National College of Ireland

A Hybrid Graph Neural Network and XGBoost Model for Telecom Customer Churn Prediction and Feature Attribution Configuration Manual

Research Practicum Part 2
Data Analytics

Muhammad Osama Hassan Khan
Student ID: x24137782@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Anderson Simiscuka

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Muhammad Osama Hassan Khan
Student ID:	x24137782@student.ncirl.ie
Programme:	Data Analytics
Year:	2025
Module:	Research Practicum Part 2
Supervisor:	Anderson Simiscuka
Submission Due Date:	11/12/2025
Project Title:	A Hybrid Graph Neural Network and XGBoost Model for Telecom Customer Churn Prediction and Feature Attribution Configuration Manual
Word Count:	2,214
Page Count:	6

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:	
Date:	11/12/2025

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):	

A Hybrid Graph Neural Network and XGBoost Model for Telecom Customer Churn Prediction and Feature Attribution Configuration Manual

Muhammad Osama Hassan Khan
x24137782@student.ncirl.ie

1 Introduction

This project implements a comprehensive Telco Customer Churn Prediction system using advanced machine learning and deep learning techniques with a particular focus on Graph Neural Networks (GNNs). Customer churn prediction is a critical business problem in the telecommunications industry, and the ability to accurately forecast churn can dramatically improve customer retention. This configuration manual accompanies the codebase and explains how to set up and run the project.

The notebook provides the following components:

- **Exploratory Data Analysis (EDA):** understanding customer behaviour patterns and feature distributions.
- **Data Preprocessing:** handling missing values, encoding categorical variables and feature engineering.
- **Traditional Machine Learning Models:** implementation of classifiers such as Random Forest, XGBoost, Long Short Term Memory (LSTM) and Decision Trees.
- **Graph Neural Networks:** advanced deep learning approaches using:
  - Homogeneous GraphSAGE: customer similarity graphs.
  - Heterogeneous GraphSAGE: multi-relational customer networks with Random Forest integration.
- **Model Evaluation:** comprehensive comparison using accuracy, precision, recall, F1-score and ROC-AUC curves.
- **Visualization:** detailed plots including confusion matrices, feature importance and model comparison dashboards.
- **Key Features:**
  - Graph-based representation of customer relationships.
  - Hybrid approach combining GNN embeddings with traditional ML classifiers.
  - Extensive hyperparameter tuning and cross-validation.
  - Production-ready code with comprehensive documentation.

2 System Requirements

2.1 Hardware

Component	Minimum	Recommended
CPU	Intel Core i5 / AMD Ryzen 5	Intel Core i7 / AMD Ryzen 7
RAM	8 GB	16 GB+
Storage	5 GB free space	10 GB+ free space
GPU	Not required	NVIDIA GPU with CUDA support (for faster training)

Table 1: Hardware requirements

2.2 Software

The code runs on Windows 10/11, macOS 10.15+ or Linux (Ubuntu 18.04+). A Python environment is required with a recommended version between 3.10 and 3.12. The following libraries must be installed:

- `numpy` $\geq$ `1.24.0`
- `pandas` $\geq$ `2.0.0`
- `matplotlib` $\geq$ `3.7.0`
- `seaborn` $\geq$ `0.12.0`
- `scikit learn` $\geq$ `1.3.0`
- `torch` $\geq$ `2.0.0`
- `torch geometric` $\geq$ `2.4.0`
- `xgboost` $\geq$ `2.0.0`
- `kagglehub` $\geq$ `0.2.0`

2.3 Installation Commands

```
# Create virtual environment (recommended)
python -m venv venv

# Activate virtual environment
# Windows:
\.\venv\Scripts\activate
# macOS/Linux:
source venv/bin/activate
```

```
# Install required packages
pip install numpy pandas matplotlib seaborn scikit-learn
pip install torch torchvision torchaudio
pip install torch-geometric
pip install xgboost kagglehub
```

The development environment is most conveniently managed with an integrated development environment (IDE) such as Visual Studio Code with the Jupyter extension. Alternatives include JupyterLab, Jupyter Notebook or PyCharm Professional.

3 Dataset

The project uses the Telco Customer Churn dataset, which contains information about telecommunications customers and whether they churned. A summary of the dataset is provided in Table 2.

Attribute	**Description**
Source	Kaggle – IBM Sample Data Sets
Records	7,043 customer records
Features	21 columns
Target variable	*Churn* (Yes/No)

Table 2: Dataset overview

3.1 Feature Categories

Customer Demographics:

- **customerID** – Unique customer identifier (string).
- **gender** – Male/Female (categorical).
- **SeniorCitizen** – Whether the customer is a senior citizen (binary).
- **Partner** – Whether the customer has a partner (categorical).
- **Dependents** – Whether the customer has dependents (categorical).

Service Information:

- **tenure** – Number of months with the company (numeric).
- **PhoneService** – Whether the customer has phone service (categorical).
- **MultipleLines** – Whether the customer has multiple lines (categorical).
- **InternetService** – Type of internet service (DSL/Fibre optic/No).
- **OnlineSecurity** – Whether the customer has online security.
- **OnlineBackup** – Whether the customer has online backup.

