Phishing Email Detection Using BERT-Based Models

Sama Balum, samabalum@mail.tau.ac.il, Department of Industrial Engineering, TAU
Joseph Glyanos, glyanos@mail.tau.ac.il , Department of Industrial Engineering, TAU
Yakov Elisian, yakove@mail.tau.ac.il , Department of Industrial Engineering, TAU

1. Introduction

Emails are integral to daily communication, making them prime targets for phishing attacks. Phishing emails deceive recipients into revealing sensitive information, leading to financial losses and data breaches. Traditional detection methods struggle with the evolving tactics of attackers, necessitating more sophisticated approaches.[1]

Researchers have explored phishing detection for years, moving from rule-based systems to advanced NLP techniques. Our research focuses on using TinyBERT and DistilBERT, which excel at analyzing email content and context. BERT based models capture contextual relationships, and offer high accuracy with greater efficiency. These models are well-suited for detecting phishing emails due to their ability to understand language nuances.[2]

In this paper, we explore how these NLP models can be applied to classify emails as phishing or legitimate. We will discuss compressing techniques as well, and present our findings on their effectiveness in detecting phishing emails.

2. Methods

In this section, we describe the methods used for classifying emails into spam and non-spam categories. The following subsections detail our dataset, the specific models used, the preprocessing steps applied to the email data, and the training and evaluation procedures.

2.1. Dataset Overview

The dataset was compiled by researchers to study phishing email tactics. It combines emails from a variety of sources to create a comprehensive resource for analysis. This dataset contains approximately 82,500 emails where there are 42,891 spam emails and 39,595 legitimate emails as shown in the histogram above.[3]

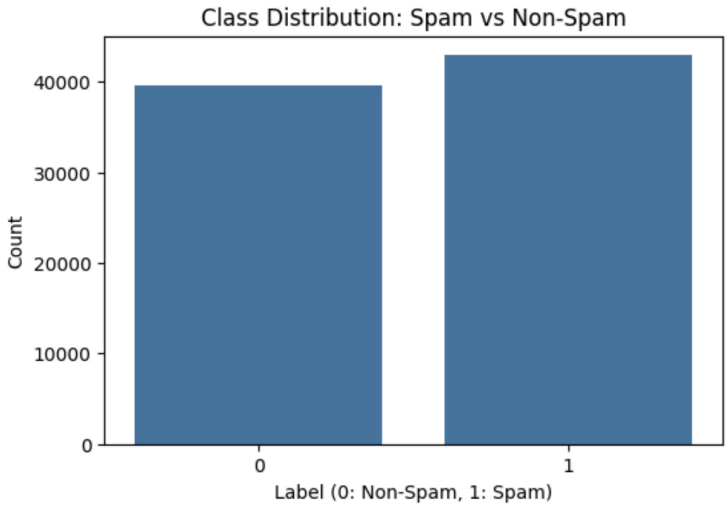

Figure 1: Distribution Of Emails Class

2.2. Pre-trained Transformer Models

To address the email classification task, we employed two transformer-based models: DistilBERT and TinyBERT, each offering distinct advantages in performance and computational efficiency:

- **DistilBERT:** A streamlined version of BERT, DistilBERT retains 97% of BERT's language understanding while being 60% faster and requiring 40% less memory. It strikes an optimal balance between accuracy and computational speed, making it ideal for scenarios where both performance and efficiency are critical.[4]
- **TinyBERT:** Even more compact, TinyBERT is specifically tailored for resource-constrained environments. Despite its reduced size, it effectively maintains much of BERT's original performance, with a significant reduction in inference time and memory usage. TinyBERT is particularly suitable for applications where minimizing resource consumption is a priority without severely compromising accuracy.[5]

2.3. Data Preprocessing

The email data was preprocessed to ensure compatibility with the transformer models. The preprocessing steps included:

- **Tokenization:** We utilized the respective tokenizers for DistilBERT and TinyBERT to convert email texts into tokenized inputs that the models can process. This step involves breaking down the text into subword tokens and padding sequences to a consistent length.
- **Text Cleaning:** The email texts were cleaned by converting them to lowercase, removing stop words, and performing stemming. This preprocessing helped standardize the text and reduce noise.

