# Phishing Email Detection Using Fine-Tuned BERT Model with Attention-Based Word Importance Highlighting

Gal Granot – gal.granot@campus.technion.ac.il
Nir Tevet – nir.tevet@campus.technion.ac.il

GitHub Repository

## Abstract

This report presents a deep learning-based approach to classifying emails as "phishing" or "safe" using a fine-tuned BERT[1] transformer model, based on the attention architecture. In addition to achieving high classification accuracy, our approach provides additional interpretability to the end-user by identifying important words and passages in each email using the attention score as calculated in the model. This work aims to contribute to the development of reliable, private and transparent phishing detection systems.

## 1. Introduction

### 1.1 Project Goal

The goal of this project is to develop an effective deep learning model for phishing email detection. By leveraging transformer models, we aim to accurately classify emails and highlight key words that contribute most to the classification decision.

### 1.2 Motivation

Phishing attacks continue to be a significant cybersecurity threat, targeting both individuals and organizations. Traditional rule-based systems often are not enough to defend the average email user from the evolving nature of phishing techniques. Machine learning models, particularly large language models based on transformers, offer a promising solution due to their ability to capture contextual information in text. Furthermore, enhancing model explainability and providing transparency into the decision-making process as seen by the user is crucial for gaining the latter's trust and ensuring that the model's decisions can be validated.

### 1.3 Previous Work

Previous research in phishing detection has primarily focused on machine learning techniques such as Support Vector Machines (SVM), Naive Bayes, and Random Forests. While these methods provide decent accuracy, they might lack the capability to capture deep semantic relationships within the text. Recent advancements in natural language processing can sometimes have superior performance in various text classification tasks.

## 2. Method

### 2.1 Algorithm Overview

We employed a pretrained BERT model for our phishing detection task. BERT is a transformer-based model designed to capture deep bidirectional context of each token by jointly conditioning on both left and right context in all layers. This enables BERT to capture rich contextual information, making it ideal for text classification tasks.

[1] BERT - Bidirectional Encoder Representations from Transformer, as presented in a 2019 article

### 2.2 Architecture

Our architecture involves fine-tuning the pre-trained BERT model on a labeled dataset of emails. The fine-tuning process includes adding a classification layer on top of the BERT model to output probabilities for two classes: "phishing" and "safe". The model architecture is depicted in Figure 1.

