

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

جامعة غرداية
University of Ghardaia

N° d'enregistrement
/.../.../.../.../...



كلية العلوم والتكنولوجيا
Faculty of Science and Technology
قسم الرياضيات و الإعلام الآلي
Département of Mathematics and Computer Science

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of
Master

Domaine : Mathematics and Computer Science
Filière : Computer Science
Spécialité : Intelligent Systems for Knowledge Extraction

Topic

Graph AI for Criminal Network Analysis

Présenté par :

Dabouz Mohamed
Tizeggagine Yacine

Soutenue publiquement le June, 2026

Devant le jury composé de :

N. Brahim	MAA	Univ.Ghardaia	President
A.Bekkair	MCB	Univ.Ghardaia	Examiner
M.Adjila	Professeur	Univ.Ghardaia	Examiner
A.Saidi	MCA	Univ.Ghardaia	Supervisor

Année universitaire 2025/2026

Acknowledgements

First and foremost, all praise and gratitude are due to **Allah**, the Almighty, for granting us the strength, patience, and knowledge to complete this work. Without His guidance and blessings, none of this would have been possible.

We would like to express our deepest gratitude to our supervisor, **Professor Ahmed Saidi**, whose unwavering support, insightful guidance, and continuous follow-up throughout every stage of this work were instrumental in bringing this project to fruition. His expertise and dedication have been a true source of inspiration to both of us.

We would also like to extend our sincere thanks to the professors who taught and guided us throughout our academic journey. Their knowledge, commitment, and passion for teaching have shaped our thinking and laid the foundation upon which this work is built.

Finally, we are grateful to all those who contributed, directly or indirectly, to the completion of this work. This achievement is the result of our combined efforts, mutual support, and shared dedication throughout this journey.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

إلى والدي الكريمين ،

اللذين غرسا في حب العلم ، وأضاءا دري بدعائهما وعطاءهما،

لا يكفي الشكر ولا يُوفيها حقهما كلام.

إلى إخوتي وأخواتي ،

رفقاء الطريق وسند الأيام.

إلى عائلتي الكريمة،

الجدور التي يرسو عليها كل نجاح.

إلى أصدقائي ،

الذين أزروني ووقفوا إلى جانبي.

إليكم جميعاً، أهدي ثمرة هذا الجهد.

الدبوز محمد

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

اللهم ارحم والدي برحمتك الواسعة، وأسكنه فسيح جناتك،

واجعل قبره نوراً وطمأنينة إلى يوم يبعثون.

إلى والدتي الغالية منبع الحب والأمان،

إلى زوجتي العزيزة، رفيقة دربي وسندي في مسيرة الحياة،

إلى كلّ من كان في قلبي ولم يُذكر قلبي،

لكم جميعاً أهدي ثمرة جهدي وعملي.

تزقخين ياسين

Abstract

Criminal Networks represent a class of complex, dynamic, and covert systems whose analysis poses significant challenges to traditional graph-based and Machine Learning (ML) methods. Existing approaches predominantly rely on single-layer static representations, which fail to capture the heterogeneous and multi-relational nature of criminal interactions, and suffer from accumulated uncertainty inherent in incomplete law enforcement data. This thesis proposes a framework that combines Heterogeneous Graph Neural Networks with Fuzzy Logic techniques to address two fundamental tasks in criminal network analysis: node classification and link prediction. The heterogeneous graph representation preserves the semantic distinctions between multiple relational layers, including social, operational, and communication ties, while fuzzy logic is employed to enhance the reliability of gathered evidence by modeling and mitigating the uncertainty inherent in criminal network data. The proposed framework is evaluated on two real-world criminal network datasets, namely the Noordin Top Terrorist Network and the Sicilian Mafia Network. Experimental results demonstrate the effectiveness of the approach, achieving a Macro-F1 score of 71% on the node classification task, outperforming existing single-layer and homogeneous graph-based baselines.

Keywords: Criminal Network Analysis, Graph Neural Networks, Heterogeneous Graphs, Fuzzy Logic, Node Classification, Link Prediction.

