Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2026.074253

ARTICLE

A Hybrid CNN-XGBoost Framework for Phishing Email Detection Using Statistical and Semantic Features

Lin-Hui Liu[1], Dong-Jie Liu[1,*], Yin-Yan Zhang[1], Xiao-Bo Jin[2], Xiu-Cheng Wu[3] and Guang-Gang Geng[1]

[1]College of Cyber Security, Jinan University, Guangzhou, China
[2]Department of Intelligent Science, Xi'an Jiaotong-Liverpool University, Suzhou, China
[3]Coremail Technology Co. Ltd., Guangzhou, China

*Corresponding Author: Dong-Jie Liu. Email: djliu@jnu.edu.cn

Received: 06 October 2025; Accepted: 26 December 2025; Published: 12 March 2026

ABSTRACT: Phishing email detection represents a critical research challenge in cybersecurity. To address this, this paper proposes a novel Double-S (statistical-semantic) feature model based on three core entities involved in email communication: the sender, recipient, and email content. We employ strategic game theory to analyze the offensive strategies of phishing attackers and defensive strategies of protectors, extracting statistical features from these entities. We also leverage the Qwen large language model to excavate implicit semantic features (e.g., emotional manipulation and social engineering tactics) from email content. By integrating statistical and semantic features, our model achieves a robust representation of phishing emails. We introduce a hybrid detection model that integrates a convolutional neural network (CNN) module with the XGBoost (Extreme Gradient Boosting) classifier, effectively capturing local correlations in high-dimensional features. Experimental results on real-world phishing email datasets demonstrate the superiority of our approach, achieving an F1-score of 0.9587, precision of 0.9591, and recall of 0.9583, representing improvements of 1.3%–10.6% compared to state-of-the-art methods.

KEYWORDS: Phishing email detection; strategic game theory; Double-S feature model; Qwen large language model; XGBoost; convolutional neural network

1 Introduction

With the rapid advancement of internet technologies, email has established itself as an indispensable communication channel in both personal and professional spheres. However, the inherent openness of email systems has also made them primary targets for cyberattacks, with phishing emails garnering particular attention due to their widespread impact and severe consequences.

The proliferation of phishing emails can be attributed to their "three-low" characteristics: low transmission costs, minimal creation barriers, and insufficient legal penalties. First, transmission costs are extremely low, enabling attackers to distribute phishing emails to thousands of users with a simple mouse click. Second, creation barriers are minimal—even attackers without technical backgrounds can easily craft convincing phishing emails using AI tools or readily available malicious programs [1]. Finally, legal penalties remain insufficient due to regulatory challenges and evolving legal frameworks surrounding phishing attacks, resulting in minimal consequences for perpetrators even when distributing numerous malicious emails [2].

Phishing email attacks represent a form of social engineering-based fraud where attackers masquerade as trusted entities to deceive users into divulging sensitive information or achieving other malicious objectives [3]. Furthermore, phishing emails frequently serve as entry points for other cyberattacks, such as embedding ransomware to initiate extortion campaigns. These attacks pose serious threats to individuals, enterprises, and even nations. For individuals, attackers may impersonate system administrators, banking personnel, or event organizers to deceive users into providing sensitive information such as identification numbers or banking credentials. For enterprises, phishing emails may entice employees to click malicious links or install malware, potentially compromising devices and threatening internal network security [4]. More critically, phishing emails can be weaponized for political purposes, affecting national security. For instance, in March 2016, the chairman of Hillary Clinton's campaign team clicked a shortened link disguised as a Google warning email and changed his password, resulting in the theft and public disclosure of confidential files, directly impacting the presidential election outcome [4].

According to quarterly reports from email company Coremail in 2024, phishing email volumes have increased progressively from 114 million in Q1 to 210 million in Q4, totaling 641 million emails for the entire year [5]. These statistics demonstrate the extensive scope and rapid growth trajectory of phishing email threats, compelling an increasing number of security researchers to develop effective detection solutions.

Traditional phishing email detection methods primarily rely on single-dimensional feature extraction, such as blacklist-based sender reputation assessment and keyword-matching content filtering [6]. However, these approaches struggle to address increasingly sophisticated phishing attack techniques. On one hand, attackers can circumvent blacklist detection through techniques such as spoofing sender addresses and utilizing dynamic domain names [7]. On the other hand, phishing email content has become increasingly diverse, making it challenging for traditional rule-based feature extraction methods to comprehensively capture semantic characteristics [8]. For instance, attackers can leverage popular AI tools to generate highly realistic phishing email content [1], thereby bypassing keyword-based detection systems. To address these evolving phishing attacks, researchers have begun exploring more advanced detection technologies. Deep learning-based models, such as CNN and long short-term memory networks (LSTMs), can process complex linguistic structures and learn deep semantic information. These models enhance phishing email recognition capabilities by analyzing multi-dimensional features including email text content, structural characteristics, and metadata [9]. However, these solutions have not comprehensively analyzed and integrated the multi-dimensional features of emails. While the emergence of large language models (LLMs) has provided new insights for semantic feature extraction, research utilizing these models for semantic feature extraction remains limited.

To address this gap and overcome the limitations of traditional methods, this paper proposes a novel Double-S (statistical-semantic) feature model that advances beyond conventional hybrid approaches through methodological and theoretical innovations. Unlike existing hybrid architectures that merely combine different feature types, Double-S introduces a game-theoretic framework for adversarial feature engineering and hierarchical feature integration, distinguishing it as a theoretically grounded approach to phishing detection. Specifically, we analyze phishing attackers' strategies through strategic game theory to derive Nash equilibrium-based statistical features from three core entities (sender, recipient, and email content), while leveraging the Qwen large language model to excavate deep semantic patterns. Building upon this, we introduce a CNN module prior to the XGBoost classification algorithm, proposing a novel phishing email detection model aimed at improving detection accuracy. This work makes the following contributions to advance the state-of-the-art in phishing email detection:

1. Unified Double-S Feature Framework: We propose a novel Double-S feature model that advances beyond conventional hybrid approaches through game-theoretic feature selection and hierarchical integration. Unlike existing methods that empirically combine features or use simple concatenation, Double-S employs strategic game theory to derive statistically optimal feature sets and implements

structured semantic-statistical integration, capturing both low-level behavioral signals and high-level contextual deception cues within a theoretically grounded adversarial framework.

2. Game-Theoretic Feature Engineering: We propose a game-theoretic approach to statistical feature extraction, modeling the adversarial dynamics between phishing attackers and defenders to derive Nash equilibrium solutions that guide the selection of statistical features most effective for detection.
3. LLM-Driven Semantic Threat Intelligence: We develop a knowledge distillation framework utilizing large language models (LLMs) to generate high-quality dialogue samples for fine-tuning smaller models (Qwen), enabling effective semantic feature extraction from phishing emails while maintaining computational efficiency.
4. Hierarchical Feature Fusion Architecture: We design the ConvXG-Net hybrid architecture that synergistically combines CNN's feature fusion capabilities with XGBoost's classification performance, where CNN learns complex interaction patterns between statistical and semantic features, and XGBoost constructs robust decision boundaries for final phishing detection.

The remainder of this paper is organized as follows: Section 2 discusses related work, Section 3 presents the proposed methodology in detail, Section 4 provides experimental results and analysis, and finally, Section 5 concludes the paper with future research directions.

2 Related Work

With the widespread adoption of the Internet and the continuous growth of Internet users, the email user community has been expanding steadily. However, within this vast email ecosystem, malicious phishing attackers exploit identity spoofing and content masquerading techniques to distribute phishing emails indiscriminately, posing serious threats to users' information security. To address this challenge, researchers from both academia and industry have proposed numerous solutions, dedicating their efforts to effectively identifying and filtering phishing emails from massive volumes of electronic mail, making significant contributions to the construction of email security protection systems.

