PHISHING EMAIL DETECTION SYSTEM USING BERT

Lee Martins
Student# 1008866835
leah.martins@mail.utoronto.ca

Asmita Chandra
Student# 1009051339
asmita.chandra@mail.utoronto.ca

Riya Kapoor
Student# 1009030558
riya.kapoor@mail.utoronto.ca

Joonseo Park
Student# 1008819281
joonseo.park@mail.utoronto.ca

ABSTRACT

This document describes the implementation of our **Bi-directional Encoder Representations from Transformers (BERT)** model, and our interpretation of its results for a phishing email detection system. – Total Pages: 9

1 INTRODUCTION

As personal and professional interactions continue to migrate online, cyberattacks in the form of phishing emails pose an increasing threat. The project's primary goal is to develop a deep-learning based system capable of differentiating between legitimate emails and deceptive "phishing" emails. This project directly addresses real-world cybersecurity threats with potential applications in the corporate industry. With the evolving nature and frequency of phishing emails - the importance of our project becomes apparent. A deep learning model is well suited as it can recognize subtle cues in email content while continuously learning from the patterns present in latest phishing methods. In contrast, traditional rule-based detection systems fail to recognize these subtle linguistic details. This is more than a project but moreover, a powerful tool to enhance digital security.

1.1 ILLUSTRATION

Figure 1, below illustrates the overall architecture of our email phishing detection project.

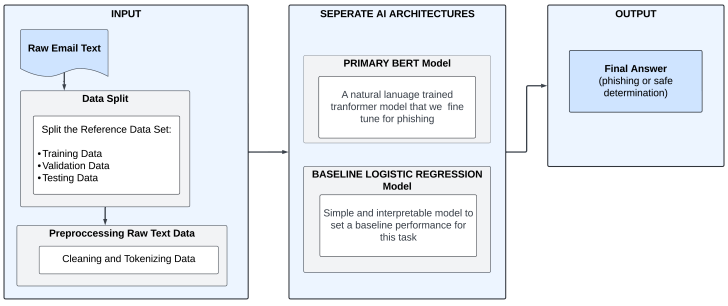

Figure 1: Explicit diagram of our AI model highlighting the BERT model

The diagram outlines the main stages of our workflow and shows our end to end process between the original email to the models phishing prediction. Note that further detail of each stage will be provided throughout this document.

2 Background and Related Work

To provide context and accuracy references for our phishing email detection system, we reviewed several related works:

1. **CNN-Based Phishing Detection Kumar et al. (2024):** This work uses a CNN–LSTM hybrid to detect phishing emails while mitigating overfitting. Although faster to train, it underperforms compared to BERT.
2. **Machine Learning Approach for Email Phishing Detection Al-Shabi et al. (2024):** This study presents a phishing detection pipeline using email header and body text features, showing that combining them improves accuracy for classifiers like LR and SVM.
3. **In-Depth Analysis of Phishing Email Detection Nguyen et al. (2025):** This study compares baseline phishing detection models, highlighting how datasets of various sizes impact performance.
4. **Phishing Email Detection Model Using Deep Learning Alqahtani et al. (2023):** The authors evaluate several popular deep learning models (CNNs, RNNs, LSTMs, hybrid BERT + LSTM model) and compare their classification accuracy. Hybrid BERT's accuracy from this journal was 99.61%.
5. **What is the BERT Model and How Does it Work Schulze (2025):** This article overviews BERT, the bidirectional transformer model for NLP, explaining its encoder architecture, MLM training, pre-training data, and modern efficiency.

3 Data Processing

Below includes data cleaning statistics, data tokenization, and accounts for our new data.

