Explainable Large Language Model (LLM)-Based Detection of Advanced Social Engineering Tactics in Phishing Emails

AI - Advanced Detection of Textual Employed SE with XAI

Gary Monaghan
X23309946
National College of Ireland

1. System Requirements

1.1 Hardware

Component	Specification
CPU	Intel Core i7-10700KF
GPU	nVidia RTX 4070Ti (12GB VRAM)
RAM	32GB DDR4
Storage	500GB

1.2 Software

Tool	Version
Windows OS	Windows 11 Pro
Python	3.12.10
VS Code	1.100
JupyterLab	4.2

2. Dependencies

"pip install" the dependencies in the table below:

torch 2.3.0	matplotlib
transformers 4.41.0	tokenizers
scikit-learn 1.4	datasets
shap	tensorboard
lime	ipywidgets

All other dependencies and libraries are pulled down when required via the code provided.

3. Project Files

File	Purpose
Generate_normal_emails	Generates “normal” office style emails
Generate_phishing_emails	Generated “phishing” emails
distilBERT_training_standard	Trains a DistilBERT model using standard parameters.
distilBERT_training_enhanced	Trains a DistilBERT model using enhanced parameters.
distilBERT_training_sentences	Trains a DistilBERT model using sentence tokenization parameters.
roBERTa_training_standard	Trains a RoBERTa model using standard parameters.
roBERTa_training_enhanced	Trains a RoBERTa model using enhanced parameters.
roBERTa_training_sentences	Trains a RoBERTa model using sentence tokenization parameters.
gptneo_training_standard	Trains a GPT-Neo model using standard parameters.
gptneo_training_enhanced	Trains a GPT-Neo model using enhanced parameters.
gptneo_training_sentences	Trains a GPT-Neo model using sentence tokenization parameters.
attention_viz	
attention_viz_v2	
model_evaluator_standard	Evaluates 3 model families trained with standard parameters (Robustness/Perturbed)
model_evaluator_enhanced	Evaluates 3 model families trained with enhanced parameters (Robustness/Perturbed)
model_evaluator_sentences	Evaluates 3 model families trained with sentence tokenization parameters (Robustness/Perturbed)
train_emails	Training split of combined normal/phishing synthetic emails.
test_emails	Testing split of combined normal/phishing synthetic emails.
validation_emails	Validation split of combined normal/phishing synthetic emails.
phishing_email	“Phish No More” public dataset.

4. Model Training

Each model family (DistilBERT, RoBERTa, GPT-Neo) was trained under three distinct regimes:

Regime Type	Description
Standard	Baseline training with standard parameters.
Enhanced	Optimised training parameters set.
Sentence Tokenized	Enhanced training with sentence-level tokenisation, implemented via sentence-piece to enhance robustness.

9x dedicated training notebooks, named according to model, training regime, each includes the following:

- Data loading and pre-processing (from local or generated datasets).
- Tokenizer and model instantiation.
- Training using Hugging Face Trainer.
- Evaluation against internal validation subset.
- Saving of model metrics.
- Attention visualiser, with PNGs saved after training.

5. Explainability Integration

Tools used to gauge interpretability/explainability:

Method	Description
LIME (All models)	Local surrogate explanations, implemented via 3x evaluator scripts contained within Jupyter Notebooks.
Attention Rollout (DistilBERT, RoBERTa)	Implemented during training of 6x models attention_viz.py
Attention Rollout (GPT-Neo 125M)	Implemented during training of 3x models via attention_viz_v2.py

attention_viz.py and attention_viz_v2.py perform the following:

- compute_rollout() - multi-layer attention propagation.
- plot_rollout() - token heatmap output with mapplotlib.
- attention_hook - extract and visualise attention for batches.
- attentionCallback - integration with Hugging Face trainer.

6. Evaluation and Robustness

Not standalone perturbation script was used. Perturbation logic is directly embedded within the following three evaluator notebooks:

- model_evaluator_standard.ipynb
- model_evaluator_enhanced.ipynb
- model_evaluator_sentences.ipynb

Perturbations applied:

- **Synonym Substitution** - Performed using WordNet.
- **Punctuation Jitter** - Randomised symbol injection (e.g. commas, periods).
- **Type Injection** - 5% chance per word (keyboard-adjacent errors injected).

These were applied to a post-training subset of emails.
ΔF1is calculated to measure robustness under adversarial interference/noise.

7. Performance Metrics

All 9x models were evaluated using the following:

Metric	**Purpose**
Precision	Specificity of prediction.
Recall	Sensitivity to phishing cues.
F1 Score	Harmonic mean of precision and recall.
ROC-AUC	Class separation capability.
ΔF1	Change in F1 after perturbation.
McNemar Test	Statistical test for model comparison.

Datasets used:

- **Synthetic hold-out split -** Internal evaluation.
- **"Phish No More" -** Public phishing dataset.
- **Perturbed Emails -** Robustness assessment.

8. Reproducibility and Data Handling

- All notebooks are self-contained, with fixed seeds (set via "random_state=").
- Synthetic data and generated emails handled locally.
- Model checkpoints saved via Hugging Face trainer.
- Output visualisations (XAI, metrics) saved to relevant directories, can be user-defined.