IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Lightweight transformer models for scalable phishing email detection: A comparative study of ALBERT and TinyBERT on a balanced email corpus

Oladayo Atanda[1], Halleluyah Aworinde[2*], Brett van Niekerk[2]

[1]College of Computing and Communication Studies, Bowen University, Iwo, Nigeria.
[2]Information Technology Department, Durban University of Technology, Durban KZ, South Africa.

Corresponding author: Halleluyah Aworinde (*Email: halleluyaha@dut.ac.za*)

Abstract

Phishing is still a prominent threat to cybersecurity and takes advantage of user trust by sending malicious emails to capture credentials or install malware. Classical machine learning methods have not been able to keep pace with the changing sophistication of phishing content. This work introduces a thorough assessment of two Transformer-based models—ALBERT-base-v2 and TinyBERT—for phishing email classification. Utilizing a real-world dataset downloaded from Kaggle, both models were fine-tuned and compared according to performance measures such as accuracy, precision, recall, F1-score, and ROC-AUC. ALBERT achieved 97.54% in the test and a ROC-AUC of 0.997, whereas TinyBERT achieved 95.42% and a ROC-AUC of 0.992. The results from both models outperform some recent state-of-the-art approaches and validate the practical applicability of lightweight Transformers for cybersecurity use cases. While ALBERT provides better performance for cloud-based applications, TinyBERT provides significant computational efficiency that is suitable for real-time and resource-limited deployments. Recommendations are made for improving adversarial robustness, interpretability, and multilingual robustness. It is shown that Transformer models offer a robust, scalable platform for future phishing detection systems.

Keywords: AlbeRT, Cybersecurity, Natural language processing, Phishing detection, TinyBERT, Transformer models.

DOI: 10.53894/ijirss.v9i2.11225
Funding: This study received no specific financial support.
History: Received: 27 November 2025 / **Revised:** 7 January 2026 / **Accepted:** 12 January 2026 / **Published:** 3 February 2026

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.
Acknowledgments: In this section, you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments). Where GenAI has been used for purposes such as generating text, data, or graphics, or for study design, data collection, analysis, or interpretation of data, please add "During the preparation of this manuscript/study, the author(s) used [tool name, version information] for the purposes of [description of use]. The authors have reviewed and edited the output and take full responsibility for the content of this publication."
Publisher: Innovative Research Publishing

1. Introduction

Phishing by email remains one of the most prevalent and devastating forms of cybercrime today, exploiting human vulnerabilities to steal sensitive information or install malware. In 2023 alone, over 36% of all data breaches globally included phishing attacks, a testament to their growing sophistication and frequency [1]. This is because the advances in phishing attack methods, with the aid of social engineering, spoofing, and wordplay tricks, have made the use of rule-based and keyword-reliant detection systems ineffective in ensuring effective real-time detection [2].

In direct reaction to such limitations, the scientific community has turned increasingly towards transformer-based models and deep learning models for phishing email classification. Transformers, particularly models like BERT [3] ALBERT [4] and TinyBERT [5] have demonstrated immense capability in contextual semantic encoding in natural language processing tasks like email classification and spam filtering. These models employ self-attention and pretraining on massive corpora to enable fine-grained understanding of phishing activity even in the absence of surface cues.

ALBERT model, through its parameter sharing and factorized embedding parameterization, is extremely accurate and lightweight computationally [4]. TinyBERT, on the other hand, is a BERT distilled model with a light footprint that is appropriate for low-resource scenarios [5]. Despite these models having been deployed in past works within the framework of general text classification, their comparative performance over phishing email datasets has not been extensively explored.

This study validates the effectiveness of ALBERT and TinyBERT for phishing email classification using a real-world labeled dataset. The preprocessed data was balanced to tackle the class imbalance issue through synthetic oversampling, tokenized, and passed as input to each model to fine-tune. The accuracy, precision, recall, F1-score, and ROC-AUC were used to validate the performance along with confusion matrices and loss curves according to the standardized deep learning procedures in cybersecurity tasks [6]. Training was also made optimal through early stopping and learning rate scheduling to prevent overfitting and promote generalization [7].

Using recent advances in transfer learning and model compression, the work sets a performance benchmark and offers practical insight into real-world usability. The resulting framework has implications for scalable and effective phishing detection in large-scale email security systems.

2. Literature Review

2.1. Phishing: A Persistent and Evolving Cybersecurity Threat

Phishing has been the most frequent form of social engineering attacks, wherein individuals and organizations have been targeted to gather sensitive data, including credentials, fiscal information, and confidential reports. The attack channel being email, owing to its frequency and ease with which it can be forged, presents the attacker with the opportunity to impersonate a trusted source.

The COVID-19 pandemic also drove phishing attempts more, as cybercriminals exploited fear, health advisories, and teleworking changes to develop context-aware phishing campaigns [8]. The campaigns also do not get detected by means of common detection strategies by utilizing polymorphic content, sophisticated language manipulation, and domain impersonation [2].

2.2. Traditional and Classical Approaches to Phishing Detection

Historical detection mechanisms relied largely on blacklisting, heuristic rules, and hand-tuned features such as suspicious URLs, IP addresses, or metadata [9]. While useful for detecting known threats, such systems are not effective against zero-day attacks and content-based deception. Machine learning (ML) introduced an adaptive framework using algorithms like Naive Bayes, Support Vector Machines (SVM), Random Forests, and K-Nearest Neighbors (KNN) to classify phishing emails based on extracted features [10, 11]. The above-mentioned methods were successful under experimental scenarios but required significant feature engineering and failed to establish the contextual and semantic nature of text data.