3.1 Data Sources

Our model is trained on four diverse datasets from Kaggle and University of Twente research:

1. Phishing No More Dataset (Kaggle), Labels: 1 = phishing, 0 = not phishing
   - Fields: body, sender, recipient, date, URLs, label Alam & Khandakar (2024)
2. University of Twente Dataset (Academic), Labels: 1 = phishing, 0 = not phishing
   - Fields: email text, email type Miltchev et al. (2024)
3. Cybercop Dataset (Kaggle), Labels: "phishing email" or "safe email"
   - Fields: email text, email type Chakraborty (2023)
4. Human-LLM Phishing Emails (Kaggle), Labels: 1 = phishing, 0 = not phishing
   - Fields: text, label Greco (2024)

3.2 Data Pre-Processing

1. **Source Collection:** All `.csv` files were loaded from a shared directory.
2. **Corruption Filtering:**
   - Files that failed to load due to encoding or malformed rows were skipped.
3. **Standardization of Columns:**
   - Column names such as `"Email Text"` were renamed to `"body"`.
   - Column names such as `"Email Type"` were renamed to `"label"`.
   - Labels `"phishing"/"safe"` were changed to binary: 1 = phishing and 0 = safe.
   - If the `urls` column was missing, it was added based on the presence of hyperlinks.
4. **Column Filtering:** Only the essential columns were retained: `body`, `urls`, and `label`.
5. **Output Format:**

- Each cleaned dataset was saved as a separate sheet in a consolidated Excel file.
- All sheets are concatenated into a single DataFrame named `phishing_df`.

6. **Final Cleaning Steps:**
   - Rows with missing values were dropped.
   - Data was shuffled randomly with a fixed seed (`random_state=42`).

After data cleaning, the final dataset contains a total of 67,126 samples, consisting of 36,604 phishing emails and 30,522 safe emails. Of these, 38,531 emails contained at least one URL, while 29,941 contained no URLs. These statistics are illustrated in Figure 2 below.

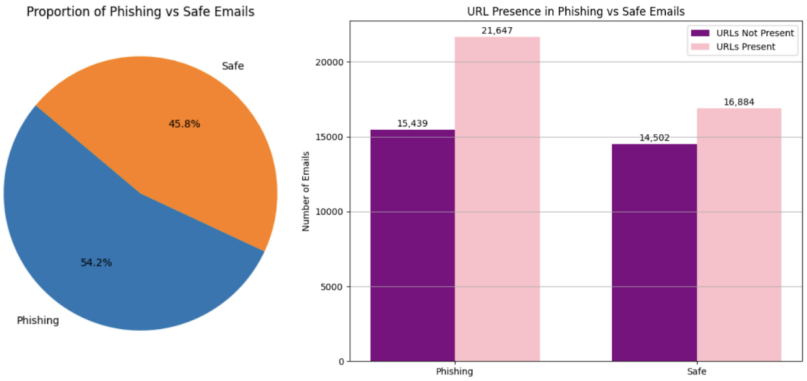

Figure 2: Distribution of phishing vs. safe emails, and URL presence across the dataset.

The cleaned data can be observed from the data frame below, which displays what the first five rows of our cleaned dataset look like.

body	urls	label
They look and feel exactly like the real thing. …	1	1
CNN Alerts: My Custom Alert …	0	1
Ranked #1 Men's Supplement by GQ in 2007, discover the secrets of thousands here http://www.risepat.com/	1	1
No girl will now resist a temptation to go with you! …	1	1
Dear Sarah, I hope this email finds you well. My name is Jennifer Smith, and I'm thrilled to offer you an …	1	0

Figure 3: Depiction of the samples in the dataframe head from our cleaned data

3.3 Data Split and Tokenization

The data is split into 70% training, 15% validation, and 15% testing. All training and validation samples come from outsourced data. Testing combines the remaining outsourced data (85%) with newly collected data (15%). A pre-trained BERT tokenizer converts each email into token IDs and corresponding attention masks. Text is lowercased, split into WordPiece tokens, truncated or padded to 128 tokens, and returned as PyTorch tensors. Each dataset sample comprises these token IDs, attention masks, a URL presence flag (binary), and its classification label (binary).

4 Architecture

The primary model for this project leveraged a customized BERT classifier built upon a transformer-based neural network known as DistilBERT (distilbert-base-uncased) Hugging Face (2025), a compressed version of BERT which maintains performance while reducing training time and number of trainable parameters of the model. This custom built classifier allows us to modify the model beyond the pre-trained default model. The model processes our preprocessed, data and training is performed over 3 epochs with a batch size of 8. The model's layers are defined in the list below.

