
Journal of Informatics and Web Engineering

Vol. 5 No. 3 (October 2026)

eISSN: 2821-370X

Calibrated Probabilistic Predictions for Phishing Email Detection Using Sentence-BERT with Machine Learning Models

Oluwasegun Abiodun Abioye^{1*}, Abraham Ewiekpaefe², Philip Odion³

¹Directorate of Information and Communication Technology, Nigerian Defence Academy, Kaduna, Nigeria.

^{2,3}Department of Computer Science, Nigerian Defence Academy, Kaduna, Nigeria.

**corresponding author: (segunabioye@nda.edu.ng; ORCID: 0009-0001-4100-9736)*

Abstract - The growing sophistication of phishing attacks requires better detection than classic conventional filters. Previous work has focused on the accuracy, precision and recall of classifiers but there is a lack in testing well-calibrated probability predictions, which are crucial for reliable decision-making using cybersecurity systems. To fill this gap, we present in this work a phishing email detection framework that combines semantically rich Sentence-BERT all-mpnet-base-v2 model embeddings with a wide variety of machine learning classifiers. A total of nine model configurations were tested, including Sentence-BERT combined with GRU, SVM, Random Forest, Logistic Regression, k-Nearest Neighbours, Gaussian Naïve Bayes, Multi-Layer Perceptron, LightGBM (LGBM), and Extra Trees. Experiments were performed on two published datasets, evaluated in terms of most common classification measures used in practice: accuracy, precision, recall and F1-score as well as Brier score for probability calibration. Results indicate that the SVM and MLP models obtain state-of-the-art accuracies of up to 0.9869 and 0.9853, respectively, alongside excellent calibration with the lowest Brier scores. Although the GRU also achieves well-calibrated probabilities, its accuracy exhibits significant variability across datasets, dropping to 0.9204 on the second dataset. Meanwhile, LGBM is shown to be an efficient candidate with strong performance and reliability, whereas tree-based ensembles achieve high prediction performance but require post-hoc calibration. The results reveal the importance of obtaining well-calibrated probabilities in order to facilitate the actual use of phishing detection models within real-world email security operations.

Keywords— *Phishing Email Detection, Sentence-BERT (SBERT), Probability Calibration, Machine Learning Models, Brier Score, Web Security, Email Security System*

Received: 6 October 2025; Accepted: 20 March 2026; Published: 16 October 2026

This is an open access article under the [CC BY-NC-ND 4.0](#) license.



1. INTRODUCTION

The cybersecurity landscape is characterized by constant change, with email phishing persisting as a complex and ongoing threat [1]. Phishing attacks employ deceptive communications, frequently masquerading as legitimate emails,

to manipulate recipients into revealing sensitive information such as financial data, login credentials, or personal identifiers [2]. As cybercriminals adopt advanced social engineering techniques and exploit both technological and psychological vulnerabilities, the frequency and sophistication of these attacks have increased significantly [3].

Phishing attacks have increased in both prevalence and sophistication in recent years. Cybercriminals now leverage AI-generated content [4], spoofed email domains [5] and cloned websites to enhance credibility and bypass traditional security filters [6]. This escalation underscores the need for continuous awareness, adaptive defence mechanisms, and proactive cybersecurity strategies. Effective risk management measures include user education, threat intelligence sharing, and the implementation of Machine Learning (ML) based approaches for threat detection [7]. While accuracy is commonly used to evaluate these systems, it provides only a limited view of model performance. In security-critical environments, calibrated probabilistic predictions are also necessary, as they quantify the confidence of classification outcomes[8].

There are numerous phishing attacks which is one of the major issues at the crossroad of web engineering and informatics. Also, the most important web-based application, the email system, is a favourite of cyberattack. Security is very essential since it determines user trust and data integrity, which are the core principles of web engineering. To be able to detect efficiently and thus be able to take real-time trustful decisions, there is a need for sophisticated informatics solutions which can handle, analyse and understand large volumes of unstructured textual data (emails) [9]. This work is a step in this direction, as it develops a calibrated, semantics, aware detection pipeline which can easily be integrated into web-based email clients and enterprise security platforms.

There has been considerable progress in phishing detection research; however, a large gap still remains. The majority of earlier works have focused on aspects like classification accuracy, precision, or recall, and have very often overlooked the probability calibration aspect in real-world applications [10]. There is only a limited number of studies that have used the Brier score in a systematic way for the evaluation of phishing detection models, particularly when deep semantic embeddings such as Sentence-BERT are combined with ML classifiers [11], [12]. The absence of this factor lowers the reliability of detection systems in operational environments where the decision-making process is based on well-calibrated probability estimates rather than accuracy alone. The research goes a step further by incorporating calibration-aware evaluation with semantically rich representations to raise the level of phishing email detection.

The study presents the use of calibrated probabilistic predictions in identifying phishing emails with Sentence-BERT and ML models. We transformed the email texts into the highly informative semantic representations using Sentence-BERT embeddings, particularly the all-mpnet-base-v2 model. To have a detailed measurement of the performance of the phishing detection, we newly run nine different model configurations including Sentence-BERT with GRU, SVM, Random Forest, Logistic Regression, KNN, Gaussian Nave Bayes, MLP, LightGBM (LGBM), and Extra Trees. Along with the Brier score as the basis evaluation metric, the introduction of the score ensures that not only accurate but also trustworthy and well-calibrated probabilistic predictions are produced by our models, which is a very important condition for the effectiveness of the modern cybersecurity systems.

2. LITERATURE REVIEW

Phishing attacks have become so sophisticated that traditional defence mechanisms (like blacklists and static filtering techniques) have become less and less effective. The research community has changed its direction towards the use of Artificial Intelligence (AI) and ML (ML) as a result of this; thus, a diverse and rapidly changing body of literature has appeared, which is focused on the development of more adaptive, accurate, and robust detection systems. This review organizes the current research into four main thematic areas: feature engineering enrichment, model architecture innovations, data challenges overcoming strategies, and real-world deployment push with an emphasis on the performance metrics that corroborate the effectiveness of these approaches.

2.1 Advancements in Feature Engineering and Representation Learning

A large part of the work in phishing detection has been locating basic features and expressions in order to discover semantic and psychological clues. Shahriar et al. [11] was the first to move in this direction by

developing the idea of Phishing Psychological Traits (PPTs) that computationally represent manipulative tactics like urgency, fear, and enticement. Their experiment led to a result that by integrating those psychological scores with contextual embeddings from BERT it was possible to improve the F1-score by 4.54% when compared to baseline models therefore it is essential to understand the attacker psychology along with linguistics. Such a great attention toward better representation learning is also reflected by the popularity of transformer-based models. Studies by Atawneh et al. [13], and AbdulNabi and Yaseen [14] carried out the most thorough comparisons and consistently concluded that models such as like BERT and hybrid are almost perfect in terms of accuracy; In fact, the hybrid model presented by Atawneh et al. reached 99.61% accuracy, 99.87% precision, and 99.55% F1-score, thereby it was able to outperform models that rely on traditional word embeddings by a large margin. Similarly, Milner and Baron [15] dealt with the issue of very short text in SMS phishing and proved that semantic sentence embeddings are better for short, text classification since their procedure which used spaCy's RoBERTa transformer and an SVC classifier achieved a highest F1-score of 0.938.

