National College of Ireland

Configuration Manual

MSc Research Project
MSc Cybersecurity

Alias Davis
Student ID: x23297859

School of Computing
National College of Ireland

Supervisor: Dr. Mosab Hamdan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Alias Davis
Student ID:	x23297859
Programme:	**Year:** 2024-25
Module:	MSc Research Project
Lecturer:	Dr. Mosab Hamdan
Submission Due Date:	15/09/2025
Project Title:	Mitigating Phishing Threats: A Practical Cybersecurity Framework for Real-World Risk Reduction Using Machine Learning
Word Count:	765 **Page Count:** 5

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: Alias Davis

Date: 15/09/2025

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

Alias Davis
Student ID: x23297859

1. Environment Setup

Recommended Software

- **Python**: 3.8+
- **Jupyter Notebook** or **JupyterLab**
- **Operating System**: Windows 10/11, macOS, or Linux (Ubuntu 20.04+)

Required Python Libraries

Install the following packages using `pip`:

```
pip install pandas numpy matplotlib seaborn scikit-learn keras tensorflow joblib tldextract
```

Optional (if using JupyterLab):

```
pip install ipywidgets
```

2. Project Folder Structure

Create a main project folder, e.g., `phishing_detection_project/`, with the following structure:

```
phishing_detection_project/
│
├── Final_integrated.ipynb      # Main implementation notebook
├── full_dataset.csv      # Primary dataset
├── Training Dataset.arff       # Secondary dataset
├── models_summary.pkl          # Auto-generated during execution
├── model_results.json          # Auto-generated log (if enabled)
└── screenshots/                # Optional: For saving graph outputs
```

Ensure both `.csv` and `.arff` datasets are in the same folder as the notebook.

3. Running the Notebook

Step-by-Step Execution

1. **Load Required Libraries**
   The notebook starts by importing necessary packages such as:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.neural_network import MLPClassifier
import joblib
```

2. **Load Dataset (`full_dataset.csv`)**
   - Contains labeled phishing and legitimate URLs.
   - The label column is usually `Result` (1 = phishing, -1 = legitimate).
3. **Feature Engineering**
   - Extracts features like length of URL, number of dots, presence of '@', etc.
   - Converts URLs into feature vectors.
4. **Train-Test Split**
   - Splits data using `train_test_split()` with test size = 0.2 (80-20 split).
5. **Model Training**
   - Models used:
     - `LogisticRegression`
     - `MLPClassifier`
     - `KerasClassifier` (Neural Net with TensorFlow backend)
     - `RandomForestClassifier`
     - `VotingClassifier` (ensemble of above)
   - Each model is trained and its accuracy stored.
6. **Model Saving**
   - All models are saved into a single `.pkl` using `joblib.dump()`:

```
joblib.dump(models_dict, 'models_summary.pkl')
```

7. **Evaluation with `.arff` Dataset**
   - Uses a secondary dataset (`Training Dataset.arff`) for validation.
   - Loads via `pandas` using `scipy.io.arff`.
8. **Simulation Mode**
   - Manually or randomly selects URLs for testing.
   - Compares human guesses with model predictions.
   - Tracks user accuracy vs. ML accuracy.
9. **Result Visualization**
   - Generates bar charts:
     - Model-wise accuracy
     - Human vs. ML comparison
   - Uses `matplotlib.pyplot.bar()` for plotting.

4. Output Files and Logs

Output File	Description
`models_summary.pkl`	Contains all trained ML models
`model_results.json`	JSON summary of model scores (optional/log only)

5. Example Output

Sample Model Accuracy Output

Model	Accuracy
LogisticRegression	92.1%
MLPClassifier	94.5%
RandomForest	96.2%
Ensemble (Voting)	96.8%

Bar Chart Sample

Bar chart compares prediction accuracy across models and simulations.

6. Troubleshooting

Issue	Solution
`ModuleNotFoundError`	Reinstall missing packages via `pip install <package-name>`
`ARFF` file not loading	Ensure `scipy` is installed and file is properly formatted
Graphs not showing	Add `%matplotlib inline` at the top of the notebook
`models_summary.pkl` not created	Rerun all cells from the top to ensure model training completes

7. Final Notes

- **No hardcoded paths**: All paths are relative, ensuring reproducibility.
- **User Interaction**: Some parts may ask for user input or simulate guesses. These can be skipped or automated by setting flags.
- **GPU Optional**: If available, TensorFlow/Keras will use GPU acceleration.