1. **Forward Pass**
   - BERT Encoder: Custom neural network classifier that is pre-trained for encoding text input and uses a Hugging Face Transformer to handle case-sensitive text
   - Dropout Layer: dropout_layer = 0.3
   - Linear Mapping: CLS token (an embedded token from the input) mapped to logit score (a value between 0 and 1 where phishing = 1, safe = 0)
   - Label smoothing: torn.nn.CrossEntropyLoss(label_smoothing=0.1) Added to our cross entropy loss to reduce overconfidence in predictions
2. **Optimizer**
   - AdamW Optimizer: learning_rate = 1e-5, weight_decay = 2e-5
   - AdamW optimizer adds adaptive learning rates
3. **Scheduler Step:** Initially starting with a low learning rate, scheduler gradually increases learning rate by 10%
4. **Track Training Loss:** Outputs error and loss each epoch
5. **Early Stopping:** Based on the error and loss values, either continue with the model or stop training if loss is increasing

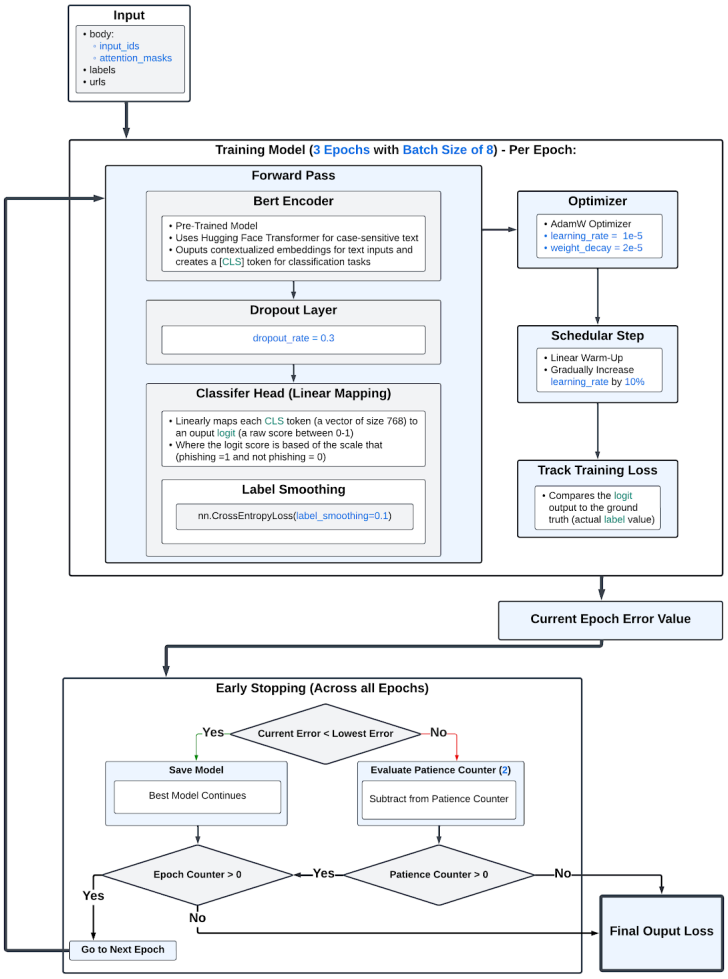

Figure 4: AI Model Architecture for BERT, (hyper-parameters are in blue text).

5 BASELINE MODEL

A baseline Binary Logistic Regression (BLR) model was implemented as it is effective and widely used for binary classification tasks. It predicts the probability that an email is phishing (class = 1) or safe (class = 0) by applying the sigmoid function to map outputs between 0 and 1, using a 0.5 threshold for final classification.

BLR was chosen as the baseline because it outputs normalized probabilities (between 0–1) that easily convert to labels, providing a clear, linearly separable decision boundary. Unlike linear regression, BLR is resistant to outliers, since it does not fit a line through all data points. Using the implementation seen in Figure 5, this model achieved a validation accuracy of 98.44% and test accuracy of 94.1%, indicating strong performance. The high accuracy suggests that with a straightforward BoW representation, the model effectively captures phishing patterns based on repeated word and phrase usage within the dataset.