2.2 Innovations in Model Architecture and Optimization

The review indicates that most of the time hybrid and ensemble models perform better than singular architectural models. One of the methods by Alsubaei *et al.* [16] was centered on a complicated hybrid framework (RNT, J) which combined ResNeXt, GRUs, autoencoders, and was optimized using the Jaya algorithm. The model achieved an accuracy of 98% while also showing fewer false positives and, thus, it outperformed the closest advanced baselines by 11, 19% in terms of both accuracy and execution time. Hybridization of different models continues to be a prevalent topic; Ullah et al. [17] designed a model that used both CNNs and RNNs for the analysis of URLs, the CNN-based model achieving an accuracy of 98.74% and a speed of over 130,000 URLs per second thus demonstrating real-time capability. Balogun et al. [18] demonstrated the potential of getting excellent results through the ensemble of functional tree algorithms: the variant BT-FT-2 attained 98.51% accuracy and a false positive rate as low as 0.015. In contrast, the work of Nayak et al. [19] focused on efficiency through the introduction of feature selection leading to a lightweight Feedforward Neural Network (FNN) achieving excellent accuracy of 94.46% using only 14 features, thus making the trade of high detection capability for the lowest computational cost.

2.3 Addressing Data Imbalance and the Need for Explainability

A significant limitation of the real-world implementation is the intrinsic class imbalance that is characteristic of the benign instances while the phishing ones are very few. Qi et al. [20] directly tackled this issue by introducing new under sampling strategies (FMPED and FMMPED) that have made it possible to attain an outstanding F1-score and accuracy of about 99.45% on a dataset with a 10:1 imbalance ratio, thus successfully eliminating model bias. While that is happening, the field of research is also grappling with the problem of the "black, box" nature of complicated AI models. Lim et al. [21] introduced a seminal contribution with the EXPLICATE framework, which comprises XAI tools and an LLM that can produce herbal language explanations. Moreover, it has reached a high accuracy of 98.4%, which is comparable to that of the best non-explainable models, at the same time offering explanations with 94.2% accuracy and 96.8% consistency, a very important step towards getting user trust.

2.4 Towards Multilingual and Real-World Deployment

The last thematic centre moves towards increasing the variety and the different applications of phishing detection systems. An et al. [22] addressed the problem of language restrictions by using Open-Source Intelligence (OSINT) features, thus leading to the improvement of a Random Forest model performance up to 97.37% accuracy in a multilingual identification task. Ultimately, the value of the research is determined by its practical use in the real world. Most of the research works have been committed to the developing of practical solutions. Adebowale et al. [23] created a hybrid CNN-LSTM model (IPDS) that reached 93.28% accuracy and was executed as a browser extension, thus, it took an average of 25 seconds for the detection process. This focus on tangible deployment ensures that the advancement of research translates to the protection of users.

Research on the use of AI to identify phishing has evolved from focus on single models to the development of more advanced and complex systems. Nowadays, research works are not only focused on accuracy but

also on aspects such as context-awareness through complex feature construction, efficiency through hybrid architecture and optimizations, trustworthiness through explainable models, and applicability through multilingual factors and real-time deployment. Published papers show that different performance metrics lead to similar results. One of the most difficult tasks is still the calibration of probabilistic prediction, which is very significant for the demonstration of trustworthy and actionable decision support as well as for operational contexts

The research papers that were analysed in the literature review have shown a great performance of the studies in terms of accuracy, precision, and F1 score. However, they have usually ignored the confidence that goes along with the model output. This is quite significant since the predictive confidence can be one of the factors that determine how trustworthy the system's decision-making is in practice. One of the metrics that can be used in this case is the Brier Score, which measures not only the accuracy of the prediction but also the reliability and calibration of the probability estimate that are given. The inclusion of such metrics would be a step towards phishing detection systems not being only effective but also trustworthy to a certain degree, thus allowing decision-making at a more granular level of risk and, hopefully, leading to the real-world processes of cybersecurity response being accelerated.

2.5 Sentence-BERT (SBERT)

Sentence-BERT (SBERT) is a variation of BERT model that is created to produce semantically meaningful sentence-level embeddings[24]. It essentially solves one of the biggest problems that the original BERT models had, which is being computationally inefficient for sentence, pair tasks like semantic similarity. SBERT, through Siamese and triplet network architectures for fine, tuning, embeds each sentence into fixed, dimensional vector spaces where semantic similarity can be quickly computed by cosine similarity or Euclidean distance without the need for pairwise input processing. Thus, it makes semantic search, clustering, and paraphrase detection scalable and at a high representational quality. SBERT has been extensively used in NLP pipelines for tasks such as document retrieval, duplicate question detection, and real-time semantic matching, and it has become a standard tool in both research and industry applications [25].

2.6 Dataset Description

This research comprised two phishing email datasets that are available to the public and were fetched from Kaggle in order to increase the generalization and robustness of the phishing detection models. The foremost dataset, the Phishing Email Detection Dataset created by Subhadeep Chakraborty, is a comprehensive text collection of emails with labels for a ML, based phishing classification research problem. It initially comprises 18,650 email samples, each containing two primary attributes: Email Text, representing the full textual content of the email body, and Email Type, indicating whether an email is classified as Safe or Phishing. After preprocessing, which involved removing entries with missing text fields and eliminating non-informative columns, the final working dataset comprises 11,322 legitimate emails (60.7%) and 7,328 phishing emails (39.3%), indicating a moderately skewed class distribution. The second dataset, Phishing Email Dataset compiled by Naser Abdullah Alam, is a large, scale phishing email corpus formed by integrating multiple well-known datasets, including Enron, Ling, CEAS, Nazario, Nigerian Fraud, and SpamAssassin. The integrated dataset is made up of close to 82, 500 emails, out of which 42, 891 are phishing (spam) and 39, 595 are legitimate (ham) thus the class distribution is almost balanced. Besides the textual content, the dataset also has the contextual metadata like the sender details and the time stamps making it a perfect fit for advanced natural language processing, ML, and deep learning, based phishing detection experiments. [26].