Blacklist-based filtering represents one of the earliest approaches to phishing email detection, which maintains lists of IP addresses previously associated with phishing emails and subsequently classifies any emails received from these IPs as phishing attempts [10]. Some researchers have also maintained databases of phishing URLs, identifying emails as phishing if their content contains previously collected malicious URLs [11]. However, attackers continuously forge their IP addresses and URLs to circumvent detection. To address the evolving nature of phishing emails, researchers subsequently proposed more sophisticated rule-based solutions that surpass blacklist approaches. These methods establish a series of expert-defined rules and calculate scores for received emails based on these rules, with emails exceeding predetermined thresholds being classified as phishing. This approach demonstrates superior generalization capabilities compared to blacklist methods [12].

With the continuous advancement of artificial intelligence, machine learning algorithms have been gradually applied to phishing email detection research, leading to significant improvements in detection accuracy. Vazhayil et al. [13] employed traditional algorithms such as Support Vector Machines (SVM), logistic regression, and random forests for phishing email detection. These approaches extract features such as email subjects and hyperlinks for detection, with limited consideration of textual features. However, attackers may also forge content to mislead detection systems [14]. With the development of Natural Language Processing (NLP) technologies, researchers have increasingly focused on textual features of emails [15]. Jain et al. [16] implemented end-to-end email classification using word embeddings and Long Short-Term Memory (LSTM) networks. Fang et al. [17] proposed the THEMIS detection model by combining email header and body features, achieving an exceptionally high accuracy of 99.848% on the

IWSPA-AP 2018 dataset. Doshi et al. [18] proposed a dual-layer architecture approach for email classification, simultaneously detecting spam and phishing emails. This method extracts features from email bodies and content, utilizing deep learning models including CNN, and Recurrent Neural Networks (RNNs) for classification. Experimental results demonstrated that this approach achieved 99.51% accuracy and an F1-score of 99.52% on the Nazario dataset. Building upon enhanced textual feature extraction, researchers have also explored innovative ensemble learning methods to combine textual and statistical features for email detection. Bountakas and Xenakis [19] proposed HELPHED (Hybrid Ensemble Learning PHishing Email Detection), a hybrid ensemble learning method that combines content and textual features using both Stacking Ensemble Learning and Soft Voting Ensemble Learning approaches, significantly improving phishing email detection performance. Their method achieved an F1-score of 99.42% and accuracy of 99.43% on the comprehensive dataset compiled from multiple publicly available sources, demonstrating exceptionally high detection accuracy and efficiency.

Concurrently, Qi et al. [20] proposed two novel algorithms: the Fisher-Markov-based Phishing Ensemble Detection (FMPED) method and the Fisher-Markov-Markov-based Phishing Ensemble Detection (FMMPED) method. These approaches effectively mitigate the impact of data imbalance on classifier performance by removing benign email samples from overlapping regions and employing Markov undersampling strategies. Experimental results demonstrated that both FMPED and FMMPED outperformed existing methods across multiple evaluation metrics including F1-score, accuracy, AUC value, G-mean, and MCC value, with the FMMPED method achieving both F1-score and accuracy of 99.45% on the HELPHED dataset.

Despite the promising results achieved by machine learning and deep learning-based NLP approaches, these methods operate primarily on surface-level textual features and cannot capture deep semantic meanings. Specifically, variations such as synonym usage and different sentence structures are difficult for machine learning-based NLP methods to detect [21].

In recent years, these models have developed rapidly, and a limited number of researchers have begun utilizing LLMs for phishing email detection tasks. The emergence of large language models like GPT-3 [22] has revolutionized natural language processing capabilities, enabling sophisticated semantic understanding that extends beyond traditional approaches. Heiding et al. [1] conducted research on both phishing email generation and detection using LLMs such as GPT, designing four specific questions to query the large language models for detection purposes, thereby demonstrating the effectiveness of LLMs in extracting semantic features from email text. Chataut et al. [23] analyzed email content and compared the performance of different LLM models, including customized versions of ChatGPT using Kaggle datasets. Recent advancements have also shown LLMs' potential in specialized domains, such as machine translation through fine-tuning techniques [24], highlighting their adaptability across diverse applications. However, research on leveraging LLMs for phishing email detection remains quite limited, with very few studies exploring the potential of large language models for semantic feature extraction in this domain, presenting significant opportunities for further investigation and innovation.

3 Methodology

3.1 Double-S Feature Model Architecture

Before delving into the technical details, we first clarify what distinguishes our Double-S model from existing hybrid architectures in phishing detection. Unlike conventional approaches that merely combine different feature types through empirical concatenation or ensemble methods, Double-S introduces four fundamental methodological and theoretical advancements:

1. Game-Theoretic Feature Derivation: Rather than empirically selecting features based on correlation analysis, Double-S employs strategic game theory to model attacker-defender dynamics. By analyzing Nash equilibria, we derive statistical features that directly correspond to optimal defensive strategies against rational adversarial behavior.

2. Hierarchical Feature Integration: Existing hybrid methods typically perform naive concatenation without considering feature interdependencies. Double-S implements structured hierarchical integration where statistical features form the foundational layer, complemented by semantic features for higher-level interpretation.

3. Advanced LLM Semantic Intelligence: While contemporary methods rely on traditional NLP techniques, Double-S harnesses large language models through knowledge distillation to extract sophisticated semantic patterns that capture nuanced social engineering tactics.

4. Theoretical Justification of Complementarity: Double-S provides rigorous theoretical grounding for feature synergy, demonstrating how statistical and semantic features capture complementary aspects of phishing threats.

These innovations establish Double-S as a theoretically grounded framework for adversarial feature engineering, rather than a simple hybrid approach.

This section presents the detailed technical implementation of our proposed Double-S feature model, focusing on the extraction mechanisms for statistical and semantic features from email data, as shown in Fig. 1.

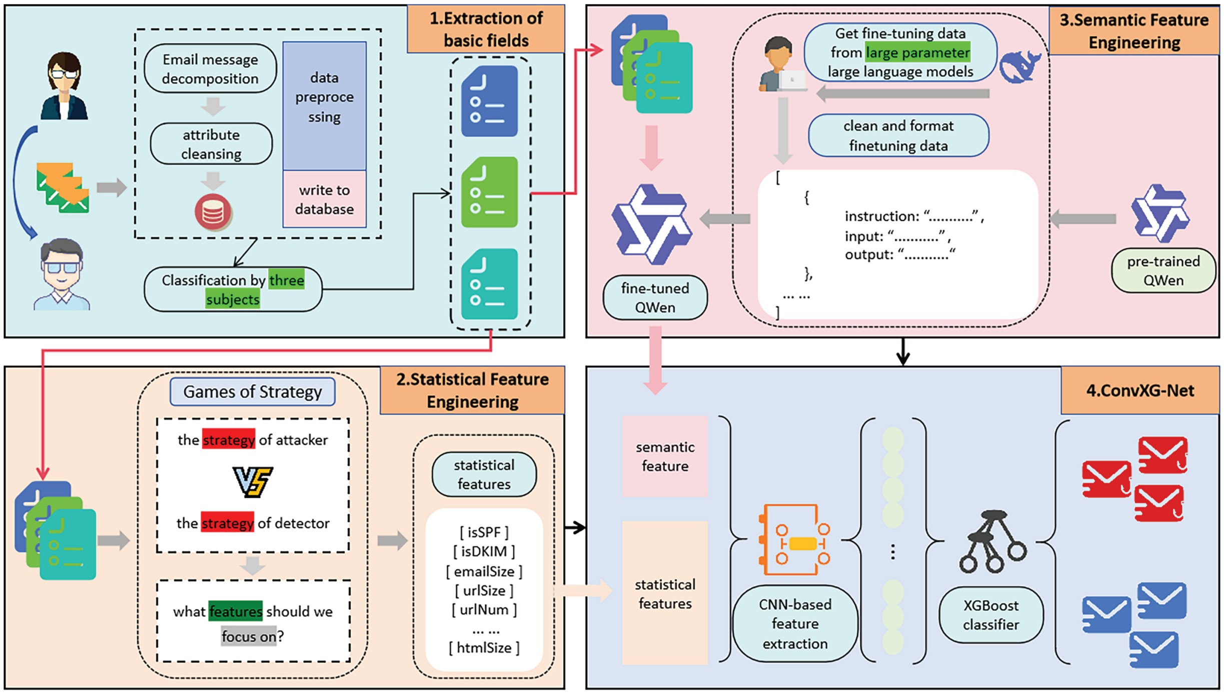


Figure 1: Double-S feature model architecture

3.1.1 Statistical Feature Extraction Based on Strategic Game Theory

In the phishing email detection scenario, we model the confrontation between defenders and attackers as a strategic game. The defender represents the email detection system, while the attacker represents the phishing email sender. Both parties aim to maximize their respective benefits.