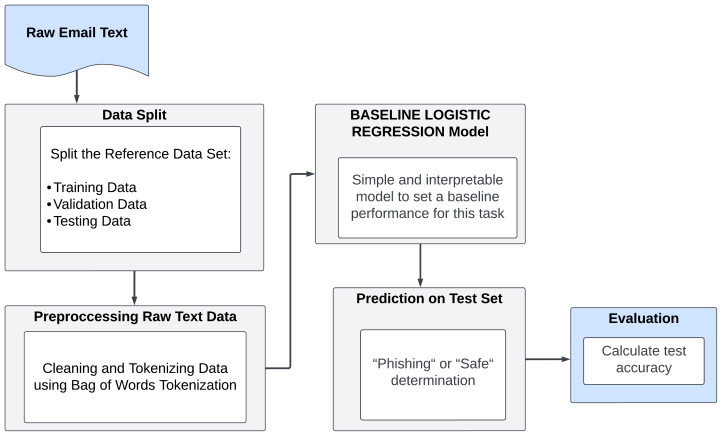

Figure 5: Baseline model flow chart

6 QUANTITATIVE ANALYSIS

Below are numerical results that indicate how our primary model performed:

Table 1: Performance metrics for training and validation datasets

	Training Dataset			**Validation Dataset**
	Ep. 1	**Ep. 2**	**Ep. 3**	
Model Accuracy (%)	99.53 (Final Accuracy)			98.98
False Positives (%)	2.83	0.61	0.39	0.56
False Negatives (%)	1.93	0.26	0.08	0.46

The DistilBERT model slightly loses its accuracy as it proceeds from training to validation. The high accuracy at the end of training (<0.5% of samples were misclassified) likely results from learning patterns unique to the training dataset. There is only 0.55% less accuracy on the validation dataset, suggesting that the model is generalizing quite well during validation. The model more often misclassified safe emails as phishing, as there are consistently more false positives.

Table 2: Model performance over training and validation datasets

	Precision	Recall	F1 Score
Training	0.9929207581639644	0.9985837320574162	0.9957441935914807
Validation	0.9897035327534174	0.9916399857701885	0.9906708129720125

The generally high (>98%) precision, recall, and F1 score produced on the training and validation datasets suggest the model's reliability on positive (phishing) samples.

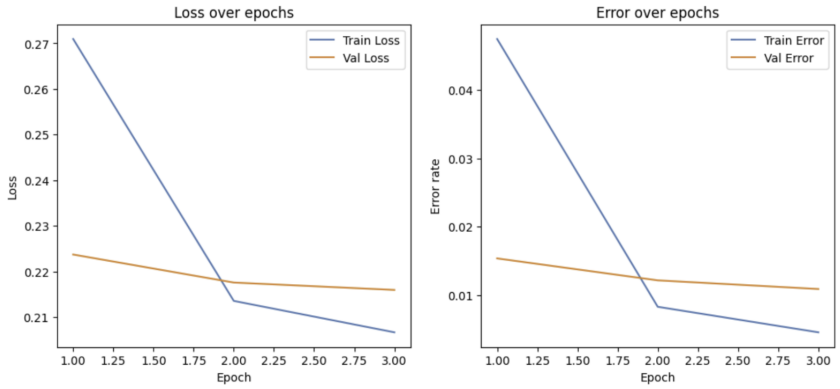

Figure 6: DistilBERT validation & training error and loss over 3 epochs

The model's loss and error decreases far more rapidly during training than during validation, likely a result of weights being changed to directly match training samples (memorization of features).

7 Qualitative Analysis

Examples of TP, TN, FP, and FN predictions are described below.

