Large Language Model Approach for Detecting Phishing and Safe Emails

Arpita Das, Ph.D. Student, tup33422@temple.edu
Department of Electrical and Computer Engineering, Temple University, USA

Abstract

Phishing attacks lead to significant financial losses and serve as entry points for ransomware, posing a major challenge for email users worldwide. While heuristic-based detection methods exist, Large Language Models (LLMs) offer a novel approach to enhance the understanding and management of this problem. LLMs have demonstrated transformative potential across various sectors especially in cyber security, making their application to email phishing detection a logical research progression. In this project, DistilBERT and RoBERTA were used to classify phishing and safe emails, with both models showing promising results in improving classification accuracy on a balanced dataset. DistilBERT, in particular, was favored for its higher accuracy and reduced computational demands, achieving an exceptional 99.52% accuracy, surpassing state-of-the-art benchmarks.

Keywords: Large language model (LLM), Phishing, Safe, Cyber Security, Fine Tuning, DistilBERT

1. Introduction

Phishing remains a pervasive issue, impacting users globally and inflicting substantial economic losses. This menace not only affects individuals but also businesses, leading to severe financial repercussions. These malicious schemes aim to manipulate people into divulging sensitive information like financial details or login credentials. Despite efforts to mitigate these threats using artificial intelligence (AI) approaches [1], traditional heuristic-based systems are still widely used. Nonetheless, the landscape is changing thanks to technological advancements and increased investment in research from both public and private sectors. This evolution has spurred the creation of new and unconventional strategies for combating phishing and enhancing security, as demonstrated by Anand et al. [2].

Transformer-based models have revolutionized the way we develop systems for classifying phishing and safe emails by processing and interpreting textual data. These models continue to evolve, incorporating various regularization methods and benefiting from attention mechanisms that enhance interpretability and facilitate understanding of classification decisions [3]. Large Language Models (LLMs), popularized by Open AI's ChatGPT, have proven effective in addressing complex problems and adapting to longstanding challenges like phishing detection. Powered by the GPT engine, ChatGPT has gained widespread use in both consumer and business contexts [4]. Other notable LLMs from companies such as Google, Meta, and academic institutions like MIT have also made significant contributions. The release of models like Llama and BERT into the open-source domain has spurred further research and development across diverse entities, from major institutions to individual consumers. Although these models are available for download, the practical deployment of pre-trained models like BERT is still in its early stages, with growing access to local GPU technology facilitating usage in platforms such as Google's Colaboratory and Hugging Face's transformers library and model hosting services.

Large Language Models (LLMs) are designed to be versatile, initially trained on a broad range of data by their developers for both commercial and non-commercial use. To train an LLM, diverse inputs are used, including web scraped data, document corpora, and texts from emails, transcribed books, discussions, or speeches. While LLMs generally perform well on broad tasks, their efficacy can be enhanced for specific applications through fine-tuning. An example of this is FinBERT [5], a variant of BERT (Bidirectional Encoder Representations from Transformers) that is specially trained on financial documents to better handle finance-related queries. Similar advancements are underway in the medical field to support both physician decision-making and patient inquiries. BERT utilizes a self-attention mechanism that allows it to understand the context and the relationships between words in any text sequence. This mechanism calculates the importance of each word, producing attention scores for all words or input tokens, which are then processed through a SoftMax function to create rich contextual embeddings. These capabilities enable BERT-based models to excel in a variety of natural language understanding tasks.

Among the variants of BERT-based models, DistilBERT and RoBERTA stand out, having been applied to tasks such as detecting fake news [6] and analyzing Twitter data for predictions [7]. Both models utilize the transformer architecture and excel in natural language processing (NLP) tasks.

DistilBERT is designed for reducing the number of parameters making it a faster and smaller version of BERT. Due to the training response time, the work has been restricted to DistilBERT and RoBERTA for the project. In this work, I utilized the DistilBERT and RoBERTA pre-trained models of the BERT family models and fine-tuned versions of both. The goal of this project is to find either

comparable or better results than the state-of-the-art methods using either of the models on Phishing and safe email datasets.

