National College of Ireland

Configuration Manual

MSc Research Project
MSc Cyber Security

Emmanuel Ikelia
Student ID: 23284153

School of Computing
National College of Ireland

Supervisor: Dr. Mosab Hamdan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Emmanuel Ikelia

Student ID: x23284153

Programme: MSc. Cyber Security **Year:** One

Module: MSc (Research) Practicum/Internship

Lecturer: Dr. Mosab Hamdan

Submission Due Date: 15/09/2025

Project Title: FDFN-SA: The Lightweight Phishing detection system for endpoint devices

Word Count: ..... **Page Count:** .....

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: Emmanuel Ikelia

Date: 11/09/2025

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

Emmanuel Ikelia
Student ID: x23284153

1. Introduction

This manual provides the end-to-end steps to reproduce the phishing URL detection framework used in the MSc research. The implementation centers on a lightweight fusion model (FDFN-SA) that combines character-level CNN signals, NLP features (TF-IDF/BoW), and URL structural features for efficient, real-time detection on modest hardware. The instructions cover hardware and software setup, datasets, project structure, and how to run the included notebooks to train, evaluate, and perform ablation studies.

2. Hardware Requirements

- CPU: Intel Core i5-8365U (4 cores @ 1.6 GHz)
- RAM: 8 GB
- GPU: Not required (CPU-only inference targeted)

3. Software Requirements

- Operating system: Windows 11 (64-bit)
- Python environment: Anaconda/Miniconda (recommended)
- Editor/IDE: Jupyter Notebook (via Anaconda or VS Code)

3.1 Python Packages (tested versions)

- pandas 2.2.2
- NumPy 1.26.4
- scikit-learn 1.4.2
- XGBoost 2.0.3
- TensorFlow 2.16.1 (Keras API) — for character-level CNN branch
- matplotlib 3.8.4
- seaborn 0.13.2
- tldextract, urllib, re (URL parsing and utilities)

3.2 Environment Setup (Conda)

- Create environment: conda create -n fdnsa python=3.11 -y
- Install core libs: pip install pandas==2.2.2 numpy==1.26.4 scikit-learn==1.4.2 xgboost==2.0.3
- Install DL/viz: pip install tensorflow==2.16.1 matplotlib==3.8.4 seaborn==0.13.2
- Utilities: pip install tldextract

4. Dataset Description

- Two datasets are included in the project folders:
- Structured dataset (with engineered features) by (Shashwat Work, 2022): Code/proposed model and FDN-SA replica/dataset.csv

  Columns include URL lexical/host features (e.g., length_url, length_hostname, ip, counts of symbols), WHOIS/web stats (domain_age, web_traffic), and label column ‘status’ (0 = legitimate, 1 = phishing).

- URL-only dataset (feature extraction in-code) by (Harisudhan411, 2022): Code/Using another dataset/dataset.csv
- Columns: url, status. Numerical features are derived from the raw URL at runtime.

4.1 Notes on Labels

- status = 1 → phishing URL
- status = 0 → legitimate URL

5. Project Structure

- Code/proposed model and FDN-SA replica/
  - FDFN-SA_Proposed model.ipynb
  - FDN-SA_original_model.ipynb
  - dataset.csv (structured features)
- Code/Ablation/
  - No_Character_Branch.ipynb
  - No_NLP_Branch.ipynb
  - No_Structural_Branch.ipynb
  - dataset.csv
- Code/Using another dataset/
  - fdnsa_Lite.ipynb
  - dataset.csv (url, status only)

6. Model Preparation and Execution

6.1 FDFN-SA Proposed Model

- Open: Code/proposed model and FDN-SA replica/FDFN-SA_Proposed model.ipynb
- Select the 'fdnsa' conda kernel (or the environment you created).
- Run the setup cells to import libraries and configure paths.
- Load dataset.csv (structured features) and verify the class distribution.
- Train the model: the notebook builds character-level CNN, NLP (TF-IDF/BoW), and structural branches, then fuses them.
- Evaluate: run provided cells to compute Accuracy, Precision, Recall, F1, ROC-AUC, confusion matrix, and curves.
- Save artifacts if required (models/vectorizers) using the supplied cells.

6.2 Original FDN-SA Replica

- Open: Code/proposed model and FDN-SA replica/FDN-SA_original_model.ipynb
- Run cells to train and evaluate the baseline for comparison.

6.3 Using URL-Only Dataset (Feature Extraction in Code)

- Open: Code/Using another dataset/fdnsa_Lite.ipynb
- Load dataset.csv (columns: url, status).
- Run the feature extraction cell(s) to compute numerical/lexical features from the raw URL.
- Train and evaluate the light variant to validate generalization when only raw URLs are provided.

7. Ablation Studies

To quantify each branch's contribution, run the ablation notebooks:

- No_Character_Branch.ipynb — removes the character-level CNN features.

- No_NLP_Branch.ipynb — removes TF-IDF/BoW features.
- No_Structural_Branch.ipynb — removes engineered URL features.

The results we compared against the full model to identify the most impactful branch.

8. Reproducibility Checklist

- Fix random seeds where provided for comparable runs.
- Use the specified package versions to avoid API drift.
- When measuring latency, run inference on CPU and report average over multiple batches.
- For fair validation, use domain-aware splits and time-based holdouts to avoid leakage.

9. Troubleshooting

- Module import errors → verify the conda environment and package versions.
- Memory errors → use smaller batch sizes or sample subsets for quick tests.
- Imbalanced classes → use class weighting or stratified splits as configured in notebooks.

References

Harisudhan411. (2022). Phishing and Legitimate URLs. Kaggle. Retrieved from https://www.kaggle.com/datasets/harisudhan411/phishing-and-legitimate-urls

Shashwat Work. (2022). Web Page Phishing Detection Dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-datase