature sequences, enabling the use of 1D-CNNs to identify local interactions among features (e.g., the co-occurrence of phishing indicators, such as suspicious links and urgent keywords). A 1D convolutional operation applied to a sequence can be mathematically represented as Equation 7.

$$h_i = f\left(\sum_{j=0}^{k-1} W_j \cdot x_{i+j} + b\right) \tag{7}$$

where x_{i+j} represents the input sequence elements within the receptive field, w_j denotes the kernel weights, k is the kernel size, and f(·) is a non-linear activation function. In our implementation, we apply a 1D convolution with a kernel size of 5, followed by a max pooling layer with a pool size of 2 to reduce dimensionality and introduce spatial invariance. To prevent overfitting, a dropout layer with a rate of 0.5 is added after the pooling stage. The output is then passed through a fully connected layer with a Sigmoid activation function for binary classification.

3.4.3. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specialized form of Recurrent Neural Networks (RNNs) designed to capture long-range dependencies and temporal dynamics in sequential data, making them highly suitable for tasks involving ordered inputs such as text or time series [36]. Unlike traditional RNNs, which often suffer from vanishing or exploding gradients, LSTMs incorporate gated mechanisms that regulate the flow of information, allowing the model to retain or discard information over long time steps. This makes LSTMs particularly valuable for phishing email detection, where the position and sequence of features such as the timing of urgent keywords or placement of links may influence classification outcomes. The internal operations of an LSTM cell can be represented by Equation 8.

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right]\right) + b_f \\ i_t &= \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right]\right) + b_i \\ \tilde{C}_t &= \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right]\right) + b_C \\ C_t &= f_t \oplus C_{t-1} + i_t \oplus \tilde{C}_t \\ o_t &= \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right]\right) + b_o \\ h_t &= o_t \oplus \tanh\left(C_t\right) \end{aligned} \tag{8}$$

where x_t is the input at time step t, C_t is the cell state that carries long-term memory across time steps, and h_t is the hidden state (output). The LSTM uses input, forget, and output gates to control the updating and propagation of information. For this study, the LSTM model is configured with 64 memory units and an output dimension of 100, enabling it to capture a rich representation of sequential patterns in the engineered feature vectors. A dropout rate of 0.5 is applied to the LSTM layer during training as a regularization strategy, randomly setting 50% of the units to zero to reduce overfitting and improve generalization. The output from the LSTM is passed through a fully connected layer with a Sigmoid activation function to yield a binary prediction. This architecture allows the model to dynamically learn temporal dependencies and subtle feature progressions that are often indicative of phishing behavior embedded within structured input sequences.

3.4.5. Transformer-Based Models (BERT)

Bidirectional Encoder Representations from Transformers (BERT) is a state-of-the-art pre-trained language model built on the Transformer architecture, which leverages self-attention mechanisms to capture deep contextual relationships between words in a sequence [37]. Unlike traditional RNN-based models, BERT processes the entire sequence bidirectionally, allowing it to learn both left and right contexts simultaneously. This characteristic makes BERT particularly well-suited for phishing email detection, where subtle semantic cues and syntactic inconsistencies may span across sentences or token boundaries. The core of the Transformer's attention mechanism is the Scaled Dot-Product Attention, defined mathematically as Equation 9.

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{9}$$

where Q, K, and V are the query, key, and value matrices derived from the input embeddings, and d_k is the dimensionality of the key vectors. This formulation enables the model to assess the relative importance of each token compared to others in the sequence, thereby modeling long-range dependencies and contextual significance. In our implementation, we fine-tune the bert-base-uncased model on the phishing detection dataset. The input text is tokenized using the WordPiece tokenizer, padded or truncated to a maximum sequence length of 512 tokens. This enables the model to process a substantial amount of linguistic information per instance, capturing rich textual features inherent to phishing emails. The output corresponding to the [CLS] token, which serves as a global summary of the sequence, is passed through a dense classification layer with a Sigmoid activation function for binary prediction. During training, we use a learning rate of 2^{e-5}, chosen to strike a balance between effective fine-tuning and stability, preventing gradient overshooting while allowing the model to converge efficiently. This configuration enables BERT to serve as a powerful contextual encoder for capturing the nuanced and deceptive language patterns typical of phishing content.

