Received: 20 January 2024 | Revised: 20 March 2024 | Accepted: 7 April 2024

DOI: 10.1002/spy2.402

RESEARCH ARTICLE

WILEY

An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach

Suhaima Jamal[1] | **Hayden Wimmer**[1] | **Iqbal H. Sarker**[2]

[1]Department of Information Technology, Georgia Southern University, Statesboro, Georgia, USA

[2]Centre for Securing Digital Futures, Edith Cowan University, Perth, Western Australia, Australia

Correspondence
Iqbal H. Sarker, Centre for Securing Digital Futures, Edith Cowan University, Perth, WA-6027, Australia.
Email: m.sarker@ecu.edu.au

Abstract

Phishing and spam have been a cybersecurity threat with the majority of breaches resulting from these types of social engineering attacks. Therefore, detection has been a long-standing challenge for both academic and industry researcher. New and innovative approaches are required to keep up with the growing sophistication of threat actors. One such illumination which has vast potential are large language models (LLM). LLM emerged and already demonstrated their potential to transform society and provide new and innovative approaches to solve well-established challenges. Phishing and spam have caused financial hardships and lost time and resources to email users all over the world and frequently serve as an entry point for ransomware threat actors. While detection approaches exist, especially heuristic-based approaches, LLMs offer the potential to venture into a new unexplored area for understanding and solving this challenge. LLMs have rapidly altered the landscape from business, consumers, and throughout academia and demonstrate transformational potential to profoundly impact the society. Based on this, applying these new and innovative approaches to email detection is a rational next step in academic research. In this work, we present IPSDM, an improved phishing spam detection model based on fine-tuning the BERT family of models to specifically detect phishing and spam emails. We demonstrate our fine-tuned version, IPSDM, is able to better classify emails in both unbalanced and balanced datasets. Moreover, IPSDM consistently outperforms the baseline models in terms of classification accuracy, precision, recall, and F1-score, while concurrently mitigating overfitting concerns.

KEYWORDS

artificial intelligence, cyber security, DistilBERT, fine tuning, large language model, model optimization, phishing, RoBERTA, spam

1 | INTRODUCTION

Phishing and spam emails are pervasive and cost business resources, both time and money. This kind of fraudulent endeavor attempts to deceive individuals into revealing sensitive and confidential information, such as financial details or login credentials. More concerning are the cyber-security implications as many breaches and attacks originate via social engineering. Threat actors use social engineering to gain an entry point via human manipulation and as a platform to launch their attacks. The majority of ransomeware attacks have been linked to entry from social engineering. While artificial intelligence (AI) approaches have attempted to assuage these issues,[1] heuristic-based systems continue to dominate.[2] Radical new approaches are necessary and have emerged due to advances in technology and increased research investment by both the public and private sector. The recent advancements in AI-based solutions have led to the development of innovative and unconventional strategies to combat spam and phishing tactics such as demonstrated by Anand et al.[3]

Transformer-based models possess a revolutionary impact on the development of spam and phishing classification models while processing, understanding, and interpreting the text data inputs. For email-based datasets, such models are continuously evolving providing additional opportunity to address the detection challenge. Furthermore, attention-based mechanisms in transformer allows model interpretability which improves the understanding of classification decisions.[4,5] Large language models made famous by Open AI's ChatGPT, have emerged triumphant in solving new problems while being adapted to well-established challenges such as phishing and spam. Open AI's ChatGPT runs on its GPT engine and has been able to make large strides in consumer and business adoption.[6] The most famous competing LLMs are available from a plethora of vendors such as Google, Meta, and MIT while the emergence of competing LLMs such as Llama and Bert have been open sourced thereby fueling research and development from large institutions all the way down to the consumer. While these models are available for download, the ability to run pre-trained models such as BERT is still in nascent stages. As more consumers have access to local GPU technology as well as organizations like Google with Collaboratory and Hugging Face with its transformer's library and model hosting the options for implementation of applications have improved.

LLMs are general in nature and pre-trained by the creators and published for commercial and non-commercial licenses. There are a multitude of inputs to train a LLM such as web scraping, document corpus, and even text sources such as email and transcribed books, discussions, or speeches. While LLMs perform well on general tasks, they can be fine-tuned to improve their performance on more specific tasks. One such example is FinBERT[7] where BERT (bidirectional encoder representations from transformers) was trained on financial specific documents and is able to better respond to use prompts on financial applications. Other such advances are in progress for medical data to aid in both physician decision making and end-user queries. BERT employs a self-attention mechanism which enables the model to capture both contextual information and dependencies among words in any text sequence. Through the self-attention method, the weights of relevant important words are calculated. Attention scores are measured for all words or input tokens and passed through SoftMax function. A rich contextual embedding can be generated by BERT based models which allow to excel in several natural language understanding tasks.

Within the family of BERT-based models, DistilBERT and RoBERTA are two promising variants and have been used for tasks such as fake news detection[8] or to make predictions via Twitter data.[9] Both models are built based on a transformer architecture while excelling in NLP processing tasks. DistilBERT is designed for reducing the number of parameters making it faster and smaller version of BERT whereas Roberta is considered a more optimized and robust version. In this work, we aimed to leverage the powerful natural language processing capabilities of LLMs to accurately classify and distinguish between these different types of emails. We present an improved phishing and spam detection model (IPSDM), a custom trained and fine-tuned version of DistilBERT and RoBERTA. The issue of spam, ham (legitimate), and phishing email detection have been addressed here by developing this fine-tuned model specifically on phishing, spam, and ham data from multiple sources. We demonstrate that our fine-tuned IPSDM outperforms basic BERT and RoBERTA on both imbalanced and balanced datasets of phishing, spam, and ham.

The contribution of our work is to demonstrate a new application of LLM technology to a common problem plaguing business and society, phishing, and spam. We illustrate how an LLM can be used to approach this and, furthermore, we illustrate how fine-tuning an existing model can improve performance. This is an important step towards the application of LLMs on a large range of challenges. As LLM technology improves, our methods can be applied to improve the performance of more advanced LLMs as they are released. The rest of the paper is outlined as follows; Section 2 describes the related current-state-of arts research works. Section 3 provides a detailed explanation of the proposed model's framework and methodology. In Section 4, the experimental outcomes and results are broadened. Furthermore, Section 5 encompasses an elaborated discussion of the results. Finally, Section 6 holds the concluding remarks and future prospects of this work.

2 | LITERATURE SURVEY

2.1 | Machine learning and deep learning based methods

Machine learning and deep learning along with artificial intelligence, are progressively making way into a wide large of industries, such as healthcare, cyber security, education, and so on.[10] Numerous machine learning and deep learning-based spam email detection and classification applications have been carried out over the past few decades by many researchers. In such studies,[11–18] authors proposed, reviewed, and evaluated spam filtering and phishing detection models where the classification models are based on traditional machine learning algorithms, that is, Naïve Bayes, XGboost method, Random Forest, SMV mostly. Govil et al.[11] have created a dictionary, named "stopwards" for removing the helping verbs from email. Then, the algorithm is executed for checking the possibility of being spam or not. A machine learning classifier, Naïve Bayes has been applied for the identification purpose where non-spam emails were classified as spam, 1 and non-spam, 0.[11]

Similarly, Chen et al.[12] evaluated machine learning algorithms for detecting spam tweets. A large dataset containing around 600 million public tweets have been collected first. Later, Trend Micro's Web Reputation System was applied for labeling the spam emails. Experiments on different data sizes revealed that TP rate is increased from 78% to 85% following KNN and 70% to 75% following Random Forest classifier. Another potential finding is the classifier could detect continuously sampled spam tweets better than randomly selected tweets.[12] In the similar context of Twitter spam detection, a WordVector is introduced by a training based model with a classification accuracy of around 80%. Average 30% higher F-measures have been achieved in this work compared to other existing models.[19] Another potential technique, long-short-term memory is combined with classical machine learning algorithms to detect fake news where high order statistical descriptor is applied for sorting the news samples.[20] Moreover, Guzella[15] reviewed the textual and image-based spam email filtering approaches focusing on designing new filters. Most common method selecting the feature is information gain and this way of collecting features might increase accuracy. In terms of datasets, SpamAssassian and LingSpam are considered as the most popular ones, whereas TREC corpora can produce more realistic online setting. Additionally, Chetty[21] proposed a deep learning-based model combining word embedding and neural network aiming to detect spams from various text documents. Naïve Bayes model is considered as the baseline model for comparing with the deep learning model. Datasets were collected from UCI machine learning repository for developing the models. For SMS dataset, the highest performance (accuracy 98.7%) is achieved from the combined model of word embedding and neural network. Apart from the supervised learning approaches, there are numerous works on unsupervised modeling as well.[22–27] Utilizing modified density-based spatial clustering of applications with noise (M-DBSCAN), 97.848% accuracy obtained by Manaa.[23] An online unsupervised spam detection scheme, SpamCampaignAssassin (SCA) could detect around 92.4% spam for DEPT trace email dataset.