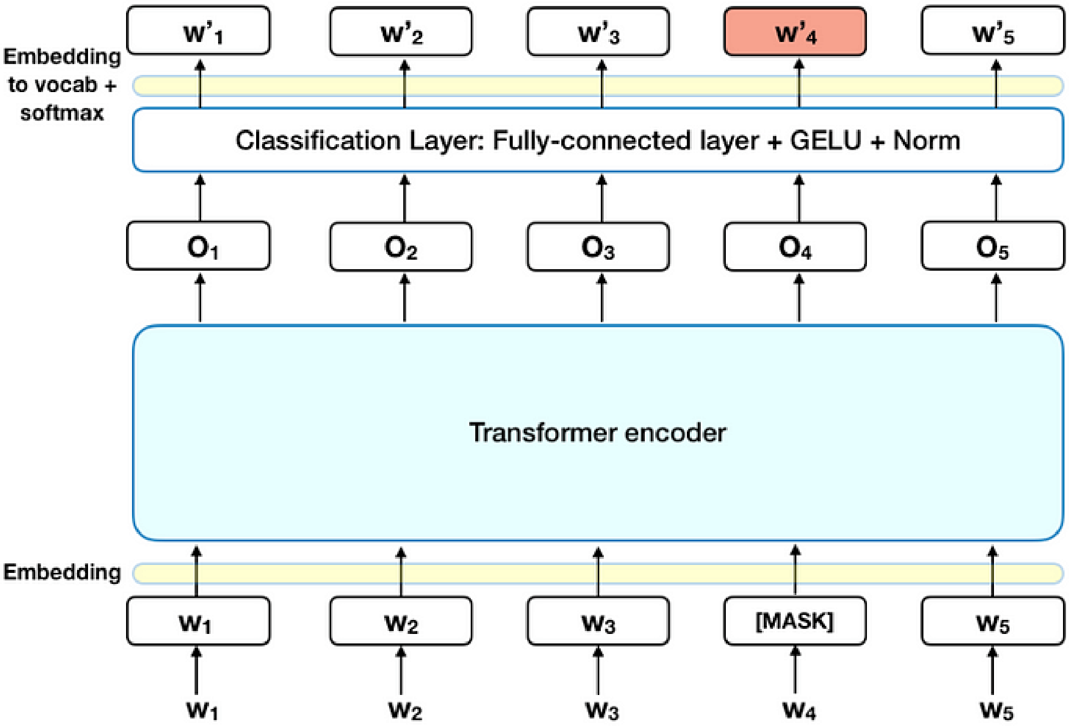
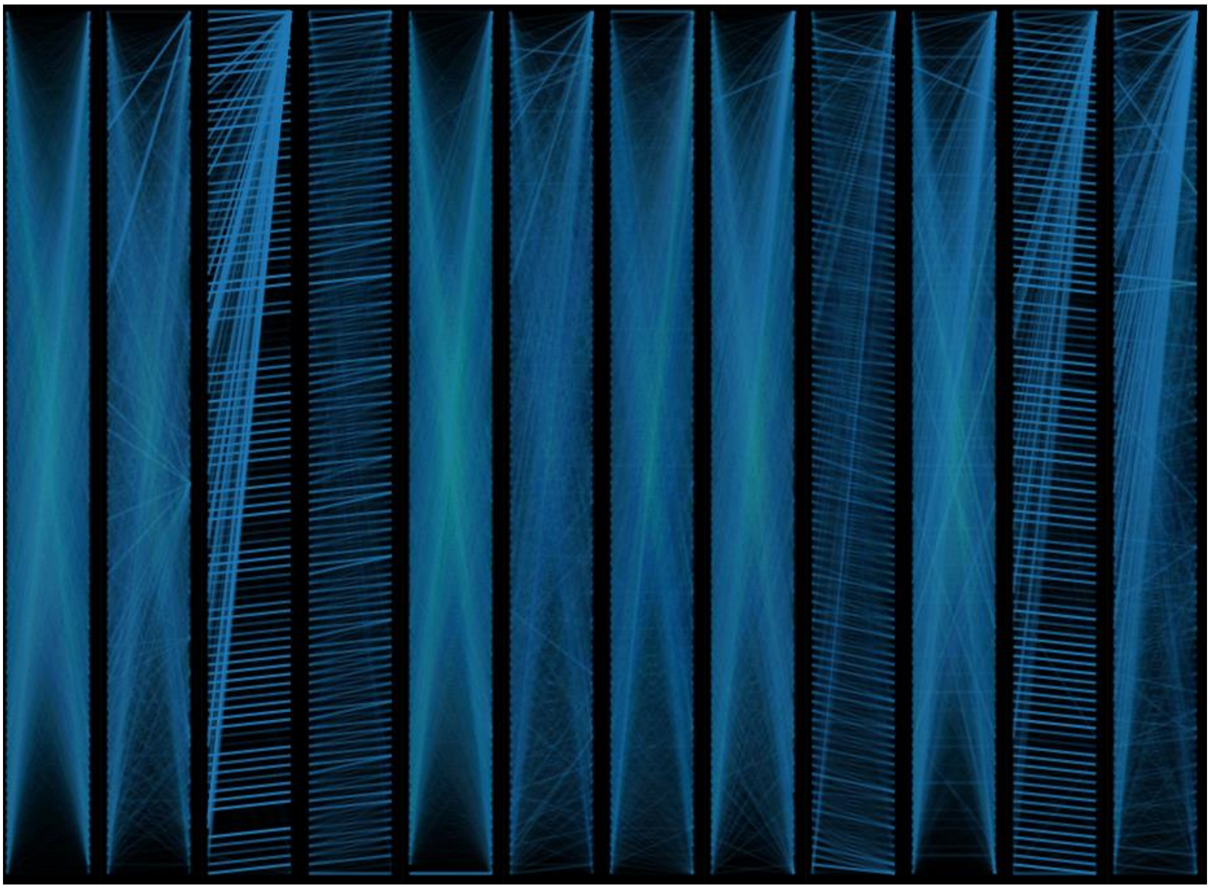