2.7 Environmental Setup

The experiments were carried out in Python version 3. 12. 3 in a computational environment like Kaggle notebooks which provided the necessary CPU and GPU resources (e. g., NVIDIA T4 GPU for PyTorch, based models). The main libraries and their versions that were used throughout the paper are as follows: sentence-transformers 2. 2. 2 (for loading and using the pre, trained all-mpnet-base-v2 model), scikit-learn 1. 2. 2, LGBM 4. 5. 0, torch 2. 4. 1 (required for the GRU implementation), numpy 1. 26. 4, pandas 2. 2. 3, matplotlib 3. 9. 2, and seaborn 0. 13. 2. Additionally, imbalanced-learn 0. 12. 3 was used both for generating synthetic minority samples and for the calculation of sample weights where necessary. Class imbalance in the datasets was handled by using built-in class weighting mechanisms

setting `class_weight='balanced'` in supported classifiers (including LGBM, Support Vector Machine, Random Forest, Extra Trees, and Multi-Layer Perceptron) or through explicit sample weights generated via `sklearn.utils.compute_sample_weight('balanced')` for models such as Gaussian Naïve Bayes.

In order to make the results fully reproducible, the random seed was set to 42 for all stochastic processes. This includes data splitting (using `train_test_split` with `random_state=42` and stratified sampling), model initializations (`random_state=42` in scikit-learn and LGBM classifiers), NumPy random operations (`np.random.seed(42)`), PyTorch randomness (`torch.manual_seed(42)` and `torch.cuda.manual_seed_all(42)` where applicable), Python's native random module (`random.seed(42)`), and CuDNN settings for deterministic behavior (`torch.backends.cudnn.deterministic = True` and `torch.backends.cudnn.benchmark = False`). There might be slight differences due to hardware-specific and non-determinism on GPUs, but the overall results remain highly consistent across runs.

3. RESEARCH METHODOLOGY

The methodology was first applied to the initial dataset, referred to as the Phishing Email Detection Dataset, which includes labelled emails that were classified as either "Safe Email" or "Phishing Email". The process consisted of four stages: data preprocessing, feature extracting, class balancing, and model training and evaluation. During preprocessing, the dataset was cleaned by removing missing email texts (`df.dropna(subset=['Email Text'])`), dropping irrelevant columns such as `Unnamed: 0`, and eliminating duplicates. The target variable, *Email Type*, was encoded numerically using `df['Label'] = df['Email Type'].map({'Safe Email': 0, 'Phishing Email': 1})`, mapping "Safe Email" to zero and "Phishing Email" to 1. In the feature extraction stage, email texts were encoded as dense numerical vectors using SBERT (*all-mpnet-base-v2*), producing embeddings such as `[0.12, -0.34, 0.56, ...]` that preserve contextual meaning. The dataset was split into the training (80%) and the testing (20%) set by stratified sampling (`train_test_split(X, y, test_size=0.20, stratify=y, random_state=42)`) in order to keep the ratio of safe to phishing emails the same as in the whole data. To solve the problem of class imbalance, we employed SMOTE for models requiring balanced dataset inputs, while class weighting was used for models like Support Vector Machines (SVM) (`class_weight='balanced'`) that increases the error of the minority phishing emails, thus forcing the model to pay more attention to them. The model training included diverse set of classifiers with key parameters as shown in Table 1.

Table 1. Models and Parameters Used for Phishing Detection

S/NO	Model	Type	Key Parameters Used
1.	Extra Trees Classifier (ETC)	Tree-based Ensemble	<code>n_estimators=600</code> , <code>criterion='gini'</code> , <code>random_state=42</code>
2.	Random Forest Classifier (RFC)	Tree-based Ensemble	<code>n_estimators=200</code> , <code>criterion='gini'</code> , <code>random_state=42</code>
3.	LGBM Classifier	Gradient Boosting	Grid Search over: <code>num_leaves=[31, 50]</code> , <code>max_depth=[-1, 10]</code> , <code>learning_rate=[0.01, 0.1]</code> , <code>n_estimators=[100, 200]</code>
4.	Logistic Regression	Linear / Probabilistic	Grid Search over: <code>C=[0.1, 1, 10]</code> , <code>solver=['liblinear', 'lbfgs']</code> , <code>max_iter=5000</code>
5.	Gaussian Naïve Bayes	Probabilistic	<code>priors=None</code> , <code>var_smoothing=1e-9</code>
6.	k-Nearest Neighbors	Instance-based	Grid Search over: <code>n_neighbors=[3, 5, 7]</code> , <code>weights=['uniform', 'distance']</code> , <code>metric=['euclidean', 'manhattan']</code>
7.	Multi-Layer Perceptron	Neural Network	Grid Search over: <code>hidden_layer_sizes=[(100,),(50,50)]</code> , <code>activation=['relu', 'tanh']</code> , <code>solver=['adam', 'sgd']</code> , <code>learning_rate=['constant','adaptive']</code> , <code>max_iter=500</code>
8.	Gated Recurrent Unit	Recurrent Neural Network	<code>embedding_dim=768</code> (SBERT output), <code>hidden_size=128</code> , <code>num_layers=1</code> , <code>batch_size=32</code> , <code>epochs=10</code> , <code>dropout=0.2</code> , <code>optimizer=Adam(lr=0.001)</code>
9.	Support Vector Machine	Kernel-based	Grid Search over: <code>C=[0.1, 1, 10]</code> , <code>kernel=['linear','rbf','poly']</code> , <code>gamma=['scale','auto']</code>

Tree-based ensembles were represented by Extra Trees, Random Forest, and LGBM that employ ensemble averaging and are less sensitive to hyperparameters. Linear and probabilistic models like Logistic Regression and Gaussian Nave Bayes were used to provide interpretable baselines. Distance-based learning was done with k, Nearest Neighbors

(kNN), while neural architectures consisted of Multi, Layer Perceptron (MLP) and Gated Recurrent Unit (GRU) networks aimed at capturing nonlinear and sequential patterns in text. Also, SVMs with linear, polynomial, and radial basis function kernels were used to take advantage of the high-dimensional SBERT embeddings.

The second dataset's workflow was comprehensively detailed in four major stages: data preprocessing, feature extraction, class balancing, and model training with evaluation. The data for preprocessing was made cleaner by filling in missing values for the subject and body fields (`df['subject'] = df['subject'].fillna("")`, `df['body'] = df['body'].fillna("")`), merging them into one text field, and removing rows with invalid labels obtained after mapping. The target variable, label, was given a numeric representation through a comprehensive mapping (`df['label'] = df['label'].map(label_map)`), where `label_map` consisted of the different forms like `{"0": 0, "safe": 0, "legitimate": 0, "ham": 0, "benign": 0, "not phishing": 0, "1": 1, "phishing": 1, "spam": 1, "malicious": 1, "phish": 1}`, thereby producing standardized binary labels (0 for safe, 1 for phishing). The feature extraction stage utilizes the pre-trained Sentence-BERT (*all-mpnet-base-v2*) model to generate 768-dimensional semantic embeddings. These embeddings not only capture deep contextual meaning but also address the vocabulary limitations of traditional frequency-based approaches like TF-IDF by representing synonyms, negation, and subtle semantic variations more effectively. To prevent data leakage, embeddings for the training and test sets are generated separately, with batch processing utilized to ensure computational efficiency.

