National College of Ireland

Adversarial Graph Neural Networks for Fair Insurance Pricing: An Integrative Framework with a Synthetic Benchmark

Data Analytics

Narendra Singh Chilwal
Student ID: x23316144

School of Computing
National College of Ireland

Supervisor: Prof Harshani Nagahamulla

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Narendra Singh Chilwal
Student ID:	x23316144
Programme:	Data Analytics
Year:	2025
Module:	Research Practicum
Supervisor:	Prof Harshani Nagahamulla
Submission Due Date:	15th September 2025
Project Title:	Adversarial Graph Neural Networks for Fair Insurance Pricing: An Integrative Framework with a Synthetic Benchmark
Word Count:	1080
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:	Narendra Singh Chilwal
Date:	15th September 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

Narendra Singh Chilwal
x23316144
MSc Data Analytics

Contents

1 Purpose and Scope

This manual explains how to set up the environment, obtain data, and reproduce all experiments for the project *Adversarial Graph Neural Networks for Fair Insurance Pricing*. It lists the required packages, hardware assumptions, repository layout, run order, configuration knobs, and high-level pseudocode for each pipeline (synthetic regression phases P0–P6 and the real-data classification prototype P7). The manual avoids code listings and focuses on *what to run*, *what it produces*, and *how to change settings*.

2 System Requirements

OS: Linux/macOS/Windows 10+. **Python:** 3.10 (recommended). **RAM:** 16 GB+ (32 GB+ preferred for large graphs). **GPU (optional):** NVIDIA CUDA 11.8+ for faster GNN training.

Recommended stack (pinned):

- PyTorch 2.3.x (`cpu` or CUDA build to match your driver)
- PyTorch Geometric 2.5.x (+ `torch_scatter`, `torch_sparse`, `torch_cluster`)
- pandas 2.2.x, numpy 1.26.x, scikit-learn 1.4.x
- networkx 3.2.x, matplotlib 3.8.x, tqdm, tensorboard, pyyaml

Listing 1: Installation Guide

```
# create env
conda create -n fair-gnn python=3.10 -y
conda activate fair-gnn

# --- Install PyTorch (pick ONE) ---
# CPU:
pip install torch==2.3.1 --index-url https://download.pytorch.org/whl/cpu
# OR CUDA 11.8:
# pip install torch==2.3.1 --index-url https://download.pytorch.org/whl/cu118

# --- Install PyG wheels that MATCH your Torch build ---
# CPU wheels:
pip install torch-geometric==2.5.3 torch-scatter torch-sparse torch-cluster \
  -f https://data.pyg.org/whl/torch-2.3.1+cpu.html
# OR for CUDA 11.8:
# pip install torch-geometric==2.5.3 torch-scatter torch-sparse torch-cluster
    \
#   -f https://data.pyg.org/whl/torch-2.3.1+cu118.html

# Core libs
pip install pandas==2.2.2 numpy==1.26.4 scikit-learn==1.4.2 \
  networkx==3.2.1 matplotlib==3.8.4 tqdm==4.66.4 tensorboard==2.17.0 pyyaml
   ==6.0.1

# Quick sanity check
```

```
python - <<'PY'
import torch, torch_geometric
print("torch:", torch.__version__)
print("pyg:", torch_geometric.__version__)
print("cuda available:", torch.cuda.is_available())
PY
```

3 Repository Layout

- `notebooks/`:
  - `policycode.ipynb` — synthetic graph generator + label-bias audit.
  - `Thesis.ipynb` — all phases (P0–P6 regression; P7 classification).
- `data/`: raw files (`synthetic_nodes.csv`, `synthetic_edges.csv`, `Insurance.arff`).

4 Data Inputs

Synthetic (Bias-on-Demand) (Baumann et al.; 2023):

- Generated by `policycode.ipynb` using knobs: n, p_A, δ_{loc}, τ, r_{NB}, α_A, M, λ_{event}, `seed`.
- Outputs: `data/synthetic_nodes.csv`, `data/synthetic_edges.csv`.

	node_id	race	is_privileged	gender	age_bucket	x_coord	y_coord	\
0	0	White	1	Male	18–25	0.544663	0.501037	
1	1	Hispanic	0	Female	60+	0.536569	0.435482	
2	2	Asian	0	Female	26–40	0.500086	0.527273	
3	3	White	1	Female	41–60	0.614519	0.583820	
4	4	White	1	Female	60+	0.637240	0.567972	

	claim_frequency
0	1
1	1
2	1
3	3
4	0

```
type
geographic    333498634
co_claim          82567
Name: count, dtype: int64
```