2.2 | Transformer model-based approaches

The research works and literature landscape on transformer-based methods are relatively limited. Transformer-based models are currently popular for addressing NLP-related challenges, complex tasks such as sentiment analysis, text generation and human emotion recognition.[28,29] Numerous researchers dedicated their efforts for developing models for the classification and identification of emotional states[30] and analysing sentiments.[31] The domain of fine-tuned transformers or attention mechanism techniques for identifying spam emails is still an emerging new field. Transformer based models are related to this specific area, Yaseen et al.[32] introduced an effective word embedding technique for spam classification. Pre-trained transformer, BERT is fine tuned to detect the spam emails from non-spam emails. Deep neural network with BiLSTM is considered as a baseline model to compare the model. Two open-source datasets from UCI machine learning repository and Kaggle have been employed to train and test the model. The proposed model achieved 98.67% classification accuracy. Similarly, In another study, a modified spam detector transformer was developed and evaluated using the publicly available datasets,[33] Spam Collection v.1 and UtkMI's Twitter Spam Detection Competition dataset. This model was able to obtain 98.92% accuracy with a recall and F1 scores rate respectively, 0.9451 and 0.9613.

Furthermore, numerous studies focused on BERT models implying the significance of self-attention mechanism.[34–36] Guo et al.[34] utilized two public datasets, Enron and simple spam email classifier dataset from Kaggle for classifying ham or spam emails using pre-trained BERT model. Similarly, an universal spam detection model (USDM) was developed and tested using four publicly available datasets which are Ling-spam dataset, spam text dataset from Kaggle, Enron dataset

and spam assassin dataset. This model has gained overall accuracy of 97% with 0.96 F1 score.[34] Moreover, for detecting phishing URL, researchers worked on fine tuning BERT based models.[37,38] Wang et al.[37] have scrapped 2.19 million pieces of URL data from PhishTank while pre-training PhishBERT model. This model exhibited 92% accuracy in detecting phishing URLs. Similarly, Maneriker[38] fine-tuned BERT and RoBERTa models and proposed a URLTran transformer. Microsoft Edge and Internet Explorer browsing telemetry data was employed for training, testing, and validating purpose. Down sampling method is applied for balancing the datasets where the final training dataset had 77 870 URLs. The final models had a true positive rate (TPR), 86.80% compared to the baseline models URL-Net[39] and Texception.[40]

In the current state of machine learning, deep learning, and transformer-based models, a notable gap persists in the literature where the fine-tuning of BERT families has not been extensively explored. Despite the remarkable achievements of BERT-based architectures in several natural language processing tasks, there remains a lack of comprehensive research on fine-tuning these models to address specific domain challenges. In our work, we aim to close this gap by implementing fine-tuning techniques on BERT variants such as DistilBERT and RoBERTA. By customizing and fine-tuning these models on datasets relevant to specific applications, we seek to enhance their adaptability and effectiveness in addressing sophisticated real-world problems in complex domains, thereby contributing to the advancement of machine learning and deep learning methodologies.

3 | METHODOLOGY

In this paper, transformer-based self-attention mechanism models are explored with an aim to improving the pre-trained baseline BERT models. Our collected and prepared dataset is used for developing and comparing models in two different settings. (1) DistilBERT and RoBERTA were pretrained using both imbalanced and balanced phishing-ham-spam dataset, and (2) the base models' training process has been improved through applying optimization and fine-tuning mechanism. We named our proposed model as improved phishing spam detection model (IPSDM). This model's classification performance is compared with the baseline models (DistilBERT and RoBERTA). At the end of the experiment, IPSDM exhibited substantial improvement in performance both for balanced and imbalanced scenarios compared to baseline models while detecting phishing and spam emails and texts. The top-level system flow diagram of this research is presented in Figure 1 and the overall methodology is illustrated in Figure 2. Later, the breakdown of detailed flow diagram of model optimization and fine-tuning are illustrated in Figures 6 and 9 of sect. 2.4.

3.1 | Data collection and preparation

The data for training, testing, and validating this experiment is developed by concatenating two opensource data sources.[41,42] One dataset has ham and spam emails which is merged with another phishing email dataset. These three categories of data has been explained in the following section:

Category 1: Spam Emails: These are unsolicited messages sent in bulk to a large number of recipients, typically for advertising or fraudulent purposes. These emails often contain irrelevant or misleading content and may include links to malicious websites or scams. Spam emails can clog up email inboxes, waste time and resources, and pose security risks to recipients by exposing them to potential malware or phishing attacks.

Category 2: Ham Emails: Ham emails refer to legitimate, non-spam messages that are relevant and solicited by the recipient. These emails can include personal or professional correspondence, newsletters, notifications, and other legitimate communications. While ham emails are not inherently harmful, the presence of spam and phishing emails can make it difficult for recipients to distinguish between legitimate and malicious messages, leading to potential security breaches or loss of trust in email communication systems.

Category 3: Phishing Emails: Phishing emails are fraudulent messages designed to deceive recipients into revealing sensitive information such as usernames, passwords, credit card numbers, or personal details. These emails often mimic legitimate communications from trusted sources, such as banks, social media platforms, or government agencies, in an attempt to trick recipients into clicking on malicious links, downloading malware, or providing confidential information. Phishing emails can lead to identity theft, financial fraud, data breaches, and other serious consequences for individuals and organizations.

The concatenated dataset has 747 spams, 189 phishing and 4825 ham samples which is highly imbalanced. Such imbalanced datasets can adversely affect the performance of machine learning models, especially in terms of accuracy, precision, and recall. By balancing the class distribution through sampling, models can better learn from and accurately classify instances from all classes, leading to improved performance metrics. Moreover, the class imbalance situation

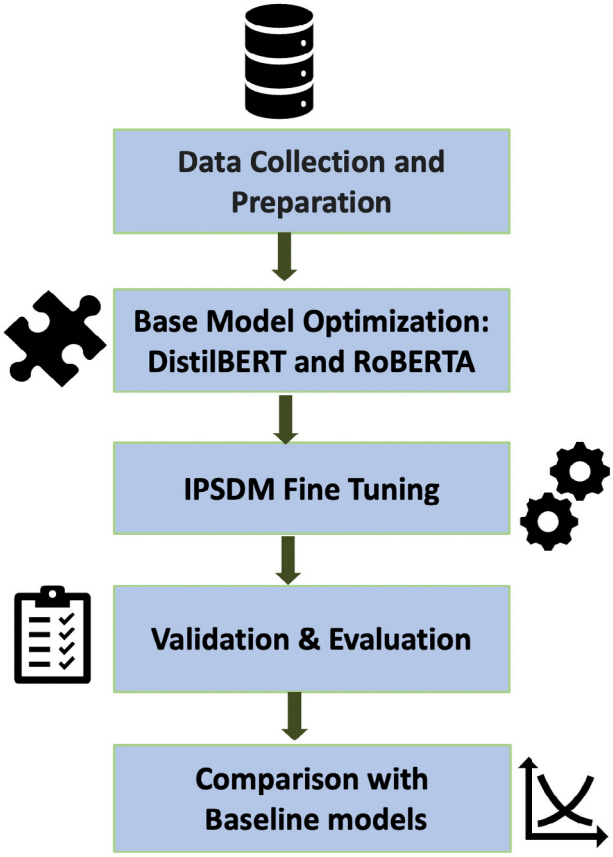

FIGURE 1 Overall system flow diagram.

can lead to biased models that perform poorly on minority classes. Sampling techniques help address this issue by either oversampling the minority class, undersampling the majority class, or generating synthetic samples to balance the class distribution.

Here, the initial dataset has been further resampled following adaptive synthetic sampling (ADASYN) technique where minor classes (ham and spam) are oversampled by generating synthetic samples with a focus on difficult-to-learn instances. This process reduces the bias towards the majority class making the overall predictive model more accurate and efficient. This versatile technique of sampling assists in mitigating the risk of overfitting as well. Figure 3 presents the feature distribution before and after sampling. ADASYN has been preferred for its adaptability, targeted synthetic sample generation, and ability to improve model performance on imbalanced datasets while mitigating the risk of overfitting. Figure 4 shows a snapshot of final dataset.

3.2 | Data splitting

The overall dataset is split into 80% (training set) and 20% (testing set). Later, from the 80% set, 60% kept for training and 20% for validation. This 20% validation set is used after the completion of each training epoch which aids in identifying the optimal model performance. It is an integral part of the development process that ensures the model's effectiveness on unseen data identification and prediction.