- **False Positive Example:** Ground Truth = Safe, Model Output = Phishing
  - **Email:** urgent! ... vacancy urgent!... please do not send in applications yet...
  - **Reasoning:** Repetition of words such as "urgent" and "application" and lack of email signature may have led to misclassification as phishing.
- **False Negative Example:** Ground Truth = Phishing, Model Output = Safe;
  - **Email:** >>> Seems in addition to wasting CPU resources its a drive hog. >>> So I put in a warning line in dot bashrc ... >> >> This will fix it ... sudo rpm -e $(rpm...
  - **Reasoning:** Casual conversational language may have contrasted with the promotional diction of other phishing emails. Far more URLs were also present in phishing email samples than not (see figure 2), likely contributing to this incorrect classification. The >> operator is also absent in many phishing emails.
- **True Positive Example:** Ground Truth = Phishing, Model Output = Phishing
  - **Email:** Adobe Creative Suite 3 Master Collection for Win http://easystoreoem.com Features: Professional page layout, image editing, vector illustration, and print production...
  - **Reasoning:** The URL does not match the Adobe domain name. There is also no personalization nor sender contact information, only generic promotional diction.
- **True Negative Example:** Ground Truth = Safe, Model Output = Safe
  - **Email:** anshuman neil , i would like to apologize for the confusion regarding anshuman ... research enron corp .1400 smith street room ebl 962 houston , tx 77002 - 7361 phone: (713) 853 3848 email : vkamins@enron.com

– **Reasoning:** The tone is direct as evident by words such as “i” and “apologize”. Specific contact information is given. There are also no clickable URLs present.

8 Evaluate on New Data

To ensure our model is evaluated on new data, team members manually collected new emails from personal Gmail and Outlook inboxes to compose 15% of the testing data we would use for our evaluation, as mentioned in Data Sourcing. Our testing data resulted in a test accuracy of 94%, where False Positives and False Negatives made up 4.37% and 1.73% of the total predictions respectively. While 94% is not as high of an accuracy as we were seeing in validation (98-99%; see section 6), we can still consider this as a high accuracy value for evaluation purposes as it compares well to the 94% we achieved in our baseline model (see section 5). We can further interpret our results with the following metrics:

- **Bert Precision = 92.3%** - meaning 92.3% of the phishing predictions were correct. 7.7% were false positives (non-phishing emails mistakenly flagged as phishing).
- **Bert recall = 96.5%** - meaning 96.5% of the real phishing emails were identified, meaning it missed about 3.5% of phishing emails (false negatives).

The 94% accuracy meets our expectations, the fact that our model predicts phishing emails more accurately than non-phishing emails is reassuring. Misclassifying phishing emails as safe poses a higher security risk, whereas mistakenly labeling legitimate emails as phishing is less harmful, mostly causing inconvenience without exposing users to malicious content. In addition to these statistics, we wanted to qualitatively demonstrate how accurately our model would be able to make phishing/non-phishing predictions on new data. Shown below are two examples of how the model made predictions on emails that were collected from team members’ family members, and were not included in any prior evaluation.