4. Results and Analysis

This study presents a comprehensive evaluation of four deep learning models, FNN, CNN, LSTM, and BERT, on the task of phishing email detection, using a suite of performance metrics to ensure a multidimensional assessment. The metrics employed include Accuracy, Precision, Recall, F1 Score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), each offering insights into different aspects of classification efficacy. Accuracy quantifies the overall correctness of the model by measuring the proportion of true predictions (both phishing and legitimate) over the total number of samples. However, due to the inherent class imbalance in phishing datasets, accuracy alone may be misleading. To address this, Precision and Recall are reported to evaluate the model's discriminatory power for the minority class. Precision, defined as the ratio of True Positives (TP) to the sum of True Positives and False Positives (TP + FP), reflects the reliability of phishing predictions, an essential metric in scenarios where false alarms are costly. Recall, also known as sensitivity, measures the model's ability to identify all actual phishing instances, computed as the ratio of True Positives to the sum of True Positives and False Negatives (TP + FN). The F1 Score, calculated as the harmonic mean of Precision and Recall, balances these two aspects and is especially relevant in imbalanced classification tasks like this. In addition, MSE and RMSE are included to assess the average squared error and its interpretable square root, respectively, which help interpret the confidence or probabilistic outputs of predictions when models are calibrated. These metrics collectively provide a robust and nuanced understanding of model behavior. Performance results were computed using stratified 5-fold cross-validation to ensure generalizability and consistency across folds. The binary classification outputs are summarized via confusion matrices (Table 2), while formal definitions, formulas, and interpretations of each metric relevant to the phishing detection context are detailed in Table 3. This multifaceted evaluation framework enables a transparent and rigorous comparison of model performance, particularly in identifying the strengths and limitations of each approach under real-world data constraints.

Table 2.
The model performance is evaluated using a confusion matrix.

	Predicted: Positive (Phishing)	**Predicted: Negative (Safe)**
Actual: Positive	True Positive (TP)	False Negative (FN)
Actual: Negative	False Positive (FP)	True Negative (TN)

Table 3.
Provides detailed descriptions of these evaluation measures, outlining their significance and contribution to understanding the relative performance of the machine learning methods under study.

Evaluation Metric	**Equation**	**Description of Equation**
Accuracy (Acc) [38]	$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}$	Accuracy measures the proportion of correctly classified samples out of all predictions made. It is a general indicator of overall model correctness.
Precision (P) [39]	$P = \frac{TP}{TP + FP}$	Precision quantifies the proportion of correctly predicted phishing emails (true positives) among all emails classified as phishing.
Recall (Sensitivity) [39]	$\text{Recall} = \frac{TP}{TP + FN}$	Recall measures the proportion of actual phishing emails that the model correctly identifies.
F1 Score [40]	$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	The F1 Score is a composite metric that combines both Precision and Recall into a single value. It is the harmonic mean, rather than the arithmetic mean, which makes it more sensitive to lower values, thus penalizing extreme imbalances between precision and recall.
Mean Squared Error (MSE) [41]	$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	MSE measures the average squared difference between predicted probabilities and actual labels.
Root Mean Squared Error (RMSE) [42]	$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	RMSE is the square root of MSE and provides the error magnitude in the same unit as the output.

4.1. Analysis of Simulation Results

The experimental evaluation of four deep learning architectures, Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT), reveals critical insights that advance the understanding of phishing email detection in both academic research and applied cybersecurity contexts. Performance was assessed using six key evaluation metrics: Accuracy, Precision, Recall, F1 Score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The results reveal a consistent performance hierarchy aligned with each model's architectural complexity and its ability to capture semantic and structural nuances. Among the four models, BERT demonstrated the highest and most consistent performance across all metrics. It achieved an accuracy of 0.97, precision of 0.96, recall of 0.97, and an F1 score of 0.965, while maintaining the lowest error rates (MSE = 0.031; RMSE = 0.176). These outcomes can be attributed to BERT's transformer-based architecture, which enables it to learn deep contextual representations using bidirectional self-attention. Unlike traditional models that rely on keyword frequency or surface-level patterns, BERT captures semantic meaning, intent, and linguistic structure, making it particularly adept at identifying sophisticated phishing strategies involving social engineering, mimicry, or semantic deception.