We define the game model as follows:

- Players: Defender (D) and Attacker (A)
- Attacker strategies: $S_A = \{s_1 : Content\text{-}rich\ emails, s_2 : Simple\ content\ emails\}$
- Defender strategies: $S_D = \{d_1 : Equal\ treatment, d_2 : Vigilant\ toward\ content\text{-}rich\}$
- Payoff functions: $U_D(s_D, s_A)$ and $U_A(s_D, s_A)$

Content-rich emails (s_1) include multimedia elements (images, videos), complex HTML formatting, multiple hyperlinks, and sophisticated layouts that are commonly used in phishing attacks to increase credibility and bypass simple text-based filters. Simple content emails (s_2) contain minimal formatting and basic text content.

The payoff matrix is shown in Table 1.

Table 1: Payoff matrix for strategic game

	Content-rich (s_1)	**Simple content (s_2)**
Equal treatment (d_1)	(U_D^{low}, U_A^{high})	(U_D^{low}, U_A^{low})
Vigilant toward content-rich (d_2)	(U_D^{high}, U_A^{low})	$(U_D^{medium}, U_A^{low})$

Where the payoff parameters represent:

- $U_A^{high} > U_A^{low}$: Attackers gain higher utility from successful content-rich phishing.
- $U_D^{high} > U_D^{medium} > U_D^{low}$: Defenders achieve highest utility when vigilant against content-rich attacks.
- U_D^{medium}: Moderate defender utility when vigilant but facing simple attacks (resource over-allocation).
- The ordinal relationships ensure realistic strategic interactions.

Nash Equilibrium Analysis: Through strategic analysis, we find that the defender's dominant strategy is d_2 (vigilant toward content-rich emails), as $U_D^{high} > U_D^{low}$ and $U_D^{medium} > U_D^{low}$ regardless of attacker's choice. Given this dominant strategy, the equilibrium outcome is (s_1, d_2), indicating that rational attackers will use content-rich emails while defenders should maintain vigilance against such sophisticated attacks.

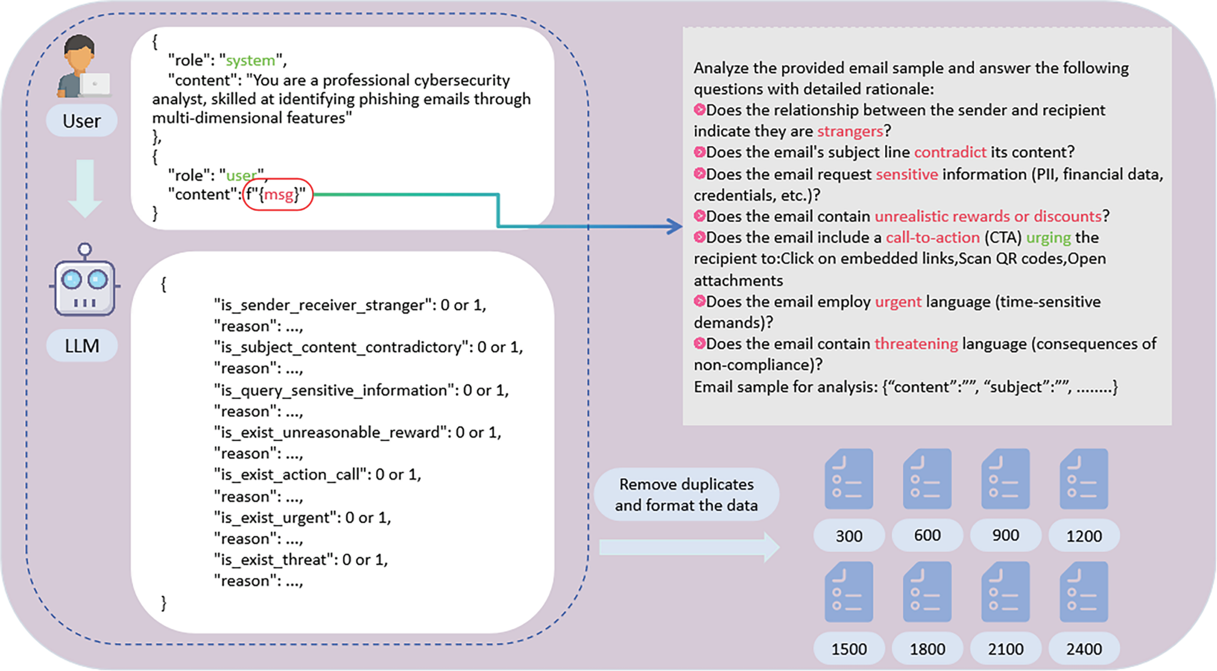
This equilibrium provides crucial strategic insight: *defense systems should prioritize detecting content-rich phishing emails*, as these represent the primary threat vector in the adversarial equilibrium.

While this game-theoretic analysis provides a conceptual framework for understanding attacker-defender dynamics, it serves as a theoretical foundation for systematic feature engineering rather than claiming quantitative precision. The ordinal relationships in the payoff structure establish strategic priorities that guide feature selection, providing a principled approach to adversarial feature design that complements empirical methods. The effectiveness of this theoretical framework is validated through our experimental results, where the game-theory-guided feature selection contributes significantly to the superior performance of our Double-S model (F1-score: 0.9587) compared to baseline approaches.

Feature Engineering Guidance: Based on the Nash equilibrium analysis, our feature extraction strategy focuses on characteristics that distinguish content-rich emails, including multimedia complexity, HTML structural sophistication, hyperlink patterns, visual presentation complexity, and attachment analysis. Through similar game-theoretic reasoning, we systematically derive other critical features, ultimately constructing a comprehensive 32-dimensional statistical feature system that aligns with the strategic equilibrium insights. The complete list of statistical features, categorized into sender features, recipient features, relationship features, and email content features, is provided in the Appendix A (Tables A1–A4).

3.1.2 Semantic Feature Extraction Based on Large Language Models

For semantic feature extraction, we employ fine-tuned large language models to capture deep semantic meanings in email content. The architecture is shown in Fig. 2.



```
{
  "role": "system",
  "content": "You are a professional cybersecurity analyst, skilled at identifying phishing emails through multi-dimensional features"
},
{
  "role": "user",
  "content": f"{msg}"
}
```

```
{
        "is_sender_receiver_stranger": 0 or 1,
        "reason": ...,
        "is_subject_content_contradictory": 0 or 1,
        "reason": ...,
        "is_query_sensitive_information": 0 or 1,
        "reason": ...,
        "is_exist_unreasonable_reward": 0 or 1,
        "reason": ...,
        "is_exist_action_call": 0 or 1,
        "reason": ...,
        "is_exist_urgent": 0 or 1,
        "reason": ...,
        "is_exist_threat": 0 or 1,
        "reason": ...,
}
```

Analyze the provided email sample and answer the following questions with detailed rationale:
- Does the relationship between the sender and recipient indicate they are strangers?
- Does the email's subject line contradict its content?
- Does the email request sensitive information (PII, financial data, credentials, etc.)?
- Does the email contain unrealistic rewards or discounts?
- Does the email include a call-to-action (CTA) urging the recipient to:Click on embedded links,Scan QR codes,Open attachments
- Does the email employ urgent language (time-sensitive demands)?
- Does the email contain threatening language (consequences of non-compliance)?

Email sample for analysis: {“content”:””, “subject”:””, ...}

Figure 2: Fine-tuned large language model architecture

We use a knowledge distillation framework to transfer knowledge from large teacher models to smaller student models, balancing performance and computational efficiency. The training samples are generated through data augmentation and manual annotation, then used to fine-tune the pre-trained language model for phishing email detection tasks.