3.3 | Model selection

3.3.1 | DistilBERT

DistilBERT is a derivation of bidirectional encoder representations from transformers (BERT) which is a transformer-based model pre-trained for developing natural language processing tasks. The idea here is to compress

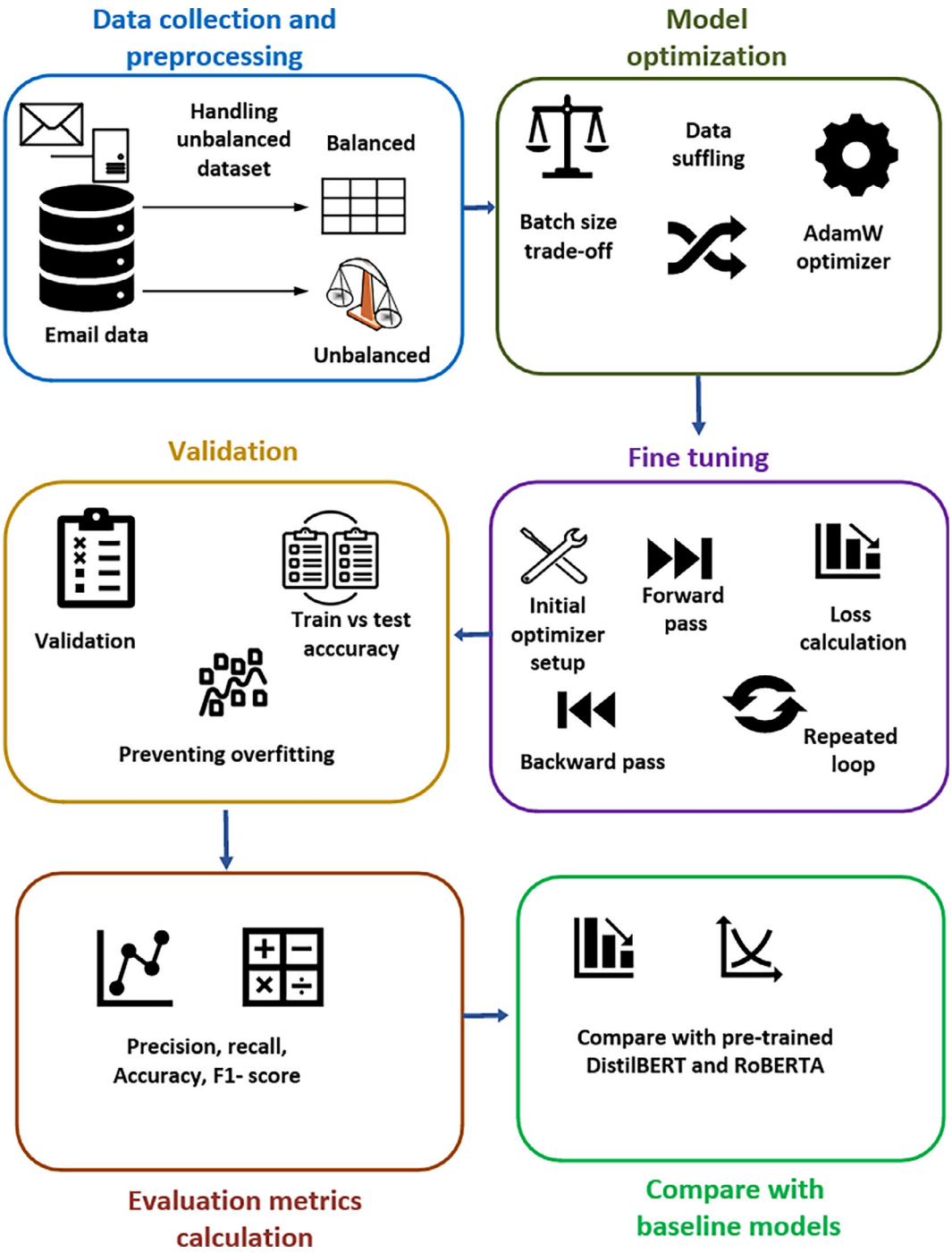

FIGURE 2 Method diagram. Illustrates data collection and preprocessing, model optimization, fine tuning, model validation, baseline model comparison and evaluation metrics.

the original model for making it more computationally efficient and faster.[43] The models can be further finetuned for any specific downstream tasks on any customized dataset. DitilBERT model achieves the compression by mimicking a teacher-student model where the customized model is trained. The input tokens are the raw text inputs that need to be preprocessed. The tokenizer uses a vocabulary to tokenize the input words into sub-words. Later, the tokenized inputs are mapped to numerical embedding. The relationships between the words are captured through the attention layer. This attention mechanism works by calculating the attention score between tokens inside a sequence allowing the model to focus more on the significant relevant words than the irrelevant ones. The pooling section indicates the entire input

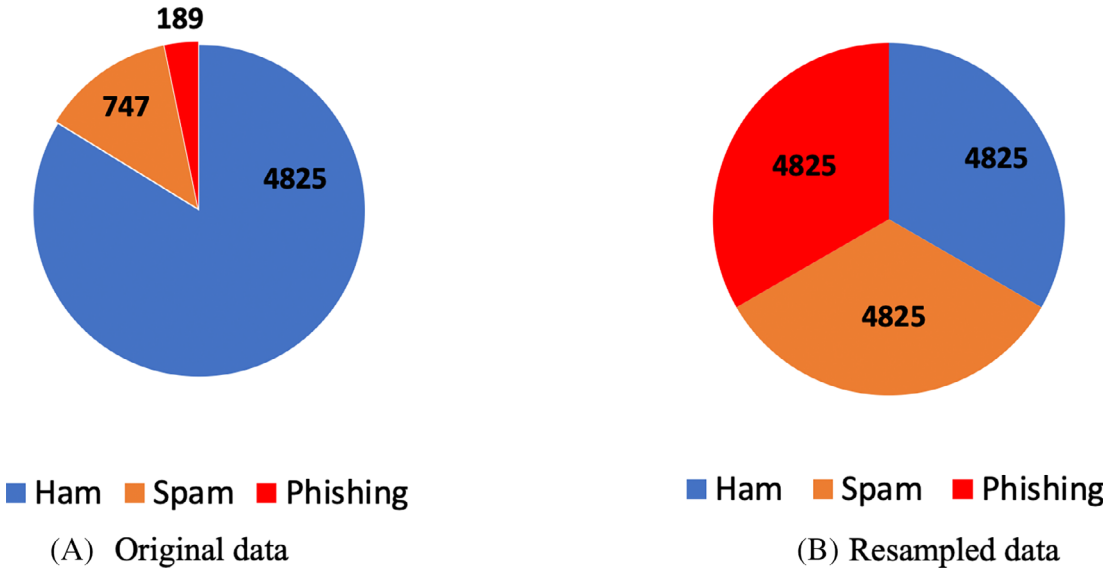

FIGURE 3 Feature distribution. (A) Original data distribution and (B) Resampled data distribution.

Email	Category
Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...	ham
Ok lar... Joking wif u oni...	ham
U dun say so early hor... U c already then say...	ham
Shop till u Drop, IS IT YOU, either 10K, 5K, v•¬£500 Cash or v •¬£100 Travel voucher, Call now, 09064011000. NTT PO Box CR01327BT fixedline Cost 150ppm mobile vary	spam
Nah I dont think he goes to usf, he lives around here though	ham
refund confirmation	phishing
Even my brother is not like to speak with me. They treat me like aids patent.	ham

FIGURE 4 A snapshot of dataset overview.

sequence having a fixed representation. The classifier head can be modified for any specific task and the final prediction layer predicts the corresponding model output. For our case, this is detecting spam/ham/phishing emails.

3.3.2 | RoBERTA

A robustly optimized BERT pretraining approach (RoBERTA) is an extended version of the transformer-based model, BERT where model can operate on large batch size and train longer sequence. The pretraining process follows improved bidirectional context-oriented mechanism while learning the masked-out tokens for longer sequences.[44] The architecture is similar as DistilBERT having transformer encoder layers with multi-head attention mechanisms. However, model has a byte-level tokenizer which is different than BERT. The dynamic masking works at different epochs and uses BPE as a subunit, not as characters. RoBERTA receives tokens as inputs and a tokenizer preprocess these. It passes through

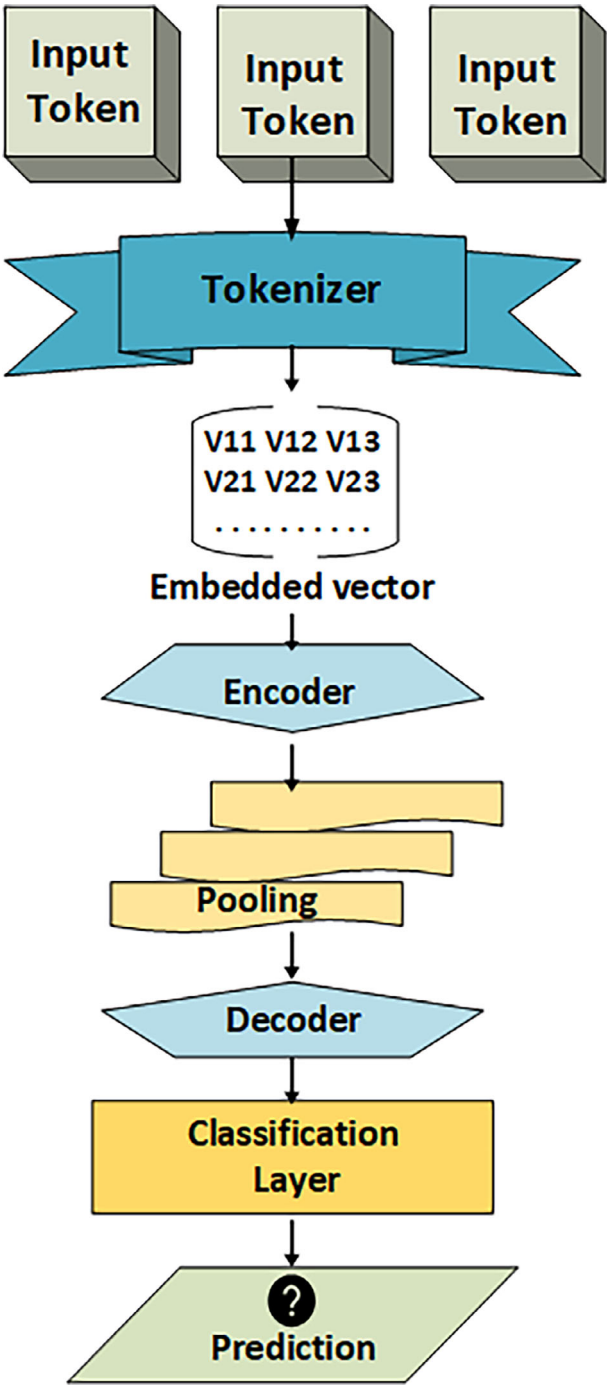

FIGURE 5 Basic architecture of DistilBERT and RoBERTA.

encoding, pooling, decoding and attention mechanism. The basic architecture of DistilBERT and RoBERTA model is similar which is illustrated in Figure 5.