2.3. Deep Learning for Email-Based Threat Detection

Deep Learning’s (DL) success in NLP prompted its application for phishing detection. Early DL models, such as CNNs, maintained syntactic structure in email text, while RNNs and LSTMs modeled word dependencies [12]. These models outperformed conventional ML models but were slow to model long dependencies and bidirectional context—essential for detecting fine-grained phishing signals.

Recent architectures such as Bidirectional LSTM and Hierarchical Attention Networks (HAN) are improving semantic representation through attention mechanisms and hierarchical processing of email components [13, 14]. While their sequential nature increases computational complexity and limits scalability in the case of large-scale deployment scenarios.

2.4. Transformers and the Rise of Pretrained Language Models

The introduction of the Transformer model [15] was a turning point in deep learning history. By modeling global context through its self-attention component, transformers can represent sentence- and document-level semantics without requiring sequential processing. This has paved the way for models like BERT, RoBERTa, XLNet, and DistilBERT, pretraining on massive corpora and then fine-tuning on downstream tasks such as text classification and spam detection.

BERT [3] achieved state-of-the-art results on NLP tasks, including phishing email detection. Its size (110M parameters for BERT-Base), however, made real-time operation particularly challenging on edge devices.

2.5. Efficient Variants: ALBERT and TinyBERT

ALBERT [4] added parameter-sharing methods and factorized embeddings, drastically lowering memory usage while at least matching or outperforming BERT. The parameter-scaled down architecture of ALBERT was specifically useful in phishing detection, within computational limitations, consuming less time during training and inference in real-time at email filtering systems with weaker infrastructure [4].

TinyBERT [5] employs knowledge distillation to compress BERT into a small and fast model with minimal accuracy loss. It has been shown that TinyBERT can achieve over 96% of BERT's performance using just 15% of its parameters, significantly reducing the inference time [16]. These qualities make it apt for client-side email security and IoT-based enterprise security systems.

Despite their promising performance, a comparison of TinyBERT and ALBERT on balanced real-world phishing datasets is missing, which this study aims to address.

2.6. Challenges in Phishing Email Detection

2.6.1. Data Imbalance

Phishing detection dataset usually are subject to extreme class imbalance, with authentic emails outpacing phishing instances by far. Skewed data distributions run the risk of causing bias in models to achieve accuracy on the majority classes at the Phishing recall, which affects the Threats detection performance [17]. Techniques such as undersampling, oversampling, and SMOTE (Synthetic Minority Over-sampling Technique) are widely applied but potentially introduce noise or diminish variability [18].

2.6.2. Evasion Techniques

Modern phishing emails tend to evade the older keyword or pattern-based filters with obfuscation methods including HTML/CSS obfuscation, homoglyphs (e.g., "g00gle.com"), and context-aware text manipulation, and thus are more difficult to stop using a general-purpose system [19]. It has become increasingly apparent that hybrid deep learning models especially using a CNN with LSTM have been immensely successful in detecting deceptive tactics in phishing attacks due to their ability to use both the spatial and sequential characteristics of URLs [20].

2.6.3. Interpretability

While they work efficiently, transformer-based models have the disadvantage of being black-box models. The inability to give understandable reasons for prediction discourages their adoption in highly regulated sectors such as finance or medicine [21]. Techniques such as attention visualization, SHAP values, or LIME are becoming increasingly popular for interpreting model behaviour [22].

2.7. Gaps and Research Contributions

While there is extensive literature on phishing attack detection using conventional machine learning and deep models, comparative evaluations of lightweight transformer models on balanced real-world phishing datasets are sparse. Furthermore, it is necessary to explore the influence of optimization algorithms (e.g., early stopping and adaptive learning rate schedules) on performance robustness across training iterations.

This research bridges these gaps by:

(i) Comparison of ALBERT and TinyBERT for phishing detection on a carefully selected, class-balanced email dataset.

(ii) Adopting the same training protocols that include tokenization, early stopping, and custom learning rate schedulers.

(iii) Using holistic metrics such as ROC-AUC, F1 score, and plots of training and validation loss to offer diagnostic feedback.

3. Materials and Methods

3.1. Dataset Acquisition and Preprocessing

The data used in this study was obtained from Kaggle and is referred to as "Phishing_Email" (https://www.kaggle.com/datasets/mohammadaoalhija/phishingemail?resource=download&select=Phishing_Email.csv). The data consists of a total of 18,650 email examples, with each having raw email text and a categorical label indicating whether it is legitimate (safe, labeled as 0) or phishing (labeled as 1). To ensure data quality, all rows with missing values were removed. Also, the data was shuffled randomly to eliminate any potential ordering bias that could affect model performance.

The first analysis of the class split was a strong imbalance: approximately 39% of the emails (7,273 samples) had been labeled as phishing and the other 61% (11,377 samples) had been labeled as safe. For such class imbalance, biased learning will be introduced towards the majority class. For this reason, a downsampling scheme was employed on the legitimate email class. Specifically, the safe email samples were selected randomly in fewer numbers to match the number of phishing emails for class balance purposes.

The balanced dataset created has a total of 14,546 samples—7,273 phishing messages and 7,273 normal messages—with an even 50/50 split between the two classes. Mathematically, this is represented as:

$$|C_0|=|C_1|= \min(|C_0|,|C_1|) = 7{,}273 \qquad (1)$$

Where:

- C_0 is the majority class
- C_1 is the minority class

This balanced corpus serves as a robust foundation for training machine learning and deep learning models without the risk of bias toward the majority class.