2. Literature Survey

The body of research on transformer-based methods, especially in the realm of fine-tuning transformers or employing attention mechanisms for phishing email identification, is still growing. Within this niche, Yaseen [8] has developed a novel word embedding technique aimed at classifying spam emails. In this approach, the pre-trained BERT model is fine-tuned to distinguish between safe and non-safe emails. A Deep Neural Network using BiLSTM serves as the baseline for comparison. For training and testing the model, two open-source datasets from the UCI Machine Learning Repository and Kaggle were utilized. The model achieved an impressive classification accuracy of 98.67%. In a similar vein, Liu, et al. [9] have crafted and tested a modified transformer model for phishing email detection using the phishing Collection v.1 and UtkMI's Twitter Phishing Detection Competition datasets. Their model reached a high accuracy of 98.92%, with recall and F1 scores of 0.9451 and 0.9613, respectively.

Guo, et al. [10] and Tida and Hsu [11] have highlighted the critical role of the self-attention mechanism in BERT models. Guo and colleagues [10] applied a pre-trained BERT model to classify emails as safe or phishing using two public datasets: the Enron dataset [12] and a simple phishing email classifier dataset from Kaggle. Similarly, the Universal Spam Detection Model (USDM) was developed and evaluated using four publicly available datasets: the Lingspam dataset [13], a spam text dataset from Kaggle, the Enron dataset, and the spam assassin dataset. This model achieved an overall accuracy of 97% and an F1 score of 0.96 [11]. There is a distinct difference between spam and phishing emails. However, due to time constraints, the focus has been put specifically on phishing and safe email classification in this work.

Additionally, there has been research on fine-tuning BERT-based models for phishing URL detection [14], [15]. Wang, et al. [14] gathered 2.19 million URL data points from PhishTank to pre-train the PhishBERT model, which demonstrated a 92% accuracy rate in identifying phishing URLs. Maneriker and colleagues [15] fine-tuned both BERT and RoBERTa models to develop the URLTran transformer, using Microsoft Edge and Internet Explorer browsing telemetry data for training, testing, and validation. A downsampling method was used to balance the datasets, resulting in a final training set of 77,870 URLs. These models showed a True Positive Rate (TPR) of 86.80%, outperforming baseline models URL-Net [16] and Texception [17].

3. Methodology

In this project, transformer-based self-attention mechanism models are explored with the aim of improving the state-of-the-art results using pre-trained BERT family models (DistilBERT and RoBERTA large). Our collected and prepared dataset is used for developing models in two different settings. 1) DistilBERT and RoBERTA were pretrained using phishing and safe emails dataset, and 2) the models' training process has been improved through applying optimization and fine-tuning mechanism. We then compared our results with the state-of-the-art results.

The top-level methodology of this research is presented in Fig. 1.

