

# Configuration Manual

MSc Research Project  
FinTech

Chaolu

Student ID: 24110442

School of Computing  
National College of Ireland

Supervisor: Dr. Noel Cosgrave

National College of Ireland  
Project Submission Sheet  
School of Computing



<b>Student Name:</b>	Chaolu
<b>Student ID:</b>	24110442
<b>Programme:</b>	FinTech
<b>Year:</b>	2025
<b>Module:</b>	MSc Research Project
<b>Supervisor:</b>	Dr. Noel Cosgrave
<b>Submission Due Date:</b>	15/9/2025
<b>Project Title:</b>	Configuration Manual
<b>Word Count:</b>	435
<b>Page Count:</b>	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.

<b>Signature:</b>	Chaolu
<b>Date:</b>	9th August 2025

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

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

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

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Chaolu  
24110442

This configuration manual specifies the essential software tools, library dependencies, and experimental settings required to successfully replicate the experimental environment used in this study. It focuses exclusively on the components and configurations that are critical for ensuring reproducibility of results, omitting generic installation procedures for standard tools. The manual provides a detailed account of the tested operating environment, dataset access requirements, Python library versions, and random seed configurations. Furthermore, it outlines the graph neural network settings, evaluation metrics, and execution considerations necessary for consistent experimentation. By adhering to these specifications, researchers and practitioners can reproduce the experimental workflow and verify the reported outcomes under comparable computational conditions. Google Colab: <https://colab.research.google.com/drive/1vVX8q3dHcfIrP-wGcoTsekDIU6UavwyH#scrollTo=d26cc59e>

## 1 Operating Environment

- **Processor:** 12th Gen Intel(R) Core(TM) i5-1235U
- **Storage:** 477 GB
- **Graphics Card:** 128 MB
- **Installed RAM:** 16.0 GB
- **Edition:** Windows 11 Home
- **Version:** 24H2
- **Installed on:** 23/03/2025
- **OS build:** 26100.4652

## 2 Dataset Access

- **Source:** Kaggle dataset <sup>1</sup> — *Elliptic Bitcoin Dataset* (`ellipticco/elliptic-data-set`).
- **Download Method:** The code uses `kagglehub` (1) to fetch the dataset automatically.

---

<sup>1</sup><https://www.kaggle.com/datasets/ellipticco/elliptic-data-set>

- **Folder Structure After Download:**

```
elliptic_bitcoin_dataset/
  elliptic_txs_classes.csv
  elliptic_txs_edgelist.csv
  elliptic_txs_features.csv
```

### 3 Required Python Libraries

The following libraries are necessary for execution. Versions should match or be higher than those tested to avoid compatibility issues. Python version: 3.11.13.

Category	Library & Version	Purpose
Core ML & Graph	torch (2.6.0+cu124) (2) torch_geometric (2.6.1) (3)  networkx (3.5) (8)	Deep learning framework Graph Neural Networks (4) (GCN (5), GAT (6), GraphSAGE (7)) Graph construction from edgelist
Data Handling	pandas (2.2.2) (9)  numpy (2.0.2) (10)	DataFrames for dataset storage Numerical operations
Traditional ML	scikit-learn (1.6.1) (11)	Baseline classifiers & pre-processing
Visualization	matplotlib (3.10.0) (12) seaborn (0.13.2) (13)	Data visualization Statistical plotting
Others	kagglehub (0.3.12) (1) warnings (built-in) copy (built-in)	Kaggle dataset download Suppress warnings Deep/shallow copy of objects

### 4 Random Seed Configuration

To ensure reproducibility of results across runs:

```
RANDOM_STATE = 42
np.random.seed(RANDOM_STATE)
torch.manual_seed(RANDOM_STATE)
```

This setting ensures consistent data splits, model initialization, and training outcomes.

### 5 Graph Neural Network Settings

- **Supported Architectures:** GCN (5), GAT (6), GraphSAGE (7)(via PyTorch Geometric).

- **Node & Edge Conversion:** NetworkX (8) graphs are converted to PyTorch Geometric (2) Data or HeteroData objects.
- **Feature Count:** The dataset contains 165 numerical features per transaction, plus transaction ID and timestep.