3.4 | Improving the training process

Employing the phishing-ham-spam dataset, base models of DistilBERT and RoBERTA have been measured first. We aim to improve the model performance and efficiency through optimization, that is, learning rate scheduling, adjusting batch size, sequence length and loss function, hyper parameter tuning, early stopping and fine tuning. Necessary measures

have been taken to handle overfitting issue. At the end of the process, it has been demonstrated that the final achieved accuracy is not affected by overfitting. This proposed methodology is also employed on imbalanced dataset which was collected initially. A noteworthy improvement is observed while developing models with imbalanced dataset as well.

3.4.1 | Model optimization

As mentioned about data preparation stage in Section 2.1, the preprocessed final phishing dataset is tokenized using Hugging Face Transformers tokenizer.[45] A sub-word-based approach is utilized by this tokenizer while breaking down the text into small unit. This allows the model to acknowledge the meaning and context of the words. The pre-trained DistilBERT and RoBERTA models are initialized with their respective pre-trained weight obtained from pre-training process. The batch size is set 32 for training data and 64 for the validation data while trading off between memory consumption and training speed. This choice balances memory consumption and training speed, ensuring efficient utilization of computational resources while maintaining model performance. Training data is shuffled in each epoch ensuring the model's visibility to the different unseen data. This will help memorizing the training dataset and mitigating overfitting issues.

Moreover, in the optimization stage of our model training pipeline, another significant technique has been employed to enhance the learning process and prevent overfitting which is choosing an optimization function. In this work, AdamW (Adam Weight Decay), an efficient optimization algorithm is used for updating the weights of pre-trained models. This algorithm computes the adaptive learning rate for each parameter by combining exponential moving gradient averages and root mean square gradients.[46,47] It adapts the learning rate for each parameter in the network. This means it can use different learning rates for different parts of the model, which can help speed up training and improve performance.

Furthermore, for L2 regularization, a weight decay mechanism has been incorporated. This regularization technique adds penalty to the loss function which is proportional to magnitude squared weights. This promotes the model to utilize small weights and mitigate overfitting risk by reducing the complexity of the acquired parameters. The model's parameter, Z is initialized with exponential decay rate, Beta1, Beta2 and epsilon with a very small value preventing division by zero. Initially, the first moment, $m_0 = 0$ and second moment, $v_0 = 0$.

In each iteration, the gradient loss is calculated as below,

$$Gradient\ loss, g = \delta_z L(z)$$

Then, the first moment is updated,

$$m_i = \beta_1 m_{i-1} + (1 - \beta_1) g$$

The updated second moment,

$$v_i = \beta_2 v_{i-1} + (1 - \beta_2) g^2$$

Later, first and second moment bias get corrected,

$$\hat{m}_i = \frac{m_i}{1 - \beta^i}$$
$$\hat{v}_i = \frac{v_i}{1 - \beta^i}$$

Finally, the parameters are updated using AdamW updating rule,

$$Z_i = Z_{i-1} - \frac{\text{learning rate}}{\sqrt{\hat{v}_i} + \epsilon}(\hat{m}_i + \text{weight decay} * Z_{i-1})$$

This weight decay regularization process assists in controlling the growth of parameters values during the training, mitigating the risk of overfitting.

In the context of loss function, as this is a multiclass classification task, cross-entropy loss is used which combines both SoftMax activation and negative log likelihood into a single loss term. The difference between ground truth label and

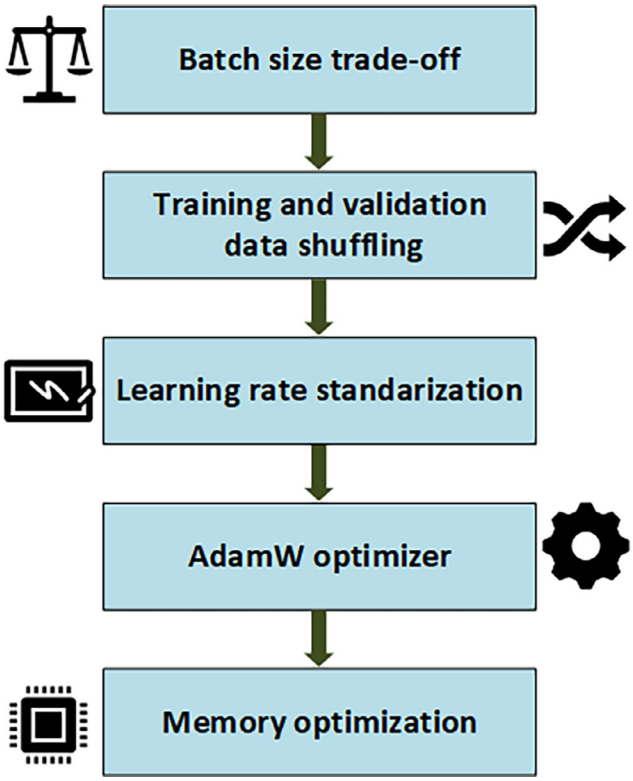

FIGURE 6 Model optimization.

probability are measured here aiming to minimize the loss during the training process. PyTorch provides cross-entropy loss implementation that handles SoftMax computation and logarithmic computation. For a single training epoch, the loss can be defined as follow,

$$\text{Loss}_i = \sum_{k=1}^{n} Z_{i,K} \cdot \log(P_{i,k})$$

Here, Z_i, k is the ground-truth label and P_i, k is predicted probability made by the model. The cross-entropy loss for the overall training is the average of individual loss,

$$\text{Loss}_t = 1/n \sum_{k=1}^{n} \text{Loss}_i$$

The models output logits for each of the class which is passed through a SoftMax activation function for converting them into class probability. The predicted probability $p_(i, k)$ is computed as below, where Z_i, k is the produced logit value.

$$p_(i, k) = e^{(Z_i,k)} \Big/ \left(\sum_{m=1}^{k} e^{(Z_i,m)} \right)$$

The optimization process diagram is presented in Figure 6.

3.4.2 | Learning rate

An ideal learning rate for model optimization and fine tuning depends on several factors, including model architecture, optimization algorithms and the specific task domain. It is a crucial parameter which controls step size during the optimization process. Having a high learning rate might lead the model in unstable mode resulting poor performance for

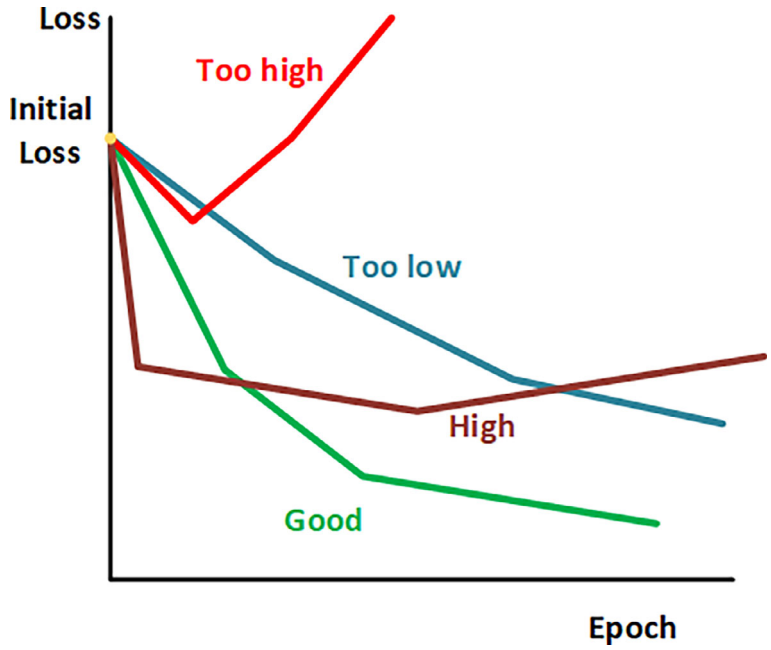

FIGURE 7 Learning rate graph.

unseen data. Again, lower learning rate can slow down the convergence process. The training process might require more epochs for achieving a good result resulting in higher computational cost. An ideal learning rate graph is presented in Figure 7.

In our experiment, a commonly accepted learning rate, 2e−5 (0.00002) is set which is standard for BERT based models, that is, RoBERTA and DistilBERT. Later, we plot validation versus test accuracy comparison to demonstrate the effectiveness of the selected learning rate.