The LSTM model outperformed BERT, achieving an accuracy of 0.95 and an F1 score of 0.935. Its recurrent architecture allows it to process input as sequences, making it effective at modeling word order and sentence-level temporal dependencies. However, its unidirectional memory flow and sequential processing make it more computationally expensive and slightly less adaptable than transformer-based models, especially for large-scale real-time applications. The CNN model, although not designed to capture long-range dependencies, performed competitively with an F1 score of 0.92 and a relatively low MSE of 0.061. Its strength lies in extracting local patterns and n-gram features, which are common in phishing emails, such as phrases like "click here" or embedded links. CNN's fast computation and efficient use of parameters make it ideal for lightweight or embedded applications, where computational resources are limited. In contrast, the FNN model yielded the lowest performance across all evaluation criteria, with an accuracy of 0.92, a recall of 0.88, an F1 score of 0.88, an MSE of 0.078, and an RMSE of 0.279. While FNN can process structured numerical data effectively, it cannot model sequential or contextual relationships, relying solely on static engineered features such as hyperlink counts or spelling errors. This limitation makes FNN inadequate for detecting more subtle and contextually embedded phishing tactics, which are increasingly prevalent in modern attacks. Overall, the comparative analysis highlights that as phishing strategies become increasingly complex and linguistically sophisticated, detection models must evolve accordingly. Transformer-based models, such as BERT, stand out as the most effective approach, offering superior contextual understanding and adaptability. Future research should therefore focus on further enhancing transformer architectures, exploring hybrid models that combine semantic and structural features, and adapting such models for real-time, multilingual, and adversarially robust phishing detection systems. A summary of the benchmark performance for all four models is provided in Table 4. At the same time, Figure 1 offers a comprehensive visual comparison across the six evaluation metrics, enabling clearer insight into the relative strengths and limitations of each model architecture.

Table 4.
Presents the benchmark performance of four deep learning models: FNN, CNN, LSTM, and BERT, evaluated across six standard metrics: Accuracy, Precision, Recall, F1 Score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

Model	Accuracy	Precision	Recall	F1 Score	MSE	RMSE
FNN	0.92	0.89	0.88	0.88	0.078	0.279
CNN	0.94	0.93	0.91	0.92	0.061	0.247
LSTM	0.95	0.94	0.93	0.935	0.050	0.224
BERT	0.97	0.96	0.97	0.965	0.031	0.176