The model assessment became extensive. Accuracy served as a measure of overall predictive capability, and precision, recall, and F1-score described the trade-offs between false positives and false negatives. Confusion matrices were employed to show the patterns of misclassification, whereas ROC-AUC curves were used to evaluate the discriminative power at different thresholds. The Brier score and calibration curves were used to assess probabilistic calibration so that the predicted probabilities reflected the actual outcomes. Additionally, 5-fold cross-validation was used to validate models further, reporting average accuracy, Standard Deviation (SD), and 95% confidence intervals to indicate predictive stability and trustworthiness. The pipeline, which was written in Python and used Scikit, learn, LGBM, and PyTorch/TensorFlow, ensured reproducibility and robustness by including code-driven preprocessing (`df.dropna`, column removal, label encoding), SBERT embeddings, class weighting, stratified splits, and systematic evaluation metrics, thus providing a solid base for phishing email detection on the first dataset.

RESULTS AND DISCUSSIONS

The differential performance of nine ML models, each trained on SBERT embeddings for phishing email detection, highlights significant differences in both classification accuracy and probability calibration between the two datasets. The model performance measures included accuracy, precision, recall, and F1-score, SD, and Confidence Intervals (CI) along with the Brier score which is indicative of the calibration of probabilistic predictions. The analysis of results reveals the variation of model efficiency and calibration from one dataset to the other, thus giving insight into their strength and ability to be applied to a different data set for the task of phishing email detection. A succinct summary of the results is available in Table 2 for the Phishing Email Detection Dataset and Table 3 for the Phishing Email Dataset.

The experimental evaluation of nine probability-based SBERT derived detection models in Table 2 offers valuable insights not only into their classification performances but also into the correctness of their probabilistic predictions. The tree-based ensemble models, Extra Trees Classifier (ETC) and Random Forest Classifier (RFC), were most effective in classification, reaching accuracies of 0.9418 and 0.9455, respectively. Both the models kept precision, recall, and F1-scores fairly balanced at about 0.93–0.96 and hence, they were consistent in the correct identification of phishing and safe emails. Yet, their Brier scores (0.0630 for ETC and 0.0683 for RFC) indicate that the predicted probabilities of these models are not as close to the real ones even though they are good classifiers. The SDs of 0.0029–0.0032 and CIs of about [0.9392, 0.9443] for ETC and [0.9427, 0.9483] for RFC, respectively, also suggest that the performance of the models is quite robust, but there is still some variation across cross-validation folds. Application of calibration methods such as Platt scaling or isotonic regression to these models can make their probability estimates more accurate and thus enhance them further. LGBM emerged as a strong candidate for deployment, as it combined a high accuracy of 0.9597 with a lower Brier score of 0.0285, thus indicating that the probabilities were well-calibrated. Precision and recall were about 0.94–0.96, and the SD was as small as 0.0027 with the 95% CI [0.9574, 0.9621], so LGBM not only makes correct classification decisions but also provides reliable confidence estimates. LGBM stands out from ETC and RFC in terms of the combination of higher accuracy and lower Brier score, which demonstrates its potential in real-world use cases where both correctness and probability confidence are important. The Logistic

Regression (LR) model reached an accuracy of 0.9550 and had a precision of 0.92, a recall of 0.97, and an F1-score of 0.94. Its Brier score of 0.0325 is indicative of fairly good probability calibration. The SD of 0.0015 and 95% CI [0.9538, 0.9563] confirm that LR predictions are stable, but the model is slightly less reliable in terms of probabilistic confidence when compared to LGBM or some other models. In general, LR is still a robust, easily understandable, and especially interpretable option when the simplicity of the model is a factor.

Table 2. Performance of SBERT and ML Models on Phishing Email Detection Dataset

Model	Accuracy	Precision	Recall	F1-Score	Brier Score	SD	CI
ETC	0.9418	0.96	0.90	0.93	0.0630	0.0029	[0.9392, 0.9443]
RFC	0.9455	0.94	0.93	0.93	0.0683	0.0032	[0.9427, 0.9483]
LGBM Classifier	0.9597	0.94	0.96	0.96	0.0285	0.0027	[0.9574, 0.9621]
Logistic Regression	0.9550	0.92	0.97	0.94	0.0325	0.0015	[0.9538, 0.9563]
Gaussian Naïve Bayes	0.9204	0.88	0.91	0.89	0.0808	0.0050	[0.9160, 0.9248]
k-Nearest Neighbors	0.9577	0.91	0.98	0.94	0.0339	0.0014	[0.9565, 0.9590]
Multi-Layer Perceptron	0.9635	0.97	0.97	0.97	0.0228	0.0024	[0.9614, 0.9656]
Gated Recurrent Unit	0.9539	0.96	0.97	0.96	0.0260	0.0030	[0.9512, 0.9565]
Support Vector Machine	0.9691	0.97	0.97	0.97	0.0251	0.0006	[0.9686, 0.9696]

On the other hand, Gaussian Naïve Bayes (GNB) was the worst performer in all metrics with an accuracy of 0.9204, precision of 0.88, recall of 0.91, and F1-score of 0.89. The Brier score of 0.0808, which is the highest for all models, indicates poor probability calibration. In addition, its SD of 0.0050 and 95% CI [0.9160, 0.9248] point to less stability across cross-validation splits. These findings indicate that GNB's independence assumptions do not match well with SBERT embeddings and thus the model is not suitable for high-stakes scenarios. The k-Nearest Neighbors (kNN) algorithm demonstrated solid classification capability with a mean cross-validation accuracy of 0.9577, precision of 0.91, and recall of 0.98, thereby achieving an F1-score of 0.94. The Brier score of 0.0339 signifies a moderate level of calibration. The SD of 0.0014 and 95% CI [0.9565, 0.9590] indicate that the performance is consistent across folds. The lower precision compared to recall indicates that kNN is less conservative, thus producing more false positives while still capturing most phishing emails. Artificial intelligence architectures, namely Multi-Layer Perceptron (MLP) and Gated Recurrent Unit (GRU), demonstrated strong performance. MLP attained an accuracy of 0.9635 with balanced precision, recall, and F1-score of 0.97, while GRU achieved an accuracy of 0.9539 with similar performance metrics. Their Brier scores (0.0228 for MLP and 0.0260 for GRU) indicate good probability estimates. With SDs of 0.0024–0.0030 and relatively tight CIs ([0.9614, 0.9656] for MLP, [0.9512, 0.9565] for GRU), these models are accurate and reasonably stable, though not the top-performing in this table. Due to their capability to model nonlinear dependencies in SBERT embeddings, they remain effective for semantic classification tasks.