3.4.3 | Fine tuning

Fine tuning process involves adapting a pre-trained model to get trained on some specific tasks and datasets. This enhances the ability of any pre-trained NLP model to perform any domain-specific task, that is, email classification for our case. The models are finetuned using training dataset, the 80% of the data which was separated beforehand. Training data is passed in each epoch as batches through the models, calculating the gradient using backpropagation method. To facilitate an efficient batching, DataLoader is used during the training. Code snippet is attached for RoBERTA model in Figure 8. A similar approach is employed for DistilBERT as well. Train_loader is configured for creating mini batches of size, 32, which promotes parallel processing and optimize the memory use. Val_loader is designed to batch of 64 samples for validation ensuring most efficient evaluation method without shuffling the data. RobertaForSequenceClassification class is used to adapt the pre-trained model for specifically email classification task. This class enables an additional classification layer for the target label prediction. The overall fine-tuning process flow diagram is illustrated in Figure 9.

4 | RESULTS

The proposed IPSDM model is validated and tested using both unbalanced and balanced datasets. The IPSDM result metrics are compared to baseline modes, that is, pretrained DistilBERT and RoBERTA models. To assess the performance more comprehensively, various key metrics including overall accuracy, precision, recall and F1-score are calculated. These provide crucial insights of the model performance.

```
train_dataset = EmailDataset(train_df, tokenizer)
val_dataset = EmailDataset(val_df, tokenizer)

train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=64, shuffle=False)

model = RobertaForSequenceClassification.from_pretrained('roberta-base', num_labels=3)

optimizer = AdamW(model.parameters(), lr=2e-5)
num_epochs = 3
```

FIGURE 8 Code snippet fine tuning.

4.1 | Evaluation metrics

4.1.1 | Precision

The ratio of true positive predictions and the total number of positive predictions is called precision. It indicated how many predicted positive samples made by the model are actually positive. The formula for precision is as follow,

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

Here, TP = True Positives; the number of instances that are correctly predicted as positive.
FP = False Positives; the number of instances that are incorrectly predicted as positive.
FN = False Negatives; are the number of instances that are incorrectly predicted as negative.
High precision value suggests that the model's predicted positive instance rate is truly positive and correct. Whereas low precision indicates about making many false positive errors by the model.

4.1.2 | Recall

Recall is the measurement of model's sensitivity for understanding true positive rate. It presents the ratio of true positive instances which is predicted as positive by the model. The formula for calculating recall is stated below,

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

Higher recall conveys that the model can successfully predict the positive samples as positive making a little false negative error. However, low recall suggests that a higher number of actual positive samples are getting missed while the model predicts the false negatives.

4.1.3 | F1- score

This is a statistical metric which is the average of precision and recall which balances these values. This provides a comprehensive view on how a model deals with imbalanced datasets by trading off between precision and recall. If either of precision or recall is low, then the overall F1 score will be lower. This metric validates the model's ability for predicting the positive rates and how many instances are actually positive. The formula is as follow,

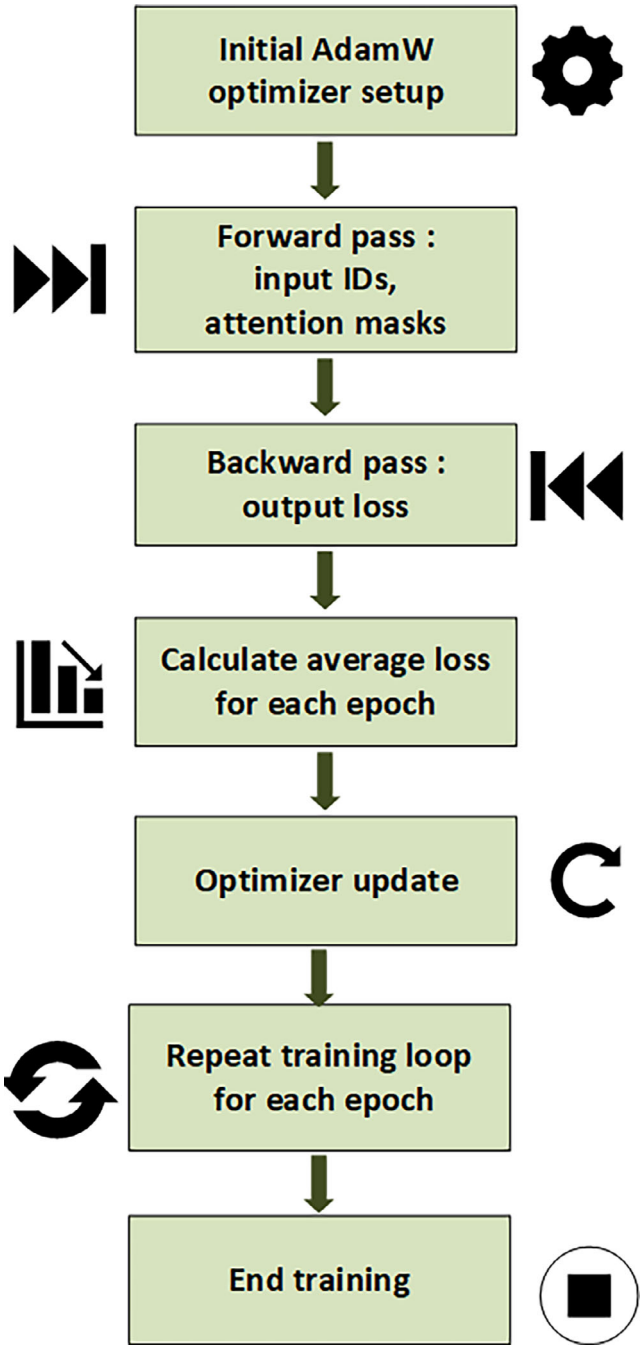

FIGURE 9 Fine tuning diagram.

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

4.1.4 | Accuracy

Accuracy is the ratio of accurately predicted samples to the total number of samples made by the model. It is calculated by following formula,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

However, some notable points need to consider carefully when interpreting the model accuracy because it suffers from some limitations while dealing with imbalanced data. Feature distribution across all the classes is required to be observed meticulously. Otherwise, it might raise a biased classification result. Hence, in this study, all of the essential metrics are calculated and combined together to interpret our proposed IPSDM results after running a vigilant examination.

4.2 | Imbalanced dataset results

This experiment was initially carried on imbalanced datasets to assess IPSDM model's performance on imbalanced dataset. The initial collected dataset was highly imbalanced having a majority class, ham (Figure 3A). Comparison tables (Tables 1 and 2) and graphs (Figures 10 and 11) between baseline model's performance and IPSDM model's performance clearly reflect that IPSDM has a better performance in the imbalanced setting. Although due to highly uneven distribution of data samples across the three classes, the model performance is a biased towards 'ham' class, still it has achieved comparatively higher values than the baseline models.

The precision values are higher than recall for both cases (DistilBERT and RoBERTA). In the context of highly imbalanced characteristics of this dataset, the model can identify the majority class, 'ham', however, for the model struggles for classifying the minor classes, 'spam' and 'phishing'.

There is a noticeable disparity between Precision and Recall values for both models. Recall values are considerably lower compared to the precision. Validation and test recall for base DistilBERT model are 0.30 and 0.31 (shown in Table 1). For base RoBERTA model, the recall values are 0.47 and 0.49 (shown in Table 2) which suggest that the models are facing challenges for identifying the minor classes, 'spam' and 'phishing' due to the imbalanced nature. However, it is noteworthy that the performance of IPSDM for both DistilBERT and RoBERTA is notably higher even the dataset is imbalanced.

TABLE 1 Baseline DistilBERT versus IPSDM performance (imbalanced dataset).

Evaluation metrics	Base DistilBERT	IPSDM
Validation accuracy	30.28%	51.32%
Test accuracy	31.60%	53.67%
Validation precision	0.841	0.972
Test precision	0.852	0.981
Validation recall	0.302	0.561
Test recall	0.311	0.582
Validation F1-score	0.432	0.613
Test F1-score	0.451	0.621

TABLE 2 Baseline RoBERTA versus IPSDM performance (imbalanced dataset).

Evaluation metrics	Base RoBERTA	IPSDM
Validation accuracy	43.78%	66.97%
Test accuracy	45.24%	67.86%
Validation precision	0.892	0.981
Test precision	0.912	0.981
Validation recall	0.465	0.693
Test recall	0.492	0.731
Validation F1-score	0.567	0.874
Test F1-score	0.581	0.893