2.4. Model Fine-tuning

Each model was fine-tuned on the email classification task using a supervised learning approach:

- **Training:** The models were fine-tuned on the labeled email dataset, with a training procedure that included the optimization of the cross-entropy loss function. Gradient accumulation and mixed precision training were employed to manage large batches and enhance computational efficiency.
- **Evaluation:** We evaluated the performance of the models on a held-out test set. The following four metrics were used to check model performance:

  - $accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives}$
  - $precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$
  - $recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$
  - $f1\ score = \frac{2 * precision * recall}{presicion + recall}$

2.5. Implementation Details

- **Hardware and Software:** The experiments were conducted using a standard computing environment with GPU and TPU support to expedite training and evaluation. The models were implemented using the Hugging Face Transformers library and PyTorch.
- **Hyperparameters:** The models were trained with a learning rate of 2e-5 and a batch size of 16. The learning rate was adjusted using a linear scheduler with warm-up steps.

2.6. Model compression

Model compression techniques such as pruning, quantization, and knowledge distillation are key strategies for making large deep learning models more efficient, here are the methods we used to compress distilbert:

- **Pruning:** involves reducing the number of parameters in a neural network by removing less important weights or entire neurons. This is done by identifying and eliminating parameters that have minimal impact on the model's predictions, effectively "trimming" the model.
- **Knowledge distillation:** is a technique where a smaller, simpler model (called the "student") is trained to replicate the behavior of a larger, more complex model (called

the "teacher"). The student model learns not just from the ground truth labels but also from the teacher's predictions, which can provide richer information.

- **Quantization:** reduces the precision of the numbers used to represent a model's parameters, typically from 32-bit floating-point (float32) to 16-bit floating-point (float16) or even 8-bit integers (int8).

3. Results and Discussion

We trained a TinyBERT and Distilbert models to classify emails as either spam or not spam. The training process involved fine-tuning the pre-trained models on our labeled email dataset. The dataset was split in 80/20 ratio into training and test (validation) sets to monitor the models performance and prevent overfitting.

3.1. TinyBERT

Training and Validation Loss: The training and validation loss curves (appendices 1) show a consistent decrease over the epochs, indicating that the model is learning effectively. The training loss decreased sharply initially and then leveled off, while the validation loss followed a similar trend but with a less steep decline, suggesting good generalization to unseen data.

Classification Performance: The classification report (appendices 2) provides detailed metrics for the model performance. The model achieved a precision, recall, and F1-score of 0.99 for both 'Not Spam' and 'Spam' classes, with support of 16498 instances. This high level of performance demonstrates the model's effectiveness in distinguishing between spam and non-spam emails.

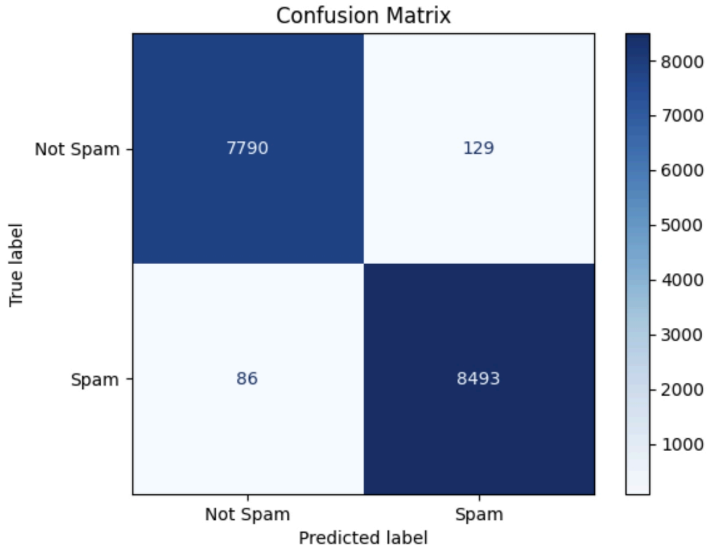

Figure 2: Confusion matrix for Test set - TinyBERT

	Precision	Recall	f1-score
Not Spam	0.99	0.98	0.99
Spam	0.99	0.99	0.99

Table 1: Performance of TinyBERT

3.2. DistilBERT

While TinyBERT offers significant advantages in resource-constrained environments, DistilBERT provides a compelling alternative that balances efficiency with high performance

Training and Validation Loss: The training and validation loss curves (appendices 3) for DistilBERT show a similar trend to those of the BERT model. The training loss decreases sharply in the initial epochs and then levels off, while the validation loss also decreases but at a slower rate. This indicates that the DistilBERT model is effectively learning from the training data and generalizing well to the validation data.