W'1
W'2
W'3
W'4
W'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
W1
W2
W3
[MASK]
W5
W1
W2
W3
W4
W5

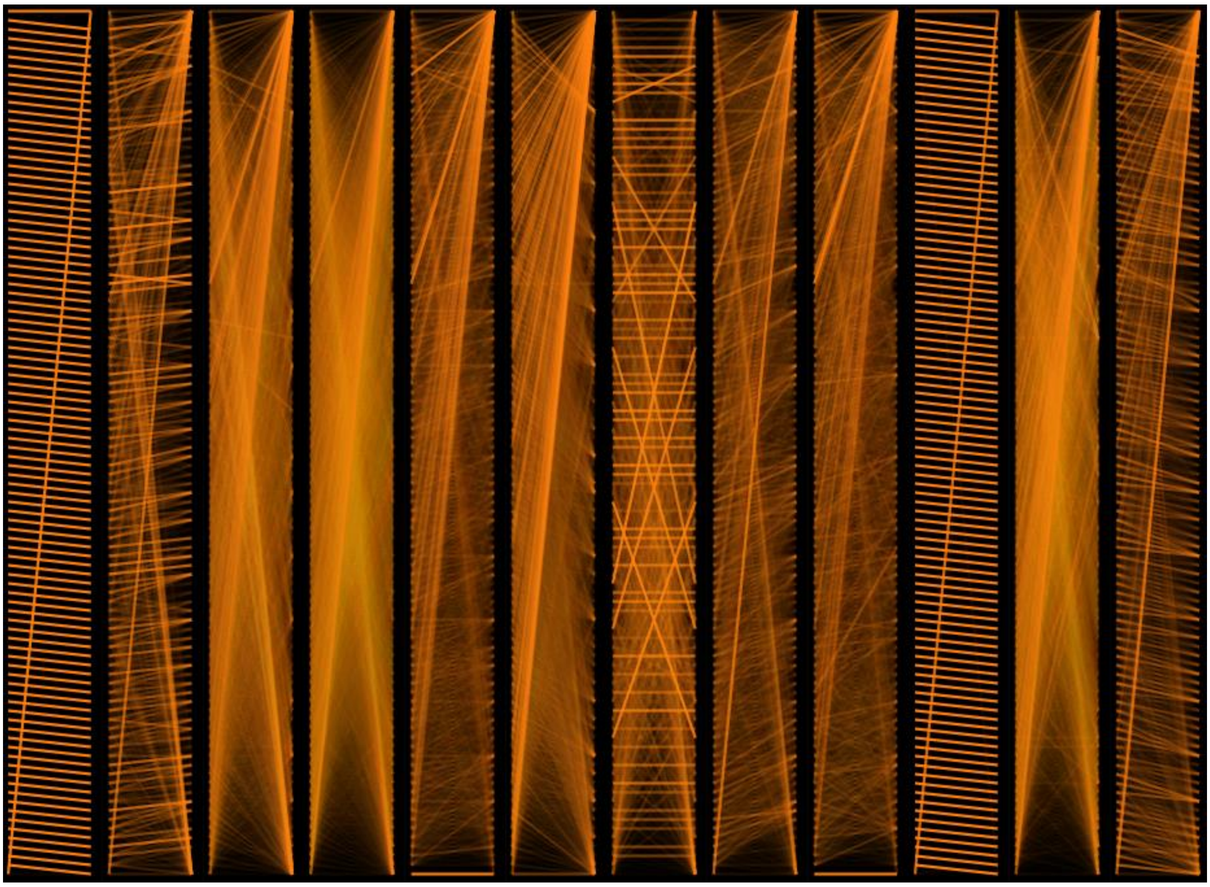
*Figure 1 – BERT model architecture*

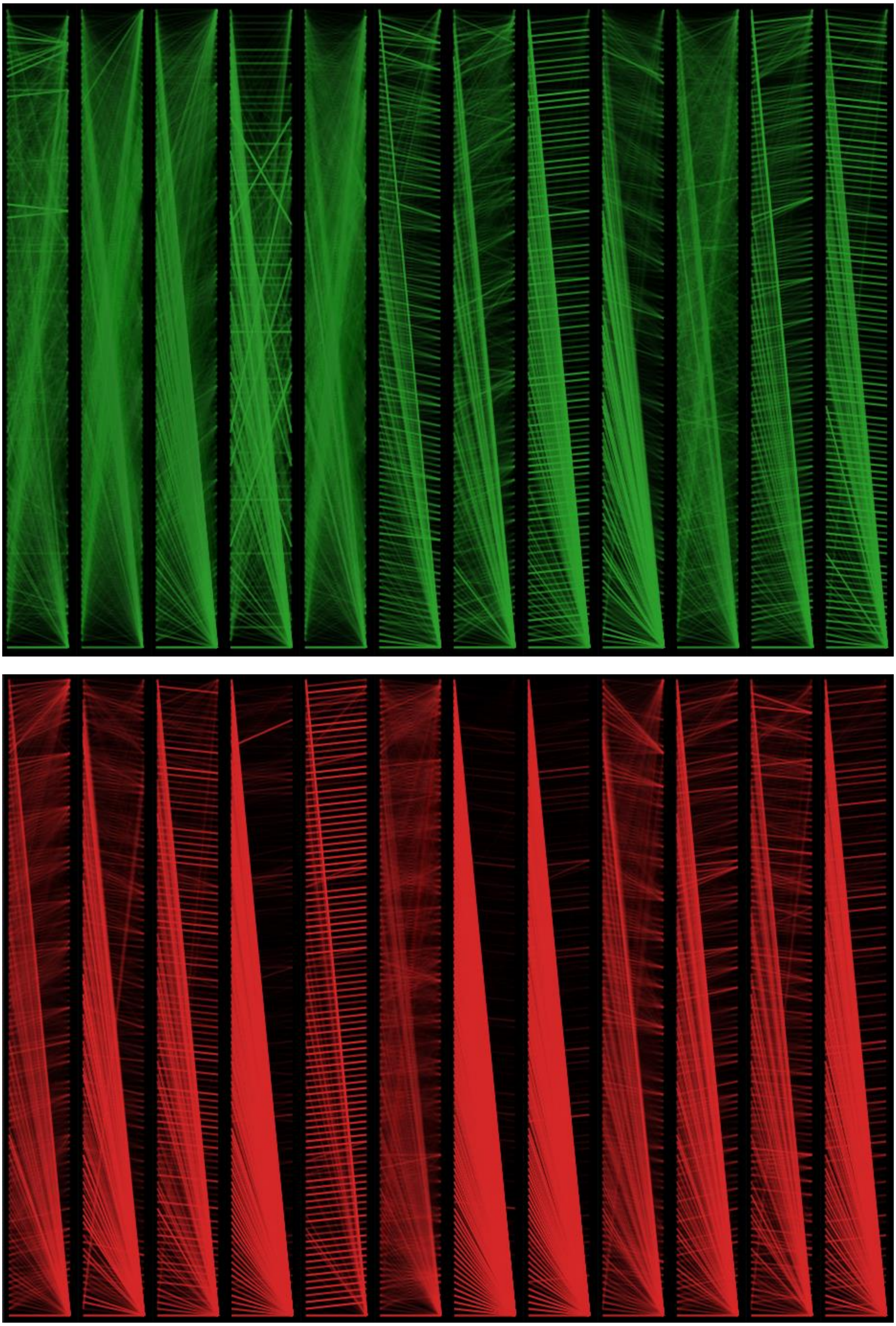
### 2.4 Attention-Based Word Importance

To further our understanding of the way the model has classified the text and keeping in mind our goal of providing explainability to the users, we analyzed the attention weights from the BERT model's attention heads. The [CLS] token, which is prepended to every input sequence, is designed to aggregate information from all other tokens, making it a useful indicator of the overall importance of words in the text. By examining the attention weights focused on the [CLS] token, we identified the most important words in each email that influenced the classification decision. These words were highlighted on a platform we developed, providing a visual representation of the model's focus during prediction. Furthermore, we've identified layers which are very similar to one another, leading us to suspect there might be some redundancy or overfitting in the model and that further work might serve to optimize performance, or that it might benefit some dropout layers.