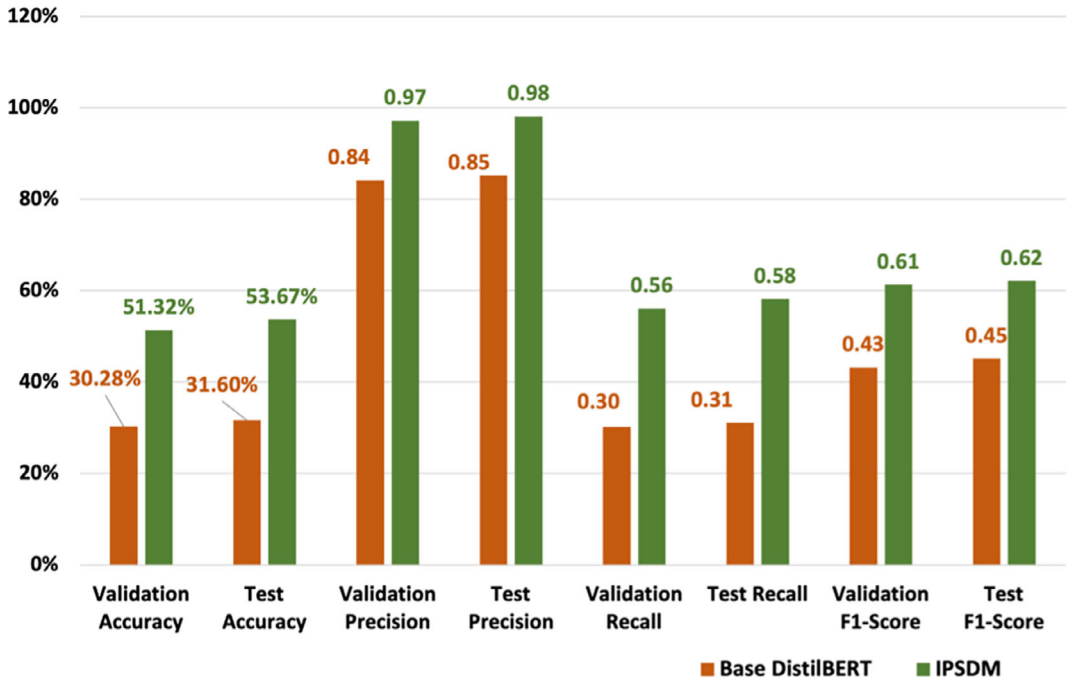

FIGURE 10 Comparison graph of baseline DistilBERT versus IPSDM performance (imbalanced dataset).

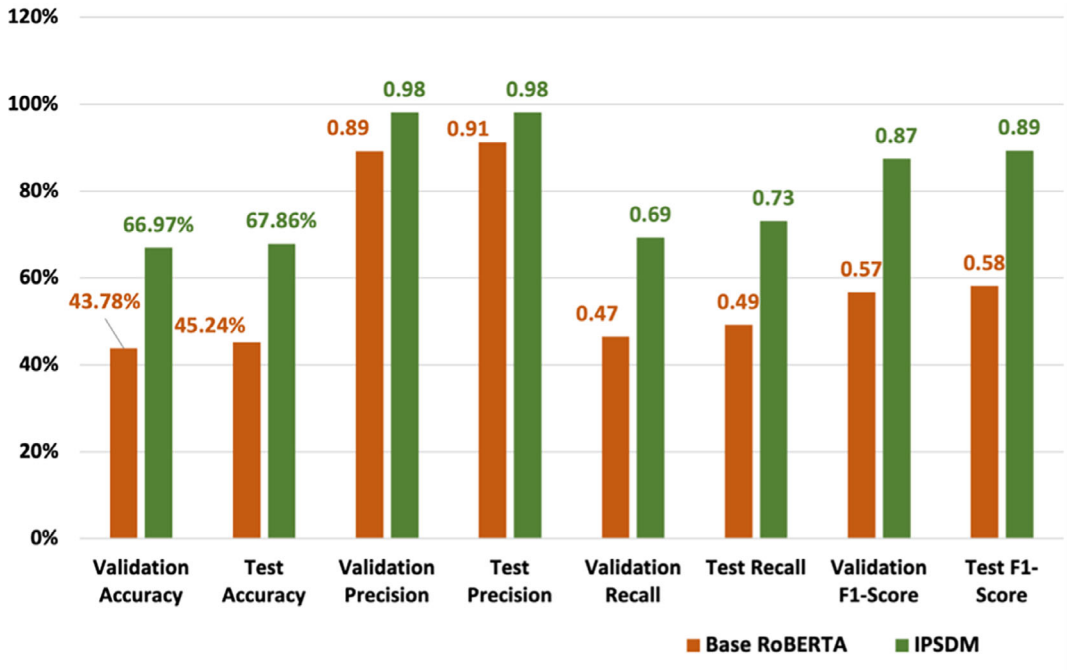

FIGURE 11 Comparison graph of baseline RoBERTA versus IPSDM performance (imbalanced dataset).

TABLE 3 Baseline DistilBERT versus IPSDM performance (balanced dataset).

Evaluation metrics	Base RoBERTA	IPSDM
Validation accuracy	82.63%	97.50%
Test accuracy	88.95%	97.10%
Validation precision	0.8543	0.9755
Test precision	0.9025	0.9716
Validation recall	0.6971	0.9750
Test recall	0.7532	0.9710
Validation F1-score	0.8867	0.9749
Test F1-score	0.8943	0.9710

TABLE 4 Baseline ROBERTA versus IPSDM performance (balanced dataset).

Evaluation metrics	Base ROBERTA	IPSDM
Validation accuracy	87.10%	98.99%
Test accuracy	93.29%	99.00%
Validation precision	0.921	0.982
Test precision	0.853	0.991
Validation recall	0.903	0.989
Test recall	0.923	0.991
Validation F1-score	0.911	0.982
Test F1-score	0.931	0.985

4.3 | Balanced dataset results

The collected email datasets have been resampled and balanced. After preparing this balanced dataset, baseline DistilBERT and RoBERTA models were trained and validated. Again, using the similar dataset, we worked on model optimization and fine tuning. The evaluation metrics of our proposed model, IPSDM and the baseline models are tabulated in Tables 3 and 4. Accuracy, precision, recall for both validation and test cases are presented here. Also, the values are illustrated in comparison graphs (Figures 12 and 13).

The evaluation metrics exhibit an increase in validation accuracy- approximately 14.87% and 11.89%; test accuracy approximately 8.15% and 5.71% respectively for base DistilBERT and RoBERTA models versus IPSDM. A consistent rise in F1scores suggests that the IPSDM has elevated performance across both cases. This score is the harmonic mean of recall and precision which is a crucial metric for assessing the balance between the crucial aspects of classification performance.

4.4 | Avoiding overfitting

A common issue in statistical modellings and machine learning is overfitting which occurs when a model is performs too well on the training dataset, however, too poorly on the new or unseen data, that is, testing dataset. Overfitting can be effectively managed in balanced situations while a model has consistent performance on validation and test datasets. A close alignment between test and validation accuracy suggests that the classification models yield good results on unseen, new data. In the balanced scenario, test and validation accuracy values indicate minimal disparity, that is, 97.10% versus 97.50% and 99. 00% versus 98.99%. Based on the Table 5 and Figure 14 data, there is not a large gap between validation and test accuracy. When training or validation accuracy is notably higher than test accuracy, there is a high chance of overfitting. Moreover, the precision, recall and F1 measures from Tables 1 to 4 also suggest a harmonic distribution of these metrics which is also a positive indication. The comparison graph for both validation and test accuracy lines are almost overlapping with each other indicating that the model is performing well on the unseen data.

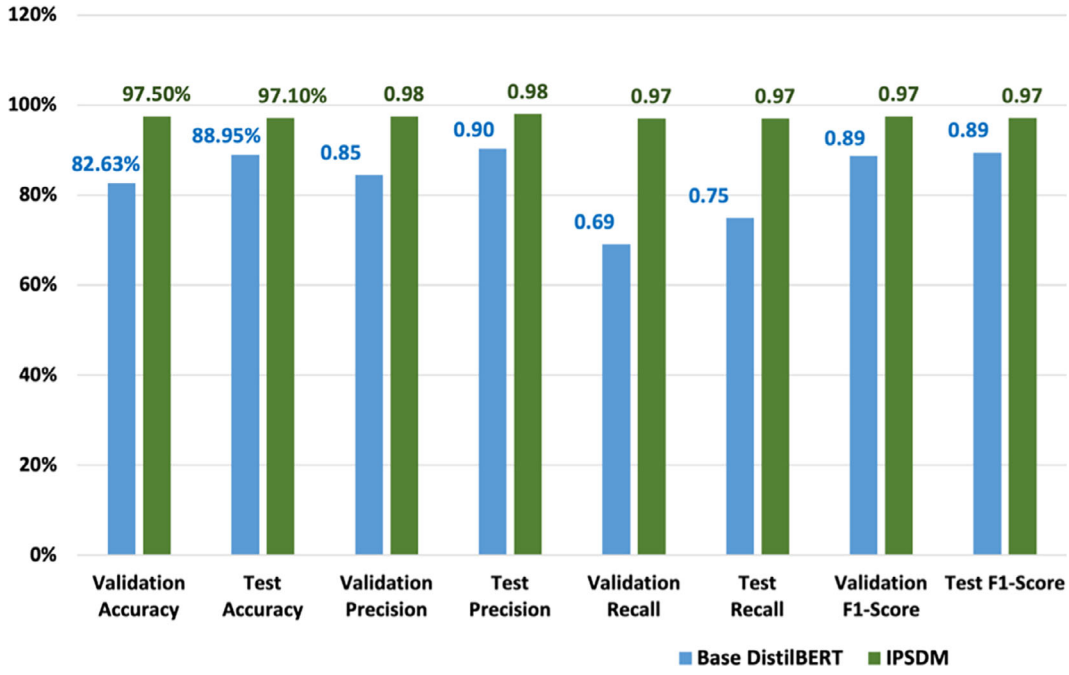

FIGURE 12 Comparison graph of baseline DistilBERT versus IPSDM performance (balanced dataset).