3.2 ConvXG-Net Hybrid Model Architecture

3.2.1 Model Architecture Design Principles

ConvXG-Net implements a hierarchical two-stage architecture that synergistically combines CNN's feature fusion capabilities with XGBoost's classification performance, as shown in Fig. 3. The design principles are grounded in the following theoretical foundations:

Let the statistical feature vector be $\mathbf{X}_{\text{stat}} \in \mathbb{R}^{d_s}$ and the semantic feature vector be $\mathbf{X}_{\text{sem}} \in \mathbb{R}^{d_l}$, where d_s and d_l represent the dimensions of statistical and semantic features, respectively. The CNN feature fusion process begins with feature concatenation, which combines complementary information from both feature types to enable joint learning of cross-modal interactions. This concatenation is crucial for capturing the synergistic effects between behavioral patterns (statistical features) and contextual deception cues (semantic features), as demonstrated in our ablation studies where the combined features achieved 3.06% higher F1-score than single-modal approaches:

$$\mathbf{X}_{\text{input}} = [\mathbf{X}_{\text{stat}}; \mathbf{X}_{\text{sem}}] \tag{1}$$

$$\mathbf{F}_{\text{fused}} = f_{\text{CNN}}(\mathbf{X}_{\text{input}}) \tag{2}$$

where $f_{\text{CNN}}(\cdot)$ represents the nonlinear transformation function of CNN that learns hierarchical feature interactions, and $[\cdot;\cdot]$ denotes the feature concatenation operation along the feature dimension. The CNN transformation captures local correlations and global dependencies between statistical and semantic features, providing robust fused representations for downstream classification.

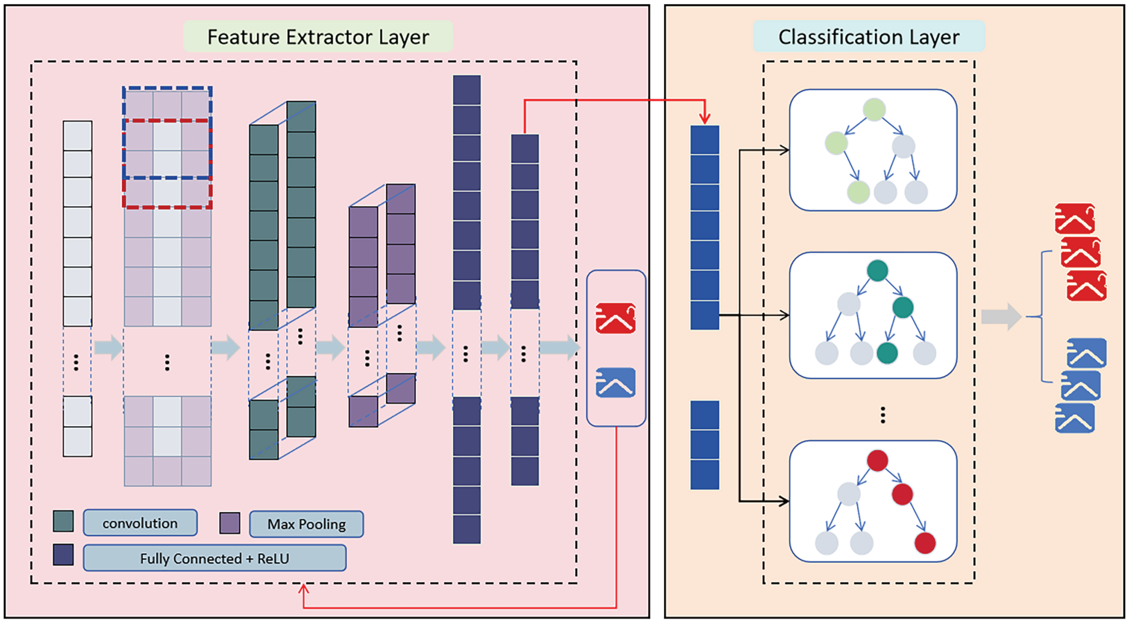


Figure 3: ConvXG-Net architecture: hybrid structure of CNN feature fusion module and XGBoost classifier

The total loss function integrates multiple objectives to ensure joint optimization of feature fusion and classification performance:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda_1 \mathcal{L}_{\text{CNN}} + \lambda_2 \mathcal{L}_{\text{XGB}} \tag{3}$$

where the classification loss employs binary cross-entropy to maximize prediction accuracy, with $\lambda_1 = 0.001$ and $\lambda_2 = 0.01$ selected through grid search to balance the contributions of different loss terms:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{4}$$

3.2.2 CNN Feature Fusion Module

The CNN module is responsible for deep fusion of statistical and semantic features in the Double-S feature model, learning interaction patterns between the two types of features. This module employs a one-dimensional convolutional structure that can effectively capture local correlations and global dependencies between features, as evidenced by the experimental results where the CNN fusion improved F1-score by 3.95% compared to direct concatenation.

First, input features undergo standardization preprocessing to ensure numerical stability and prevent feature dominance:

$$\mathbf{X}_{\text{norm}} = \text{StandardScaler}([\mathbf{X}_{\text{stat}}; \mathbf{X}_{\text{sem}}]) \tag{5}$$

Subsequently, multiple convolutional kernels are employed for hierarchical feature extraction, with $K = 64$ kernels of size 3 to capture local patterns in the concatenated feature space:

$$\mathbf{H}_i = \text{ReLU}(\mathbf{W}_i * \mathbf{X}_{\text{norm}} + b_i) \tag{6}$$

where $\mathbf{W}_i \in \mathbb{R}^{1\times3}$ is the i-th convolutional kernel, $b_i \in \mathbb{R}$ is the bias term, and $*$ denotes the valid convolution operation that preserves spatial relationships.

Next, max pooling operations are performed to reduce dimensionality and extract the most salient features, using a pooling size of 2:

$$\mathbf{P}_i = \text{MaxPool}(\mathbf{H}_i) \tag{7}$$

Finally, pooled features are concatenated and input to the fully connected layer, which learns non-linear combinations of the multi-scale features:

$$\mathbf{P} = \text{Flatten}([\mathbf{P}_1; \mathbf{P}_2; \ldots; \mathbf{P}_K]) \tag{8}$$

$$\mathbf{F}_{\text{fused}} = \text{ReLU}(\mathbf{W}_f \mathbf{P} + \mathbf{b}_f) \tag{9}$$

where $\mathbf{W}_f \in \mathbb{R}^{128\times d_f}$ and $\mathbf{b}_f \in \mathbb{R}^{128}$ are the fully connected layer weights and biases, with d_f being the flattened feature dimension after pooling operations.

3.2.3 CNN-to-XGBoost Integration Pipeline

To ensure seamless feature transfer between the CNN feature extractor and XGBoost classifier, we implement a carefully designed integration pipeline that handles dimensionality reduction, feature flattening, and input formatting:

1. Feature Dimensionality Management: The CNN feature extractor outputs a 128-dimensional feature vector per sample ($\mathbf{F}_{\text{fused}} \in \mathbb{R}^{128}$). This dimensionality is determined by the final fully connected layer size and provides a compact yet informative representation that balances computational efficiency with feature richness.

2. Feature Flattening and Normalization: Raw CNN features undergo flattening to create a one-dimensional vector suitable for XGBoost input. No additional normalization is applied at this stage, as tree-based models like XGBoost are robust to feature scaling variations. The flattened features maintain their original scale from the CNN's ReLU activation, preserving the learned non-linear relationships.

3. Batch Processing and Memory Management: Features are processed in batches of 64 samples to optimize GPU memory usage during training. Each batch of CNN-extracted features is directly fed to XGBoost without intermediate storage, ensuring efficient pipeline execution.

4. Input Format Standardization: The integrated pipeline ensures consistent input formatting between training and inference phases. During training, features from the CNN are immediately passed to XGBoost's fit method, while during prediction, the same extraction and formatting pipeline is applied to test samples.