(a) Nodes/edges preview

```
Race means:
 race
Hispanic    0.979644
Asian       0.991906
Other       0.999319
Black       1.004774
White       2.718360
Name: claim_frequency, dtype: float64
Label DP_race gap: 1.739

Geo means:
 geo_quad
0    1.148810
2    2.230231
1    2.294612
3    2.723551
Name: claim_frequency, dtype: float64
Label DP_geo gap: 1.575
```

(b) Bias audit (DP)

Figure 1: Synthetic-dataset checks from `policycode.ipynb`.

Quick checks. Real dataset (P7):

```
FINAL SUMMARY
================================================================
Dataset: 15,000 nodes, 319,874 edges
Model: 13,105 parameters, 3-layer GCN
Test AUC: 0.4867 (Needs Improvement)
Test Accuracy: 0.6169
Average Fairness Gap: 0.225 (Moderate Bias)
```

(a) Console summary for P7.

Item	Value
Dataset size	15,000 nodes; 319,874 edges
Model	3-layer GCN; 13,105 parameters
AUC	0.4867
Accuracy	0.6169
Avg fairness gap	0.225
$\mathrm{DP}_{\mathrm{City}}$	0.750
$\mathrm{DP}_{\mathrm{Accommodation\ Type}}$	0.066
$\mathrm{DP}_{\mathrm{Is\ Spouse}}$	0.025
$\mathrm{DP}_{\mathrm{Reco\ Insurance\ Type}}$	0.057

(b) P7 (real-data) test-set summary.

Figure 2: Phase 7 snapshot: metrics table and console output.

- `data/Insurance.arff` (public, de-identified), sourced from OpenML (Dataset ID 45064) (Lee; 2023). Used as a 15k stratified sample for classification.
- One-hots for low-cardinality categoricals (incl. `City_Code`); 3 numeric scaled; total $\sim$71 dims.

5 Reproduction Steps

1. **Generate synthetic data:** open `notebooks/policycode.ipynb`, run all cells. Verify on-screen the head of `nodes_df` and edge counts; the label-bias audit should print DP gaps.
2. **Train/evaluate phases:** open `notebooks/Thesis.ipynb`.
   - **P0–P6 (regression):** set `task: regression`, pick $\lambda \in \{0, 0.1, 0.5, 1, 2\}$ and the adversary mode (single/multi/intersectional). View RMSE/DP results inline; optionally save the plot via Jupyter's 'Save image' menu.
   - **P7 (classification):** set `task: classification`, load `Insurance.arff`, build the 15k graph (feature-similarity + city proximity + light random), train a 3-layer GCN. Export AUC/Accuracy and DP by groups.

6 Configuration

```
seed: 42
task: regression  # or classification

data:
  synthetic_nodes: data/synthetic_nodes.csv
  synthetic_edges: data/synthetic_edges.csv
  insurance_arff:  data/Insurance.arff

bias_knobs:
  n_nodes: 50000
```

```
  p_privileged: 0.60
  delta_loc: 0.05
  tau: 0.05
  r_nb: 1.0
  alpha_priv: 1.0
  M_events: 10000
  lambda_event: 4.0

model:
  gnn: GCN
  hidden_dim: 128
  layers: 2   # 3 for P7
  dropout: 0.0

train:
  epochs: 50
  lr: 0.01
  weight_decay: 1e-4
  lambda_adv: [0, 0.1, 0.5, 1, 2]
  adversaries: [race, gender, age, geo]  # 'multi' or 'racexgender'
```

7 Notebook Responsibilities

policycode.ipynb

- Samples demographics and locations (*representation bias* via δ_{loc}).
- Simulates claim frequencies with NB(r=1) and privileged log-odds boost α_A (*measurement bias*).
- Builds two edge layers: geographic radius τ and co-claim cliques from M events (*structural bias*).
- Saves nodes/edges CSV and prints a label-bias audit (group means and DP gaps).

Thesis.ipynb

- Loads graph; splits masks; builds 2×GCN encoder with regression head.
- Attaches GRL-based adversarial heads (single/multi/intersectional).
- Trains across λ values; logs RMSE + DP for race/gender/age/geo; exports frontier plot.
- P7 mode: builds 15k real-data graph, 3-layer GCN classifier, reports AUC/Accuracy and DP by groups.