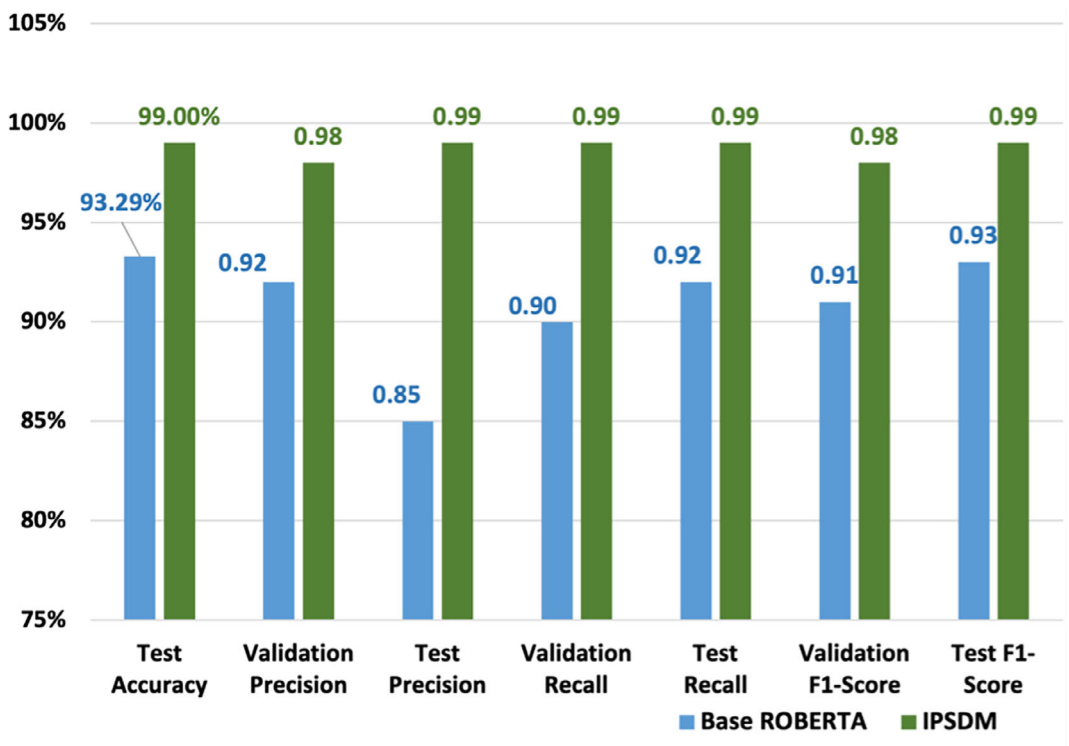

FIGURE 13 Comparison graph of baseline RoBERTA versus IPSDM performance (balanced dataset).

TABLE 5 Validation versus test accuracy.

Model name	Validation accuracy	Test accuracy
Balanced_ IPSDM/ DistilBERT	97.50%	97.10%
Balanced_ IPSDM/ RoBERTA	98.99%	99.00%
Imbalanced_ IPSDM/ DistilBERT	51.32%	53.67%
Imbalanced_ IPSDM/ RoBERTA	66.97%	67.86%

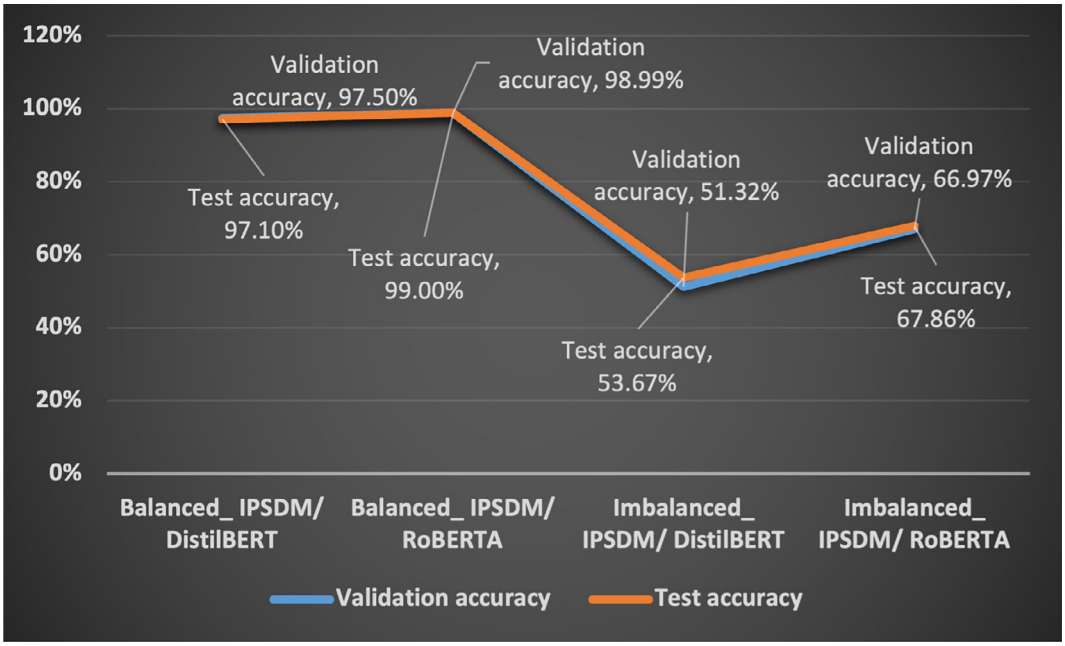

FIGURE 14 Validation versus test accuracy graph.

5 | DISCUSSION

The results from both imbalanced and balanced settings depict an enhancement in performance for IPSDM model. Validation and test accuracy are separately measured to understand if there is any overfitting issue persist. Baseline DistilBERT has 82.63% of validation accuracy and 88.95% of testing accuracy whereas IPSDM has 97.50% and 97.10% validation and test accuracy respectively. The baseline model's accuracy variation is 6 (±32)% in training and testing performance reveal that base DistilBERT is exhibiting a minor overfitting problem. However, this has been effectively handled during the development of IPSDM DistilBERT version. Again, a similar trait is visible for base RoBERTA and IPSDM for RoBERTA as well. Base RoBERTA model's validation and testing accuracy gap is around 6 (±19)% whereas IPSDM has 0.01% of difference between these two values.

Such evaluation has been also extended to imbalanced dataset to analyze how IPSDM is performing in challenging scenario. In the imbalanced setting, Tables 1 and 2 present that our proposed model has outperformed the baseline models in this scenario as well. While the IPSDM model has demonstrated significant improvements in performance compared to baseline models, there are still areas for improvement. One limitation is the potential bias towards the majority class ('ham') due to the heavy skewness of the sample distribution. This presents that the model is predicting most of the instances as 'ham'. This bias can result in higher precision but lower recall, indicating that the model may be overly conservative in classifying instances as 'ham'. The results show that precision values for both baseline and IPSDM models are notably higher for base DistilBERT and IPSDM (consecutively 0.85 and 0.98); for base RoBERTA and IPSDM respectively 0.92 and 0.98. To address this, future research could explore techniques to balance the dataset further or modify the model architecture to better handle class imbalances. However, later applying ADASYN, an advanced sampling technique, this class imbalanced situation is handled at the initial stage. A prominent change in performance is hence demonstrated in IPSDM models both for DistilBERT and RoBERTA both for balanced and imbalanced datasets scenarios.

Our work shows how emergent large language model (LLM) technology can be leveraged to solve existing issues presented in the phishing and spam problem. While NLP and other traditional machine learning approaches are viable, using LLMs has vast potential as LLMs advance and continue to transform society. We illustrate how a LLM can be custom trained to improve results. In our case, we show that we can have an impact in improving phishing and spam detection. While the future of LLM is vast, we are at the nascent stages of applying LLMs to existing problems. Our results can help secure cyberspace and save businesses time and money by detecting phishing and spam thereby improving their cyber-defense and improving their overall cyber-health. Furthermore, our method can be extended to new emerging LLM models which are improving at an astounding pace. Future researchers can base studies by applying our method to newly emerging LLMs and apply our method to a wide array of other challenges facing business and researchers.

6 | CONCLUSION AND FUTURE DIRECTIONS

Solving long standing societal issues via radical new approaches, specifically LLMs, shows great promise to improving the lives and experiences of computing users the world over. Phishing and Spam have long since been an issue causing lost time and straining financial resources of consumers and organizations. We demonstrate how leveraging new technology can be applied to these persistent challenges. LLMs offer society great benefits and we have only scratched the surface on their potential. In the future, improving the quality of life via multiple dimensions will be realized such as medical diagnoses, chat-bots, education, and security to name a few. This work demonstrates how LLMs can be leveraged to detect phishing and spam by leveraging LLMs and then presenting our fine-tuned version, IPSDM. Following the proposed mechanism, modified DistilBERT could achieve 97.50% of validation and 97.10% of test accuracy with a F1-score of 0.97. Again, the modified RoBERTA model obtained 98.99% of validation and 99.00% of test accuracy including a F1-score of 0.98. The result of this study presents the effectiveness of IPSDM model while reducing the overfitting issues and handling imbalanced datasets. The attained accuracy has surpassed the existing state-of-the art models.