This integration approach eliminates potential information loss during feature transfer while maintaining computational efficiency, enabling end-to-end training of the hybrid CNN-XGBoost architecture.

3.2.4 XGBoost Classifier Design

XGBoost serves as the second-stage classifier, receiving CNN-fused features $\mathbf{F}_{\text{fused}}$ for final binary classification decisions. XGBoost iteratively constructs multiple weak learners to progressively optimize prediction performance, achieving superior performance on the imbalanced phishing dataset compared to other classifiers (F1-score of 0.9344 vs. 0.9182 for MLP).

The XGBoost prediction function aggregates outputs from multiple decision trees, with $T = 100$ trees selected through cross-validation to balance complexity and performance:

$$\hat{y}_i = \sum_{k=1}^{T} f_k(\mathbf{F}_{\text{fused},i}) \tag{10}$$

where f_k represents the k-th decision tree and T is the total number of trees.

Each tree's training objective minimizes a regularized loss function to prevent overfitting while maintaining predictive power:

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{T} \Omega(f_k) \tag{11}$$

where $\ell(\cdot)$ is the binary cross-entropy loss function, and $\Omega(f_k)$ is the regularization term that penalizes model complexity:

$$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \sum_{j=1}^{T_k} w_j^2 \tag{12}$$

where $\gamma = 0.1$ is the leaf node complexity penalty coefficient, $\lambda = 1.0$ is the L2 regularization coefficient, T_k is the number of leaf nodes in the k-th tree, and w_j is the weight of the j-th leaf node.

XGBoost uses second-order Taylor expansion to approximate the loss function and guides tree construction through gradient information. The final predicted probability for phishing classification is obtained through sigmoid transformation:

$$P(\text{phishing}|\mathbf{F}_{\text{fused},i}) = \frac{1}{1 + \exp(-\hat{y}_i)} \tag{13}$$

Through CNN-XGBoost collaboration, ConvXG-Net fully leverages the advantages of the Double-S feature model: the CNN module effectively integrates complex interaction relationships between statistical and semantic features, while the XGBoost classifier constructs robust decision boundaries based on fused features, ultimately achieving high-precision phishing email detection.

4 Experiments and Evaluation

4.1 Dataset

Existing public datasets available for email detection experiments suffer from issues such as small dataset sizes and early collection times. Phishing email formats evolve with the times and continuously iterate alongside technological developments. Therefore, the forms and characteristics of phishing emails in current network environments may differ significantly from those in existing datasets. Using these datasets for research may not adequately address contemporary email detection tasks.

To address this issue, our experimental data comes from real email data collected by the servers of Coremail, a renowned Asian email company. This dataset has undergone preprocessing, with individual EML files parsed into JSON format to facilitate subsequent detection tasks. In addition to the original fields such as email subjects and content, the dataset includes behavioral features and fingerprint features statistically derived by the email server. The dataset comprises a total of 200,000 emails, consisting of 150,000 normal emails and 50,000 phishing emails, representing a realistic distribution that reflects the imbalanced nature of phishing detection scenarios in real-world applications.

4.2 Experimental Setup and Hyperparameters

Having established our dataset characteristics, we now detail the experimental setup and model configurations used in our evaluation. To ensure reproducibility and enable other researchers to replicate our results, we provide comprehensive specifications of the Double-S ConvXG-Net architecture and hyperparameters in Tables 2 and 3. Our implementation follows a two-stage pipeline: first, a CNN module performs feature fusion on the Double-S feature representations, followed by an XGBoost classifier for final prediction. All hyperparameters were selected through grid search cross-validation on a held-out validation set.

Table 2: CNN feature fusion module architecture and hyperparameters

Component	Parameter	Value
Input processing	Normalization	Min-max scaling
Convolutional layer	Kernel size	3
	Input channels	1
	Output channels	32
	Padding	1
	Activation function	ReLU
Pooling layer	Pooling type	Max pooling
	Kernel size	2
	Stride	2
Fully connected layers	Hidden layer size	32
	Output layer size	2 (binary classification)
	Activation function	ReLU (hidden), Softmax (output)
Training parameters	Optimizer	Adam
	Learning rate	0.01
	Batch size	64
	Epochs	5
	Loss function	Cross-entropy

Table 3: XGBoost classifier hyperparameters

Parameter	Value	Description
n_estimators	100	Number of boosting rounds
Max_depth	3	Maximum depth of each tree
Learning_rate	0.1	Step size shrinkage
Objective	binary:logistic	Binary classification objective
Random_state	42	Random seed for reproducibility
Booster	gbtree	Tree-based models
Gamma	0	Minimum loss reduction for split
Min_child_weight	1	Minimum sum of instance weight
Max_delta_step	0	Maximum delta step for each leaf
Subsample	1.0	Subsample ratio of training instances
Colsample_bytree	1.0	Subsample ratio of columns for each tree

The CNN-to-XGBoost integration is implemented as a seamless pipeline where CNN-extracted features (128-dimensional vectors) are directly fed to XGBoost without intermediate processing, ensuring efficient end-to-end training and consistent feature representation between training and inference phases.

4.3 Performance Metrics

The confusion matrix is a fundamental tool in the evaluation of classification models, and is defined as Table 4:

Table 4: The basic structure of the confusion matrix

	Actual positive	Actual negative
Predict positive	True Positive (TP)	False Negative (FN)
Predict negative	False Positive (FP)	True Negative (TN)

- True Positives (TP): The number of actual positive instances correctly identified by the model as positive.
- False Positives (FP): The number of actual negative instances incorrectly identified by the model as positive.
- True Negatives (TN): The number of actual negative instances correctly identified by the model as negative.
- False Negatives (FN): The number of actual positive instances incorrectly identified by the model as negative.

In evaluating the model performance, we use Precision, Recall, and F1-Score (denoted as F1) as the evaluation metrics.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

Precision represents the proportion of correctly predicted anomalies among the predicted anomaly samples, with a higher value indicating a lower false positive rate. Recall represents the proportion of correctly predicted anomalies among the true anomaly samples, with a higher value indicating a lower false negative rate. F1 combines Precision and Recall, with a higher value indicating a better overall detection performance. To highlight the comprehensive performance of the proposed method, this paper considers F1 as the primary evaluation metric.

4.4 Experimental Results and Analysis

Using the experimental setup and hyperparameters detailed in the previous section, we conducted comprehensive evaluations of our Double-S ConvXG-Net model. To ensure robust and reliable performance estimation, we employed a customized cross-validation approach that balances statistical rigor with practical considerations for large-scale email datasets.

Our validation scheme divides the dataset into ten equal parts and randomly selects seven parts (70% of data) for training and three parts (30% of data) for testing in each iteration. This process is repeated five times with different random selections, and final results are averaged across all runs.

This 7:3 split ratio is specifically chosen to reflect real-world machine learning deployment scenarios where models are typically trained on substantial datasets and evaluated on meaningful test sets. Compared to traditional 10-fold cross-validation, our approach provides more reliable performance estimates for large datasets by ensuring adequate test set sizes for stable metric computation while maintaining sufficient training data for model convergence. The random selection across iterations ensures diverse training-test combinations, mitigating potential biases from fixed fold assignments while maintaining statistical validity through multiple repetitions.

4.4.1 Comparative Experiment 1: Selection of Semantic Feature Models

First, this experiment employs CNN algorithms to perform binary classification tasks on semantic features extracted from eight fine-tuned large language models and one original large language model concatenated with statistical features, to determine the most suitable fine-tuned model—that is, the model that requires the least fine-tuning data while achieving excellent fine-tuning results. The experimental results are shown in Table 5.