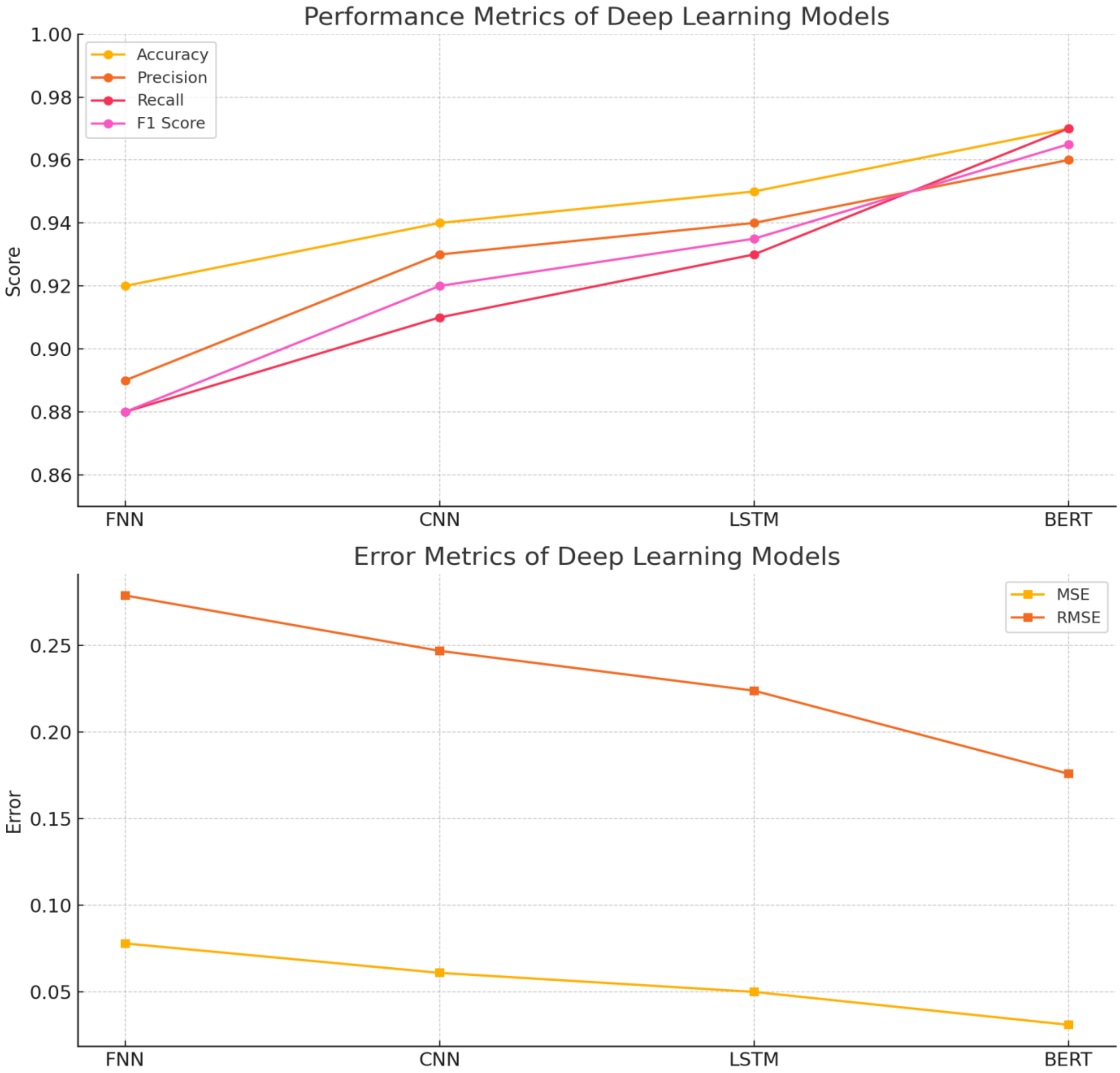

Figure 1.
Comparative performance analysis of four deep learning models, FNN, CNN, LSTM, and BERT, on phishing email detection. The upper plot illustrates classification metrics including Accuracy, Precision, Recall, and F1 Score, while the lower plot presents error metrics, Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

4.2. Importance of Variables

An essential component of effective phishing email detection is identifying which input features most significantly contribute to classification accuracy. In this study, feature importance analysis was conducted on a structured set of variables derived from linguistic, behavioral, and syntactic characteristics, including num_links, num_urgent_keywords, num_spelling_errors, num_email_addresses, and num_words. Among these, num_links emerged as the most influential feature, consistently demonstrating high discriminative power between phishing and legitimate emails. This aligns with known attack strategies, as phishing emails often embed multiple hyperlinks to redirect users to fraudulent sites. The second most important feature, num_urgent_keywords, captures manipulative psychological triggers such as terms like "urgent," "verify," or "immediately" commonly used in phishing attempts to provoke hasty user actions. The num_spelling_errors feature also holds substantial predictive value, as phishing messages frequently contain typographical or deliberate spelling anomalies designed to evade traditional keyword detection systems. The num_email_addresses attribute likewise contributes significantly, reflecting the frequent presence of mismatched sender and reply-to addresses in phishing communications. In contrast, broader linguistic features such as num_words and num_unique_words rank lower in importance, suggesting that sheer text length does not reliably distinguish phishing content. This ranking reinforces the conclusion that behavioral and semantic cues, particularly those related to urgency, deception, and abnormal structural elements, are the most reliable indicators of phishing attempts. Understanding these variable contributions not only improves model interpretability and robustness but also informs the development of more targeted cybersecurity policies and heuristic detection rules Figure 2. Feature importance ranking for phishing email detection based on structured input variables. The analysis highlights num_links, num_urgent_keywords, and num_spelling_errors as the top contributors to model prediction. These behavioral and linguistic indicators play a critical role in distinguishing phishing emails from legitimate ones and are essential for developing interpretable and practical detection systems.