While the IPSDM model has shown promising results in both balanced and imbalanced dataset scenarios, further evaluation and validation are necessary to ensure its robustness and generalizability across different datasets and settings. This could involve testing the model on larger and more diverse datasets, as well as conducting cross-validation experiments to assess its performance under various conditions. Future work entails further refinement of IPSDM via incorporation of additional tuning techniques as well as hyper-parameter tuning and combining with ensemble modeling. Applying data augmentation such as text rotation, contrastive learning, synonym replacement might also assist in increasing the diversity and improving the training performance. Furthermore, the field of Large Language Models has attracted substantial investment from industry and consumers causing it to develop rapidly with new open-source models being released nearly daily. We aim to experiment with further LLMs such as Meta's Llama and Llama 2. Infusing such solutions into chatbot, web applications and other real-world practical systems would serve society in numerous valuable ways.

DATA AVAILABILITY STATEMENT

The data for training, testing, and validating this experiment is developed by concatenating two opensource data sources.[41,42] Repository links: https://www.kaggle.com/datasets/shantanudhakadd/email-spam-detection-dataset-classification and https://github.com/TanusreeSharma/phishingdata-Analysis/blob/master/1st%20data/PhishingEmailData.csv. The data that support the findings of this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGMENT

Open access publishing facilitated by Edith Cowan University, as part of the Wiley - Edith Cowan University agreement via the Council of Australian University Librarians.

ORCID

Iqbal H. Sarker https://orcid.org/0000-0003-1740-5517

REFERENCES

1. Basit A, Zafar M, Liu X, Javed AR, Jalil Z, Kifayat K. A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommun Syst*. 2021;76:139-154.
2. Aslam M. AI and cybersecurity: an ever-evolving landscape. *Int J Adv Eng Technol Innov*. 2024;1(1):52-71.

3. Anand P, Bharti A, Rastogi R. Time efficient variants of twin extreme learning machine. *Intell Syst Appl*. 2023;17:200169.
4. Han K, Xiao A, Wu E, Guo J, Xu C, Wang Y. Transformer in transformer. *Adv Neural Inf Process Syst*. 2021;34:15908-15919.
5. Yao Y, Duan J, Xu K, Cai Y, Sun Z, Zhang Y. A survey on large language model (llm) security and privacy: the good, the bad, and the ugly. High-confidence. *Comput Secur*. 2024;100211.
6. Roumeliotis KI, Tselikas ND. ChatGPT and open-AI models: a preliminary review. *Future Internet*. 2023;15(6):192.
7. Araci D. Finbert: Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv:1908.10063 2019.
8. Khan JY, Khondaker MTI, Afroz S, Uddin G, Iqbal A. A benchmark study of machine learning models for online fake news detection. *Mach Learn Appl*. 2021;4:100032.
9. Deb S, Chanda AK. Comparative analysis of contextual and context-free embeddings in disaster prediction from twitter data. *Mach Learn Appl*. 2022;7:100253.
10. Jamal S, Cruz MV, Chakravarthy S, Wahl C, Wimmer H. Integration of EEG and eye tracking technology: a systematic review. *SoutheastCon*. 2023;2023:209-216.
11. Govil N, Agarwal K, Bansal A, Varshney A. A machine learning based spam detection mechanism. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC) IEEE; 2020:954-957.
12. Chen C, Zhang J, Xie Y, et al. A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE Trans Comput Soc Syst*. 2015;2(3):65-76.
13. Kumar S, Gao X, Welch I, Mansoori M. A machine learning based web spam filtering approach. 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), IEEE, 973-980. 2016.
14. Baaqeel H, Zagrouba R. Hybrid SMS spam filtering system using machine learning techniques. 2020 21st International Arab Conference on Information Technology (ACIT), IEEE, 1-8. 2020.
15. Guzella TS, Caminhas WM. A review of machine learning approaches to spam filtering. *Expert Syst Appl*. 2009;36(7):10206-10222.
16. Dada EG, Bassi JS, Chiroma H, Adetunmbi AO, Ajibuwa OE. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*. 2019;5(6).
17. Mahajan S. Phishing uniform resource locator detection using machine learning: a step towards secure system. *Secur Priv*. 2023;6(6):e311.
18. Almousa M, Zhang T, Sarrafzadeh A, Anwar M. Phishing website detection: how effective are deep learning-based models and hyperparameter optimization? *Secur Priv*. 2022;5(6):e256.
19. Wu T, Liu S, Zhang J, Xiang Y. Twitter spam detection based on deep learning. Proceedings of the Australasian Computer Science Week Multiconference, 1-8. 2017.
20. Madani M, Motameni H, Mohamadi H. Fake news detection using deep learning integrating feature extraction, natural language processing, and statistical descriptors. *Secur Priv*. 2022;5(6):e264.
21. Chetty G, Bui H, White M. Deep learning based spam detection system. 2019 International Conference on Machine Learning and Data Engineering (iCMLDE), IEEE, 91-96. 2019.
22. Qian F, Pathak A, Hu YC, Mao ZM, Xie Y. A case for unsupervised-learning-based spam filtering. *ACM SIGMETRICS Perform Eval Rev*. 2010;38(1):367-368.
23. Manaa M, Obaid A, Dosh M. Unsupervised approach for email spam filtering using data mining. *EAI Endors Trans Energy Web*. 2021;8(36).
24. Cabrera-León Y, García Báez P, Suárez-Araujo CP. E-mail spam filter based on unsupervised neural architectures and thematic categories: design and analysis. International Joint Conference on Computational Intelligence Springer, 239-262. 2016.
25. Jaya T, Kanyaharini R, Navaneesh B. Appropriate detection of HAM and spam emails using machine learning algorithm. 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), IEEE, 1-5. 2023.
26. Karim A, Azam S, Shanmugam B, Kannoorpatti K. Efficient clustering of emails into spam and ham: the foundational study of a comprehensive unsupervised framework. *IEEE Access*. 2020;8:154759-154788.
27. Ghiassi M, Lee S, Gaikwad SR. Sentiment analysis and spam filtering using the YAC2 clustering algorithm with transferability. *Comput Ind Eng*. 2022;165:107959.
28. Graterol W, Diaz-Amado J, Cardinale Y, Dongo I, Lopes-Silva E, Santos-Libarino C. Emotion detection for social robots based on NLP transformers and an emotion ontology. *Sensors*. 2021;21(4):1322.
29. Acheampong FA, Nunoo-Mensah H, Chen W. Transformer models for text-based emotion detection: a review of BERT-based approaches. *Artif Intell Rev*. 2021;1-41.
30. Jamal S, Cruz MV, Kim J. Cloud-based human emotion classification model from EEG signals. 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE; 2023.
31. Jiang M, Wu J, Shi X, Zhang M. Transformer based memory network for sentiment analysis of web comments. *IEEE Access*. 2019;7:179942-179953.
32. Yaseen Q. Spam email detection using deep learning techniques. *Procedia Comput Sci*. 2021;184:853-858.
33. Liu X, Lu H, Nayak A. A spam transformer model for SMS spam detection. *IEEE Access*. 2021;9:80253-80263.
34. Guo Y, Mustafaoglu Z, Koundal D. Spam detection using bidirectional transformers and machine learning classifier algorithms. *J Comput Cognit Eng*. 2023;2(1):5-9.
35. Tida VS, Hsu S. Universal spam detection using transfer learning of BERT model. arXiv preprint arXiv:2202.03480 2022.
36. Saifullah K, Khan MI, Jamal S, Sarker IH. Cyberbullying text identification based on deep learning and transformer-based language models. *EAI Endors Trans Ind Netw Intell Syst*. 2024;11(1):e5.
37. Wang Y, Zhu W, Xu H, Qin Z, Ren K, Ma W. A large-scale pretrained deep model for phishing URL detection. ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE; 2023:1-5.

38. Maneriker P, Stokes JW, Lazo EG, Carutasu D, Tajaddodianfar F, Gururajan A. URLTran: improving phishing URL detection using transformers. MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM), IEEE; 2021:197-204.
39. Le H, Pham Q, Sahoo D, Hoi SC. URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv preprint arXiv:1802.03162 2018.
40. Tajaddodianfar F, Stokes JW, Gururajan A. Texception: a character/word-level deep learning model for phishing URL detection. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE; 2020:2857-2861.
41. Dhakad S. Email Spam Detection Dataset (classification). 2023.
42. Sharma T. PhishingEmailData. Year of dataset publication. 2022.
43. Liu Z, Lin W, Shi Y, Zhao J. A robustly optimized BERT pre-training approach with post-training. China National Conference on Chinese Computational Linguistics, Springer; 2021:471-484.
44. Jain SM. Hugging face. *Introduction to Transformers for NLP: with the Hugging Face Library and Models to Solve Problems*. Springer; 2022:51-67.
45. Zhuang Z, Liu M, Cutkosky A, Orabona F. Understanding adamw through proximal methods and scale-freeness. arXiv preprint arXiv:2202.00089 2022.
46. Jamal S, Wimmer H. Performance analysis of machine learning algorithm on cloud platforms: AWS vs azure vs GCP. International Scientific and Practical Conference on Information Technologies and Intelligent Decision Making Systems, Springer; 2022:43-60.
47. Yao Z, Gholami A, Shen S, Mustafa M, Keutzer K, Mahoney M. Adahessian: an adaptive second order optimizer for machine learning. *Proc AAAI Conf Artif Intell*. 2021;35:10665-10673.

How to cite this article: Jamal S, Wimmer H, Sarker IH. An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach. *Security and Privacy*. 2024;7(5):e402. doi: 10.1002/spy2.402