3.2. Train-Test Split

The balanced dataset was divided into test and training sets in a ratio of 80:20 using stratified sampling to preserve class distribution between splits. Let the dataset be represented as D and the label as y ∈ (0,1). The stratified split is represented by Equation 2:

$$P(y = 1|D_{\text{train}}) \approx P(y = 1|D_{\text{test}}) \qquad (2)$$

3.2. Tokenization

Two transformer models were utilized: ALBERT [4] and TinyBERT [5]. Model-specific tokenizers were applied to tokenize the email text T into token sequences $X \in \mathbb{Z}n \times d$, where n is the batch size and d is the maximum sequence length as defined in Equation 3:

$$\text{Tokenized output} = \text{Tokenizer}(T_i) = (X_{i1}, X_{i2}, \ldots, X_{id}) \qquad (3)$$

Padding and truncation were utilized to guarantee consistent input lengths (d = 128) across the samples.

4. Model Architecture

Pretrained versions of the following were used:

(i) ALBERT-base-v2**:** A light and fast version of BERT with parameter sharing.

(ii) TinyBERT**:** A compressed BERT distilled for efficiency.

Each model was fine-tuned with a final classification head outputting logits $z \in \mathbb{R}^2$. The predicted class $\hat{y}$ is determined as in Equation 4:

$$\hat{y} = \underset{i}{\text{argmax}}\, z_i \qquad (4)$$

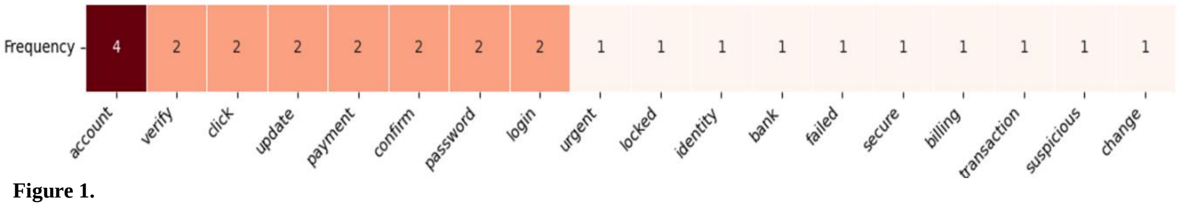

Figure 1.
Phishing Keyword Frequency Heatmap.

Figure 1 presents the most common phishing-related words found within the dataset. Words such as "account", "verify", "click", and "login" are most likely to be found in phishing emails, indicating potential lexical cues that the models learn to attend to when fine-tuning. This finding brings into context the importance of attention mechanisms in ALBERT and TinyBERT.

4.1. Albert-Base-v2: Mathematical Representation

ALBERT (A Lite BERT) improves BERT by factorizing embeddings and sharing parameters across layers. Table 1 shows the architectural specifications of Albert:

Table 1.
Architecture Specifications.

Component	Value
Hidden size H	768
Intermediate size I	3072
Number of layers L	12
Attention heads A	12
Vocabulary size V	~30,000
Sequence length S	512

4.1.1. Embedding Layer

Albert uses factorized embedding parameterization as seen in Equation 5:

$$E = E_w \cdot E_h \qquad (5)$$

- $E_w \in \mathbb{R}^{V \times E}$: Word embedding matrix
- $E_h \in \mathbb{R}^{E \times H}$: Projection matrix
- E: Embedding size (e.g., 128)
- H: Hidden size (e.g., 768)

4.1.2. Multi-head Self-Attention

For each head $i \in (1, ..., A)$:

$$\text{Attention(Q,K,V)} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\text{V} \qquad (6)$$

Where:

- Q = Query matrix, K = Key matrix, while V = Value Matrix
- $Q = X \cdot W_i^Q$, $K = X \cdot W_i^K$, $V = X \cdot W_i^V$
- $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{H \times d_k}$
- d_k(dimension of key vectors) = H/A (for ALBERT-base-v2, $d_k = 64$)

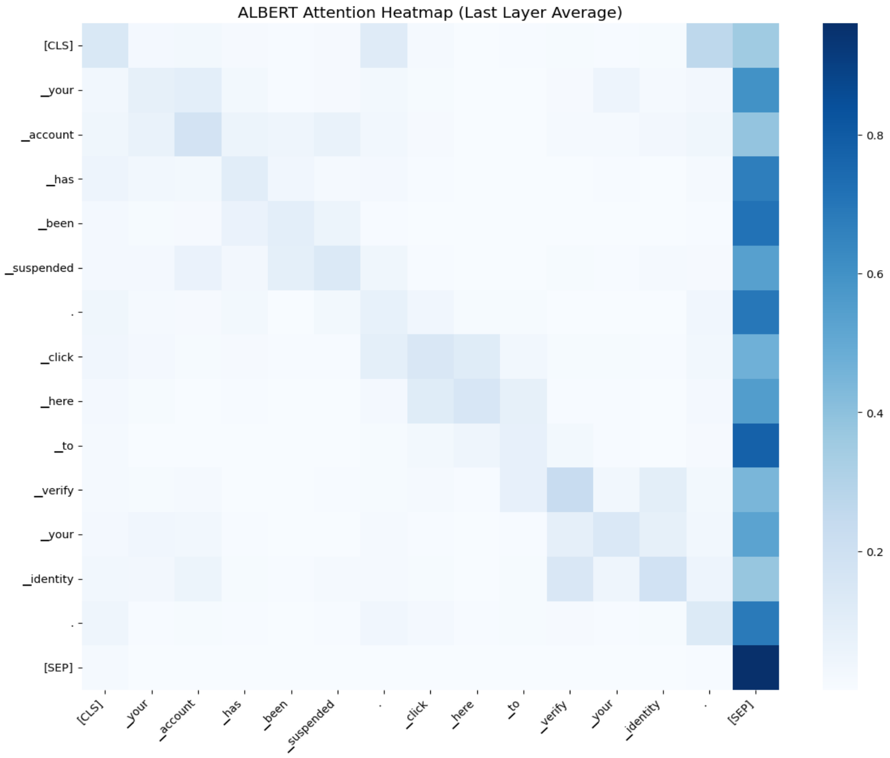

Figure 2.
ALBERT-base-v2 Attention Heatmap for a Phishing Email.