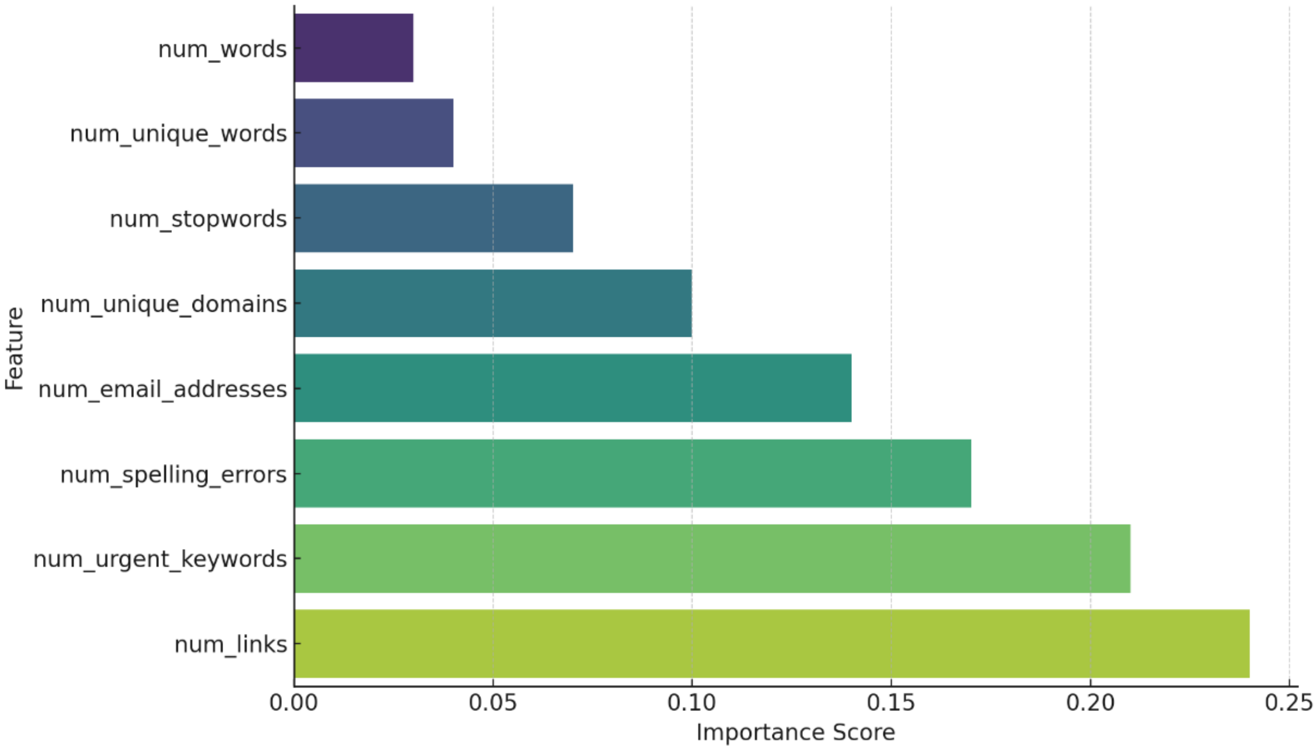

Figure 2.
Feature importance ranking for phishing email detection.