In summary, most models show strong classification capabilities; however, significant differences in probability calibration and reliability can be observed. The Support Vector Machine (SVM), and the Multi-Layer Perceptron (MLP), together with the Gated Recurrent Unit (GRU) provides an excellent balance between accuracies (around 0.95–0.97), low Brier scores (around 0.02–0.03), and small variations, hence, they are among the most appropriate to be used in risk scenarios. LGBM is also a strong candidate with good calibration (Brier = 0.0285), high mean cross-validation accuracy, and stable predictions. Tree-based ensembles, namely Extra Trees (ETC) and RFC, provide good mean cross-validation accuracies, but their Brier scores are higher (around 0.06–0.07), which indicates a requirement for calibration before deployment. Logistic Regression (LR) is a comprehensible, interpretable model with reasonable

calibration, while k-Nearest Neighbors (kNN) has strong mean cross-validation accuracy but less reliable probability estimates. Lastly, Gaussian Naïve Bayes (GNB) is the model that performed the worst in this study, having both lower accuracy and the highest Brier score, thus, it is an unsuitable method for detecting phishing based on semantic embedding. This assessment emphasizes the need to consider not only accuracy but also probability calibration when deciding on models to be deployed in the wild, especially in cybersecurity where confidence and reliability of predictions are critical.

The calibration plots depicted in Figures 1-9 represent how well the nine S-BERT, based models could predict in the Phishing Email Detection Dataset. These illustrations are instrumental in shedding light on the models' performances beyond the conventional metrics like accuracy and F1-score, as they focus on the concordance between the models' predicted probabilities and the actual outcomes. The subsequent discussion goes through these graphs and interprets their features in terms of the performance metrics shown in Table 2. In this way, it becomes possible to assess the models exhaustively, not only are they accurate, but also their confidence levels can be trusted.