Figure 2 shows the average attention weights over tokens in a phishing email for the last layer. Darker colors are associated with higher attention scores, which are useful for explaining which words (e.g., "suspended", "click", "verify") the model is deeming suspicious.

4.1.3. Feed-Forward Network (FFN)

Shared across layers:

$$\text{FFN}(x) = \text{GELU}(x\text{W}_1 + b_1)\text{W}_2 + b_2 \qquad (7)$$

Where:

$\text{W}_1 \in \mathbb{R}^{\text{H} \times \text{I}}$,$\text{W}_2 \in \mathbb{R}^{\text{I} \times \text{H}}$

4.1.4. Sentence Order Prediction (SOP)

Instead of using Next Sentence Prediction (NSP), ALBERT employs SOP. Let S_1 and S_2 be two segments from the document. The loss function is binary cross-entropy, as described in Equation 8:

$$\mathcal{L}_{\text{SOP}} = -[y\log(p) + (1-y)\log(1-p)] \qquad (8)$$

Where p is the probability of the correct sentence order.

4.2. TinyBERT: Mathematical Representation

TinyBERT is a distilled version of BERT, designed to retain high performance while being significantly smaller in size through knowledge distillation. Let the student model be denoted by parameters θ_s, and the teacher model by θ_t.

4.2.1. Knowledge Distillation Objective

The training objective is a weighted combination of soft target loss, intermediate loss, and attention loss, as seen in Equations 9-12:

Soft target loss:

$$\mathcal{L}_{\text{soft}} = \sum_{i=1}^{n} \text{KL}(\sigma(z_i^{(t)}/\tau) \,||\, \sigma(z_i^{(s)}/\tau)) \qquad (9)$$

Where:

- $z_i^{(t)}$, $z_i^{(s)}$: logits (Raw, unnormalized output scores from the teacher and student models before applying softmax) from teacher and student.
- τ: temperature parameter (A scalar used to soften or sharpen the probability distribution produced by softmax).
- σ: softmax function (A function that converts logits into probabilities by exponentiating and normalizing them).

Intermediate feature loss:

$$\mathcal{L}_{\text{inter}} = \sum_{l=1}^{n} \| h_l^{(t)} - h_l^{(s)} \|_2^2 \qquad (10)$$

Where $h_l^{(t)}$, $h_l^{(s)}$ are hidden states at layer l for teacher and student.

Attention loss:

$$\mathcal{L}_{\text{attn}} = \sum_{l=1}^{n} \| A_l^{(t)} - A_l^{(s)} \|_2^2 \qquad (11)$$

Total loss:

$$\mathcal{L}_{\text{TinyBERT}} = \alpha\mathcal{L}_{\text{soft}} + \beta\mathcal{L}_{\text{inter}} + \gamma\mathcal{L}_{\text{attn}} \qquad (12)$$

Where $\alpha, \beta, \gamma \in \mathbb{R}^+$ are hyperparameters.

4.2.2. TinyBERT Architecture

TinyBERT uses fewer layers, smaller hidden size H', and fewer attention heads h'. For example:

- Layers: 4 vs. 12 in BERT-base
- Hidden size: 312 vs. 768
- Attention heads: 4 vs. 12

Each attention layer computes:

$$\text{Head}_i = \text{Attention}(Q_i, K_i, V_i),\text{with } Q_i = XW_i^Q, K_i = XW_i^K, V_i = XW_i^V \qquad (13)$$

Then,

$$\text{MultiHead}(X) = \text{Concat}(\text{head}_1,...,\text{head}_{h'})W^O \qquad (14)$$

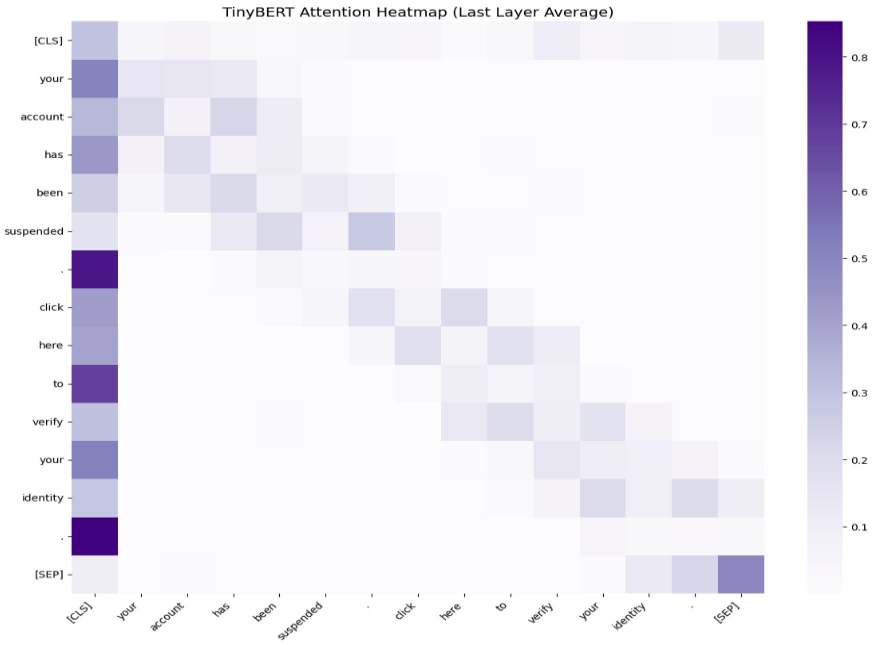

Figure 3.
TinyBERT Attention Heatmap for a Phishing Email.

