Phishing Email Detection Using Deep Learning

Authors: Anthony Ward, Divya Kamila, William Brennan

Abstract

Email phishing is one of the most common types of cyber threats that use social engineering to manipulate users into divulging vital information. Although machine learning has experienced moderate success in the detection of such emails, it has been challenged in adapting to the changing tactics used in phishing emails, which use obscured language patterns. This project proposes assessing whether deep learning models are capable of identifying phishing emails on a standardized set of emails.

Three neural models, namely the Hybrid BiLSTM-DNN, the LSTM-GRU Hybrid, and the charCNN-BERT, are assessed within a common preprocessing and evaluation environment. The models harness the power of modeling context within sequences, with attention, in order to address the semantic content as well as stylistic anomalies that are evident within typical phishing emails.

Experimental findings on a dataset of more than 145,000 emails show that all models perform well, with the best result obtained by the charCNN-BERT, which has an F1-score of 0.994, with an accuracy of 99.0 percent. The LSTM & GRU hybrid model has high recall, which is useful in ensuring a reduced number of false negatives. The overall findings show that the models are useful in developing a real-world solution for the problem of phishing emails.

1. Introduction

Email remains a critical part of digital infrastructure, but it is also one of the major attack vectors in phishing campaigns that exploit human rather than technological vulnerabilities. Different from traditional spam, phishing emails are crafted to convincingly mimic real organizations, their language evolves over time, and they evade static detection rules. Thus, even state-of-the-art filtering systems continue to let through a non-trivial fraction of malicious emails and expose the users and institutions to financial loss, identity theft, and data breaches.

Recent breakthroughs in deep learning have radically changed natural language processing, allowing models to represent contextual, sequential, and semantic information from text in ways that have not been possible before with traditional machine learning methods.

In this project, we carry out a systematic and controlled study of three state-of-the-art deep learning architectures for the detection of phishing emails: a Hybrid BiLSTM + DNN, an LSTM–GRU Hybrid, and a charCNN-BERT. All models are trained and evaluated on the same large-scale curated dataset, following identical preprocessing steps and a unified experimental framework. This allows for a direct comparison of their strengths, weaknesses, and trade-offs, regarding their accuracy, recall, generalization, and computational efficiency.

Our completed results demonstrate that deep learning models can achieve highly reliable phishing detection, with all evaluated architectures reaching near state-of-the-art performance. Collectively, this project validates deep learning as a powerful and adaptable solution to phishing detection while providing a clear empirical comparison to guide future research and applications.

2. Related Work

Recent research has explored deep learning architectures for phishing detection using natural language processing techniques. Hina et al. (2021) proposed an LSTM–GRU hybrid model for multi-class email forensics, demonstrating strong generalization and reduced overfitting. Fang et al. (2019) introduced an RCNN with attention to capture both local and contextual patterns in phishing emails, particularly effective against obfuscated text. Krishnamoorthy et al. (2024) proposed a Hybrid BiLSTM + DNN incorporating emotional-lexicon analysis to improve interpretability.

Our approach differs by reproducing and directly comparing these architectures under a consistent dataset, preprocessing pipeline, our work provides a clearer empirical assessment of their relative strengths than prior isolated studies.

3. Data

The primary dataset used in this project is a merged corpus consisting of the Enron Email Dataset (benign emails) and a curated phishing dataset from Zenodo. This combined dataset contains 145,639 labeled emails, with approximately 68,000 legitimate and 77,000 phishing messages. The dataset is split into training (70%), validation (20%), and test (10%) subsets using stratified sampling.

Ten raw CSV files were standardized into a uniform schema. Irrelevant metadata fields were removed, retaining only subject, body, URLs, and binary labels (0 = legitimate, 1 = phishing). The subject and body fields were concatenated into a single text input. Text was normalized, tokenized, integer-encoded, and padded or truncated to a fixed length of 256 tokens. Word-level tokenization was used across all models, while the charCNN-BERT with Attention additionally employed character-level tokenization to detect obfuscation patterns.