The next 3 pages contain examples of the head attentions visualizations we've generated using a tool called BertViz (link to repository). Notable is the similarity between the red and purple layers (suggesting possible redundancy) and the tendency on the heads to focus on the [SEP] and [CLS] tokens (special added tokens by the tokenizer).







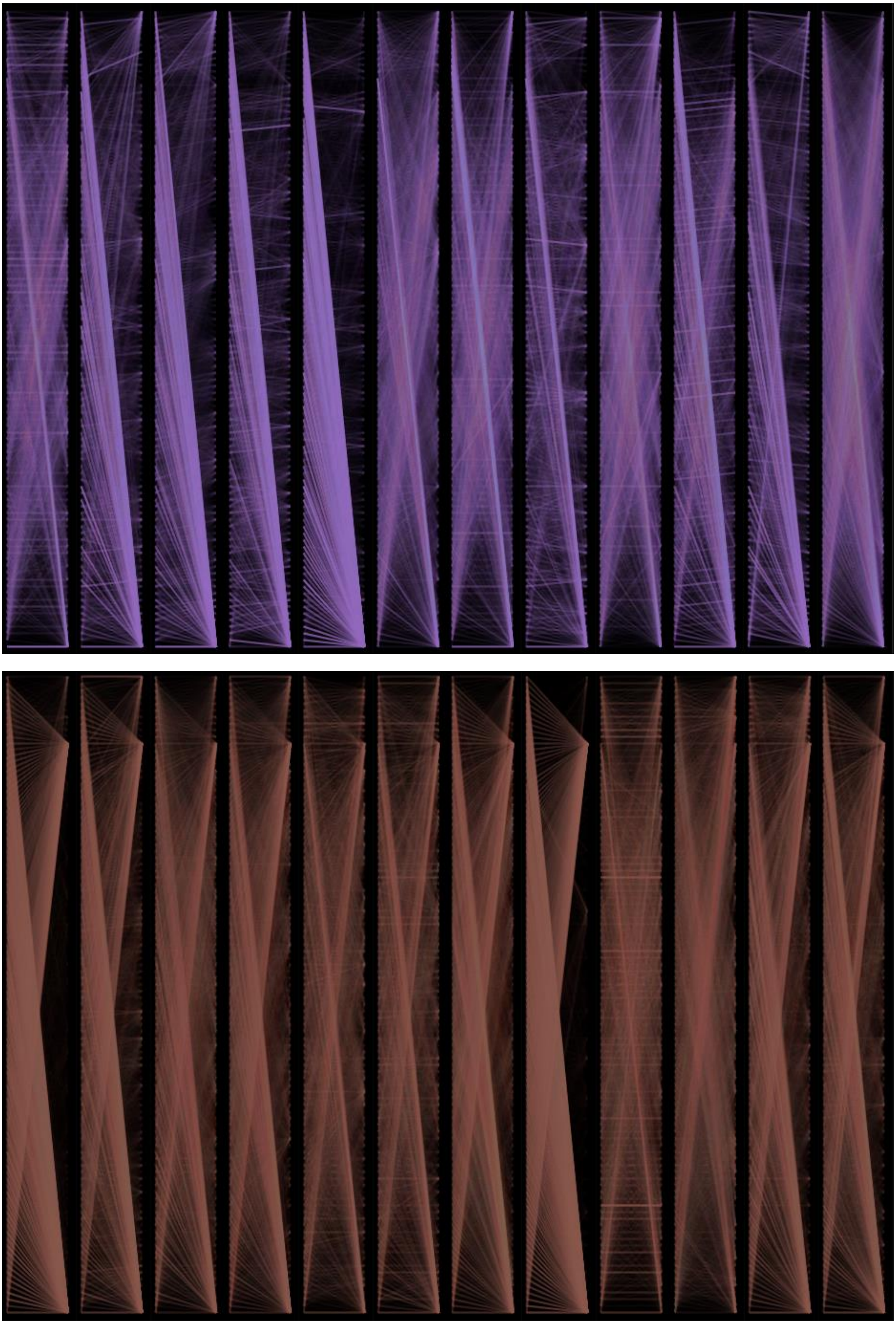


*Figure 2 – visualization of the attention mechanism across different heads and layers*

### 2.5 Hyperparameter Search

To optimize the performance of our BERT-based model, we surveyed various hyperparameter options. Key hyperparameters tuned included the learning rate, batch size, and the number of epochs.