Classification Performance: The classification report (appendices 4) for DistilBERT shows high performance metrics, with precision, recall, and F1-score all close to 0.99 for both 'Not Spam' and 'Spam' classes. The support for each class is consistent with the dataset used, demonstrating that DistilBERT is also highly effective in distinguishing between spam and non-spam emails.

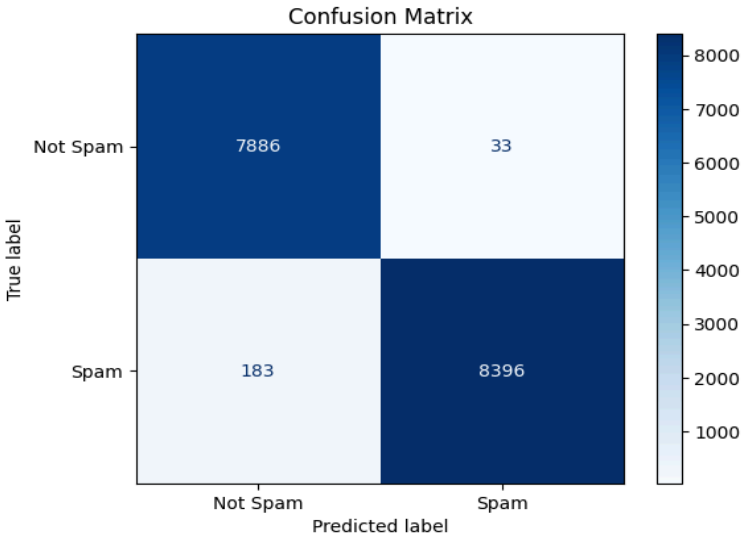

Figure 3: Confusion matrix for Test set - DistilBERT

	Precision	Recall	f1-score
Not Spam	0.98	1	0.99
Spam	1	0.98	0.99

Table 2: Performance of DistilBERT

What can be observed from these two models is that the results are very impressive, and they are capable of providing particularly high accuracy rates. If they had not succeeded, we might have tried to create another model (such as XGBoost, for example) and feed it the prediction results of the model along with the other features we generated during the EDA. However, this was not necessary due to the impressive results.

Both models perform at a high accuracy, but DistilBERT shows slightly better performance in classifying Not Spam emails, with fewer False Positives (33 vs. 129). TinyBERT, on the other

hand, performs slightly better in classifying Spam emails, with fewer False Negatives (86 vs. 183).

3.3. Model Compression

Three compressions were done for the DistilBERT model. Pruning, Knowledge distillation and Quantization. The performance results are shown in the following table.

	Training accuracy	Training F1 score	Training loss	Test accuracy	Test F1 score	Test loss	epochs
Pruned	0.9988	0.9988	0.0041	0.9930	0.9930	0.0278	7
KD	-	-	-	0.9869	0.9869	0.0453	1
Quantization	0.9898	0.9898	0.0442	0.9796	0.9796	-	1

Table 3: Performance of compressing the DistilBERT model

Pruning was performed over seven epochs, while knowledge distillation and quantization were limited to one epoch due to runtime disconnection issues. The results indicate high performance across both training and test sets, with the pruned DistilBERT model achieving the highest scores and quantization the lowest. This discrepancy can be attributed to the fact that the pruned model was trained for seven epochs. Additionally, some values in Table 3 are missing due to these runtime disconnection issues.

4. Conclusion

In this project, we classified emails into spam and non-spam categories using TinyBERT, DistilBERT, and compressed variants of DistilBERT. Our results indicate that TinyBERT excels in identifying spam emails, leading to a lower rate of false negatives. Conversely, DistilBERT is more effective at classifying non-spam emails, resulting in fewer false positives. Among the compressed models, the pruned DistilBERT demonstrated the highest performance.

The choice between TinyBERT and DistilBERT should be guided by the application's specific needs—whether prioritizing the reduction of false positives or false negatives. Additionally, model compression does not significantly impact performance in our results, suggesting that compressed models can achieve comparable effectiveness while offering potential benefits in terms of efficiency.