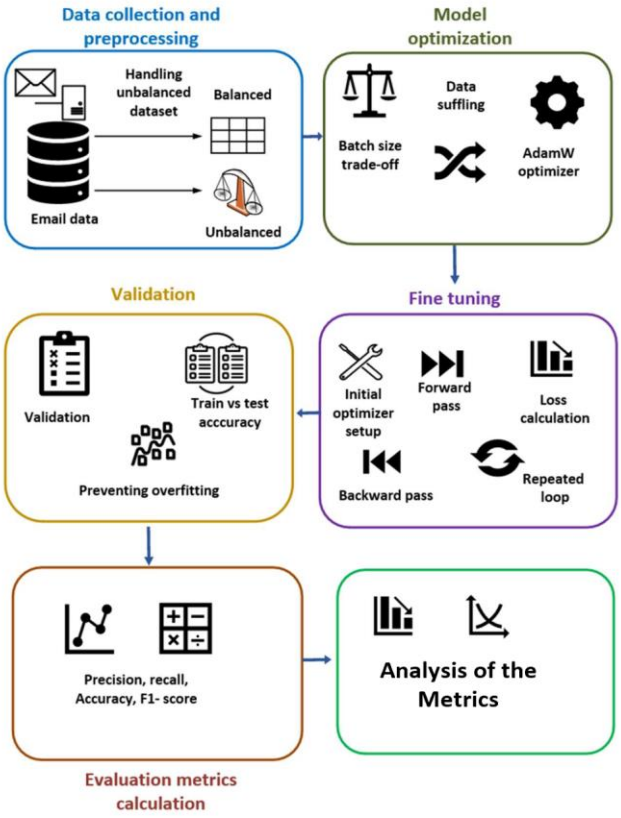

Fig. 1. Overall Methodology

3.1 Data Collection and Preparation

The data for training, testing, and validating this experiment is taken from an open-source dataset that has phishing and safe emails.[1] The dataset contains 18,634 emails with 61% safe emails and 39% phishing emails. The minority class then has been oversampled randomly to address unbalanced issues and to prevent overfitting. It generally du-

[1] https://github.com/arpita-das-CUET/Large-Language-Model-Approach-to-Detect-Phishing-and-Safe-Emails

plicates some of the original samples of the minority class. Fig. 2 shows a snapshot of the final dataset.

Email Text	Email Type
re : 6 . 1100 , disc : uniformitarianism , re ...	Safe Email
the other side of * galicismos * * galicismo *...	Safe Email
re : equistar deal tickets are you still avail...	Safe Email
\nHello I am your hot lil horny toy.\n I am...	Phishing Email
software at incredibly low prices (86 % lower...	Phishing Email
...	...
date a lonely housewife always wanted to date ...	Phishing Email
request submitted : access request for anita	Safe Email
re : important - prc mtg hi dorn & john , as y...	Safe Email
press clippings - letter on californian utilit...	Safe Email
empty	Phishing Email

Fig. 2. A snapshot of dataset overview

3.2 Data Splitting

The entire dataset is divided into two segments: 80% for training and 20% for testing. Within the training segment, further subdivision is made where 60% is allocated for direct training purposes and the remaining 20% is set aside for validation. This validation set is employed at the end of each training epoch to help pinpoint the model's optimal performance. Such a structure is crucial in the development process as it confirms the model's ability to effectively identify and predict outcomes on data it has not previously encountered.

3.3 Models Selection

A. DistilBERT

DistilBERT is a compact version of the original BERT model (Bidirectional Encoder Representations from Transformers), which itself is based on transformer architecture and pre-trained to tackle various natural language processing tasks. The purpose behind DistilBERT is to refine and compress BERT, enhancing its speed and computational efficiency [18]. This is achieved through a teacher-student training approach, where the smaller, "student" model learns to replicate the performance of the larger, "teacher" model.

The process begins with raw text inputs, which must be preprocessed. These inputs are tokenized using a vocabulary that breaks words down into sub-words. These tokenized inputs are then converted into numerical embeddings. The relationships between the words are understood through an attention mechanism in the attention layer. This mechanism calculates scores that assess the relevance of each word concerning others within the sequence, helping the model to focus more on important words and less on irrelevant ones.

The model includes a pooling section that represents the entire input sequence uniformly. Depending on the specific requirements, the classifier head can be adapted, and the final prediction layer outputs the results, such as classifying emails as safe or phishing in this case.

B. RoBERTA Large

RoBERTA (A Robustly Optimized BERT Pretraining Approach) is an enhancement of the transformer-based BERT model, designed to handle larger batch sizes and train with longer sequences. The pretraining phase of RoBERTA uses an improved bidirectional context-oriented mechanism that focuses on learning masked-out tokens across extended sequences [19]. While RoBERTA shares a similar architecture to DistilBERT, including transformer encoder layers with multi-head attention mechanisms, there are distinct differences.