- We experimented with learning rates $\eta \in [0.1 \cdot 10^{-5}, 5 \cdot 10^{-5}]$ in $0.5 \cdot 10^{-5}$ intervals. The best performance was achieved with a learning rate of $\eta = 1 \cdot 10^{-5}$.
- We tested batch sizes of 16, 32, and 64. A batch size of 32 provided the best balance between computational efficiency and model performance.
- The model was trained for 3, 4, and 5 epochs. The optimal number of epochs was 4, where the model showed the best performance on the validation set without overfitting.

## 3. Experiments and Results

### 3.1 Dataset

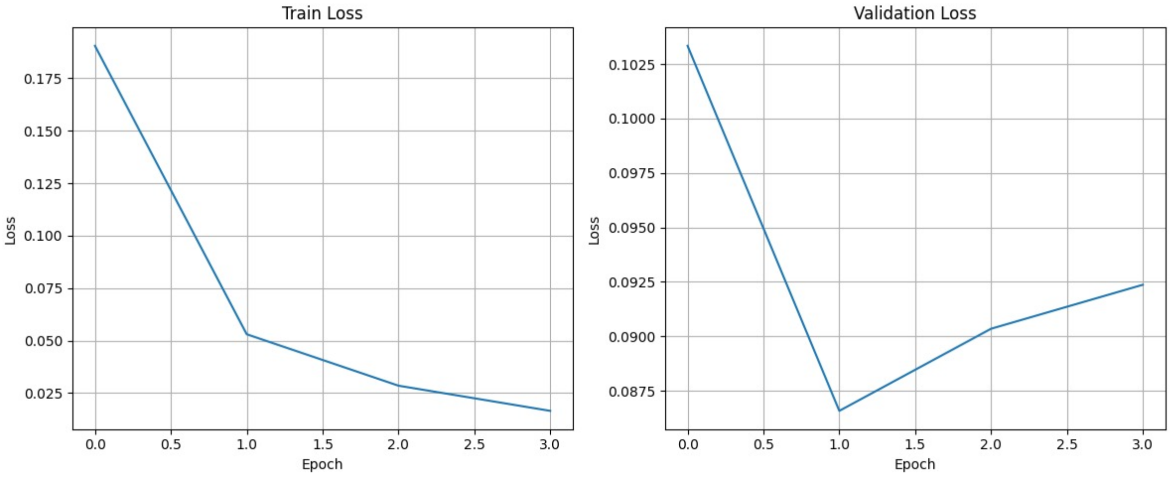
We used the "phishingemails" dataset from Kaggle, which consists of labeled emails categorized as either phishing or non-phishing. The dataset was split into training, validation, and test sets, with an 80-10-10 split ratio. We applied several pre-processing techniques to balance the classes and improve model generalization, such as removing links, repeating special characters and unreasonably long words. We've also optimized the maximum the sequence length for training length performance.

### 3.2 Workflow

Our workflow consisted of using the pre-trained BERT model with an added classification layer and training it (while maintaining some weights as no-decay) on the email database. After conducting the hyperparameter search, we trained the optimized model on the dataset, validated it during training using the validation set, while trying not to overfit the model in order to improve generalization.

### 3.3 Results

Our model achieved an accuracy rate of 98% on the test dataset, in addition to highlighting critical words that contributed to the model's decision-making process.