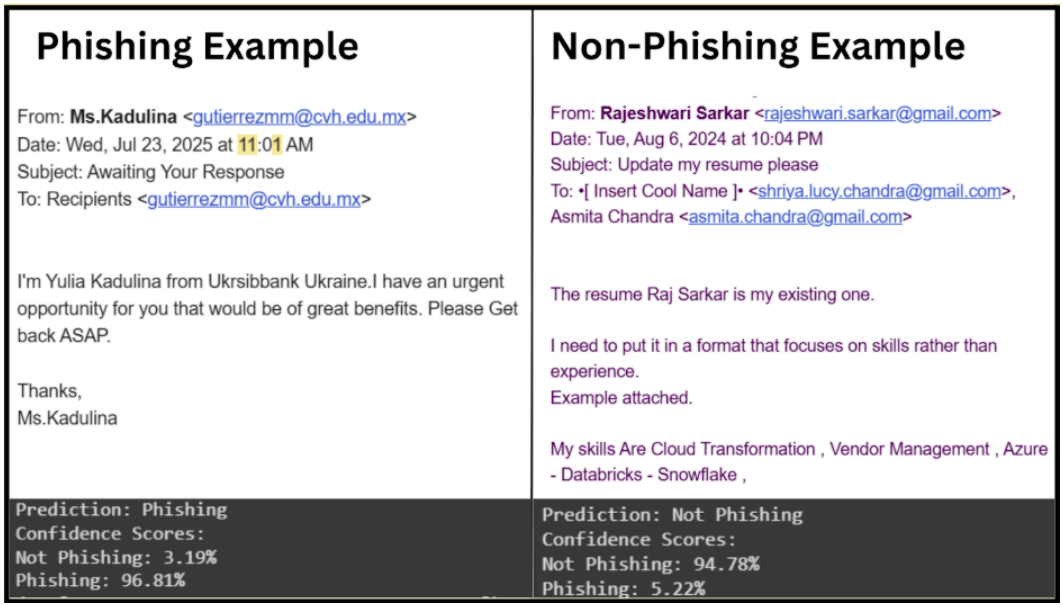

Figure 7: Side by side comparison of phishing and non-phishing emails (new unseen data)

9 Discussion

Our model achieved high accuracy on both the training and validation datasets, effectively distinguishing phishing emails from legitimate ones. However, as seen in Figure 8, testing accuracy was slightly lower, an expected drop when evaluating unseen data. Likely contributors include: potential manual labeling errors, and the domain-specific nature of inbox samples, which differ from the training data.

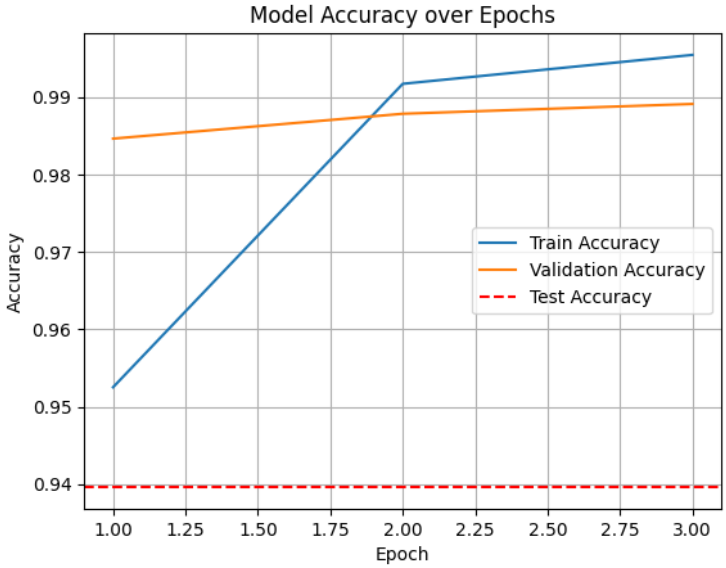

Figure 8: DistilBERT accuracy comparison across all sets of data

These results exceeded initial expectations. Given transformer-based models like DistilBERT were introduced late in the course, we anticipated a steep learning curve with modest performance. The model trained successfully on the first attempt and achieved competitive results with larger architectures in related work (see section 2), while offering faster runtimes and reduced overfitting risk.

Unexpectedly, our baseline BLR model achieved similar accuracy on the test dataset to DistilBERT. BLR analyzes high-frequency patterns in the data while DistilBERT excels at generalizing complex semantic relationships. This suggests, for our dataset, simpler models can match accuracies in complex architectures, performing remarkably well. Nonetheless, transformers like DistilBERT retain a critical advantage in handling nuanced, varied, and evolving phishing strategies crucial for robust real-world deployment.

The first thing we learned from this project was how split ratios affect model performance; adjusting the split from 70-20-10 to 70-15-15 produced an approximately 2% accuracy difference for both models. We then also witnessed the benefits of parameter and memory efficiency; despite DistilBERT's reduced size, it delivered comparable accuracy to larger models with faster training and lower overfitting risk. Finally, we expanded our knowledge of overfitting mitigation beyond standard techniques taught in class (e.g., L2 regularization, batch normalization) by applying strategies such as: changing our base model, label smoothing, employing learning rate schedulers, and early stopping.

Future implementations would classify emails into more categories (phishing, shopping adverts, receipts, etc). Technical improvements would include incorporating more overfitting mitigation strategies that were not implemented due to time restrictions such as data augmentation and POS tagging.