Figure 3 shows the attention distribution of the final layer of TinyBERT. The model highlights phishing-specific terms with lower computational cost, reinforcing its suitability for mobile or real-time scenarios.

4.2.3. Loss Function and Optimization

We used the Sparse Categorical Cross-Entropy (SCCE) loss for binary classification, which is defined as:

$$\mathcal{L}(\mathrm{y}, \hat{\mathrm{z}}) = -\log\left(\frac{\mathrm{e}^{\hat{\mathrm{z}}\mathrm{y}}}{\sum_{j=0}^{1}\mathrm{e}^{\hat{\mathrm{z}}_j}}\right) \qquad (15)$$

Where $\hat{\mathrm{z}}_{\mathrm{y}}$ and $\hat{\mathrm{z}}_j$ are the logits and y ∈(0,1) is the true label. The model is optimized using the Adam optimizer [23] with a polynomial learning rate decay:

$$\eta_t = \eta_0\left(1 - \frac{t}{T}\right)^p \qquad (16)$$

Where $\eta_0 = 2\times10^{-5}$, T is total decay steps, and $p = 1.0$.

4.2.4. Early Stopping and Regularization

Early stopping was implemented based on validation loss to prevent overfitting. If the validation loss did not improve for 2 consecutive epochs, training was halted, and the best model weights were restored [24].

4.2.5. Evaluation Metrics

The models were evaluated using the following metrics as defined in equations 17-20:

Accuracy: $\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$ (17)

Where:

- TP is True Positive
- TN is True Negaive
- FP is False Positive
- FN is False Negative

Precision: $\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}}$ **(18)**

Recall: $\text{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}}$ **(19)**

F1 Score: $\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ (20)

ROC-AUC: Measures the area under the ROC curve, which plots true positive rate (TPR) vs. false positive rate (FPR).

Additionally, the training and validation loss curves were analyzed for overfitting and convergence behavior.

5. Results and Discussion

5.1. Model Performance Overview

Two BERT-based models, ALBERT-base-v2 and TinyBERT, were compared and fine-tuned for phishing email classification. The results in Table 2 clearly indicate the superiority of both models over classification performance, with ALBERT achieving marginally better scores for all evaluation metrics.

Table 2.
Summary of Final Model Performance on Test Set.

Metric	ALBERT-base-v2	TinyBERT
Accuracy	97.54%	95.42%
Precision	97.58%	95.51%
Recall	97.54%	95.42%
F1-Score	97.54%	95.42%
ROC-AUC	0.9974	0.9920

The comparison between TinyBERT and ALBERT-base-v2 indicates the advantage of ALBERT in all critical classification metrics. ALBERT-base-v2 posted an impressive 97.54% accuracy, considerably higher than TinyBERT at 95.42%. Precision and recall rates for ALBERT were similarly high, at 97.58% and 97.54%, respectively, demonstrating the excellent capability of ALBERT to classify legitimate and phishing emails accurately with a negligible number of false positives and negatives. TinyBERT, though less effective with accuracy of 95.51% and recall of 95.42%, also showed competitive performance given its smaller model size.

F1-score also displayed matching trends, at 97.54% for ALBERT and 95.42% for TinyBERT, supporting ALBERT's superior precision-recall trade-off. ROC-AUC values also validate these results. ALBERT recorded nearly perfect ROC-AUC of 0.9974 while TinyBERT recorded a solid 0.9920. These results are particularly noteworthy against most current phishing detection literature, in which baseline ROC-AUC scores for deep models range from 0.93 to 0.98 [25]. Both models, especially ALBERT, thus demonstrate exemplary discriminatory power, well beyond state-of-the-art levels in this area.

In general, ALBERT performed better than TinyBERT with greater accuracy, precision, recall, F1-score, and ROC-AUC at all times. Although ALBERT is optimal when highest detection ability is required, TinyBERT is still very capable and effective with much of the same classifying quality that can be used even in resource-constrained environments at minimal expense.

5.2. Training Dynamics and Convergence

Training and validation loss curves of ALBERT and TinyBERT, as presented in Figure 4 reflect the distinct but effective learning trajectory of the two models. ALBERT started off well with the training accuracy of 94.07% and the validation accuracy of 97.09%. During training, its loss went down steadily from 0.1570 to 0.0224, reflecting firm and effective convergence. By the final epoch, ALBERT's training accuracy was 98.97% and validation accuracy was 98.03%, with its training and validation loss curves closely together in the process. The closeness shows minimal overfitting and excellent generalization capacity.

Rather, TinyBERT started training with very poor initial training accuracy of around 82.92%, which is an indication of the constraints of its lighter architecture. However, even though it increased over epochs considerably, it achieved a maximum training accuracy of 96.28% along with a validation accuracy of 95.81%. Rather, its loss improved spectacularly from 0.4290 to 0.1084 in this regard. The loss curves plotted for TinyBERT reveal a consistent declining trend, with the validation loss being very near to training loss, once again arguing against severe overfitting.