Résumé

Les réseaux criminels constituent une classe de systèmes complexes, dynamiques et clandestins dont l'analyse représente un défi majeur pour les méthodes traditionnelles fondées sur les graphes et l'apprentissage automatique. Les approches existantes reposent principalement sur des représentations statiques à couche unique, qui ne parviennent pas à capturer la nature hétérogène et multi-relationnelle des interactions criminelles, et souffrent d'une incertitude accumulée inhérente à l'incomplétude des données judiciaires. Cette thèse propose un cadre combinant les réseaux de neurones sur graphes hétérogènes et les techniques de logique floue pour traiter deux tâches fondamentales de l'analyse des réseaux criminels : la classification des nœuds et la prédiction de liens. La représentation par graphe hétérogène préserve les distinctions sémantiques entre les différentes couches relationnelles, incluant les liens sociaux, opérationnels et de communication, tandis que la logique floue est utilisée pour renforcer la fiabilité des preuves collectées en modélisant et atténuant l'incertitude inhérente aux données des réseaux criminels. Le cadre proposé est évalué sur deux jeux de données réels : le réseau terroriste de Noordin Top et le réseau de la Mafia sicilienne. Les résultats expérimentaux démontrent l'efficacité de l'approche, atteignant un score Macro F1 de 71% sur la tâche de classification des nœuds, surpassant les méthodes existantes à couche unique et à graphes homogènes.

Mots-clés : Analyse de Réseaux Criminels, Réseaux de Neurones sur Graphes, Graphes Hétérogènes, Logique Floue, Classification de Nœuds, Prédiction de Liens.

ملخص

تُمثّل الشبكات الإجرامية فئةً من الأنظمة المعقدة والديناميكية والمتخفية، التي يُشكّل تحليلها تحدياً جوهرياً أمام الأساليب التقليدية المستندة إلى نظرية البيانات والتعلم الآلي. تعتمد الأساليب القائمة بصفة رئيسية على تمثيلات ثابتة أحادية الطبقة، التي تعجز عن استيعاب الطابع المتغير ومتعدد العلاقات للتفاعلات الإجرامية، فضلاً عن معاناتها من تراكم حالة عدم اليقين الناجمة عن نقص بيانات التحقيقات. تقترح هذه الأطروحة إطاراً يجمع بين شبكات الأعصاب الاصطناعية على البيانات غير المتجانسة وتقنيات المنطق الضبابي، لمعالجة مهمتين أساسيتين في تحليل الشبكات الإجرامية: تصنيف العُقد والتنبؤ بالروابط. يحافظ تمثيل البيان غير المتجانس على الفروق الدلالية بين طبقات العلاقات المتعددة، بما فيها الروابط الاجتماعية والتشغيلية والتواصلية، في حين يُوظف المنطق الضبابي لتعزيز موثوقية الأدلة المجمعة من خلال نمذجة حالة عدم اليقين والتخفيف منها. يُقيّم الإطار المقترح على مجموعتي بيانات حقيقيتين: شبكة الإرهابي نوردين توب وشبكة المافيا في 71% بنسبة Macro F1 الصقلية. تُثبت النتائج التجريبية فعالية المقاربة المقترحة، إذ حققت درجة مهمة تصنيف العُقد، متفوقةً على الأساليب القائمة على تمثيلات أحادية الطبقة والبيانات المتجانسة.

الكلمات المفتاحية: تحليل الشبكات الإجرامية، شبكات الأعصاب على البيانات، البيانات غير المتجانسة، المنطق الضبابي، تصنيف العُقد، التنبؤ بالروابط، الشبكات متعددة الطبقات.

Contents

1	Introduction	15
1.1	General Introduction to the Research Context	15
1.2	Challenges in Criminal Network Analysis	15
1.3	Research Objectives	16
1.4	Thesis Organization	18
2	Background and Theoretical Foundations	20
2.1	Network Science Fundamentals	20
2.1.1	History of Network Science and Graph Theory Roots	20
2.1.2	Basic Concepts	21
2.1.3	Graph Types	23
2.1.4	Graph Metrics	25
2.1.5	Network Models	28
2.1.6	Multilayer and Temporal Networks	29
2.2	Graph Data and Network Types	31
2.3	Criminal Network Data	32
2.3.1	Characteristics of Criminal Network Data	32
2.4	Fuzzy Logic and Uncertainty Modelling	34
2.4.1	Definition and Historical Context	34
2.4.2	Fuzzy Sets and Membership Functions	35
2.4.3	Fuzzy Logic Operations	36
2.4.4	Fuzzy Aggregation	37
2.4.5	Fuzzy Logic in Graph-Based Criminal Network Analysis	38

2.5	Machine Learning	39
2.5.1	Definition	39
2.5.2	Applications to Network Data	40
2.5.3	Applications in Criminal Network Analysis	40
2.6	Graph Neural Networks	41
2.6.1	Definition	41
2.6.2	Message Passing Mechanism	41
2.6.3	Limitations	42
2.7	Graph Convolutional Networks	42
2.7.1	Definition	42
2.7.2	Mechanism	43
2.8	Graph Attention Networks	44
2.8.1	Definition	44
2.8.2	Attention Mechanism	44
2.9	Conclusion	46