5. Reference

[1] Al-Subaiey, A., Antora, K. F., Al-Thani, M., Khandakar, A., Alam, N. A., & Zaman, S. A. U. (Year). *Novel Interpretable and Robust Web-based AI Platform for Phishing Email Detection*. 2405.11619 (arxiv.org)

[2] Lee, Y., & Saxe, J. (2020). *CATBERT: Context-Aware Tiny BERT for Detecting Social Engineering Emails* (A Preprint). arXiv. https://arxiv.org/abs/2010.03484.

[3] Al-Subaiey, A., Al-Thani, M., Alam, N. A., Antora, K. F., Khandakar, A., & Zaman, S. A. U. (2024, May 19). Novel Interpretable and Robust Web-based AI Platform for Phishing Email Detection. ArXiv.org. https://arxiv.org/abs/2405.11619

[4] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv. https://arxiv.org/abs/1910.01108

[5] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). *TinyBERT: Distilling BERT for Natural Language Understanding*. arXiv. https://arxiv.org/abs/1909.10351

6. Appendices

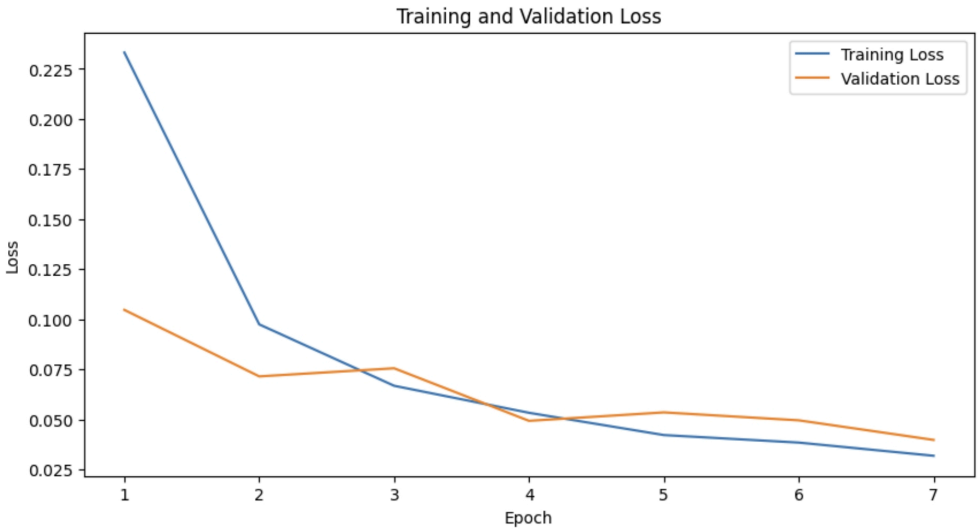

Figure 1: TinyBERT Training and Validation Loss over Epochs

Classification Report:

	precision	recall	f1-score	support
Not Spam	0.99	0.98	0.99	7919
Spam	0.99	0.99	0.99	8579
accuracy			0.99	16498
macro avg	0.99	0.99	0.99	16498
weighted avg	0.99	0.99	0.99	16498