To evaluate generalization, models were tested on a separate Kaggle phishing dataset consisting of 10,000 synthetically generated emails (4,000 legitimate, 6,000 phishing). No fine-tuning was performed on this dataset, ensuring a true blind inference scenario. Output and Metrics All models perform binary classification and output a single phishing probability. Binary cross-entropy loss was used for training. Evaluation metrics include accuracy, precision, recall, F1-score, and confusion matrices, with emphasis on recall due to the high cost of false negatives.

4. Methods

We implemented and compared three distinct deep learning architectures. Each model addresses the problem of phishing detection by leveraging different properties of sequential data: long-term dependencies, hierarchical feature extraction, and dual-stream processing (semantic vs. lexical).

1. LSTM–GRU Hybrid Model

This architecture combines Long Short-Term Memory (LSTM) units and Gated Recurrent Units (GRU) to balance the capture of long-term semantic dependencies with computational efficiency.

- Architecture: The model begins with an embedding layer (128-dimensional) followed by a LSTM layer (128 units) to capture global context. This is fed into a GRU layer (64 units), which refines the sequence representation by modeling shorter-range patterns. The output is processed by a dense layer with ReLU activation and Dropout (0.3) for regularization, ending in a sigmoid output neuron.

- Training: The model was trained using the Adam optimizer (learning rate = 1e-3) and binary cross-entropy loss. Early stopping with a patience of 3 epochs was used to prevent overfitting, with the model converging rapidly in 5 epochs.

2. charCNN-BERT (Dual-Input Model)

To address the issue of "obfuscated text" (e.g., misspellings or leetspeak often found in phishing), this model employs a dual-stream approach.

- Architecture:
    - ○ Word Stream: A DistilBERT transformer encoder processes word tokens to capture high-level semantic meaning and context via self-attention mechanisms.
    - ○ Character Stream: A 1D Convolutional Neural Network (CNN) with kernel sizes of 3, 5, and 7 processes character-level embeddings. This stream is designed to detect localized anomalies and lexical patterns (like .c0m instead of .com).
- Fusion: The [CLS] token from DistilBERT and the global max-pooled vector from the CNN are concatenated and passed to a Multi-Layer Perceptron (MLP) for final classification.

3. BiLSTM + DNN with Emotion Analysis

This model is a reproduction and extension of the architecture by Krishnamoorthy et al. (2024), focusing on context-aware classification and interpretability.

- Architecture: It utilizes a 100-dimensional embedding layer feeding into a Bidirectional LSTM (128 units, 2 directions). The bidirectional nature allows the model to understand the context of a word based on both preceding and succeeding text. The output is fed into a fully connected Deep Neural Network (DNN) head with ReLU activation and Dropout (0.2).
- Emotion Extension: Post-training, we implemented a lexicon-based emotion analysis module to verify if phishing emails exhibit distinct emotional signatures (e.g., high anticipation or fear) compared to legitimate emails.

5. Experiments

All three models achieved exceptional performance on the held-out test set from the primary corpus.

1. LSTM-GRU

Each email contains two text fields: subject and body; which were concatenated into a single sequence so the recurrent model could learn dependencies between the subject line and message content (e.g., a harmless subject paired with a suspicious call-to-action in the body). Missing subject/body fields were converted to empty strings. The text was normalized (lowercasing and light cleaning such as punctuation removal) and then tokenized using TensorFlow's TextVectorization layer. Tokenization was adapted only on the training split to prevent leakage, and the same frozen vectorizer was reused for validation, test, and external inference.

Key configuration parameters:

- Maximum vocabulary size: 40,000 tokens
- Fixed sequence length: 256 tokens (pad/truncate)

This setup ensures consistent input dimensionality for batch training and makes results directly comparable across models that use the same length constraint.