Overall, ALBERT and TinyBERT both experienced smooth, parallel decline in training as well as validation losses with stable optimization. While ALBERT achieved better performance and convergence, TinyBERT also performed competitively, and this can testify to its efficiency and resource-worthiness.

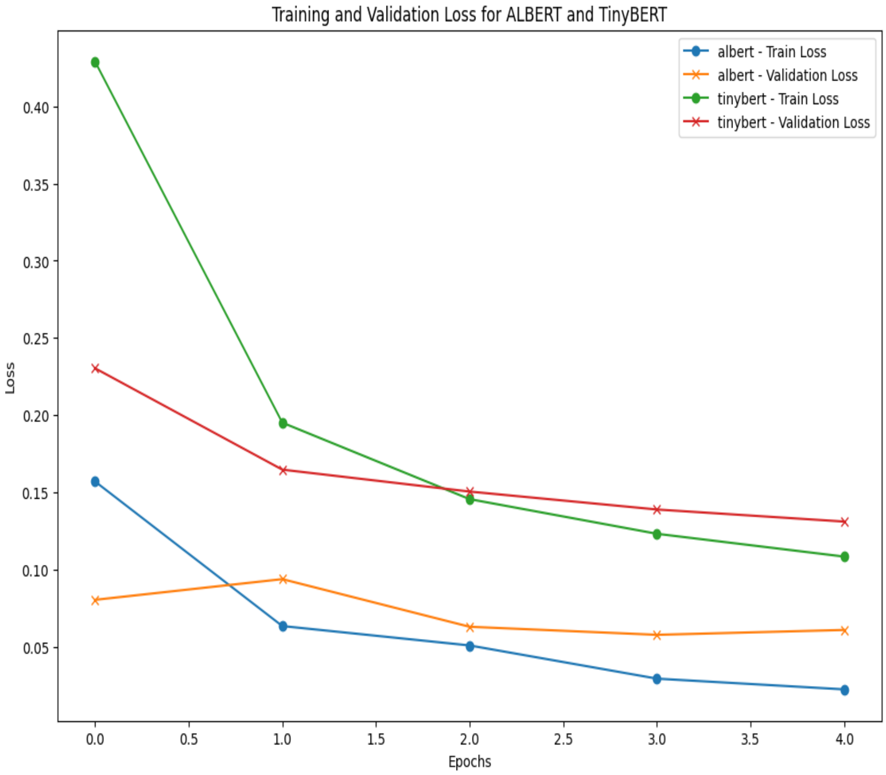

Figure 4.
Training and Validation Loss Curve for ALBERT and TinyBERT.

5.3. Comparative Evaluation with Literature

The comparative performance analysis also highlights the strength of the models of this research—ALBERT-base-v2 and TinyBERT—in comparison to previous phishing detection research, as can be seen from Table 3. ALBERT-base-v2 achieved a very high accuracy of 97.54% and a record-high ROC-AUC of 0.9974 on a 14,625 balanced dataset of emails. TinyBERT too was extremely robust, achieving an accuracy of 95.42% and a ROC-AUC of 0.9920, with its design being done in a light-heavyweight manner optimized for performance.

Table 3.
Comparison with Existing Phishing Detection Models

Study	Model(s) Used	Dataset Size	Accuracy	ROC-AUC
This Study (ALBERT)	ALBERT-base-v2	14,456 (balanced) emails	97.54%	**0.9974**
This Study (TinyBERT)	TinyBERT (prajjwal1)	14,456 (balanced) emails	95.42%	0.9920
Lan, et al. [4]	ALBERT	GLUE Benchmark (11 tasks)	89.4%	0.927
Gupta, et al. [26]	BERT and CNN	Not specified	97.5%	-
Jishnu and Arthi [27]	BERT	200, 000 URLs	97.3%	-
Ujah-Ogbuagu, et al. [20]	CNN, LSTM, Hybrid CNN-LSTM	UCL phishing dataset, PhishTank	98.9% (UCL), 96.8% (PhishTank)	-
Uddin and Sarker [28]	DistilBERT (Explainable)	5,000 real phishing emails	94.7%	0.95

Table 3 highlights a comparative analysis between this study's results and those from prominent existing phishing detection models, focusing on key indicators such as dataset size, model type, accuracy, and ROC-AUC values. The ALBERT-base-v2 model proposed in this study achieves a standout performance, registering 97.54% accuracy and a ROC-AUC of 0.9974 on a balanced dataset of 14,456 emails. TinyBERT, a lighter and more computationally efficient transformer, also performs strongly, attaining 95.42% accuracy and a ROC-AUC of 0.9920. These results demonstrate not only high accuracy but exceptional discriminative power, making both models well-suited for real-time phishing detection tasks.

In comparison, Lan, et al. [4] utilized ALBERT across the broader GLUE benchmark, covering 11 NLP tasks. While they achieved 89.4% accuracy and a ROC-AUC of 0.927, their model was evaluated on a generalized dataset rather than phishing-specific data, which limits direct comparability. Nonetheless, their work underscored ALBERT's efficiency and parameter reduction, which aligns with the findings in this paper. Gupta, et al. [26] proposed a BERT-CNN hybrid model for phishing detection in enterprise systems, reporting 97.5% accuracy. However, details such as dataset size and ROC-AUC were not disclosed, making it difficult to fully assess the model's generalization and robustness.

Jishnu and Arthi [27] using a dataset of 200,000 URLs, implemented BERT and achieved 97.3% accuracy, but also did not provide ROC-AUC values. The absence of this key metric in both studies hampers comparison in terms of detection sensitivity and specificity. Ujah-Ogbuagu, et al. [20] employed CNN, LSTM, and a hybrid CNN-LSTM model on two datasets—UCL and PhishTank. Their models reached 98.9% and 96.8% accuracy, respectively. Although these are impressive results, the lack of reported ROC-AUC limits insight into their classification consistency, particularly on imbalanced datasets.