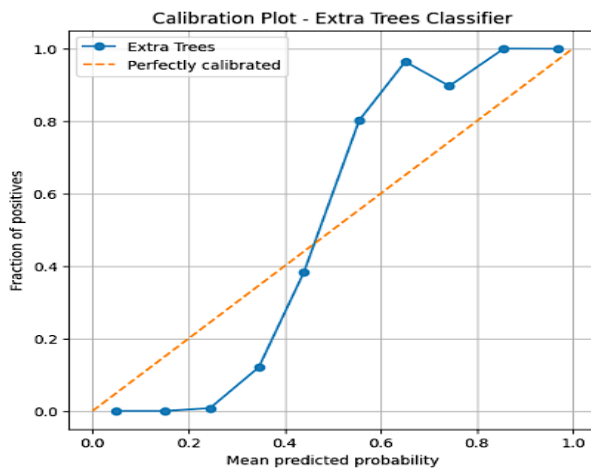


Figure 1. Calibration Plot Using ETC

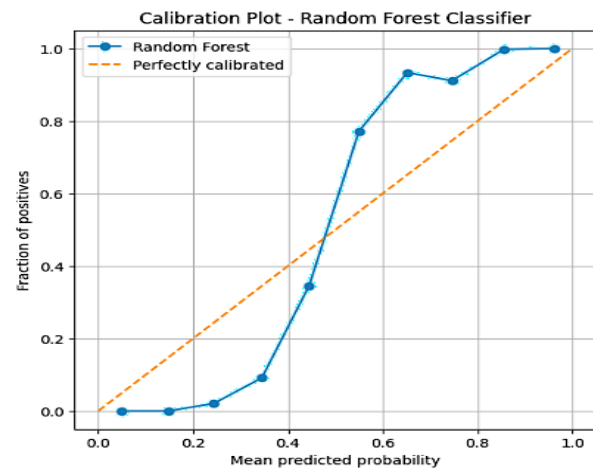


Figure 2. Calibration Plot Using RFC

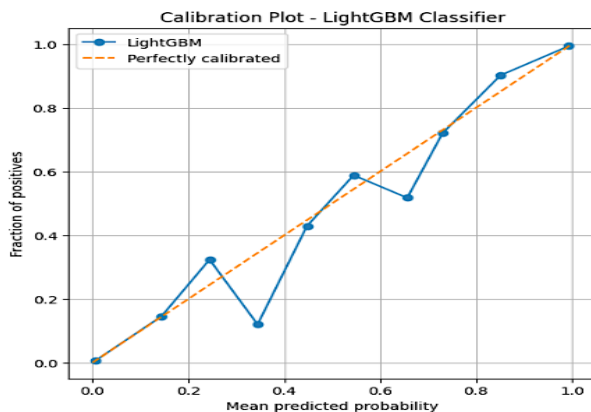


Figure 3. Calibration Plot Using LGBM Classifier

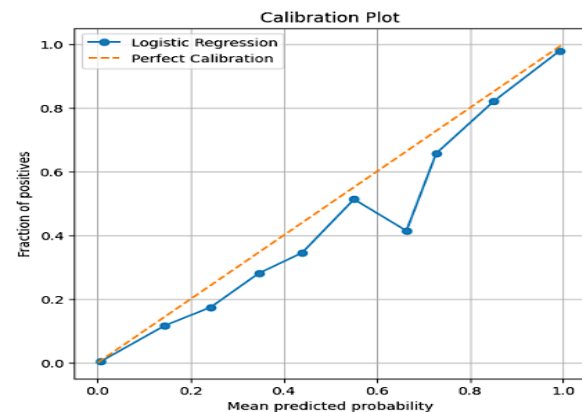


Figure 4. Calibration Plot Using Logistic Regression Classifier

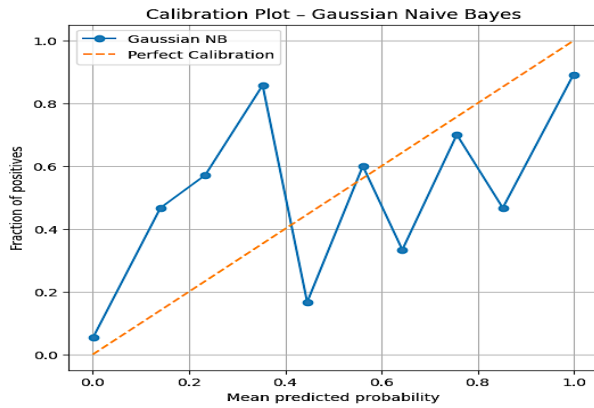


Figure 5. Calibration Plot Using Gaussian Naïve Bayes Classifier

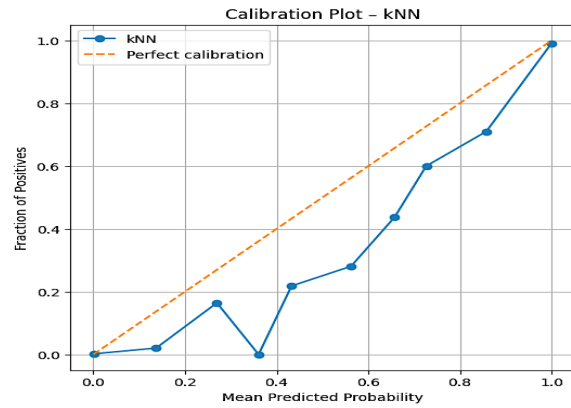


Figure 6. Calibration Plot Using k-Nearest Neighbors Classifier

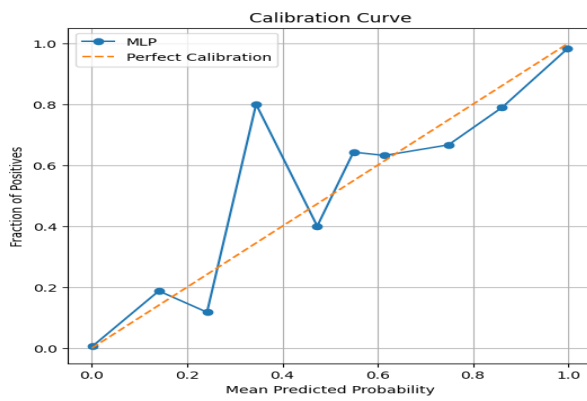


Figure 7. Calibration Plot Using MLP Classifier

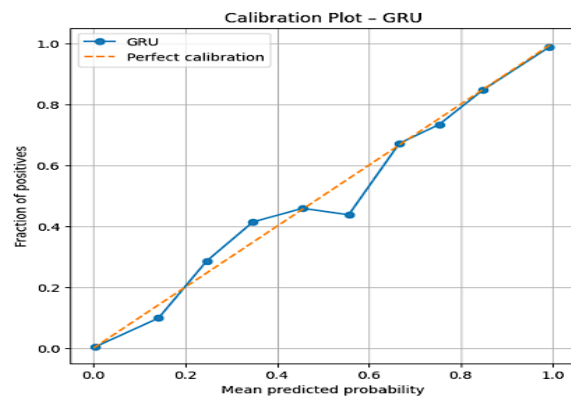


Figure 8. Calibration Plot Using GRU Classifier

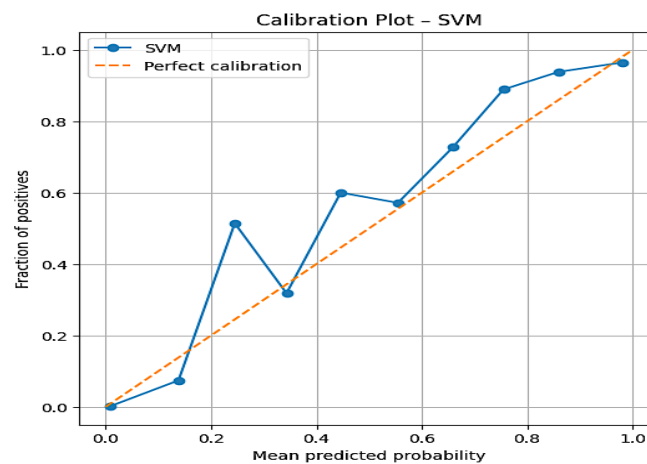


Figure 9. Calibration Plot Using SVM Classifier

The calibration plots in Figures 1–9 provide meaningful diagnostic information regarding the probabilistic reliability of the nine SBERT-based models evaluated on the Phishing Email Detection Dataset. These visualizations complement the Brier scores in Table 2 by revealing the specific patterns of deviation from perfect calibration, where the diagonal line represents ideal alignment between mean predicted probabilities and observed positive frequencies.

The calibration plots for the Phishing Email Detection Dataset reveal distinct reliability patterns across the models. SVM, MLP, and GRU show excellent calibration, with curves closely following the diagonal and minimal deviations, corresponding to the lowest Brier scores (0.0228–0.0260), making them highly suitable for deployment with reliable threshold-based actions (e.g., auto-quarantining high-probability phishing emails). LGBM and Logistic Regression exhibit good but slightly conservative calibration, with mild under-confidence at intermediate probability levels (0.4–0.7) and Brier scores of approximately 0.0285–0.0325; this tendency helps reduce false positives in cybersecurity settings. kNN achieves solid calibration overall, though with minor over-confidence at higher probabilities (>0.8), aligning with its Brier score of approximately 0.0339.

In contrast, tree-based ensembles (Extra Trees and Random Forest) display typical sigmoid-shaped curves, featuring over-confidence near probability 1.0 and under-confidence in intermediate ranges, resulting in higher Brier scores approximately (0.0630–0.0683) and necessitating post-hoc calibration (e.g., isotonic regression) for practical use. Gaussian Naïve Bayes performs poorest, with severe probability compression and under-confidence, yielding the highest Brier score approximately (0.0808) and limited probabilistic trustworthiness. Overall, the plots confirm that neural models and SVM offer the most dependable probability estimates for real-world phishing detection, while tree-based models require additional calibration to mitigate risks like alert fatigue or undetected threats despite their strong model classification effectiveness.

The evaluation of nine models for phishing email detection in Table 3 demonstrates distinct differences in classification performance, probability calibration, and reliability. Among the tree-based ensembles, ETC and RFC achieved mean cross-validation accuracies of 0.9722 and 0.9683, respectively, with balanced precision, recall, and F1-scores ranging from 0.96 to 0.98. Their Brier scores (0.0409 for ETC and 0.0404 for RFC) indicate moderately calibrated probability estimates, while low SDs (0.0013–0.0015) and narrow CIs ([0.9710, 0.9733] for ETC, [0.9669, 0.9696] for RFC) suggest stable performance across folds. LGBM and Logistic Regression slightly outperform the ensembles in calibration, with LGBM achieving an accuracy of 0.9778 and Logistic Regression with 0.9747, alongside lower Brier scores (0.0178 and 0.0196), demonstrating well-calibrated predictions and consistent performance across cross-validation. k-Nearest Neighbors (kNN) also shows strong performance with 0.9832 accuracy, recall of 0.99, and a low Brier score (0.0127), indicating both high sensitivity and reliable probabilistic outputs. Neural network models, the Multi-Layer Perceptron (MLP) achieved a very high accuracy of 0.9853 and F1-score of 0.99, with a low Brier score of 0.0115. In contrast, the Gated Recurrent Unit (GRU) achieved a much lower mean accuracy of 0.9204, though it maintained high precision, recall, and F1-scores (all 0.99) alongside an excellent Brier score of 0.0105. While MLP is highly stable (SD = 0.0010, CI [0.9848, 0.9858]), GRU exhibits greater variability (SD = 0.0315, CI [0.8928, 0.9481]), suggesting inconsistent performance despite some strong classification metrics, reducing confidence in its reliability. Support Vector Machine (SVM) achieved the highest mean accuracy (0.9869) and matches the top F1-scores of 0.99, showing the best probability calibration (Brier = 0.0078) with minimal variability (SD = 0.0006, CI [0.9860, 0.9879]), making it one of the most reliable models overall. In contrast, Gaussian Naïve Bayes (GNB) performed the weakest, with accuracy of 0.9149, lower recall (0.88), and the highest Brier score (0.0800), indicating poor probability calibration despite low variability (SD = 0.0010).