3 State of the Art in Criminal Network Analysis 48

3.1	Criminal Network Layers	48
3.2	Criminal Network Representations	50
3.2.1	Single-Layer Criminal Networks	50
3.2.2	Heterogeneous Graph Representation	50
3.2.3	Multilayer Criminal Networks	51
3.3	Criminal Network Analysis Tasks	51
3.3.1	Node Classification	51
3.3.2	Link Prediction	52

3.4	Traditional and Statistical Methods	52
3.4.1	Centrality-Based Approaches	52
3.4.2	Similarity-Based Approaches	53
3.5	Machine Learning Approaches	54
3.6	Graph Neural Network Approaches	56
3.6.1	Graph Convolutional Networks for Node Classification	56
3.6.2	Graph Attention Networks for Link Prediction	57
3.7	Discussion and Research Gaps	58
3.8	Conclusion	60
4	Experimental Study	62
4.1	Experimental Objectives	63
4.2	Datasets	65
4.2.1	Description of Criminal Network Datasets	65
4.2.2	Dataset Statistics	67
4.3	Preprocessing Steps	68
4.3.1	Noordin Top Terrorist Network	68
4.3.2	Sicilian Mafia Network	68
4.4	Graph Construction	69
4.4.1	Layer Representation	69
4.4.2	Heterogeneous Graph Construction	70
4.5	Fuzzy Reliability Modelling	71
4.6	Add Features	72
4.7	Data Splitting	73
4.8	Baseline Methods	74

4.8.1	Machine Learning Baselines	74
4.9	Proposed Method	76
4.9.1	Fuzzy Reliability Modelling	76
4.9.2	Heterogeneous Learning Framework	76
4.10	Experimental Tasks	79
4.10.1	Leader Identification (Node Multi-Classification)	79
4.10.2	Link Prediction (Future Collaboration Inference)	79
4.11	Evaluation Metrics	80
4.12	Results	81
4.12.1	Node Classification: Comparison Against Baseline Approaches	81
4.12.2	Link Prediction: Comparison of GNN Encoders	83
4.13	Discussion	85
4.13.1	Interpretation of Results	85
4.13.2	Advantages of the Proposed Approach	86
4.13.3	Limitations	87
4.14	Conclusion	88

Conclusion and Future Directions	90
---	-----------

List of Figures

2.1 A path $P = P^6$ in $G^{[13]}$ 22

2.2 A simple graph with three clusters^[15]. 23

2.3 Directed graph. 23

2.4 Weighted graph. 24

2.5 Multigraph. 24

2.6 Bipartite graph. 25

2.7 A low-degree vertex with high betweenness^[19]. 26

3.1 Architecture of the Graph Convolutional Network for node role classification^[45]. 57

4.1 Experimental Study Pipeline. 64

4.2 Architecture of the proposed Heterogenous Graph Convolutional Network (HeteroGCN) model. 78

List of Tables

3.1	Summary of Related Work in Criminal Network Analysis	59
4.1	Summary Statistics for the Criminal Network Datasets.	67
4.2	Stratified data split for node classification experiments.	74
4.3	Node Multi-Classification Results on the Test Set (29 nodes).	82
4.4	Confusion matrix for HeteroGCN on the test set (29 nodes).	83
4.5	Link Prediction Results. GNN models evaluated on the held-out test edge set (918 edge-label pairs); ML baselines via 5-fold cross-validation. . . .	84

List of Abbreviations

AUC Area Under the ROC Curve.

CNN Convolutional Neural Network.

CNs Criminal networks.

GAT Graph Attention Network.

GCN Graph Convolutional Network.

GNNs Graph Neural Networks.

GraphSAGE Graph Sample and Aggregated.

HeteroGAT Heterogenous Graph Attention Network.

HeteroGCN Heterogenous Graph Convolutional Network.

HeteroGraphSAGE Heterogenous Graph Sample and Aggregated.

ML Machine Learning.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

SNA Social Network Analysis.

Chapter 1

Introduction

1 Introduction

1.1 General Introduction to the Research Context

Criminal networks (CNs) operate as complex, adaptive systems characterized by covert relationships that transcend traditional hierarchical structures. The socio-economic impact of these networks is profound, human trafficking alone affects millions and generates hundreds of billions of dollars annually^[1], while illicit financial flows move trillions across the global economy^[2]. Consequently, disrupting these ecosystems requires robust and scalable analytical approaches.