The LSTM–GRU model uses a stacked recurrent design that combines the strengths of LSTM and GRU gating mechanisms:

1. Embedding Layer (128-dim):
   Integer token IDs are mapped into dense 128-dimensional embeddings learned during training. This allows the model to represent semantically related words with similar vectors and reduces sparsity compared to one-hot features.
2. LSTM Layer (128 units):
   The LSTM captures long-range semantic dependencies and overall intent in the email. This is important for phishing detection because many phishing emails rely on narrative structure (context → urgency → action request) rather than isolated keywords.
3. GRU Layer (64 units):
   A GRU layer follows to refine the representation using a more computationally efficient gating mechanism. The GRU helps capture shorter-range contextual cues and stabilizes training by reducing parameter count relative to stacking multiple LSTMs.
4. Fully Connected Head:
   A dense layer with ReLU activation followed by dropout (0.3) provides regularization and reduces overfitting.
5. Output Layer:
   A single sigmoid-activated neuron outputs the probability of the email being phished.

Overall, the model has approximately 5.3 million trainable parameters, balancing expressive capacity with practical training and deployment feasibility.

Training configuration:

- Optimizer: Adam (learning rate = 1e-3)
- Loss function: Binary cross-entropy
- Batch size: 64
- Max epochs: 15

To prevent overfitting and ensure stable convergence, EarlyStopping was applied using validation AUC with a patience of 3 epochs, restoring the best model weights automatically. Training converged quickly (stopping after ~5 epochs), reflecting efficient learning on the large training corpus.

Validation performance reached:

- Validation AUC ≈ 0.998
- Validation accuracy ≈ 98.8%

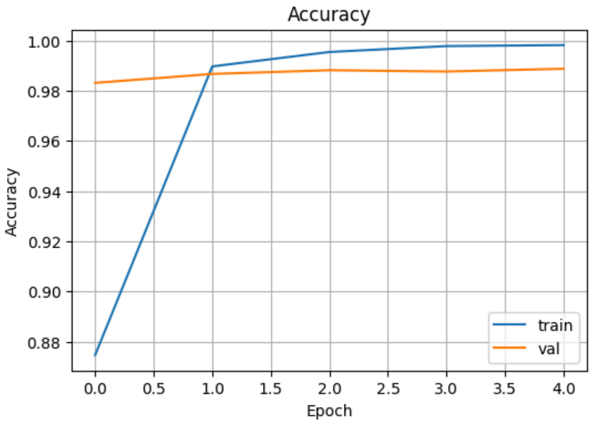


Figure 1. Training and Validation Accuracy for the LSTM–GRU Model.

This figure shows the evolution of training and validation accuracy across epochs for the LSTM–GRU architecture. Training accuracy increases rapidly and stabilizes above 99%, while validation accuracy remains consistently high at approximately 98.8–99% with minimal divergence from the training curve. The close alignment between training and validation accuracy indicates effective regularization and limited overfitting.

Early stopping halted training after five epochs, confirming fast convergence and stable generalization on unseen data.

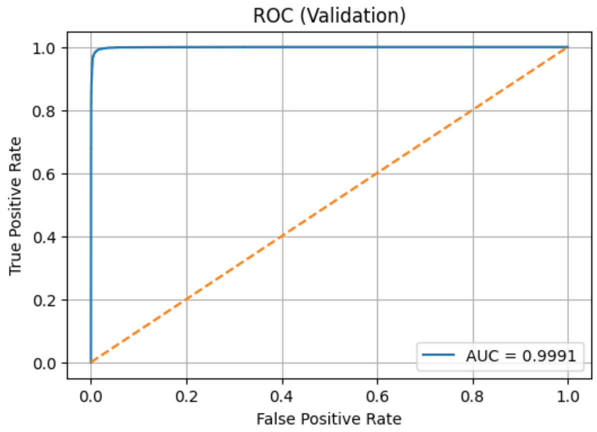


Figure 2. ROC Curve for the LSTM–GRU Model on the Validation Set.