- **DeviceProtection** – Whether the customer has device protection.
- **TechSupport** – Whether the customer has tech support.
- **StreamingTV** – Whether the customer has streaming TV.
- **StreamingMovies** – Whether the customer has streaming movies.

Account Information:

- **Contract** – Contract term (Month to month / One year / Two year).
- **PaperlessBilling** – Whether the customer has paperless billing.
- **PaymentMethod** – Payment method.
- **MonthlyCharges** – Monthly charge amount (numeric).
- **TotalCharges** – Total charges to date (numeric).

Target Variable:

- **Churn** – Whether the customer churned (Yes/No, binary).

3.2 Data File Location

The dataset is located in a sub directory `telco/` with the main Jupyter notebook `telco_v3_irllll_V2.ipynb` and the dataset file `WA_Fn-UseC_-Telco-Customer-Churn.csv`. To obtain the dataset:

- **Direct download:** download from Kaggle – Telco Customer Churn dataset.
- **Using KaggleHub:**

```
import kagglehub
path = kagglehub.dataset_download("blastchar/telco-customer-churn")
```

4 Directory Structure

The project repository is organised as follows:

```
telco/
 telco_v3_irllll_V2.ipynb              # Main Jupyter notebook (latest version)
 WA_Fn-UseC_-Telco-Customer-Churn.csv  # Dataset file
 README.md                             # Project read-me
```

Additional directories may be created for model outputs, figures or logs during execution.

5 Code Execution

This section explains how to run the Jupyter notebook that accompanies this manual.

5.1 Step 1: Environment Setup

1. Navigate to the project directory:

```
cd c:\Users\XT\AppData\Local\Programs\Python\Python312\genv\telco
```

2. Activate the virtual environment (if using one):

```
# Windows
..\venv\Scripts\activate
# macOS/Linux
source ../venv/bin/activate
```

3. Verify the Python and package versions:

```
python --version
pip list | grep -E "torch|pandas|sklearn"
```

5.2 Step 2: Launch Jupyter Environment

- **Option A – Visual Studio Code (recommended):** open the `telco/` folder and run the notebook `telco_v3_irllll_V2.ipynb`.
- **Option B – Jupyter Notebook:**

```
jupyter notebook telco_v3_irllll_V2.ipynb
```

- **Option C – JupyterLab:**

```
jupyter lab
```

5.3 Step 3: Execute Notebook Cells

The notebook is organised into logical sections. Execute the cells in order as summarised in Table 3.

Section	Cell Range	Description
Importing Libraries	1–2	Load required packages (˜5 s)
Loading Dataset	3–10	Load and clean data (˜10 s)
EDA & Visualisation	11–15	Exploratory analysis (˜30 s)
Data Preprocessing	16–25	Feature engineering (˜20 s)
Traditional ML Models	26–45	Train baseline models (2–5 min)
Homogeneous GraphSAGE	46–55	Train GNN model (3–5 min)
Heterogeneous GraphSAGE	56–70	Advanced GNN + RF (5–10 min)
Model Comparison	71–80	Compare all models (˜1 min)
Visualisation & Results	81–87	Generate final plots (˜30 s)

Table 3: Notebook sections and estimated execution times

You can run all cells at once using the *Run All* command, execute cells individually with `Shift+Enter` or run selected sections by highlighting multiple cells.

5.4 Step 4: Monitor Execution

Watch the output of each cell for progress and errors. If using a GPU, monitor availability with the following commands:

```
import torch
print(f"CUDA␣Available:␣{torch.cuda.is_available()}")
print(f"Device:␣{torch.cuda.get_device_name(0)␣if␣torch.cuda.is_available()␣
    else␣'CPU'}")
```

5.5 Step 5: Verify Results

After execution, verify that:

- The pre processed dataset has shape `(7043, 21)` and no missing values.
- Accuracy, precision, recall, F1 score and ROC AUC are reported for each model.
- Confusion matrices, ROC curves, feature importance plots and model comparison dashboards are generated.
- The best performing models are typically Heterogeneous GraphSAGE + Random Forest, XGBoost and Random Forest.

6 Troubleshooting

Common issues and solutions are summarised in Table 4. Refer to this table if you encounter errors during execution.

Issue	Solution
ModuleNotFoundError	Install the missing module using `pip install <module_name>`.
FileNotFoundError for dataset	Ensure that `telco_customer_churn.csv` is in the same directory as the notebook.
CUDA out of memory	Reduce the batch size or switch to CPU execution.
Kernel dies during execution	Increase system RAM or reduce the data size.
Long execution time	Use a GPU or reduce the hyperparameter search space.

Table 4: Troubleshooting common issues

7 Saving Results

After execution, save the notebook with outputs to preserve the results. In Visual Studio Code you can press `Ctrl+S`, whereas in Jupyter Notebook you can choose *File* → *Save and Checkpoint*. The notebook can also be exported as HTML, PDF or a Python script using the *Export As* menu. Be sure to commit any changes to version control if you are tracking your project in a repository.