10 ETHICAL CONSIDERATIONS

The following describes ethical considerations during data collection, model training, and model usage:

1. **Bypassing Phishing Filters:**
   - Attackers could test their phishing emails against our model to find and exploit weaknesses, bypassing detection
   - **Improve by:** Regularly updating our test data and tune the model to handle evolving phishing strategies.
2. **Privacy and Data Protection:**
   - Deployment requires the model to process sensitive and confidential personal information which risks misuse and bias when features or labels are extracted from personal emails.
   - **Improve by:** Using anonymized, consented training data.
3. **Representation Bias in Training Data:**
   - Dataset was sourced from a limited number of resources, which possibly all come from specific organizations , languages or even email styles.
   - A limited dataset may under-represent emails from other countries, organizations, languages, or email styles, causing accuracy drops for unfamiliar phishing attempts.
   - **Improve by:** Diversifying sources, adding country-specific examples, and monitoring performance across demographics.
4. **Fairness and Disparate Impact:**
   - A model trained on a limited dataset can unintentionally correlate phishing emails with senders of certain demographics, causing higher false positives for some groups.
   - **Improve by:** Conducting regular fairness audits on error rates across users.
5. **Model Misuse for Disruption:**
   - Malicious operators tuning or misusing the model so that it flags legitimate business emails as phishing, thus slowing down operations or damaging trust between partners.
   - **Improve by:** Lock the weights and hyperparameters in our BERT model to avoid malicious tampering.

COLAB LINK

The Google Colab notebook for this project can be accessed here: https://colab.research.google.com/drive/14V0_e972xoqFXIJRX6JS1sEyE_CR5CJG?usp=sharing

GITHUB LINK

The GitHub repository for this project can be accessed here: https://github.com/leahmdmartins10/Phishing-Email-Detection-System-BERT.git

REFERENCES

Mohammad Al-Shabi, Abdul Monem Mohammed Ameen, and Mohammed M. Hamza. Email phishing detection using bert-based deep learning model. *Procedia Computer Science*, 235:1580–1587, 2024. doi: 10.1016/j.procs.2024.04.265. URL https://www.sciencedirect.com/science/article/pii/S1877050924034136.

Naser Abdullah Alam and Amith Khandakar. Phishing no more dataset, 2024. URL https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset. Accessed: 2025-07-11.

Abdulaziz Alqahtani, Hamoud Binsalleeh, Ibrahim Alzahrani, Muhammad Khan, and Fawaz Alsolami. Phishing email detection model using deep learning. *Electronics*, 12(20):4261, 2023. doi: 10.3390/electronics12204261. URL https://www.mdpi.com/2079-9292/12/20/4261.

Subhadeep Chakraborty. Cybercop phishing emails dataset, 2023. URL https://www.kaggle.com/datasets/subhajournal/phishingemails. Accessed: 2025-07-11.

Francesco Greco. Human-llm generated phishing and legitimate emails. https://www.kaggle.com/datasets/francescogreco97/human-llm-generated-phishing-legitimate-emails, 2024. Accessed: 2025-07-11.

Hugging Face. Distilbert model documentation. https://huggingface.co/docs/transformers/en/model_doc/distilbert, 2025. Accessed: 2025-08-11.

Pankaj Kumar, Akash Singh, and Ramesh Thakur. Email phishing detection using convolutional neural networks. *2024 International Conference on Computational Intelligence and Smart Communication (CISC)*, 2024. doi: 10.1109/CISC59793.2024.10872755. URL https://ieeexplore.ieee.org/document/10872755.

Radoslav Miltchev, Dimitar Rangelov, and Genchev Evgeni. Phishing email validation dataset. https://research.utwente.nl/en/datasets/phishing-validation-emails-dataset, 2024. Accessed: 2025-07-11.

Thanh Nguyen, Hoang Nguyen, Minh Tran, Lan Pham, and Quang Le. In-depth analysis of phishing email detection. *Applied Sciences*, 15(6):3396, 2025. doi: 10.3390/app15063396. URL https://www.mdpi.com/2076-3417/15/6/3396.

Jessica Schulze. What is the bert model and how does it work? https://www.coursera.org/articles/bert-model, 2025. Updated Jul 23, 2025; Accessed Aug 11, 2025.