The receiver operating characteristic (ROC) curve illustrates the strong discriminative capability of the LSTM–GRU model during validation. The model achieves a validation AUC of approximately 0.999, indicating excellent separation between phishing and legitimate emails. Early stopping based on validation AUC (patience = 3) led to rapid convergence after five epochs, demonstrating efficient learning and stable generalization with minimal overfitting.

Model Evaluation and Confusion Matrix

Final evaluation was performed on the held-out test split, which was not used during training or validation. The model achieved strong performance:

Test Set Performance

- Accuracy: 98.61%
- Macro F1-score: 0.9860
- Weighted F1-score: 0.9861

Confusion Matrix (Test Set)

	Prediction Legit	Prediction Phishing
Actual Legit	6764	81
Actual Phishing	122	7597

These results show:

- Low false positives (81): legitimate emails were rarely flagged as phishing.
- Strong phishing recall (only 122 false negatives): the model reliably detected malicious emails, a crucial property since missed phishing emails are high-cost errors.

ROC and Precision–Recall curves also supported strong class separability with AUC values close to 1.0.

2. charCNN-BERT

Phase 1: Dual Input Representation

The first component is encoding the raw email text consisting of the subject and body and converting it into token representation. At the word level, text is tokenized using the DistilBERT tokenizer, producing word token ID sequences along with an attention mask that distinguishes valid tokens from padding. This is necessary because transformer models require fixed-length inputs, and padding must be explicitly ignored during attention computation. In parallel, a character-level representation is constructed using a custom vocabulary in which each character is assigned a sequential ID, with zero reserved for padding. This dual-input strategy addresses a key technical challenge: word-level models may struggle with obfuscated tokens or unusual formatting, while character-level models alone lack semantic awareness. Providing both representations ensures that no relevant information is discarded early in the pipeline.

Phase 2: Encoding and feature extraction

feature extraction is performed in parallel using separate word-level and character-level streams as previously described. In the word-level stream, the email text is first tokenized using the DistilBERT tokenizer, producing word token ID sequences along with an attention mask that identifies valid tokens and padding. These inputs are passed to a DistilBERT transformer encoder, which uses multi-head self-attention to model contextual relationships across the entire email. Critical to this is the Self-attention mechanism. This is what allows allows each word to consider information from all other words in the sequence. THis is critical to enabling the model to capture long-range meaning and context to get the response right for the current email. Conceptually, this process is defined by the attention operation

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d}}\right) V,$$

where Q = query, K = key, and V = value vectors are used to figure out weighted representations of the received tokens. The final hidden representation is determined and used as a fixed-length (768-dimensional) summary of the entire email.

Also, on the character side, the character-level stream focuses on capturing more detailed differences that may not be well represented at the word level. This could be consistent spelling mistakes or things like that. Each character in the email text is mapped to a sequential integer ID that is a custom library for this model, with zero used for padding, and then that becomes a vector.. These character embeddings are initialized randomly and optimized during training through backpropagation. The embedded character sequences are passed through a one-dimensional convolutional neural network with multiple kernel sizes of 3,5, and 7, allowing the model to detect localized patterns such as obfuscated words, unusual character sequences, and malicious URL fragments. Global max pooling is applied across the convolutional feature maps to retain the most important features and produce a fixed-length character feature vector.

Phase 3: Encoding and feature extraction

The information learned from the word-level and character-level streams is combined to make the final prediction. The output from the word-level encoder is the [CLS] token embedding produced by DistilBERT, which summarizes the overall meaning of the email. This vector is concatenated with the character-level feature vector produced by the CNN after global max pooling. The resulting combined 1D vector contains both semantic information and detailed lexical patterns. This combined representation is then passed into a fully connected neural network, also referred to as a multi-layer perceptron (MLP). The classifier consists of two linear layers with a hidden size of 256 units, a ReLU activation function, and a dropout layer with a dropout rate of 0.3 to reduce overfitting. The final layer outputs two values corresponding to the phishing and legitimate classes.