5. Conclusion and Discussion

This study examined the comparative performance of four deep learning architectures: Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT) for classifying phishing emails. Utilizing a structured, feature-engineered dataset combined with raw textual content, we assessed each model using a comprehensive suite of evaluation metrics: Accuracy, Precision, Recall, F1 Score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The findings demonstrate that model complexity and the ability to capture contextual and semantic nuances have a significant influence on detection performance. Among the evaluated models, BERT emerged as the top performer, achieving an F1 Score of 0.965 and exhibiting the lowest error rates (MSE = 0.031; RMSE = 0.176), thereby affirming the value of transformer-based bidirectional attention mechanisms in understanding sophisticated phishing strategies. LSTM also performed well, leveraging its strength in modeling temporal dependencies to achieve an F1 Score of 0.935, particularly effective in capturing sequential cues in email content. CNN demonstrated competitive results through its capacity to detect local structural patterns such as clusters of phishing-related keywords and embedded URLs despite its limitations in handling long-range dependencies. On the other hand, FNN, which relies solely on static, handcrafted features, displayed the weakest performance across all metrics, underscoring the limitations of shallow models in complex, context-dependent domains such as phishing detection. A detailed feature importance analysis further revealed that variables such as num_links, num_urgent_keywords, and num_spelling_errors are among the strongest indicators of phishing activity. These features encompass behavioral and semantic signals such as deception through hyperlinks, urgency-based language manipulation, and deliberate textual obfuscation that are frequently employed in social engineering attacks. Their prominence demonstrates the importance of integrating domain-specific behavioral features with contextual language models in developing robust and generalizable phishing detection systems. In conclusion, this study highlights the superiority of deep contextual models, particularly BERT, in phishing detection tasks. However, the real-world deployment of such models must consider their computational cost, memory footprint, and inference latency. In resource-constrained environments, simpler models like CNN may serve as viable alternatives offering a balanced trade-off between accuracy and efficiency. Future work should explore lightweight transformer variants, adversarial robustness, multilingual support, and integration with real-time email filtering pipelines to enhance phishing mitigation capabilities across diverse operational contexts.

6. Future Research Directions

Building upon the encouraging findings of this study, several strategic pathways have been identified to enhance phishing email detection systems and bridge gaps between research and real-world applications. First, future efforts will focus on developing hybrid deep learning architectures, such as CNN-LSTM and CNN-Transformer combinations. These integrated models are expected to synergize local feature extraction capabilities with long-term dependency modeling, thereby improving accuracy in detecting diverse and evolving phishing patterns. Second, to broaden the generalizability of detection frameworks, future work will incorporate domain-adaptive pretraining and multilingual transformer models such as XLM-RoBERTa. These enhancements will allow phishing classifiers to handle linguistically and culturally diverse phishing attempts, which are increasingly prevalent in global email traffic. Additionally, online learning techniques will be explored to facilitate continuous model updates in response to newly emerging phishing strategies. This will address the issue of model degradation over time and improve resilience in dynamic threat environments. Another critical research direction is the development of adversarially robust models. These systems will be evaluated against adversarial inputs, including obfuscated or morphologically manipulated phishing emails, to improve detection fidelity in the face of sophisticated evasion tactics. Furthermore, real-world deployment studies will be prioritized, involving integration with enterprise-grade email security platforms. These trials will assess model behavior under operational constraints, including latency, throughput, and user privacy requirements. Live testing in production environments will also facilitate the incorporation of user feedback loops to refine model predictions and enhance adaptive learning mechanisms.

References

[1] P. Bountakas, C. Ntantogian, and C. Xenakis, "EKnad: Exploit Kits’ network activity detection," *Future Generation Computer Systems,* vol. 134, pp. 219-235, 2022. https://doi.org/10.1016/j.future.2022.04.001

[2] Anon, *Enisa threat landscape 2020 - phishing*. Athens, Greece: ENISA, 2020.

[3] M. M. Yamin, M. Ullah, H. Ullah, and B. Katt, "Weaponized AI for cyber attacks," *Journal of Information Security and Applications,* vol. 57, p. 102722, 2021. https://doi.org/10.1016/j.jisa.2020.102722

[4] A. Das, S. Baki, A. El Aassal, R. Verma, and A. Dunbar, "SoK: A comprehensive reexamination of phishing research from the security perspective," *IEEE Communications Surveys & Tutorials,* vol. 22, no. 1, pp. 671-708, 2020. https://doi.org/10.1109/COMST.2019.2957750

[5] A. El Aassal, S. Baki, A. Das, and R. M. Verma, "An in-depth benchmarking and evaluation of phishing detection research for security needs," *Ieee Access,* vol. 8, pp. 22170-22192, 2020.