Table 3. Performance of SBERT and ML Models on Phishing Email Dataset

Model	Mean CV Accuracy	Precision	Recall	F1-Score	Brier Score	SD	CI
ETC	0.9722	0.98	0.97	0.97	0.0409	0.0013	[0.9710, 0.9733]
RFC	0.9683	0.98	0.96	0.97	0.0404	0.0015	[0.9669, 0.9696]
LGBM Classifier	0.9778	0.98	0.98	0.98	0.0178	0.0016	[0.9764, 0.9791]
Logistic Regression	0.9747	0.98	0.98	0.98	0.0196	0.0010	[0.9738, 0.9755]
Gaussian Naïve Bayes	0.9149	0.95	0.88	0.92	0.0800	0.0010	[0.9141, 0.9158]
k-Nearest Neighbors	0.9832	0.98	0.99	0.98	0.0127	0.0005	[0.9828, 0.9837]
Multi-Layer Perceptron	0.9853	0.99	0.99	0.99	0.0115	0.0010	[0.9848, 0.9858]
Gated Recurrent Unit	0.9204	0.99	0.99	0.99	0.0105	0.0315	[0.8928, 0.9481]
Support Vector Machine	0.9869	0.99	0.99	0.99	0.0078	0.0006	[0.9860, 0.9879]

In general, the study indicates that a few models can be raised as highly accurate in terms of classification, but the differences in calibration and stability being factors deciding the issue: SVM, MLP, and kNN are the best-performing models in terms of mean cross-validation accuracy (0.9869, 0.9853, and 0.9832, respectively), probability

reliability, and consistent performance, thus they can be most efficiently deployed, while GNB is the least clearly suitable.

4. CONCLUSION

The study outlines a powerful methodology for the identification of phishing emails by coupling the semantically rich Sentence-BERT (all-mpnet-base-v2) embeddings with an assortment of ML classifiers tested on two publicly available datasets. One of the major contributions is the focus on probability calibration mostly judged by the Brier score together with the standard metrics of accuracy, precision, recall, and F1-score, thus going beyond a markedly different scenario of research subfield that had been addressed primarily by accuracy alone.

The findings across the two datasets demonstrate that the Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) are the most robust models of choice, consistently delivering high accuracies (up to 0.9869 for SVM) and excellent calibration with low Brier scores (ranging from 0.0078 to 0.0251). Although the Gated Recurrent Unit (GRU) also achieves exceptionally low Brier scores (0.0105–0.0260) and high F1-scores, it exhibits significant performance variability across datasets; its accuracy falls to 0.9204 on the second dataset with a high standard deviation (0.0315) and a wide confidence interval [0.8928, 0.9481], which reduces confidence in its consistency. LGBM serves as a highly practical alternative for deployment in resource-constrained settings; it achieves high accuracies (0.9597 to 0.9778) paired with strong calibration (Brier scores of 0.0178 to 0.0285) and minimal variability. In contrast, tree-based ensembles (Extra Trees and Random Forest) achieve strong classification accuracy (0.9418 to 0.9722) but are limited by higher Brier scores (0.0404 to 0.0683), highlighting a critical need for post-hoc calibration prior to deployment. Logistic Regression provides a balanced, highly interpretable alternative with reasonable calibration, whereas k-Nearest Neighbors offers high classification accuracy but yields less reliable probability estimates. Lastly, Gaussian Naïve Bayes (GNB) represents the weakest approach for semantic-embedding-based phishing detection, consistently exhibiting lower accuracies (0.9149 and 0.9204, respectively) and the worst calibration (Brier scores of 0.0800 to 0.0808). Ultimately, these well-calibrated probabilities enable email security systems to make more informed and nuanced operational decisions. For instance, systems can set dynamic thresholds tailored to specific risk tolerances such as aggressively auto-quarantining emails scored above 90% as phishing to minimize escapes, while routing lower-confidence predictions for manual security review. This approach minimizes false positives that erode user trust, optimizes resource allocation in security operations centres, and supports cost-sensitive actions where the damage of an undetected phishing email far outweighs the cost of flagging a legitimate one. Consequently, the research highlights that reliable confidence estimates are just as vital as accurate classifications in real-world cybersecurity settings. By recommending models like SVM, MLP, and LGBM, while suggesting that GRU be used with caution due to its variability, the study advances the development of more dependable, deployable solutions to counter evolving phishing threats and establishes a valuable benchmark for future semantic text classification efforts in cybersecurity.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for the suggestions to improve the paper.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Oluwasegun Abiodun Abioye: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;

Abraham Ewwiekpaefe: Project Administration, Supervision, Writing – Review & Editing;

Philip Odion: Project Administration, Supervision – Review & Editing.

CONFLICT OF INTERESTS

No conflicts of interest were disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. <https://publicationethics.org/>

DATA AVAILABILITY

The data that support the findings of this study are openly available in Kaggle at <https://www.kaggle.com/datasets/subhajournal/phishingemails> (DOI: 10.34740/kaggle/dsv/6090437) and <https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset> (DOI: 10.34740/kaggle/ds/5074342). These data were derived from sources in the public domain.

REFERENCES

- [1] M. A. Daniel, S.-C. Chong, L.-Y. Chong, and K.-K. Wee, "Optimising Phishing Detection: A Comparative Analysis of Machine Learning Methods with Feature Selection," *Journal of Informatics and Web Engineering*, vol. 4, no. 1, pp. 200–212, Feb. 2025, doi: 10.33093/jiwe.2025.4.1.15.
- [2] T. S. Tajamul and K. Aman, "Unveiling the Efficacy of AI-based Algorithms in Phishing Attack Detection," *Journal of Informatics and Web Engineering*, vol. 3, no. 2, pp. 116–133, Jun. 2024, doi: 10.33093/jiwe.2024.3.2.9.
- [3] H. Faotu and O. N. Asheshemi, "Human Vulnerabilities in Cybersecurity: Analyzing Social Engineering Attacks and AI-Driven Machine Learning Countermeasures," *Journal of Science and Technology*, vol. 30, no. 1, pp. 72–84, Dec. 2025, doi: 10.20428/jst.v30i1.2597.
- [4] F. Serpa and D. V. Voinea, "Synthetic Media: Deepfakes, AI-Generated Content, and Authenticity in the Digital Society," *Encyclopedia 2026, Vol. 6, Page 175*, vol. 6, no. 8, p. 175, Aug. 2026, doi: 10.3390/ENCYCLOPEDIA6080175.
- [5] N. Fatin, N. M. Nazlan, S. Fadzlun, and M. Salleh, "SpoofSentry: Email Spoofing Detection Tool Using Domain-Based Authentication," *Applied Information Technology And Computer Science*, vol. 7, no. 1, pp. 1165–1185, Jul. 2026, doi: 10.30880/aitcs.2026.07.01.063.
- [6] A. Wong, A. Abuadbba, M. Almashor, and S. Kanhere, "PhishClone: Measuring the Efficacy of Cloning Evasion Attacks," Sep. 2022, [Online]. Available: <http://arxiv.org/abs/2209.01582>
- [7] A. Basit, M. Zafar, and X. Liu, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun Syst*, vol. 76, no. 1, pp. 139–154, Jan. 2021.
- [8] R. O. Lane, "A comprehensive review of classifier probability calibration metrics," 2025.
- [9] M. K. Moizuddin, M. Kabeer, and M. Misbahuddin, "Cyber-Phishing Analysis offering Cyber Security for Social Networks," in *2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, Pune, India: IEEE, Oct. 2024.