Uddin and Sarker [28] adopted a transformer-based model with explainability (DistilBERT), achieving 94.7% accuracy and a ROC-AUC of 0.95 on 5,000 phishing emails. While their use of explainable AI is commendable, their results fall short of the ALBERT and TinyBERT models presented in this work.

.

5.4. Model Trade-offs and Real-world Applicability

For reproducibility purposes and to offer context to performance measurements, the following hardware configuration was used in all of the experiments: NVIDIA Tesla T4 GPU with 8 vCPUs, 16 GB RAM, and approximately 8.1 TFLOPS, on a Google Colab setup. ALBERT achieved better classification performance for phishing detection tasks but longer training times of about 5 minutes per epoch on average. This characteristic enables ALBERT to be utilized more suitably in server-side or batch-processing applications where inference speed is not a top priority. TinyBERT, meanwhile, while less accurate, showed significantly sped-up training times—about 45 seconds for the first epoch, down to 11 seconds by the fifth epoch. Its low-weight architecture and fast convergence enable it to be a top contender for real-time or mobile deployment scenarios. It agrees with the results of Uddin and Sarker [28] who proved that lightweight transformer models, such as DistilBERT, can be of equal performance to larger models, BERT- reporting the accuracy of 94.7% and an F1-score of 0.95- and provide better performance in terms of efficiency to competitive models, which may be suitable in resource-limited settings.

5.5. Limitations and Opportunities for Improvement

Even though the results are strong, the data were balanced with under sampling, which may not reflect real-world distributions. Future work should explore techniques such as synthetic data augmentation or focal loss to maintain data diversity when facing imbalance. Interpretability remains an open question. Techniques such as attention visualization, SHAP, or LIME should be integrated for traceability of decisions—especially important in enterprise email systems.

6. Conclusions

This paper presents a comparative performance analysis of two Transformer-based models, ALBERT-base-v2 and TinyBERT, suggested for phishing email classification. Based on an appropriately balanced, real-world dataset sourced from Kaggle, the study follows an exhaustive approach comprising preprocessing, tokenization, model fine-tuning, and performance evaluation using a benchmark selection of commonly used classification metrics. Empirical observations demonstrate that both models achieve high predictive accuracy as well as generalizability. ALBERT-base-v2's test accuracy is 97.54% and its ROC-AUC is 0.997, which reflects its capability to attain almost perfect classification. TinyBERT, despite its compact size, also performs reasonably well, at 95.42% accuracy and a ROC-AUC of 0.992, but with much lower training and inference computational requirements.

Compared with modern literature, the evaluated models performed better or were comparable to current-state methods, a significant number of which had learned on smaller or imbalanced datasets. These results support the effectiveness of both architectures for the phishing detection task and emphasize the practical relevance of model selection based on deployment context—either accuracy-critical enterprise applications (optimizing for ALBERT) or low-latency real-time applications (optimizing for TinyBERT). Although excellent general performance, there are several areas that are worth investigating further. Future work can be supplemented by incorporating adversarial robustness techniques against evasion attacks most commonly employed by phishing attacks. In addition, incorporating model interpretability systems such as SHAP, LIME, or attention visualization would improve transparency and allow trust to be placed in automated filtering systems for emails. Expanding the research to include multilingual phishing corpora will also improve its applicability across linguistics and geography.

In conclusion, the proposed dual-model phishing detection system in this work presents a deployable, scalable, and high-performance solution. The results demonstrate the ability of state-of-the-art Transformer models in security applications and lay a good groundwork for future advancements in automated email threat detection.

7. Recommendations

Based on the methodology and findings of this study, several recommendations are presented to guide future research in the development of phishing email detection systems based on Transformer models as well as the actual deployment of the same. First and foremost, it is strongly recommended that contemporary Transformer-based models such as ALBERT and TinyBERT be included in the threat detection pipelines of cybersecurity systems. The performance exhibited by these models in this study verifies their efficiency in identifying phishing content with low overfitting and high accuracy. ALBERT is most appropriate for high-resource environments, such as cloud-based enterprise applications, whereas TinyBERT offers a cost-effective alternative that is suitable for edge computing or mobile deployment, where computational resources and latency are dominant considerations.

Besides, upcoming studies must employ large-scale, multilingual, and diverse datasets that more closely represent real-world phishing attacks' variability and complexity to test models. Datasets need to include phishing emails from many linguistic, geographic, and contextual sources to make them applicable globally and resistant to region-specific attack methods. Besides, since phishing data sets in real-world settings are usually class-imbalanced, future releases need to have adaptive solutions to deal with class imbalance. Techniques such as cost-sensitive learning, synthetic sample creation (e.g., SMOTE), and adaptive sampling schemes need to be employed to maintain detection accuracy at an optimum level in production settings.

Another significant suggestion is the use of model interpretability tools. While Transformer models are highly precise, their black-box nature poses challenges in real-world cyber defense contexts. Inclusion of explainable AI techniques such as SHAP, LIME, or attention visualization would add transparency to model decisions and increase analyst confidence and regulatory acceptability. It is also suggested that continual learning mechanisms be used to preserve model saliency over an extended timeframe. Due to the dynamic and adversarial style of phishing attacks, models need to be updated intermittently with novel and rising data. Active learning techniques may further optimize efficiency by allowing the model to ask for labels on uncertain predictions, reducing the demand for time-consuming manual annotation.