8 Pseudocode

A. Bias-on-Demand Graph Generator

1. Sample N policyholders with attributes: race (with privileged share p_A), gender, age bucket.
2. Place each node in $\mathbb{R}^2$ with mean shifted by $\pm\delta_{\text{loc}}$ for (un)privileged groups.
3. Draw claim counts from NB(r=1) with log-mean $\alpha_0 + \alpha_A \cdot \mathbb{1}\{\text{priv}\}$.
4. Add geographic edges for pairs with Euclidean distance $< \tau$.
5. Simulate M multi-vehicle events with Poisson(λ_{event}) participants; add clique edges.
6. Output `synthetic_nodes.csv`, `synthetic_edges.csv`; compute and log label DP gaps.

B. Adversarial GNN (Regression, P0–P6)

1. Build 2-layer GCN encoder; regression head predicts $\hat{y}$ (claim count).
2. Attach GRL + classifiers for targeted attributes; choose mode: single / multi / intersectional.
3. Train with loss $\text{MSE}(y, \hat{y}) - \lambda \sum_k \text{CE}(s_k, \hat{s}_k)$.
4. For each λ, report RMSE (test) and DP gaps by attribute; export frontier plot.

C. Real-Data Prototype (Classification, P7)

1. Load `Insurance.arff`; build 15k stratified sample; scale numeric, one-hot categoricals.
2. Construct graph (feature-similarity + city proximity + light random edges).
3. Train 3-layer GCN classifier (weighted BCE); report AUC, Accuracy, and DP by City and other groups.

9 Troubleshooting & FAQs

1. **PyG / PyTorch install errors.** Version mismatches between `torch` and `torch-geometric` cause import failures. Reinstall using the wheels that match your Torch/CUDA (or CPU) version as shown in the install snippet; restart the kernel after installation.
2. **"Insurance.arff not found."** Place `Insurance.arff` in the same folder as `Thesis.ipynb` (or edit the path at the top of the notebook). Rerun from the start.
3. **Kernel dies when building the 50k synthetic graph.** The geographic edge stage is memory-heavy. Reduce `n_nodes` (e.g., to 10k) or increase τ sparsity; the paper's results use a 10k induced subgraph for tractability.

4. **City fairness table shows many tiny groups.** Small cells inflate the DP gap. For Phase 7 we report the *unpruned* DP (upper bound). If desired, prune groups with < 20 samples before computing DP (not required to reproduce the report).

5. **Plots don't save to disk.** The notebooks display plots inline only. If you need files for the manual, use Jupyter's "Save image" from the output cell, or add a single line `plt.savefig('figs/<name>.png', dpi=300)` right after plotting.

6. **Results differ slightly from the report.** Confirm the fixed seed (42) and that you use the same train/validation/test splits stated in the notebooks. Minor drift can occur across library versions.

7. **CPU vs. GPU.** The notebooks run on CPU. If you have CUDA, ensure all PyG wheels match the Torch CUDA version; otherwise keep Torch CPU to avoid clashes.

8. **Relative paths fail on Windows.** Avoid spaces in folder names; run the notebook from the project directory so relative paths (e.g., `data/`, `synthetic_nodes.csv`) resolve correctly.

9. **Order of execution.** Run `policycode.ipynb` first to generate `synthetic_nodes.csv` and `synthetic_edges.csv`. Then run `Thesis.ipynb` for Phases P0–P6 (regression) and P7 (classification).

10 Reproducibility Notes

Seeds. All runs use `seed=42`; masks/splits are fixed and saved.

Common issues:

- *PyG install fails:* ensure your PyTorch build (CPU/CUDA) matches the PyG wheel URL.
- *Out-of-memory on full graphs:* switch to the 10k induced subgraph (regression) or the 15k P7 graph; or use PyG `NeighborLoader`.

References

Baumann, P., Renz, J. and Schmid, S. (2023). Bias-on-demand: A controlled benchmark for fairness experiments.
URL: *https://arxiv.org/abs/2302.07890*

Lee, Y. (2023). Insurance (openml dataset 45064), OpenML. Version 1; ARFF; accessed 2025-08-10.
URL: *https://www.openml.org/d/45064*