Configuration Manual

MSc Research Project
MSc Cybersecurity

Aslam Malik Abdul Azeez
Student ID: x23183098

School of Computing
National College of Ireland

Supervisor: Mr. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Aslam Malik Abdul Azeez		
Student ID:	X23183098		
Programme:	MSc in Cybersecurity	**Year:**	2024-2025
Module:	Practicum		
Lecturer:	Mr. Imran Khan		
Submission Due Date:	29/01/2025		
Project Title:	Leveraging Large Language Models (LLM) for the Detection of Spear-phishing Emails as Indicators of Advanced Persistent Threats (APTs)		
Word Count:	905	**Page Count: 4**	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Aslam Malik Abdul Azeez

Date: 29/01/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Aslam Malik Abdul Azeez
Student ID: x23183098

Leveraging Large Language Models (LLM) for the Detection of Spear-phishing Emails as Indicators of Advanced Persistent Threats (APTs)

1. System Configuration

1.1 Hardware Requirements

- **Processor:** Quad-core processor (Intel i5 or AMD Ryzen equivalent).
- **Memory:** Minimum 16 GB RAM.
- **Storage:** 5 GB free disk space for datasets, models, and dependencies.
- **GPU:** Recommended for BERT fine-tuning; CUDA-compatible GPU (e.g., NVIDIA GTX 1660 or higher).
- **Operating System:** Windows 10, macOS 10.15+, or Linux (Ubuntu 20.04+).

1.2 Software Requirements

- **Python Version:** Python 3.10 or higher.
- **Development Environment:** Jupyter Notebook, Google Colab, or VS Code.

2. Python Library Dependencies

Below are the required libraries for data preprocessing, machine learning, and BERT-based NLP tasks:

Library	**Version**	**Purpose**
pandas	>=2.2.2	Data manipulation and analysis.
numpy	>=1.26.4	Numerical computations.
scikit-learn	>=1.5.2	Gradient Boosting and evaluation metrics.
matplotlib	>=3.8.0	Data visualization.
seaborn	>=0.13.2	Enhanced statistical visualizations.
tensorflow	>=2.17.1	Deep learning and BERT integration.
transformers	>=4.46.2	BERT model and tokenizer.

datasets	>=3.1.0	Hugging Face datasets for BERT.
joblib	>=1.4.2	Model serialization.
nltk	>=3.9.1	Text preprocessing.
torch	>=2.5.1	PyTorch backend for BERT.
wordcloud	>=1.9.4	Visualizing frequent words in text data.

3. Dataset Information

3.1 Data Sources

- **Phishing Dataset:** Contains email URLs and metadata labeled as phishing or legitimate.
- **Legitimate Email Dataset:** Includes legitimate email metadata and text.

3.2 Data Insights

- **Phishing Dataset Fields:**
    - url: URL found in phishing emails.
    - target: Entity targeted by the phishing attempt.
    - Labels: 1 for phishing and 0 for legitimate emails.
- **Legitimate Dataset Fields:**
    - message: Raw email text.
    - Labels: 0 for legitimate emails.

4. Data Preprocessing

1. **Cleaning and Labelling:**
    - **Phishing Data:**
        - URLs were cleaned to remove special characters and extract domains.
        - Label: 1 (phishing).
    - **Legitimate Data:**
        - Text data was cleaned of HTML tags, special characters, and non-alphabetic symbols.
        - Label: 0 (legitimate).
2. **Combining Datasets:**
    - Unified the cleaned phishing and legitimate data into a single data frame.

- Added datatype field to distinguish between phishing and legitimate.

3. **Text Vectorization:**
    - Applied **TF-IDF** (Term Frequency-Inverse Document Frequency) with a maximum of 1000 features.

5. Machine Learning Models

5.1 Gradient Boosting Classifier

- **Vectorization:** TF-IDF for converting text to numerical features.
- **Data Split:** 80% training, 20% testing.
- **Evaluation:**
    - Metrics: Confusion matrix, accuracy, precision, recall, and ROC-AUC.
    - Model saved as phishing_detector.pkl.

5.2 BERT-based Classification

- **Model:** DistilBERT and BERT (bert-base-uncased) for sequence classification.
- **Tokenizer:** Converts text to token IDs compatible with BERT.
    - Padding and truncation enabled for uniform input sizes (max tokens: 512).
- **Training:**
    - Framework: Hugging Face Trainer API.
    - Batch size: 4 (training), 8 (evaluation).
    - Learning rate: 2e-5.
    - Epochs: 1 (with fine-tuning capability for downstream tasks).
- **Evaluation:**
    - Metrics: Accuracy, precision, recall, and F1 score (all achieved 1.0 on test data).

6. Configuration and Execution Steps

6.1 Installation Commands

Install the required Python libraries using the following command:

- **pip install pandas numpy scikit-learn nltk tensorflow transformers seaborn matplotlib wordcloud joblib datasets torch**

6.2 Execution

1. **Preprocess datasets:**
    - Clean phishing URLs and legitimate email text.
    - Combine datasets with proper labeling.

2. **Train Gradient Boosting:**
    - Split data into training and testing sets.
    - Train the model using GradientBoostingClassifier ().
3. **Train BERT-based Classifier:**
    - Tokenize text data with BertTokenizer.
    - Fine-tune BERT using the Hugging Face Trainer API.

6.3 Saving Models

- Gradient Boosting Model: phishing_detector.pkl.
- BERT Model: bert_phishing_model.
- TF-IDF Vectorizer: tfidf_vectorizer.pkl.

7. Results and Observations

Model	Accuracy	Precision (Phishing)	Recall (Phishing)	F1-Score (Phishing)
Gradient Boosting	93.5%	93%	92%	92.5%
BERT-based Classifier	100.0%	100%	100%	100%

8. Notes and Recommendations

- **Best Practice:** Use BERT-based classifiers for highly accurate phishing detection.
- **Future Work:** Integrate additional data sources and explore multilingual phishing datasets.
- **Deployment:** Models can be deployed using Flask or FastAPI with saved model artifacts.