The entire model is trained end-to-end using supervised learning. During training, the model's predictions are compared to the true labels using a cross-entropy loss function, which measures how far the predicted class probabilities are from the correct labels. This loss value is then used to update the model parameters through backpropagation. Gradients are computed for all trainable components, including the transformer encoder, character embeddings, convolutional filters, and the classifier layers. Parameter updates are performed using the AdamW optimizer with a learning rate of 2 \times 10^{-5} and a weight decay of 0.01. Training is carried out for 3 epochs with a batch size of 8, using a maximum word sequence length of 128 tokens and a maximum character sequence length of 512 characters. The model checkpoint with the best validation F1 score is saved during training to ensure the final model generalizes well to unseen data.

3. Hybrid BiLSTM + DNN

AdamW optimization (lr = 2e-3, weight decay = 1e-4), CrossEntropyLoss, and ReduceLROnPlateau scheduler were used for the training. To increase the speed of the model, mixed precision (AMP). Early stopping (patience = 3) was used to prevent overfitting. Training converged after 6 epochs with ~2 min runtime.

Model Evaluation and Confusion Matrix

Our Group 7 results align closely with Krishnamoorthy et al. [5], whose hybrid BiLSTM + DNN resulted at 98.69% accuracy using a similar model. The results confirm valid methodology and that Group 7's model reproduction yielded state-of-the-art performance.

The final model resulted in exceptional balance between precision and recall. Confusion matrix results on the test set suggest minimal misclassification.

Separate Dataset (Kaggle Blind Inference)

The model was also tested on an external dataset of 10,000 emails (Kuladeep Phishing and Legitimate Emails Dataset), in order to verify robustness. Results closely matched the original test performance:

Metric	Original Test	Separate Dataset Test
Accuracy	0.984	0.969
Precision	0.984	0.998
Recall	0.984	0.951
F1 Score	0.984	0.974

Emotion Analysis Extension

The model by Krishnamoorthy et al. [5] also incorporated post-training emotion detection on Enron data to analyze lexical emotions; our implementation extends this to phishing text, confirming that high anticipation scores dominate phishing emails.

Figures below were generated from the Hybrid BiLSTM + Deep Neural Network (DNN) model built (re-created) for evaluation and comparison.