- [10] H. Ghalechyan, E. Israyelyan, A. Arakelyan, G. Hovhannisyan, and A. Davtyan, “Phishing URL detection with neural networks: an empirical study,” *Sci. Rep.*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-74725-6.
- [11] S. Shahriar, A. Mukherjee, and O. Gnawali, “IMPROVING PHISHING DETECTION VIA PSYCHOLOGICAL TRAIT SCORING,” 2022.
- [12] T. Jiang, S. Huang, Z. Luan, D. Wang, and F. Zhuang, “Scaling Sentence Embeddings with Large Language Models,” Jul. 2023, [Online]. Available: <http://arxiv.org/abs/2307.16645>
- [13] S. Atawneh and H. Aljehani, “Phishing Email Detection Model Using Deep Learning,” *Electronics (Switzerland)*, vol. 12, no. 20, Oct. 2023, doi: 10.3390/electronics12204261.
- [14] I. AbdulNabi and Q. Yaseen, “Spam email detection using deep learning techniques,” in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 853–858. doi: 10.1016/j.procs.2021.03.107.
- [15] H. Milner and M. Baron, “Establishing an Optimal Online Phishing Detection Method: Evaluating Topological NLP Transformers on Text Message Data,” *Journal of Data Science and Intelligent Systems*, vol. 2, no. 1, pp. 37–45, Jul. 2023, doi: 10.47852/bonviewjdsis32021131.
- [16] F. S. Alsubaei, A. A. Almazroi, and N. Ayub, “Enhancing Phishing Detection: A Novel Hybrid Deep Learning Framework for Cybercrime Forensics,” *IEEE Access*, vol. 12, pp. 8373–8389, 2024, doi: 10.1109/ACCESS.2024.3351946.
- [17] A. Ullah, R. A. Shah, S. A. Nawaz, N. Ahmad, and M. H. Malik, “Enhancing Phishing Detection, Leveraging Deep Learning Techniques”, *JCBI*, Feb. 2024. .
- [18] A. O. Balogun et al., “Improving the phishing website detection using empirical analysis of Function Tree and its variants,” *Heliyon*, vol. 7, no. 7, Jul. 2021, doi: 10.1016/j.heliyon.2021.e07437.
- [19] G. S. Nayak, B. Muniyal, and M. C. Belavagi, “Enhancing Phishing Detection: A Machine Learning Approach With Feature Selection and Deep Learning Models,” *IEEE Access*, vol. 13, pp. 33308–33320, 2025, doi: 10.1109/ACCESS.2025.3543738.
- [20] Q. Qi, Z. Wang, Y. Xu, Y. Fang, and C. Wang, “Enhancing Phishing Email Detection through Ensemble Learning and Undersampling,” *Applied Sciences (Switzerland)*, vol. 13, no. 15, Aug. 2023, doi: 10.3390/app13158756.
- [21] B. Lim, R. Huerta, A. Sotelo, A. Quintela, and P. Kumar, “EXPLICATE: Enhancing Phishing Detection through Explainable AI and LLM-Powered Interpretability,” Mar. 2025, [Online]. Available: <http://arxiv.org/abs/2503.20796>
- [22] P. An, R. Shafi, T. Mughogho, and O. A. Onyango, “Multilingual Email Phishing Attacks Detection using OSINT and Machine Learning,” Jan. 2025, [Online]. Available: <http://arxiv.org/abs/2501.08723>
- [23] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, “Intelligent phishing detection scheme using deep learning algorithms,” *Journal of Enterprise Information Management*, vol. 36, no. 3, pp. 747–766, Apr. 2023, doi: 10.1108/JEIM-01-2020-0036.
- [24] F. Khalid, M. Yaseen, G. Rahman, N. Mazhar, M. A. Nauman, and A. Mustapha, “Hybrid Feature Learning with TF-IDF and SBERT for Ambiguous Requirement Classification.,” *International Journal of Advanced Computer Science & Applications*, vol. 17, no. 4, p. 570, Apr. 2026.

- [25] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." [Online]. Available: <https://github.com/UKPLab/>
- [26] Subhadeep Chakraborty, "Phishing Email Detection," Kaggle. Accessed: Sep. 08, 2025. [Online]. Available: <https://www.kaggle.com/datasets/subhajournal/phishingemails>

BIOGRAPHIES OF AUTHORS

	Oluwasegun Abiodun Abioye is a System Analyst/Programmer with Directorate of Information and Communication Technology, Nigerian Defence Academy, Kaduna, Nigeria. He has core efficiency in the field of Artificial Intelligence, Machine Learning, and Deep Learning. He holds MSc in Information Technology and is currently a PhD student in Computer Science. He can be contacted at email: segunabioye@nda.edu.ng.
	Abraham Evwiekpaefe is an Associate Professor in the Department of Computer Science, Nigerian Defence Academy, Kaduna, Nigeria. He holds BSc, MSc, and PhD degrees in Computer Science from the University of Benin, Nigeria. He served as Head of Department and is a member of the Nigeria Computer Society (NCS) and CPN. His research focuses on software engineering, data science, and IT adoption. He can be contacted via email: aeevwiekpaefe@nda.edu.ng.
	Philip Odion is an Associate Professor in the Department of Computer Science, Nigerian Defence Academy, Kaduna. He obtained a B.Sc. from the University of Benin (1996), M.Sc. from Abubakar Tafawa Balewa University (2006), and Ph.D. from NDA (2013). He has held roles such as Examination Officer and Postgraduate Coordinator and served as Head of Department (2015–2022). His research focuses on software engineering and project management. He is a member of NCS and he can be contacted via email: poodion@nda.edu.ng.