Configuration Manual

MSc Research Project
Cloud Computing

Ammar Yousuf Abrahani
Student ID: 23341696

School of Computing
National College of Ireland

Supervisor: Dr Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ammar Yousuf Abrahani
Student ID:	23341696
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Dr Giovani Estrada
Submission Due Date:	05/08/2025
Project Title:	Configuration Manual
Word Count:	336
Page Count:	3

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:	Ammar Yousuf Abrahani
Date:	9th August 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

Ammar Yousuf Abrahani
23341696

1 Project Repository

The complete source code, datasets, and scripts for this project are available at: `https://github.com/ammarabrahani/deep-q-learning-anomaly-detection`

2 Python Version

This project was developed and tested using:

- **Python Version:** 3.12.6

3 Tools and Libraries

The project uses the following tools and libraries:

- **Programming Language:** Python 3.12.6
- **Development Environment:** Jupyter Notebook, VS Code
- **Core Libraries:**
  - `numpy`, `pandas` - Data preprocessing and manipulation
  - `scikit-learn` - Dataset generation, anomaly detection models, evaluation metrics
  - `tensorflow.keras` - Deep learning model implementation (Autoencoder, MLP)
  - `matplotlib` - Data visualization
- **Version Control:** Git, Git LFS for large files

4 Installation Steps

Ensure you are running Python 3.12.6. Run the following commands in your terminal to set up the environment:

```
# Create and activate a virtual environment
python -m venv venv_tf
source venv_tf/bin/activate   # On Windows: venv_tf\Scripts\activate
```

```
# Upgrade pip
pip install --upgrade pip

# Install required packages
pip install numpy pandas scikit-learn tensorflow matplotlib jupyter git-lfs
```

Install Git LFS for dataset handling:

```
git lfs install
git lfs track "*.csv"
```

5 Project Structure

```
project_root/
 data/
    raw/                # Original datasets (Credit Card Fraud)

 notebooks/              # Jupyter notebooks for experiments
    01_isolation_forest_anomaly_detection.ipynb
    02_compare_models_Is_local.ipynb
    03_mlp_classifier.ipynb
    04_autoencoder_pipeline.ipynb
    05_autoencoder_pipeline_credit_card_fraud_detection.ipynb
    06_model_comparison_full.ipynb
    07_deep_autoencoder_visualization.ipynb
    comparison_with_real_dql.ipynb

 models/                 # Saved trained models
 scripts/                # Python scripts (real_dql_model.py, utils.py)
 requirements.txt        # Python dependencies
 README.md               # Project overview
```

6 Dataset Configuration

- **Primary Dataset:** Credit Card Fraud Detection (Kaggle) – `creditcard.csv`
- **Synthetic Datasets:** Generated using `scikit-learn` (blobs, moons, uniform random)
- Store all datasets in `data/raw/`

7 Model Configuration

7.1 Autoencoder

- Architecture: 32 → 16 → 8 → 16 → 32 → Output

- Activation: ReLU (hidden), Sigmoid (output)
- Loss: Mean Squared Error (MSE)

7.2 MLP Classifier

- Binary classification with class imbalance handling
- Loss: Binary Cross-Entropy

7.3 Deep Q-Learning (Custom Wrapper)

- Implemented in `real_dql_model.py`
- Reward-penalty mechanism for anomaly classification
- Compatible with synthetic datasets

8 Evaluation Metrics

- **Macro Average F1-Score** – effective for imbalanced datasets
- Classification report including precision, recall, and F1-score

9 Reproducibility Notes

- Fixed random seed: `random_state=42`
- Consistent training parameters across experiments
- Recommended execution order:

  1. `04_autoencoder_pipeline.ipynb`
  2. `03_mlp_classifier.ipynb`
  3. `06_model_comparison_full.ipynb`
  4. `07_deep_autoencoder_visualization.ipynb`
  5. `comparison_with_real_dql.ipynb`