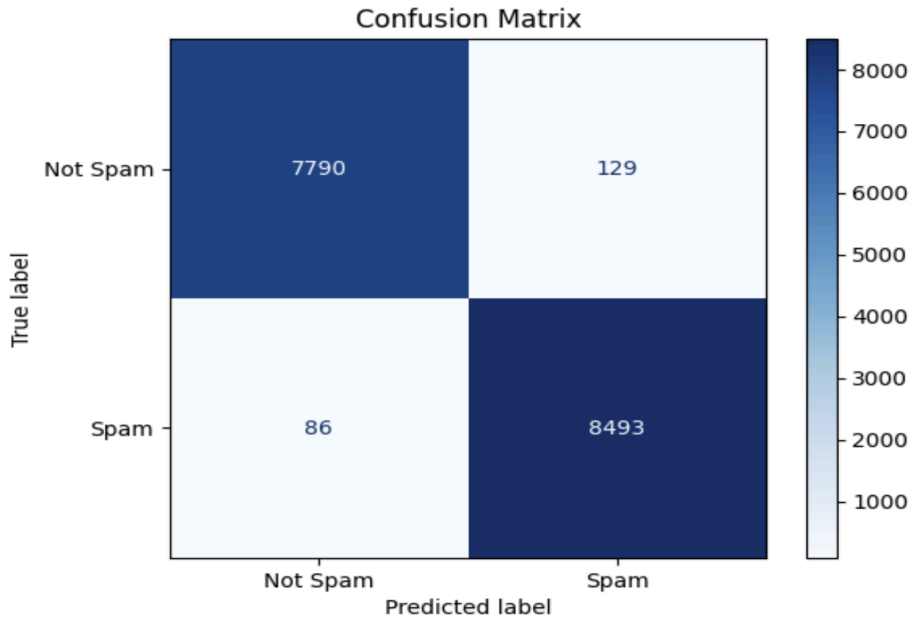

Figure 2: TinyBERT Classification Report

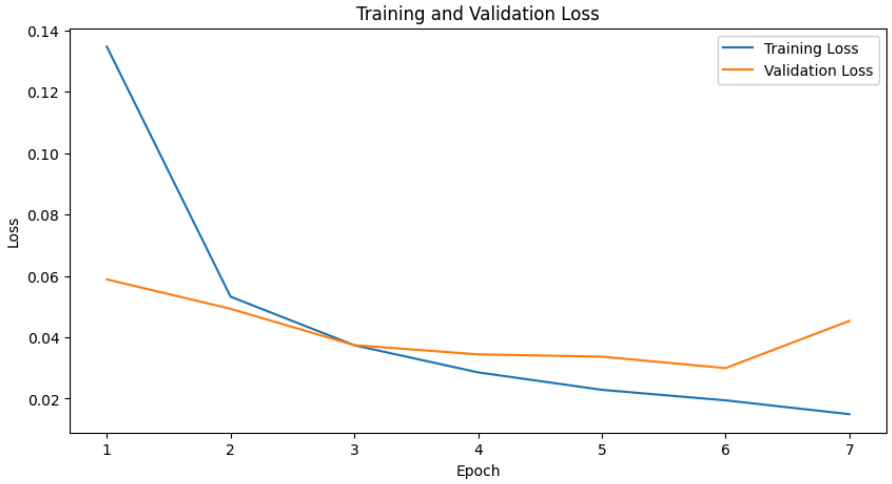

Figure 3: DistilBERT Training and Validation Loss over Epochs

```
Classification Report:
              precision    recall  f1-score   support

    Not Spam       0.98      1.00      0.99      7919
        Spam       1.00      0.98      0.99      8579

    accuracy                           0.99     16498
   macro avg       0.99      0.99      0.99     16498
weighted avg       0.99      0.99      0.99     16498
```

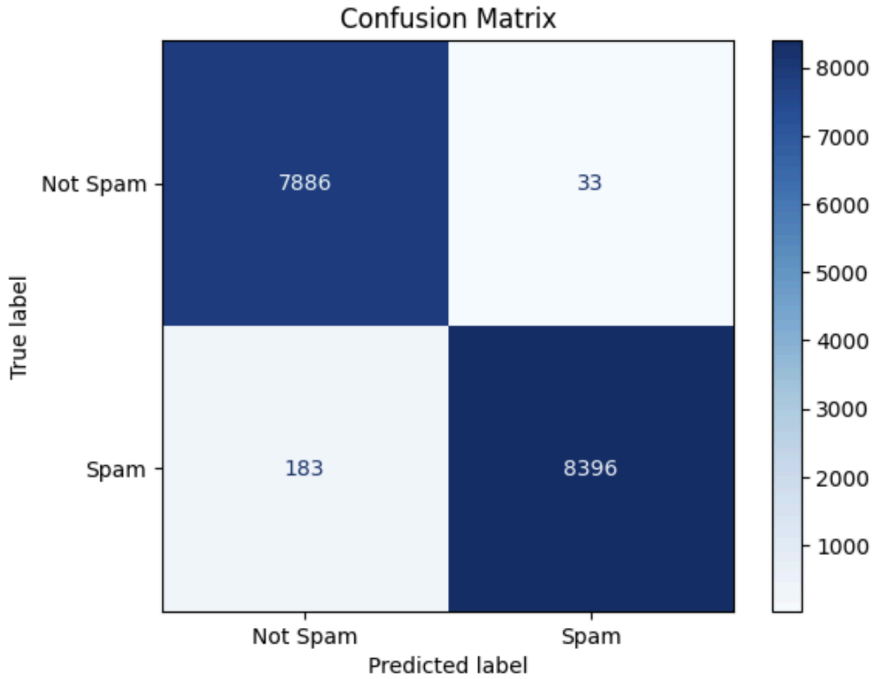

Figure 4: DistilBERT Classification Report