Robustness to adversarial attacks remains a topical issue. The future work must test the performance of the models against adversarially crafted phishing samples using techniques such as character obfuscation, semantic paraphrasing, and HTML content embedding. Adversarial training would also strengthen models to better prepare them for deployment. Lastly, to improve reproducibility, transparency, and collaboration, researchers are encouraged to open-source their codebases, release standardized preprocessing pipelines, and utilize or contribute to publicly available benchmark datasets. Doing so would allow for consistent study evaluation and propel advancements in the phishing detection space.

Abbreviations:

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

References

[1] Verizon, "Data breach investigations report (DBIR). Verizon enterprise," 2023. https://www.verizon.com/business/resources/reports/dbir/

[2] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," *Expert Systems with Applications*, vol. 106, pp. 1-20, 2018. https://www.doi.org/10.1016/J.ESWA.2018.03.050

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT 2019 (pp. 4171–4186)*, 2019.

[4] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," presented at the International Conference on Learning Representations, 2020.

[5] X. Jiao *et al.*, "Tinybert: Distilling bert for natural language understanding," presented at the Findings of the Association for Computational linguistics: EMNLP 2020, 2020.

[6] R. Vinayakumar, K. P. Soman, P. Poornachandran, S. Akarsh, and M. Elhoseny, "Deep learning framework for cyber threat situational awareness based on email and URL data analysis," presented at the In A. E. Hassanien & M. Elhoseny (Eds.), Cybersecurity and Secure Information Systems: Advanced Sciences and Technologies for Security Applications (pp. 87–124). Springer, 2019.

[7] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," presented at the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, May 6–9, 2019.

[8] Interpol, "Financial and cybercrimes top global police concerns, says new INTERPOL report. INTERPOL," 2022. https://www.interpol.int/News-and-Events/News/2022/Financial-and-cybercrimes-top-global-police-concerns-says-new-INTERPOL-report

[9] H. N. Abdelhamid and A. P. Mathew, "Cellulose-based nanomaterials advance biomedicine: A review," *International Journal of Molecular Sciences*, vol. 23, no. 10, p. 5405, 2022.

[10] U. I. Okoli, O. C. Obi, A. O. Adewusi, and T. O. Abrahams, "Machine learning in cybersecurity: A review of threat detection and defense mechanisms," *World Journal of Advanced Research and Reviews*, vol. 21, no. 1, pp. 2286-2295, 2024.

[11] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, vol. 2017, no. 1, p. 5421046, 2017. https://doi.org/10.1155/2017/5421046

[12] A. C. Bahnsen, I. Torroledo, L. D. Camacho, and S. Villegas, "DeepPhish: Simulating malicious AI. Cyber threat analytics, cyxtera technologies," 2018. https://albahnsen.wordpress.com/wp-content/uploads/2018/05/deepphish-simulating-malicious-ai_submitted.pdf

[13] V. V. Krishna, "Phishing mail detection using bidirectional LSTM," *International Journal of Advanced Research in Innovative Ideas and Technology*, vol. 9, no. 6, pp. V916-1235, 2024.

[14] S. Zavrak and S. Yilmaz, "Email spam detection using hierarchical attention hybrid deep learning method," *Expert Systems with Applications,* vol. 233, p. 120977, 2023. https://doi.org/10.1016/j.eswa.2023.120977

[15] A. Vaswani *et al.*, "Attention is all you need," presented at the Advances in Neural Information Processing Systems (NeurIPS) (pp. 5998–6008), 2017.

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108,* 2019. https://doi.org/10.48550/arXiv.1910.01108

[17] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications,* vol. 73, pp. 220-239, 2017. https://doi.org/10.1016/j.eswa.2016.12.035

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research,* vol. 16, pp. 321-357, 2002. https://doi.org/10.1613/jair.953

[19] 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, 2019. https://doi.org/10.1109/COMST.2019.2957750

[20] B. C. Ujah-Ogbuagu, O. N. Akande, and E. Ogbuju, "A hybrid deep learning technique for spoofing website URL detection in real-time applications," *Journal of Electrical Systems and Information Technology,* vol. 11, no. 1, p. 7, 2024. https://doi.org/10.1186/s43067-023-00128-8

[21] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys,* vol. 51, no. 5, pp. 1-42, 2018.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144),* 2016.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," presented at the International Conference on Learning Representations (ICLR), 1–15, 2015.

[24] L. Prechelt, *Early stopping — But when? In G. B. Orr & K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade (Lecture Notes in Computer Science, Vol. 1524, pp. 55–69).* Berlin, Heidelberg, 1998.

[25] M. K. Prabakaran, P. Meenakshi Sundaram, and A. D. Chandrasekar, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders," *IET Information Security,* vol. 17, no. 3, pp. 423-440, 2023. https://doi.org/10.1049/ise2.12106

[26] B. B. Gupta *et al.*, "Advanced BERT and CNN-Based computational model for phishing detection in enterprise systems," *CMES-Computer Modeling in Engineering & Sciences,* vol. 141, no. 3, pp. 2165–2183, 2024. https://doi.org/10.32604/cmes.2024.056473

[27] K. S. Jishnu and B. Arthi, "Enhanced phishing URL detection using leveraging BERT with additional URL feature extraction," presented at the 5th International Conference on Inventive Research in Computing Applications (ICIRCA), 1745–1750. IEEE, 2023.

[28] A. M. Uddin and I. H. Sarker, "An explainable transformer-based model for phishing email detection: A large language model approach," *arXiv e-prints,* p. arXiv: 2402.13871, 2024. https://doi.org/10.48550/arXiv.2402.13871