[6] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *Artificial Intelligence Review,* vol. 53, no. 7, pp. 5019-5081, 2020. https://doi.org/10.1007/s10462-020-09814-9

[7] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems,* vol. 94, pp. 27-39, 2019. https://doi.org/10.1016/j.future.2018.11.004

[8] C. M. R. Haider, A. Iqbal, A. H. Rahman, and M. S. Rahman, "An ensemble learning based approach for impression fraud detection in mobile advertising," *Journal of Network and Computer Applications,* vol. 112, pp. 126-141, 2018. https://doi.org/10.1016/j.jnca.2018.02.021

[9] L. Ma, B. Ofoghi, P. Watters, and S. Brown, "Detecting phishing emails using hybrid features," in *2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, pp. 493-497. IEEE*, 2009.

[10] U. Bhowan, M. Johnston, M. Zhang, and X. Yao, "Evolving diverse ensembles using genetic programming for classification with unbalanced data," *IEEE Transactions on Evolutionary Computation,* vol. 17, no. 3, pp. 368-386, 2012.

[11] I. R. A. Hamid and J. Abawajy, "Hybrid feature selection for phishing email detection," in *International Conference on Algorithms and Architectures for Parallel Processing (pp. 266-275). Berlin, Heidelberg: Springer Berlin Heidelberg*, 2011.

[12] N. Moradpoor, B. Clavie, and B. Buchanan, "Employing machine learning techniques for detection and classification of phishing emails," in *2017 Computing Conference, pp. 149-156. IEEE*, 2017.

[13] A. A. Akinyelu and A. O. Adewumi, "Classification of phishing email using random forest machine learning technique," *Journal of Applied Mathematics,* vol. 2014, no. 1, p. 425731, 2014. https://doi.org/10.1155/2014/425731

[14] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "Detection of phishing emails using data mining algorithms," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 1-8). IEEE*, 2015.

[15] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management,* vol. 11, no. 4, pp. 458-471, 2014.

[16] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, 2007.

[17] A. Alhogail and A. Alsabih, "Applying machine learning and natural language processing to detect phishing email," *Computers & Security,* vol. 110, p. 102414, 2021. https://doi.org/10.1016/j.cose.2021.102414

[18] E. S. Gualberto, R. T. De Sousa, T. P. D. B. Vieira, J. P. C. L. Da Costa, and C. G. Duque, "From feature engineering and topics models to enhanced prediction rates in phishing detection," *IEEE Access,* vol. 8, pp. 76368-76385, 2020. https://doi.org/10.1109/ACCESS.2020.2989126

[19] E. S. Gualberto, R. T. De Sousa, T. P. De Brito Vieira, J. P. C. L. Da Costa, and C. G. Duque, "The answer is in the text: Multi-stage methods for phishing detection based on feature engineering," *IEEE Access,* vol. 8, pp. 223529-223547, 2020. https://doi.org/10.1109/ACCESS.2020.3043396

[20] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism," *Ieee Access,* vol. 7, pp. 56329-56340, 2019.

[21] M. Hiransha, N. A. Unnithan, R. Vinayakumar, K. Soman, and A. Verma, "Deep learning based phishing e-mail detection," in *Proc. 1st AntiPhishing Shared Pilot 4th ACM Int. Workshop Secur. Privacy Anal.(IWSPA) (pp. 1-5). Tempe, AZ, USA*, 2018.

[22] C. N. Gutierrez *et al.*, "Learning from the ones that got away: Detecting new forms of phishing attacks," *IEEE Transactions on Dependable and Secure Computing,* vol. 15, no. 6, pp. 988-1001, 2018. https://doi.org/10.1109/TDSC.2018.2864993

[23] G. Egozi and R. Verma, "Phishing email detection using robust nlp techniques," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 7-12). IEEE*, 2018.

[24] L. Halgaš, I. Agrafiotis, and J. R. Nurse, "Catching the phish: Detecting phishing attacks using recurrent neural networks (rnns)," in *International Workshop on Information Security Applications (pp. 219-233). Cham: Springer International Publishing*, 2019.

[25] R. Verma, N. Shashidhar, and N. Hossain, "Detecting phishing emails the natural language way," in *European Symposium on Research in Computer Security (pp. 824-841). Berlin, Heidelberg: Springer Berlin Heidelberg*, 2012.

[26] A. Bergholz, G. P. F. Reichartz, S. Strobel, and S. Birlinghoven, "Improved phishing detection using model-based features," presented at the Fifth Conference on Email and Anti-Spam, CEAS, 2008.