Notably, RoBERTA utilizes a byte-level tokenizer, unlike BERT's tokenization approach. It employs dynamic masking that varies across different training epochs and uses Byte Pair Encoding (BPE) as a subunit rather than characters. In RoBERTA's workflow, tokens are inputted and preprocessed by the tokenizer, followed by encoding, pooling, decoding, and an attention mechanism. Despite these differences, the fundamental structure of both the DistilBERT and RoBERTA models remains similar, as depicted in Fig. 3.

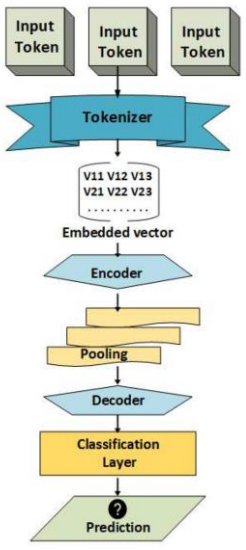

Fig. 3. Basic architecture of DistilBERT and RoBERTA

There are two types of RoBERTA models – RoBERTA Large and RoBERTA Base. The main differences between RoBERTA Large and RoBERTA base are the number of parameters and the number of hidden layers. RoBERTA Large has twice as many parameters and 1.5 times as many hidden layers as RoBERTA base. This allows RoBERTA Large to learn more complex representations and perform tasks with higher accuracy.

The specific differences between RoBERTA Large and Roberta_base are as follows:

Number of parameters: RoBERTA Large has 137B, RoBERTA base has 60B

Number of hidden layers: RoBERTA Large has 24, RoBERTA base has 12

Training data: RoBERTA Large has 1.3B words, RoBERTA base has 300M words

Task accuracy: RoBERTA Large achieves higher accuracy on all tasks than RoBERTA base. RoBERTA Large is a more accurate language model than RoBERTA base, but it also takes longer to train and requires more computational resources.

Since the work is bound by the training time constraints, RoBERTA Large has been used instead of RoBERTA base.

3.3 Improving the Training

The work aimed to improve the pre-trained models through model optimization i.e., learning rate scheduling, adjusting batch size, sequence length and loss function, hyper parameter tuning, early stopping and fine tuning.

A. Model Optimization

The balanced phishing and safe emails dataset is tokenized using the Hugging Face Transformers tokenizer [20], which adopts a sub-word-based method to break down the text into smaller units. This tokenization process helps the model understand the meanings and contexts of words more effectively. Both the pre-trained DistilBERT and RoBERTA models are then loaded with their respective weights, which were acquired during their pre-training phases. The training process is configured with a batch size of 16 for the training data and 32 for the validation data, balancing memory usage with training efficiency. To enhance the model's exposure to diverse data and prevent overfitting, the training data is shuffled at the beginning of each epoch.

In this experimental setup, the AdamW (Adam Weight Decay) optimization algorithm is utilized to update the weights of the pre-trained models. AdamW adjusts the learning rate for each parameter individually by incorporating both the averages of exponential moving gradients and the square roots of these gradients [21]. Additionally, it integrates L2 regularization, a weight decay method that introduces a penalty into the loss function based on the square of the magnitude of the weights. This encourages the use of smaller weights, thus helping to reduce model complexity and lower the risk of overfitting.

The model's parameter, Z, is initialized with the exponential decay rates, $\beta 1$ and $\beta 2$, and a very small value ε to prevent division by zero. Initially, the first moment ($m_0 = 0$) and the second moment ($v_0 = 0$) are set to zero. During each training iteration, the loss gradient is calculated as follows:

$$\text{Gradient loss, } g = \nabla_z L(z)$$

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 gets corrected,

$$\widehat{m_l} = \frac{m_i}{1 - \beta 1^i}$$

$$\hat{v}_l = \frac{v_i}{1 - \beta 2^i}$$

Finally, the parameters are updated using the AdamW updating rule,

$$Z_i = Z_{i-1} - \frac{\text{learning rate}}{\sqrt{\hat{v}_l} + \varepsilon} \cdot (\widehat{m_l} + \text{ weight decay } * Z_{i-1})$$

The weight decay regularization technique helps manage the increase in parameter values throughout training, reducing the likelihood of overfitting.