Table 5: Comparative experimental results of fine-tuned models

Model	Precision	Recall	F1
Qwen-2.5 7B original	0.9347	0.8968	0.9118
Qwen-2.5 7B-300	0.9319	0.9330	0.9306
Qwen-2.5 7B-600	0.9231	0.9258	0.9234
Qwen-2.5 7B-900	0.9468	0.9388	0.9455
Qwen-2.5 7B-1200	0.9470	0.9391	0.9454
Qwen-2.5 7B-1500	0.9400	0.9473	0.9433
Qwen-2.5 7B-1800	0.9428	0.9459	0.9442
Qwen-2.5 7B-2100	0.9449	0.9404	0.9441
Qwen-2.5 7B-2400	0.9413	0.9387	0.9399

The results are visualized in Fig. 4. As can be observed, in the initial fine-tuning phase (300 samples), the model's F1-score (0.9306) improved compared to the original model (0.9118), indicating that even small amounts of data can optimize model performance. When the sample size increased to 900, model performance improved notably with an F1-score reaching 0.9455, demonstrating that increased data volume helps the model learn more stable features. At 1200 samples, the F1-score remained stable (0.9454), suggesting the model may have reached an optimal state under the current data distribution. At 1500 samples, recall (0.9473) peaked while precision slightly decreased (0.9400), likely due to increased data noise or class imbalance causing the model to favor improving recall. As the sample size further increased (1800–2400), model performance stabilized with F1-scores fluctuating around 0.944, showing no significant improvement. This indicates that under current tasks and data distribution, the model has approached its performance ceiling, and continued sample size increases yield limited marginal benefits. In this dataset, the Qwen-2.5 7B model achieved strong performance (F1 ≈ 0.945) with 900–1200 fine-tuning samples. Therefore, subsequent experiments use semantic features extracted from the model fine-tuned with 900 data points to ensure effectiveness while minimizing fine-tuning costs.

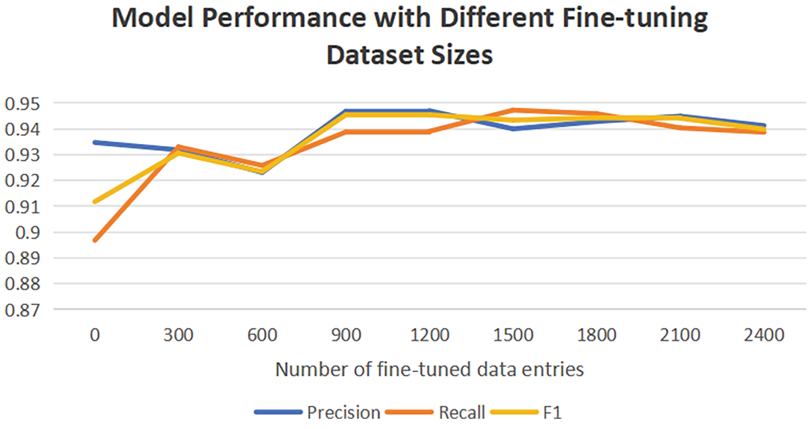


Figure 4: Comparative experimental results of different fine-tuned models

4.4.2 Baseline Comparison Experiments

To ensure fair evaluation, all baseline algorithms were implemented and evaluated on the same unified dataset (200,000 emails with 50,000 phishing and 150,000 legitimate samples) using identical training and testing splits. To fully demonstrate the superiority of our proposed model, we employ common binary classification algorithms on our constructed model for binary classification tasks. The experimental results are shown in Table 6 and visualized in Fig. 5. The experimental results indicate that traditional machine learning models (such as decision trees, logistic regression, etc.) exhibit overall weak performance, with F1-scores all below 0.90. Among them, although decision trees have relatively high recall (0.9329), they have the lowest precision (0.8187), demonstrating sensitivity to noise and tendency to overfit. In contrast, deep learning models show outstanding performance in precision (CNN: 0.9468, MLP: 0.9603), but MLP's recall significantly decreases (0.8635), indicating feature extraction bias issues. Ensemble methods (XGBoost, Random Forest) achieve F1-scores near 0.93-0.94, but still do not surpass the Double-S-ConvXG-Net method proposed in this paper.

Our proposed method demonstrates strong performance across all evaluation metrics: the F1-score reaches 0.9587, representing improvements of 1.3% 10.6% over other models; precision (0.9591) and recall (0.9583) are nearly identical, achieving excellent balance between accuracy and coverage. While these improvements may appear modest in percentage terms, they are highly meaningful in the context of large-scale email processing. With our dataset containing 200,000 emails (50,000 phishing and 150,000 legitimate), even a small 1% improvement in F1-score translates to correctly classifying hundreds of additional emails. This practical significance is particularly important in phishing detection, where reducing false positives can significantly decrease user frustration and security team workload. Similar evaluation metrics (precision, recall, F1-score) are widely adopted in phishing detection literature, as demonstrated in recent works such as those comparing HELPHED, FMMPED, and other hybrid approaches. This advantage primarily stems from the model design: the CNN branch effectively extracts local spatial features to ensure high precision, while the XGBoost branch integrates global statistical features to optimize recall. The complementary nature of these features successfully overcomes the limitations of single models, such as MLP's recall degradation. This hybrid architecture not only improves classification performance but also maintains strong model robustness.

Table 6: Comparative experimental results of different binary classification algorithm models

Model	Precision	Recall	F1
Decision tree	0.8187	0.9329	0.8720
Logistic regression	0.8110	0.8875	0.8476
K-nearest neighbors	0.9215	0.8754	0.8979
Naive bayes	0.8933	0.8756	0.8844
CNN	0.9468	0.9388	0.9455
MLP	0.9603	0.8635	0.9182
XGBoost	0.9400	0.9289	0.9344
Random forest	0.9301	0.9060	0.9179
Double-S-ConvXG-Net	0.9591	0.9583	0.9587

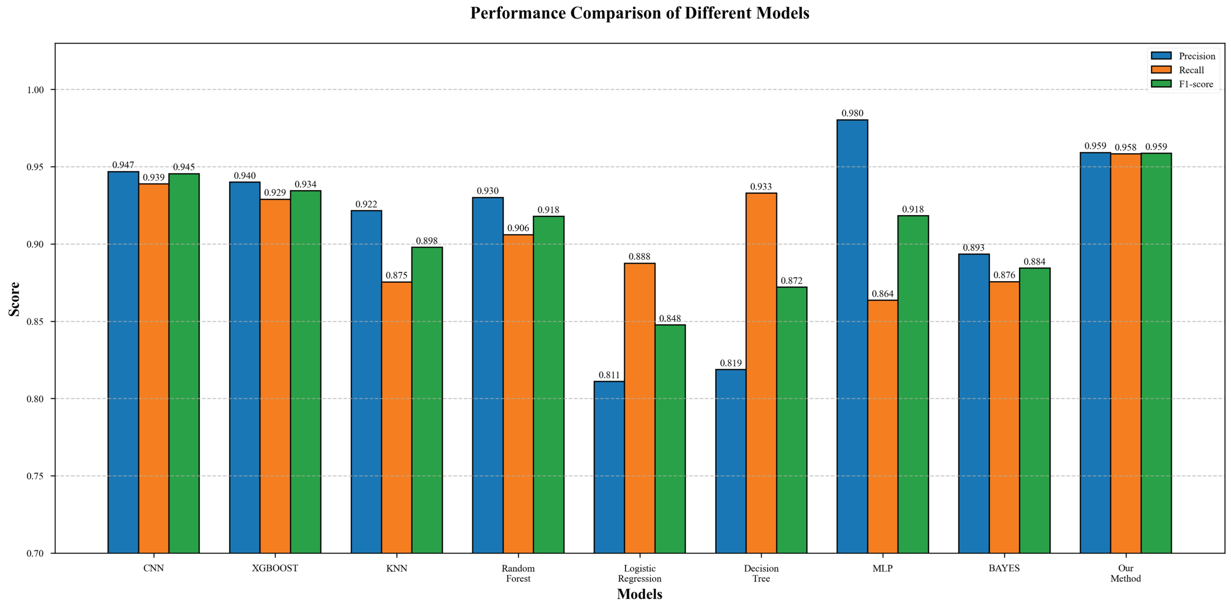


Figure 5: Comparative experimental results chart of different binary classification algorithm models

4.4.3 Comparison with State-of-the-Art Methods

To ensure fair comparison, we reproduced the implementations of state-of-the-art methods from the literature and evaluated all approaches on our unified dataset (200,000 emails with 50,000 phishing and 150,000 legitimate samples) using identical experimental settings. This approach eliminates dataset variability as a confounding factor in performance comparison. We also conducted comparative experiments with advanced feature models and algorithmic models from other researchers, as shown in Table 7.