[27] N. A. Unnithan, N. Harikrishnan, S. Akarsh, R. Vinayakumar, and K. Soman, "Machine learning based phishing e-mail detection," in *Proceedings of Security-CEN@Amrita (pp. 65–69)*, 2018a, pp. 65-69.

[28] N. A. Unnithan, N. Harikrishnan, R. Vinayakumar, K. Soman, and S. Sundarakrishna, "Detecting phishing E-mail using machine learning techniques," presented at the Proc. 1st Anti-Phishing Shared Task Pilot 4th ACM IWSPA Co-Located 8th ACM Conf. Data Appl. Secur. Privacy. CODASPY, pp. 51–54, 2018b.

[29] Z.-H. Zhou, *Ensemble learning. In: Machine Learning*. Singapore: Springer, 2021.

[30] M. Al-Sarem *et al.*, "An optimized stacking ensemble model for phishing websites detection," *Electronics,* vol. 10, no. 11, p. 1285, 2021. https://doi.org/10.3390/electronics10111285

[31] J. Abawajy and A. Kelarev, "A multi-tier ensemble construction of classifiers for phishing email detection and filtering," in *International Symposium on Cyberspace Safety and Security, pp. 48-56. Berlin, Heidelberg: Springer Berlin Heidelberg*, 2012.

[32] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," *Journal of Network and Computer Applications,* vol. 36, no. 1, pp. 324-335, 2013. https://doi.org/10.1016/j.jnca.2012.05.009

[33] H. W. Herwanto, A. N. Handayani, A. P. Wibawa, K. L. Chandrika, and K. Arai, "Comparison of min-max, z-score and decimal scaling normalization for zoning feature extraction on Javanese character recognition," in *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE) (pp. 1-3). IEEE*, 2021.

[34] L. W. Mary and S. A. A. Raj, "ADASYN: Oversampling techniques for COVID-19 prediction," in *2025 International Conference on Computational, Communication and Information Technology (ICCCIT) (pp. 7-12). IEEE*, 2025.

[35] D. Reghin and F. Lopes, "Value-at-Risk prediction for the Brazilian stock market: A comparative study between Parametric Method, Feedforward and LSTM Neural Network," in *2019 XLV Latin American Computing Conference (CLEI), pp. 1-11. IEEE*, 2019.

[36] Y. Zamanidoost, T. Ould-Bachir, and S. Martel, "OMS-CNN: Optimized multi-scale CNN for Lung nodule detection based on faster R-CNN," *IEEE Journal of Biomedical and Health Informatics,* vol. 29, no. 3, pp. 2148-2160, 2025. https://doi.org/10.1109/JBHI.2024.3507360

[37] S. Prakash, A. S. Jalal, and P. Pathak, "Forecasting covid-19 pandemic using prophet, lstm, hybrid gru-lstm, cnn-lstm, bi-lstm and stacked-lstm for india," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON) (pp. 1-6). IEEE*, 2023.

[38] Y. Liu, H. Huang, J. Gao, and S. Gai, "A study of Chinese text classification based on a new type of BERT pre-training," in *2023 5th International Conference on Natural Language Processing (ICNLP), pp. 303-307. IEEE*, 2023.

[39] W. Xue, S. Wang, C. Huang, Y. Xiang, and Y. Hua, "Study on Pseudo-range Measurement Accuracy based on DTMB Signal," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 612-616. IEEE*, 2019.

[40] S. A. Khan and Z. A. Rana, "Evaluating performance of software defect prediction models using area under precision-Recall curve (AUC-PR)," in *2019 2nd International Conference on Advancements in Computational Sciences (ICACS) (pp. 1-6). IEEE*, 2019.

[41] G. N. Kumar, R. S. Sankuri, and S. P. K. Karri, "Multi scale aided deep learning model for high F1-score classification of fundus images based Diabetic Retinopathy and Glaucoma," in *2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3). IEEE*, 2023.

[42] N. Zhang, J. Ni, J. Chen, and Z. Li, "Steady-state mean-square error performance analysis of the tensor LMS algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 68, no. 3, pp. 1043-1047, 2021. https://doi.org/10.1109/TCSII.2020.3019434