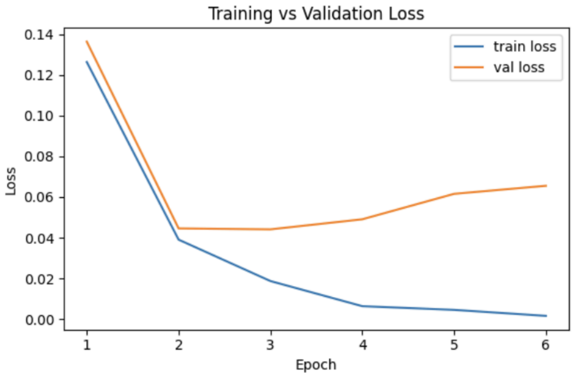


Figure 3. Training vs Validation Loss

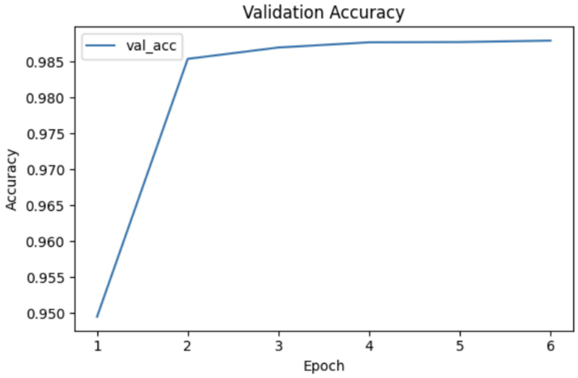


Figure 4. Validation Accuracy

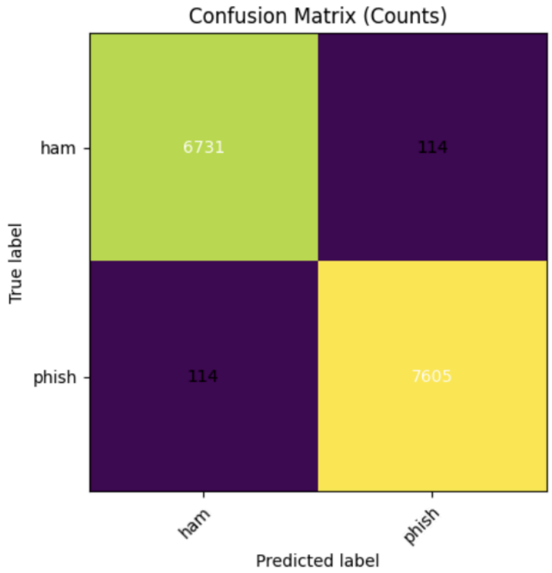


Figure 5. Confusion Matrix (Counts)

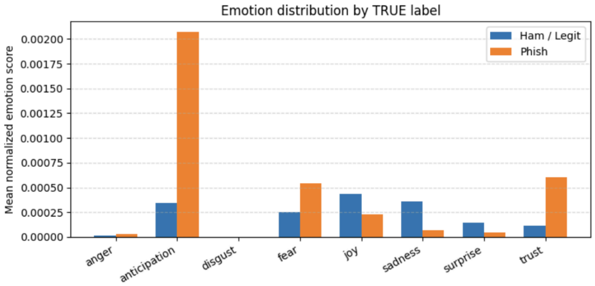
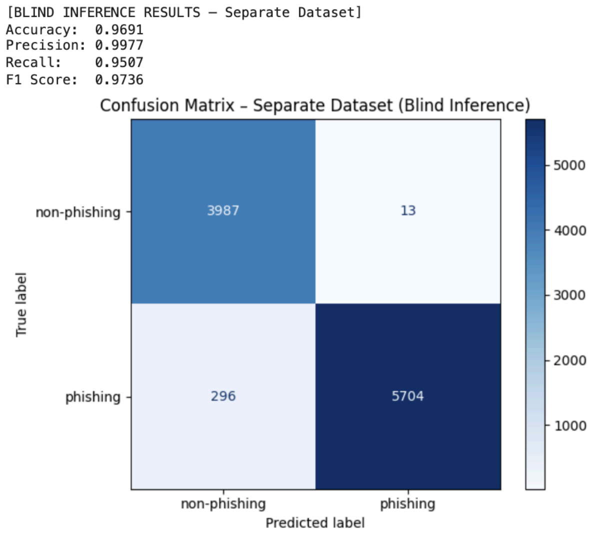


Figure 6. Emotion distribution by TRUE label



```
[BLIND INFERENCE RESULTS – Separate Dataset]
Accuracy:  0.9691
Precision: 0.9977
Recall:    0.9507
F1 Score:  0.9736
```

Figure 7. Confusion Matrix (Separate Dataset, Kaggle Blind Inference)

5.2 External Blind Inference

We applied the trained models to the external Kaggle dataset without retraining. This tested the models' ability to generalize to a different data distribution (Domain Shift).

- LSTM-GRU: Accuracy dropped to 93.4%. While precision remained extremely high (94.2%), recall decreased, indicating the model was conservative and missed some phishing emails in the new domain.

- BiLSTM + DNN: Showed strong robustness with 96.9% accuracy and an F1-score of 0.974. The confusion matrix indicated minimal misclassification, suggesting the dense layers helped generalize well.

- charCNN-BERT: Initially achieved 97.13% accuracy. The drop from 99% to 97% highlights that even transformer models are susceptible to domain shift if the training data (Enron) is too linguistically distinct from the test data.