Table 7: Comparative experimental results of different algorithmic models

Method	Precision	Recall	F1
Rastenis et al. [25]	0.9087	0.8514	0.8898
Rahman & Ullah [26]	0.9335	0.9423	0.9224
McGinley & Monroy [27]	0.9357	0.9435	0.9233
Qi et al. [20]	0.9233	0.9353	0.9323
Doshi et al. [18]	0.9329	0.9234	0.9334
Bountakas & Xenakis [19]	0.9222	0.9357	0.9388
Double-S-ConvXG-Net	0.9591	0.9583	0.9587

Rastenis et al. [25] employed traditional TF-IDF feature-based methods, which achieved certain results but were limited by shallow feature representation capabilities, with an F1 value of 0.8898. Methods based on deep semantic features (Rahman and Ullah [26] using word embeddings + CNN-BiLSTM, F1 = 0.9224; McGinley and Monroy [27] using character-level CNN, F1 = 0.9233) automatically learn semantic representations through neural networks, showing improved performance.

Recent advanced methods have demonstrated higher performance: Qi et al. [20] proposed the Fisher-Markov undersampling-based ensemble learning methods FMPED and FMMPED, specifically optimized for imbalanced datasets, achieving good results in addressing the severe imbalance between normal and phishing emails in real-world scenarios, with an F1 value of 0.9323. Doshi et al. [18] proposed a dual-layer architecture approach that simultaneously handles phishing and spam email detection tasks, integrating email content and body features for classification, with an F1 value of 0.9334, though their method primarily focuses on the breadth of email classification rather than deep semantic understanding. Bountakas and Xenakis [19] developed the HELPHED hybrid ensemble learning framework, combining hybrid representations of content and textual features using soft voting ensemble learning strategies, with an F1 value of 0.9388. Despite innovations in feature fusion, their semantic feature extraction still relies on traditional Word2Vec methods, lacking the ability to capture deep semantic relationships.

In contrast, the Double-S-ConvXG-Net model proposed in this paper innovatively integrates statistical features with deep semantic features and employs a hybrid architecture of CNN feature fusion and XGBoost classification, achieving an F1 value of 0.9587. This result not only confirms the necessity of multi-modal feature fusion for comprehensive text characteristic representation but also reveals the advantages of hybrid algorithmic architectures in simultaneously modeling local statistical patterns and global semantic dependencies in text. Analysis shows that although traditional hybrid feature methods (such as HELPHED) and dual-layer architecture methods (such as Doshi et al.'s work) have innovations in feature combination and architectural design, they are primarily limited by insufficient semantic feature extraction capabilities, unable to fully mine deep semantic information from text.

4.4.4 Ablation Studies

Finally, to characterize the role of modules in feature models and algorithmic models, we conducted ablation experiments, as shown in Table 8. The ablation experimental results indicate that the synergistic effect between feature representation and algorithmic architecture has a significant impact on model performance. From the table, it can be seen that adopting the Double-S feature model demonstrates performance improvements across various algorithmic frameworks: in the single CNN architecture, Double-S features improve the F1-score by 3.95 percentage points; in the XGBoost framework, the F1-score improves by 3.24 percentage points. Particularly noteworthy is that when combining the hybrid architecture of CNN and XGBoost, Double-S features achieve optimal performance (F1 = 0.9587), improving by 3.06 percentage points compared to single-feature methods. This validates the complementarity of statistical and semantic features—statistical features ensure model robustness while semantic features enhance fine-grained pattern recognition capabilities. Additionally, algorithm-level ablation shows that the CNN+XGBoost hybrid architecture's performance gain increases by 2.42 percentage points, greater than the improvement amplitude of single algorithms, proving that the synergistic effect of deep feature extraction and ensemble learning can more fully exploit the potential of multi-modal features. This finding provides important insights for feature engineering and model architecture design in complex classification tasks: integrating complementary feature representations with heterogeneous algorithmic frameworks can produce significant performance improvements.

Table 8: Ablation study results

Algorithm	Feature	Precision	Recall	F1
CNN	Statistical	0.9256	0.8961	0.9096
	Double-S	0.9468	0.9388	0.9455
XGBoost	Statistical	0.9390	0.8679	0.9021
	Double-S	0.9400	0.9289	0.9344
CNN+XGBoost	Statistical	0.9319	0.9243	0.9281
	Double-S	0.9591	0.9583	0.9587

5 Conclusion

As network phishing attack techniques continue to evolve, traditional email detection methods have become inadequate for addressing increasingly complex threats. This paper proposes a phishing email detection method based on the Double-S feature model and ConvXG-Net hybrid architecture. By innovatively integrating statistical and semantic features and employing a collaborative classification strategy combining CNN and XGBoost, we have significantly improved phishing email detection accuracy.

Experimental results demonstrate that our proposed method achieves strong performance across all metrics: F1-score of 0.9587, precision of 0.9591, and recall of 0.9583, showing notable improvements compared to existing advanced methods. These achievements not only validate the effectiveness of multi-modal feature fusion in text classification tasks but also demonstrate the practical value of knowledge distillation for enabling semantic understanding in resource-constrained deployment scenarios. By transferring advanced semantic capabilities from large language models to efficient hybrid architectures, our approach provides a scalable solution for phishing detection that balances performance with computational feasibility.

Future work can be further optimized in several directions: first, exploring multi-dimensional feature representations, such as graph structure-based email network features; second, investigating adaptive feature fusion mechanisms to achieve dynamic weight adjustment in different scenarios; finally, considering extending this method to multilingual environments and real-time detection scenarios to enhance the practicality and generalization capability of the approach.

Acknowledgement: The authors would like to thank Coremail Corporation for providing the email dataset used in this research. We also acknowledge the support from our institution and colleagues who provided valuable feedback during the research process.

Funding Statement: This work is supported by the National Key Research and Development Program of China (No. 2023YFB3105700).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Lin-Hui Liu and Dong-Jie Liu; methodology, Lin-Hui Liu and Dong-Jie Liu; software, Lin-Hui Liu; validation, Lin-Hui Liu, Dong-Jie Liu and Yin-Yan Zhang; formal analysis, Lin-Hui Liu and Xiao-Bo Jin; investigation, Lin-Hui Liu and Yin-Yan Zhang; resources, Xiu-Cheng Wu and Guang-Gang Geng; data curation, Lin-Hui Liu and Xiu-Cheng Wu; writing—original draft preparation, Lin-Hui Liu; writing—review and editing, Dong-Jie Liu and Yin-Yan Zhang; visualization, Lin-Hui Liu; supervision, Dong-Jie Liu and Guang-Gang Geng; project administration, Dong-Jie Liu; funding acquisition, Dong-Jie Liu and Guang-Gang Geng. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Due to the nature of this research, the original email dataset contains sensitive user information from Coremail Corporation and cannot be shared publicly for privacy and confidentiality reasons. However, to enhance research transparency and reproducibility, we have provided comprehensive methodological details and statistical characteristics:

- **Dataset Statistics:** Complete information on data volume (200,000 emails: 150,000 benign, 50,000 phishing) and class distribution to enable result contextualization.
- **Complete Model Specifications:** Detailed CNN and XGBoost architecture parameters and hyperparameters in Section 4.2, enabling exact replication of our experimental setup.
- **Feature Extraction Documentation:** Complete algorithms and normalization techniques for all 32 statistical features documented in the Appendix A.
- **Reproducible Evaluation Framework:** Detailed cross-validation setup and performance metrics for external validation of our results.

These comprehensive methodological disclosures maximize research transparency while respecting privacy constraints. Researchers can reproduce our experimental setup, validate our methodology on similar datasets, and assess the generalizability of our findings without accessing the original data.