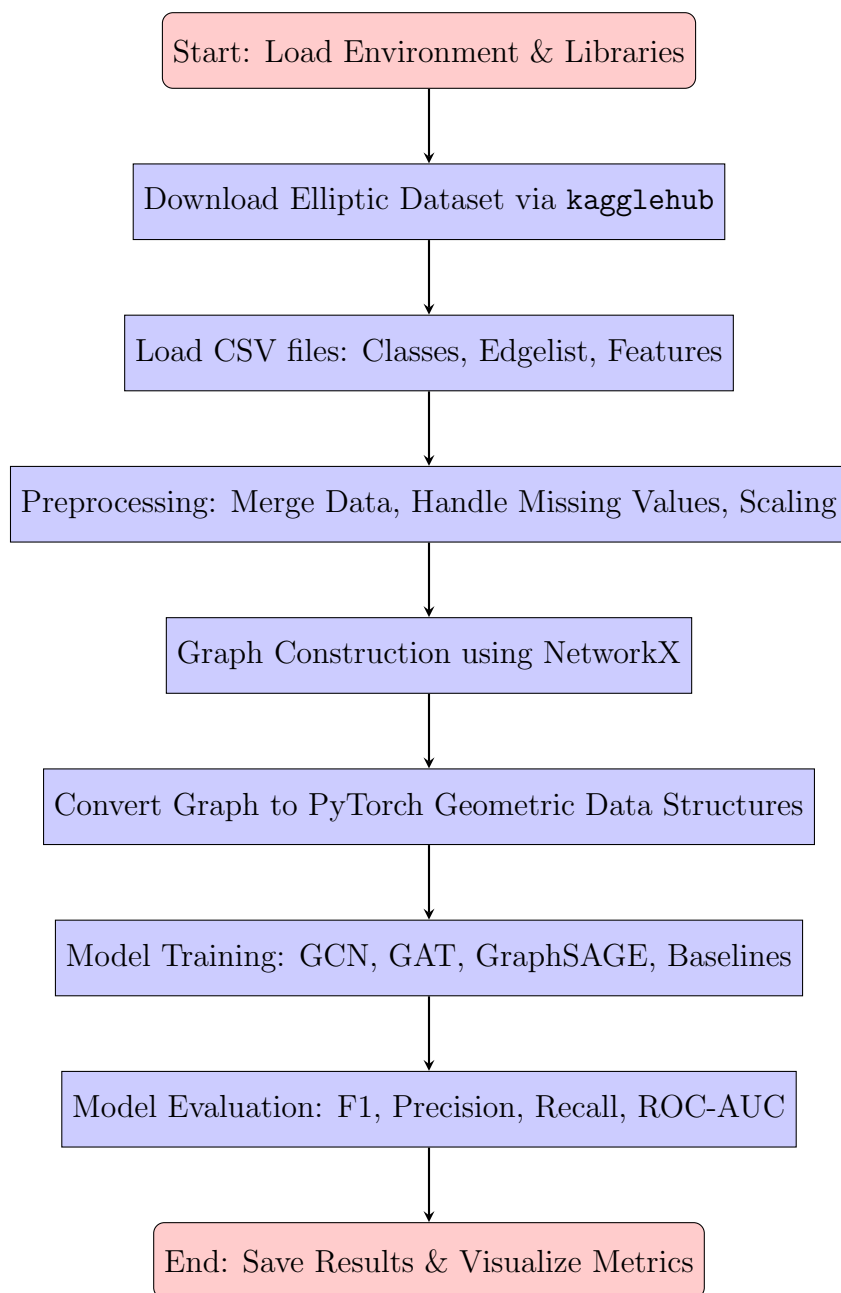
## 6 Evaluation Metrics

The setup computes: **F1-score, Precision, Recall, AUC-ROC, AUC-PR**. Metrics are computed using `scikit-learn` (11) functions.

## 7 Execution Notes

- Ensure CUDA (14) is available and correctly configured for GPU acceleration; otherwise, PyTorch (2) will default to CPU, which may significantly increase training time.
- Dataset download via `kagglehub` (1) requires a valid Kaggle API token set in the environment.
- Large datasets may require increasing the Python memory limit or adjusting `pandas` (9) read parameters.

## 8 Experimental Workflow Diagram



## References

- [1] Kaggle, “Kagglehub: The official Kaggle library to interact with Kaggle models, datasets, and more.” <https://github.com/Kaggle/kagglehub>, 2024. Accessed: 2025-08-09.
- [2] The PyTorch Foundation, “PyTorch.” <https://pytorch.org/>, 2024. Accessed: 2025-08-09.
- [3] PyTorch Geometric Contributors, “PyTorch Geometric.” <https://pytorch-geometric.readthedocs.io/en/latest/>, 2024. Accessed: 2025-08-09.

- [4] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, pp. 61–80, 2009.
- [5] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations (ICLR)*, 2017.
- [6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *International Conference on Learning Representations (ICLR)*, 2018.
- [7] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.
- [8] NetworkX Developers, “NetworkX.” <https://networkx.org/>, 2024. Accessed: 2025-08-09.
- [9] The pandas development team, “pandas: powerful python data analysis toolkit.” <https://pandas.pydata.org/>, 2024. Accessed: 2025-08-09.
- [10] Harris, Charles R. and Millman, K. Jarrod and van der Walt, Stéfan J. and Gommers, Ralf and Virtanen, Pauli and Cournapeau, David and Wieser, Eric and Taylor, Julian and Berg, Sebastian and Smith, Nathaniel J. and Kern, Robert and Picus, Matti and Hoyer, Scott and van Kerkwijk, Marten H. and Brett, Matthew and Haldane, Alex and del Río, Jaime Fernández and Wiebe, Mark and Peterson, Pearu and Gérard-Marchant, Pierre and Sheppard, Kevin and Reddy, Tyler and Weckesser, Warren and Abbasi, Hameer and Grout, Christoph and Cronin, Antony and Oliphant, Travis E., “Array programming with NumPy.” <https://numpy.org/>, 2020. Accessed: 2025-08-09.
- [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [12] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [13] M. L. Waskom, “seaborn: statistical data visualization,” *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.
- [14] NVIDIA Corporation, “NVIDIA CUDA C++ Programming Guide.” <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>, 2025. Version 13.0.