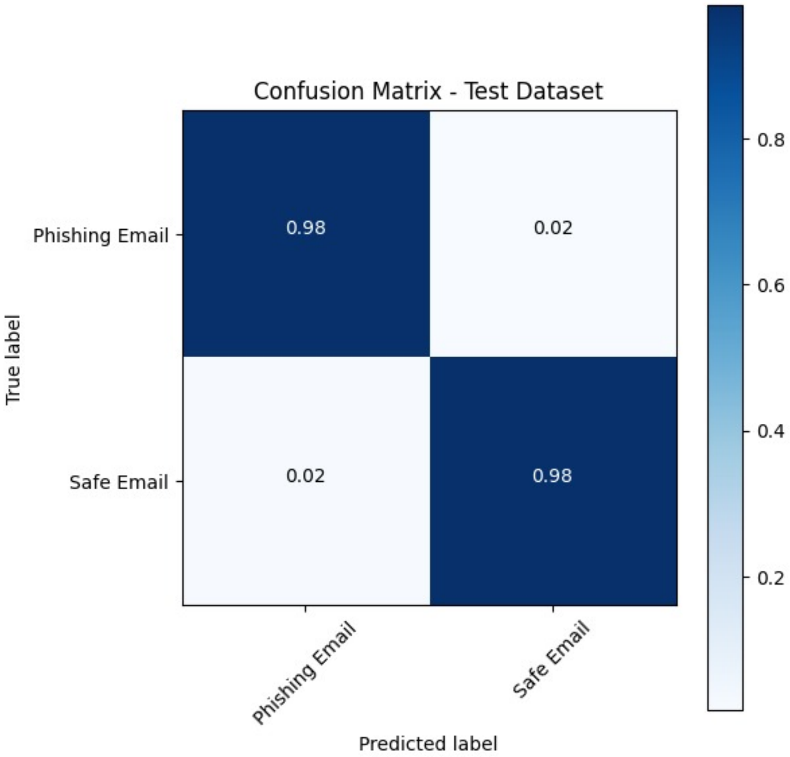


*Figure 3 – prediction confusion matrix*

Out of 1312 datapoints in the test dataset, the model successfully classified 1286 datapoints. The false-negative and false-positive rates are very comparable (11 false-negatives and 15 false-positives), which suggests strong confidence in the predictions as well as a strong generalization ability.

## 4. Alternatives Solutions

After finalizing the work on the BERT-based transformer, we used an SVM-based approach with a TF-IDF algorithm to handle text-based input using sklearn based tools. We've reached a 97% accuracy on the test dataset, which is comparable with the BERT-based model. This is notable – natural language processing can certainly be done with more "classical" machine learning algorithms. Future work could include comparisons between the algorithms on different databases, learning their (possibly) distinct use cases and differentiating where they each excel. Furthermore, we could try to incorporate the TF-IDF method perhaps as an exit layer in a neural network of a pretrained transformer, perhaps with positive results.

## 5. Conclusion and Future Work

In this report we've demonstrated the effectiveness of using transformers on natural language processing contexts on an email classification task. Our results show that transformers can be very reliable in this context, and our experimentation with the trained model led us to believe there are many different enhancements and configurations to be explored with it, even without training it again.

Furthermore, we were impressed with the versatility of the pre-trained BERT model – many of the model's parameters were frozen during training, but the model still performed extremely well on the database and other text we've given it to process. This means the time-to-market and cost of training with different applications of transformers can both be greatly reduced by using pre-trained models with different enhancements.

In conclusion, our results indicate that transformer models are well-suited for this task, offering both high accuracy and explainability. As for possible avenues for future work, we'd like to explore how we might provide better support for privacy by hashing the email before and unhashing them after the model receives them, and how would that affect the model's performance and training regime. Additionally, given the model's performance on this given task, we'd like to expand its role as a general email AI assistant – providing context clues, directing users towards important passages and generating emails for the users itself.

## 5. References

- **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017).** Attention is All You Need. *Advances in Neural Information Processing Systems*, 30

  This paper introduces the Transformer architecture, which serves as the foundation for models like BERT.

- **Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019).** BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT 2019*, 4171-4186.

  This paper describes the BERT model used in our project for text classification tasks.
- **Kaggle Phishing Emails Dataset.**

  The dataset used in our experiment, available here.