For this multiclass classification task, the Cross-Entropy Loss function is employed. This function effectively merges SoftMax activation and negative log likelihood into a single term of loss. The goal is to minimize the discrepancy between the actual labels and the predicted probabilities during training. PyTorch offers an implementation of cross-entropy loss that automatically performs both the SoftMax computation and the logarithmic calculations. The loss for each training epoch can be described as follows:

$$\text{Loss}_i = -\sum_{k=1}^{n} Z_i, k * \log\left(p_{i,k}\right)$$

Here, Z_i, k is the ground-truth label, and $p_{i,k}$ is the predicted probability made by the model.

The cross-entropy loss for the overall training is the average of individual loss,

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

The models generate logits for each class, which are then converted into class probabilities using a SoftMax activation function. The predicted probability $p_{i,k}$ is computed as below, where Z_i, k is the produced logit value.

$$p_{i,k} = \frac{e^{Z_{i,k}}}{\sum_{m=1}^{k} e^{Z_{i,m}}}$$

The optimization process diagram is presented in Fig. 4.

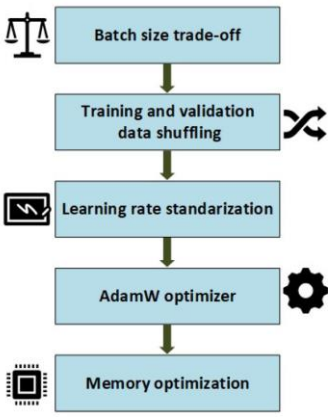

Fig. 4. Model Optimization

B. Learning Rate

The optimal learning rate for model optimization and fine-tuning varies based on factors like the model's architecture, the optimization algorithms used, and the task at hand. This rate is vital as it determines the size of the steps taken during the optimization process. A high learning rate may cause the model to become unstable, leading to subpar performance on new data. Conversely, a low learning rate may decelerate the model's convergence, necessitating more training epochs to reach satisfactory results, thereby increasing computational expenses.

In our experiment, a standard learning rate, 1e-2 (0.01) is set for BERT-based models, i.e., RoBERTA and DistilBERT.

C. Fine Tuning

The fine-tuning process tailors a pre-trained model to specialize in particular tasks and datasets, thus improving the model's performance on domain-specific activities, such as email classification in our scenario. During fine-tuning, the model is trained using 80% of the dataset designated earlier for training purposes. In each training epoch, the data is fed into the model in batches, and the gradient is calculated using the backpropagation technique.

The overall fine-tuning process flow diagram is illustrated in Fig. 5.

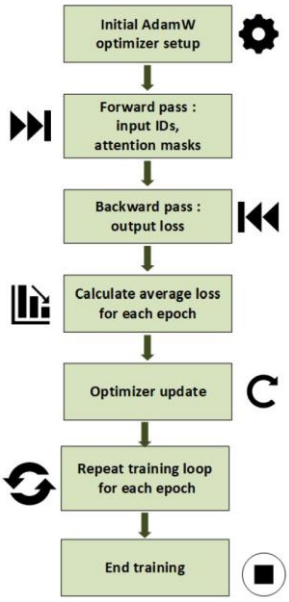

Fig. 5. Fine-tuning process flow

4. Results and Discussions

The setup has been implemented using Google Colab TPU with an average training time of 4.5 hours.[1] The performance may be evaluated by its computational speed, memory requirement, and its evaluation metrics. However, here the work is concentrated on the evaluation metrics since large language models in general require much computational time and memory resources. To evaluate the model's performance thoroughly, several important metrics are computed, including overall accuracy, precision, recall, and F1-score. These metrics offer essential insights into how effectively the model is performing.

4.1 Comparison of DistilBERT and RoBERTA

Table 1 suggests that both DistilBERT and RoBERTA has trade-off results in the sense that one model has better results in one metric, whereas the other showed better in another metric. The judgment lies in the user which metric to consider as the final evaluation parameter and depends largely on the specific context of the problem and the nature of the data.