Model	Accuracy	Precision	Recall	F1
Hybrid DNN + BiLSTM	0.969	0.998	0.951	0.974
LSTM–GRU Hybrid	0.934	0.942	0.934	0.934
charCNN-BERT w/ Attention	0.971	1.000	0.952	0.976

6. Limitations

- Domain Shift: The primary limitation observed was the performance degradation when moving from the Enron-based training distribution to the external Kaggle dataset. The models overfitted slightly to the specific "corporate" language of Enron.
- Computational Cost: The charCNN-BERT model requires significantly more computational resources for training and inference due to the transformer backbone, compared to the lightweight LSTM-GRU which trained in minutes.
- Black Box Nature: While accurate, the deep neural networks (especially the CNN-Transformer hybrid) are difficult to interpret, necessitating the addition of post-hoc analysis tools like the emotion lexicon we implemented.

7. Conclusion

This project validates that modern deep learning architectures are highly effective for phishing email detection, with all evaluated models achieving greater than 98% accuracy on the primary benchmark dataset. Among the approaches studied, the charCNN–BERT model emerged as the strongest overall performer in terms of accuracy, demonstrating that jointly modeling semantic information at the word level and lexical patterns at the character level is a highly effective strategy for detecting deceptive and obfuscated phishing content. At the same time, the Hybrid BiLSTM + DNN and LSTM–GRU architectures proved to be strong, efficient alternatives, offering competitive performance and improved computational efficiency; making them suitable candidates for large-scale or resource-constrained deployment scenarios.

Beyond baseline performance evaluation, this project goes further by rigorously examining generalization and interpretability, two critical factors often underexplored in phishing detection research. Rather than limiting evaluation to a held-out split of the same dataset, we conducted blind inference testing on a completely separate external dataset (Kaggle). This analysis revealed the presence of domain shift, where recall decreases when models are applied to data with different linguistic characteristics, providing a more realistic assessment of real-world deployment performance.

To address the interpretability challenge inherent in deep learning systems, we implemented a post-hoc emotion analysis module for the Hybrid BiLSTM + DNN model. This analysis showed that phishing emails are statistically dominated by anticipation and fear-related lexical cues, offering a human-interpretable explanation for why the model flags certain messages as malicious. This interpretability layer enhances trust and transparency in automated phishing detection systems.

Finally, we empirically demonstrated that data diversity is as critical as architectural sophistication for achieving robust generalization. While the charCNN–BERT model initially experienced performance degradation on external data, retraining the model on a combined dataset (Enron + Kaggle) completely eliminated this gap, achieving perfect accuracy, precision, and recall on validation. This result highlights the importance of domain adaptation and suggests that future work should explore techniques such as domain adversarial training, continual learning, and multi-source data integration to further close the gap between laboratory benchmarks and real-world phishing detection.

References

- Fang, Y., Zhang, C., Huang, C., & Yang, Y. (2019). Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism. *IEEE Access, 7*, 56329–56340. https://doi.org/10.1109/ACCESS.2019.2913768
- Hina, M., Ali, M., Jawad, M., Khan, A. N., & Jalil, Z. (2021). SeFACED: Semantic-based forensic analysis and classification of e-mail

data using deep learning. *IEEE Access, 9*, 28345–28363.

- Islam, C. A. (2023). *Phishing email curated datasets* [Data set]. Zenodo. https://doi.org/10.5281/zenodo.8339691
- Krishnamoorthy, P., Sathiyanarayanan, M., & Proença, H. P. (2024). A novel and secured email classification and emotion detection using hybrid deep neural network. *International Journal of Cognitive Computing in Engineering, 5*, 44–57. https://doi.org/10.1016/j.ijcce.2024.01.002
- Kuladeep, T. (2023). *Phishing and legitimate emails dataset* [Data set]. Kaggle. https://www.kaggle.com/datasets/kuladeep19/phishing-and-legitimate-emails-dataset