Ethics Approval: Not applicable. This study does not involve human subjects. The email data used was anonymized and provided by Coremail Corporation with appropriate data usage agreements in place.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Appendix A Statistical Features Table

This appendix lists all statistical features used in our Double-S model. Features are extracted from preprocessed email metadata and content using standard text processing and metadata analysis techniques. Detailed extraction algorithms and source code are available upon reasonable academic request to ensure reproducibility while maintaining manuscript conciseness.

Table A1: Sender features

Feature	Description
isAuthor	Sender registration status
emailsInADayOfSender	Daily sender email volume
emailsInFifteenMinsOfSender	15-min sender email volume
emailsInADayOfSenderIp	Daily sender IP volume
emailsInFifteenMinsOfSenderIp	15-min sender IP volume
sendTime	Email transmission time
isChangeEmail	Sender email consistency
isSenderContainsStrangeCharacter	Sender address suspicious characters

Table A2: Recipient features

Feature	Description
receiverNum	Number of recipients
isZoneInDanger	Recipient region phishing risk

Table A3: Relationship features

Feature	Description
isSRFromSameDomain	Same domain (sender-recipient)
isSRFromSameZone	Same region (sender-recipient)
isEmailUrlLikeReceiver	URL domain similarity to recipient

Table A4: Email content features

Feature	Description
isSPF	SPF configuration status
isDKIM	DKIM configuration status
isDMARC	DMARC configuration status
attachNum	Attachment count
attachPngNum	Image attachment presence
attachDangerSuffixNum	Executable attachment presence
htmlSize	HTML content length
htmlImgVideoNum	HTML image tag count
htmlUrlNum	HTML URL link count
urlNum	Total URL count
urlLength	Total URL length
isUrlUsuallyUsedAsPhishing	URL personal site indicator
isUrlHttps	URL HTTPS status
contentSize	Text content size
size	Total email size
subjectSize	Subject length

(Continued)

Table A4 (continued)

Feature	Description
imageFingerprintCNT	Image fingerprint count
attachmentFingerprintCNT	Attachment fingerprint count
subjectFingerprintCNT	Subject fingerprint count

References

1. Heiding F, Schneier B, Vishwanath A, Bernstein J, Park PS. Devising and detecting phishing emails using large language models. IEEE Access. 2024;12:42131–46. doi:10.1109/access.2024.3375882.
2. Biró G, Kiss M. Phishing and some possibilities of its prevention. J Financ Econ. 2024;11(4):371–91.
3. Sturman D, Auton JC, Morrison BW. Security awareness, decision style, knowledge, and phishing email detection: moderated mediation analyses. Comput Secur. 2025;148(1):104129. doi:10.1016/j.cose.2024.104129.
4. Jiaping M, Yangwen X, Tao M. Analysis of phishing emails in social engineering attacks. Res Inf Secur. 2021;7(2):166.
5. King LMD. DeepSeek unlocks insights: the 2024 Q4 enterprise email security report [Internet]. [cited 2025 Feb 7]. Available from: https://www.mailabc.cn/blog/2025/02/07/deepseek.
6. Fette I, Sadeh N, Tomasic A. Learning to detect phishing emails. In: WWW '07: Proceedings of the 16th International Conference on World Wide Web. New York, NY, USA: ACM; 2007. p. 649–56.
7. Yue Z. CANTINA: a content-based approach to detecting phishing web sites. In: WWW '07: Proceedings of the 16th International Conference on World Wide Web. New York, NY, USA: ACM; 2007. p. 639–48.
8. Abu-Nimeh S, Nappa D, Wang X, Nair S. A comparison of machine learning techniques for phishing detection. In: Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit. Piscataway, NJ, USA: IEEE; 2007. p. 60–9.
9. Divakarla U, Chandrasekaran K. Predicting phishing emails and websites to fight cybersecurity threats using machine learning algorithms. In: 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON). Piscataway, NJ, USA: IEEE; 2023. p. 1–10.
10. Sankhwar S, Pandey D. A comparative analysis of anti-phishing mechanisms: email phishing. Int J Adv Res Comput Sci. 2017;8(3):567–74.
11. Chiba D, Akiyama M, Yagi T, Hato K, Mori T, Goto S. DomainChroma: building actionable threat intelligence from malicious domain names. Comput Secur. 2018;77(1):138–61. doi:10.1016/j.cose.2018.03.013.
12. Surwade AU. Blocking Phishing e-mail by extracting header information of e-mails. In: 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC). Piscataway, NJ, USA: IEEE; 2020. p. 151–5.
13. Vazhayil A, Harikrishnan N, Vinayakumar R, Soman K, Verma A. PED-ML: phishing email detection using classical machine learning techniques. In: Proceedings of the 1st AntiPhishing Shared Pilot at 4th ACM International Workshop on Security and Privacy Analytics (IWSPA 2018); 2018 Mar 21; Tempe, AZ, USA. p. 1–8.
14. Castillo E, Dhaduvai S, Liu P, Thakur KS, Dalton A, Strzalkowski T. Email threat detection using distinct neural network approaches. In: Proceedings for the First International Workshop on Social Threats in Online Conversations: Understanding and Management. Stroudsburg, PA, USA: ACL; 2020. p. 48–55.
15. Das A, Baki S, El Aassal A, Verma R, Dunbar A. SoK: a comprehensive reexamination of phishing research from the security perspective. IEEE Commun Surv Tutor. 2019;22(1):671–708. doi:10.1109/comst.2019.2957750.
16. Jain G, Sharma M, Agarwal B. Optimizing semantic LSTM for spam detection. Int J Inf Technol. 2019;11:239–50. doi:10.1007/s41870-018-0157-5.
17. Fang Y, Zhang C, Huang C, Liu L, Yang Y. Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism. IEEE Access. 2019;7:56329–40. doi:10.1109/access.2019.2913705.

18. Doshi J, Parmar K, Sanghavi R, Shekokar N. A comprehensive dual-layer architecture for phishing and spam email detection. Comput Secur. 2023;133(1):103378. doi:10.1016/j.cose.2023.103378.

19. Bountakas P, Xenakis C. Helphed: hybrid ensemble learning phishing email detection. J Netw Comput Appl. 2023;210:103545. doi:10.1016/j.jnca.2022.103545.

20. Qi Q, Wang Z, Xu Y, Fang Y, Wang C. Enhancing phishing email detection through ensemble learning and undersampling. Appl Sci. 2023;13(15):8756. doi:10.3390/app13158756.

21. Gutierrez CN, Kim T, Della Corte R, Avery J, Goldwasser D, Cinque M, et al. Learning from the ones that got away: detecting new forms of phishing attacks. IEEE Trans Dependable Secur Comput. 2018;15(6):988–1001. doi:10.1109/tdsc.2018.2864993.

22. Floridi L, Chiriatti M. GPT-3: its nature, scope, limits, and consequences. Minds Mach. 2020;30:681–94. doi:10.1007/s11023-020-09548-1.

23. Chataut R, Gyawali PK, Usman Y. Can ai keep you safe? A study of large language models for phishing detection. In: 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC). Piscataway, NJ, USA: IEEE; 2024. p. 0548–54.

24. Elshin D, Karpachev N, Gruzdev B, Golovanov I, Ivanov G, Antonov A, et al. From general LLM to translation: how we dramatically improve translation quality using human evaluation data for LLM finetuning. In: Proceedings of the Ninth Conference on Machine Translation. Stroudsburg, PA, USA: ACL; 2024. p. 247–52.

25. Rastenis J, Ramanauskaitė S, Suzdalev I, Tunaitytė K, Janulevičius J, Čenys A. Multi-language spam/phishing classification by email body text: toward automated security incident investigation. Electronics. 2021;10(6):668. doi:10.3390/electronics10060668.

26. Rahman SE, Ullah S. Email spam detection using bidirectional long short term memory with convolutional neural network. In: 2020 IEEE Region 10 Symposium (TENSYMP). Piscataway, NJ, USA: IEEE; 2020. p. 1307–11.

27. McGinley C, Monroy SAS. Convolutional neural network optimization for phishing email classification. In: 2021 IEEE International Conference on Big Data (Big Data). Piscataway, NJ, USA: IEEE; 2021. p. 5609–13.