Precision is crucial if we want to minimize the number of legitimate (safe) emails incorrectly flagged as phishing. A high precision means that when an email is labeled as phishing, there is a high likelihood that it truly is phishing. This is particularly important in business contexts where

missing important emails due to incorrect classification can lead to missed opportunities or disrupted communications.

Recall becomes significant when it is critical to catch as many phishing attempts as possible. For instance, in security-sensitive industries, failing to identify a phishing email could lead to severe security breaches. A high recall ensures that most phishing emails are correctly identified, even if some safe emails get caught in the filter.

Since both missing phishing emails and misclassifying safe emails carry significant costs, the F1-Score is a useful metric in phishing detection. It balances the need for precision and recall, providing a more comprehensive view of the model's performance when both false positives and false negatives are costly.

Accuracy can give a quick snapshot of overall model performance especially when the dataset is balanced. Thus, in this case, the focus was put on accuracy to evaluate the performance. Although RoBERTA performed better than DistilBERT in terms of F1-Score, the latter outperformed the former based on accuracy. Thereby, it is concluded that DistilBERT serves as a better model than RoBERTA in regard to accuracy. Also, the former uses fewer parameters than the latter utilizing fewer resources with less computational time.

Table 1. DistilBERT vs. RoBERTA performance

Metrics	DistilBERT	RoBERTA
Val Accuracy	82.63%	81.64%
Test Accuracy	**99.52%**	**94.67%**
Val Precision	0.8543	0.9212
Test Precision	0.9025	0.9474
Val Recall	0.6971	0.9031
Test Recall	0.7532	0.9470
Val F1-Score	0.8867	0.9820
Test F1-Score	0.8943	0.9467

4.2 Comparison with State-of-the-art Methods

Table 2 clearly shows that the implemented DistilBERT and RoBERTA surpassed all the previous results of the state-of-the-art methods except for [22] where their RoBERTA model outperformed the implemented RoBERTA model. However, this work's DistilBERT is better than theirs. This could be because of the optimization algorithm or different fine-tuning parameters like learning rate, batch size, etc. Also, large language models are sensitive to the dataset in the sense that different work can preprocess their data differently. They are also sensitive to the imbalanced dataset leading to overfitting. The model's performance can also be a trade-off between hyper-tuning parameters.

Table 2. Accuracy comparison with the literature

Methods	Accuracy (%)
Yaseen [8]	98.67
Liu et. al. [9]	98.92
Tida and Hsu [11]	97.00
Wang et. al. [14]	92.00
Jamal et. al. [22] (RoBERTA)	**96.43**
Jamal et. al. [22] (DistilBERT)	99.00
This work (RoBERTA)	94.67
This work (DistilBERT)	**99.52**

5. Conclusions, Challenges, and Future Work

Utilizing innovative solutions, particularly Large Language Models (LLMs), holds considerable promise for addressing enduring societal challenges and enhancing the daily experiences of computer users worldwide. Issues like phishing and spam have historically consumed significant time and drained the financial resources of both individuals and organizations. Our work illustrates the potential of applying cutting-edge technologies to these persistent problems. LLMs present substantial advantages to society, and we are just beginning to explore their full capabilities. Looking ahead, LLMs are expected to significantly improve quality of life across various areas including medical diagnosis, conversational agents, education, and security, among others. This work implemented pre-trained DistilBERT and RoBERTA models and fine-tuned them to achieve 99.52% accuracy for DistilBERT and 94.67% accuracy for RoBERTA. It was concluded that DistilBERT serves as a better model than RoBERTA in terms of accuracy. Also, the former uses fewer parameters than the latter utilizing fewer resources with less computational time. The implemented DistilBERT model also outperformed the state-of-the-art results, especially that of [22], although the improvement is minimal.

The only challenges were in computational time. Even with the Google Colab resources, the setup took much training time. This slowed down the process while debugging through the codes.

Future efforts may enhance the models through advanced tuning methods, hyper-parameter optimization, and ensemble modeling. Data augmentation techniques like text rotation and synonym replacement can be used to improve training. The rapidly evolving field of Large Language Models, bolstered by significant industry and consumer investment, may see experimenting with newer models like Meta's Llama if not constrained by the training resources. These technologies can be integrated into chatbots, web applications, and other practical systems to provide widespread societal benefits.

References :

[1] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," Telecommunication Systems, vol. 76, pp. 139-154, 2021.

[2] P. Anand, A. Bharti, and R. Rastogi, "Time efficient variants of Twin Extreme Learning Machine," Intelligent Systems with Applications, vol. 17, p. 200169, 2023.

[3] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," Advances in Neural Information Processing Systems, vol. 34, pp. 15908-15919, 2021.

[4] K. I. Roumeliotis and N. D. Tselikas, "ChatGPT and Open-AI Models: A Preliminary Review," Future Internet, vol. 15, no. 6, p. 192, 2023.

[5] D. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," arXiv preprint arXiv:1908.10063, 2019.

[6] J. Y. Khan, M. T. I. Khondaker, S. Afroz, G. Uddin, and A. Iqbal, "A benchmark study of machine learning models for online fake news detection," Machine Learning with Applications, vol. 4, p. 100032, 2021.

[7] S. Deb and A. K. Chanda, "Comparative analysis of contextual and context-free embeddings in disaster prediction from Twitter data," Machine Learning with Applications, vol. 7, p. 100253, 2022.

[8] Q. Yaseen, "Safe email detection using deep learning techniques," Procedia Computer Science, vol. 184, pp. 853-858, 2021.

[9] X. Liu, H. Lu, and A. Nayak, "A safe transformer model for SMS safe detection," IEEE Access, vol. 9, pp. 80253-80263, 2021.

[10] Y. Guo, Z. Mustafaoglu, and D. Koundal, "Safe detection using bidirectional transformers and machine learning classifier algorithms," Journal of Computational and Cognitive Engineering, vol. 2, no. 1, pp. 5-9, 2023.

[11] V. S. Tida and S. Hsu, "Universal safe detection using transfer learning of BERT model," arXiv preprint arXiv:2202.03480, 2022.

[12] I. Androutsopoulos, V. Metsis, and G. Paliouras, "The Enron-safe datasets," Accessed: Oct, vol. 11, p. 2019, 2006.

[13] I. Androutsopoulos, "Ling-safe," Aueb. gr. Accessed: Oct, vol. 11, p. 2019, 2000.

[14] Y. Wang, W. Zhu, H. Xu, Z. Qin, K. Ren, and W. Ma, "A Large-Scale Pretrained Deep Model for Phishing URL Detection," in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023: IEEE, pp. 1-5.

[15] P. Maneriker, J. W. Stokes, E. G. Lazo, D. Carutasu, F. Tajaddodianfar, and A. Gururajan, "URLTran: Improving phishing URL detection using transformers," in MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM), 2021: IEEE, pp. 197-204.

[16] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL detection," arXiv preprint arXiv:1802.03162, 2018.

[17] F. Tajaddodianfar, J. W. Stokes, and A. Gururajan, "Texception: a character/word-level deep learning model for phishing URL detection," in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020: IEEE, pp. 2857-2861.

[18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.

[19] Z. Liu, W. Lin, Y. Shi, and J. Zhao, "A robustly optimized BERT pre-training approach with post-training," in China National Conference on Chinese Computational Linguistics, 2021: Springer, pp. 471-484.

[20] S. M. Jain, "Hugging face," in Introduction to Transformers for NLP: With the Hugging Face Library and Models to Solve Problems: Springer, 2022, pp. 51-67.

[21] Z. Zhuang, M. Liu, A. Cutkosky, and F. Orabona, "Understanding adamw through proximal methods and scale-freeness," arXiv preprint arXiv:2202.00089, 2022.

[22] Jamal, S., Wimmer, H., & Sarker, I. H. (n.d.). An Improved Transformer-based Model for Detecting Phishing, Spam, and Ham-A Large Language Model Approach.