However, the empirical analysis of CNs presents significant methodological challenges. Because these organizations are inherently deceptive, the available relational data is frequently fragmented, noisy, and uncertain^[3;4]. Furthermore, CNs exhibit high structural plasticity, enabling rapid reorganization in response to law enforcement interventions^[5]. This dynamic evolution of roles and actor dependencies complicates traditional statistical modeling, necessitating advanced analytical frameworks capable of capturing complex, multi-relational topologies^[6;7].

1.2 Challenges in Criminal Network Analysis

The review of existing approaches highlights several persistent challenges that hinder effective analysis of CNs:

- **Loss of Layer Semantics.** Fusing multiple relational layers into a single aggregated graph inevitably discards the semantic distinctions between interaction types. The resulting representation obscures the functional roles of individual layers and may produce structurally misleading conclusions.

- **Limitations of Single-Model Approaches.** Existing methods predominantly rely on a single model architecture applied uniformly across all tasks and representations. This monolithic design limits adaptability, as no single model is equally suited to the heterogeneous structural properties exhibited by different criminal network datasets.
- **Accumulated Uncertainty in Multilayer Fusion.** Each individual layer of a CNs is inherently subject to a degree of uncertainty, stemming from incomplete observations, hidden links, and data collection biases. Merging multiple such layers into a single unified graph compounds these individual uncertainties. The resulting representation therefore exhibits substantially lower reliability than any constituent layer considered in isolation.

1.3 Research Objectives

Motivated by the challenges identified above, the present work pursues the following objectives:

- **Enhancing Evidence Reliability.** To increase the reliability of gathered criminal evidence through the integration of fuzzy logic techniques, which provide a principled mechanism for handling uncertainty and imprecision inherent in CNs data.
- **Preserving Layer Semantics.** To overcome the semantic loss associated with layer fusion by converting the multilayer CNs into a heterogeneous graph representation, wherein each layer is preserved as a distinct relational type, maintaining its structural and functional identity.

- **Improving Predictive Performance.** To achieve superior accuracy in downstream analytical tasks—namely node classification and link prediction—by leveraging the proposed architecture, and to rigorously validate these improvements using the Micro F1-score to ensure robustness against imbalanced class distributions.

1.4 Thesis Organization

This thesis is structured to provide a systematic progression from theoretical foundations to empirical validation. Chapter 2 establishes the necessary background by introducing fundamental concepts in graph theory, network science, and ML. Chapter 3 provides a comprehensive review of the state-of-the-art, identifying critical gaps and methodological limitations in existing CNs analysis approaches. Chapter 4 forms the core contribution of this research. It details the architecture of the proposed fuzzy-based heterogeneous Graph Neural Networks (GNNs), presents a rigorous empirical evaluation, describes the datasets employed, and provides an in-depth discussion of the experimental results.

Chapter 2

Background and Theoretical Foundations

2 Background and Theoretical Foundations

This chapter establishes the theoretical foundations upon which the proposed framework is built. We begin with the fundamental concepts of Network Science and Graph Theory, covering basic graph definitions, graph types, structural metrics, and network models, before extending these notions to multilayer and temporal networks. We then introduce the core principles of ML and their natural application to graph-structured data. Finally, we present the family of Graph Neural Network architectures that constitute the backbone of the present work, with particular emphasis on Graph Convolutional Network (GCN) and Graph Attention Network (GAT).

2.1 Network Science Fundamentals

2.1.1 History of Network Science and Graph Theory Roots

The mathematical study of networks traces its origins to 1736, when Leonhard Euler resolved the *Seven Bridges of Königsberg* problem, laying the foundational principles of graph theory. During the nineteenth century, the field was further advanced by Gustav Kirchhoff (1847), who applied graph-theoretic concepts to electrical circuits, and by Arthur Cayley (1878), who exploited tree structures in chemical analysis. In the 1930s, Jacob Moreno introduced the *sociogram*, marking the emergence of Social Network Analysis (SNA) as a formal discipline. A pivotal theoretical contribution followed in 1959, when Paul Erdős and Alfréd Rényi proposed the random graph model, providing the first rigorous probabilistic framework for network analysis.

In 1973, Mark Granovetter published his seminal work on the *strength of weak ties*, demonstrating the structural significance of sparse inter-community connections in

social networks^[10]. The field underwent a decisive transformation in 1998, when Duncan Watts and Steven Strogatz introduced the small-world network model^[11], followed in 1999 by Albert-László Barabási and Réka Albert, who formulated the scale-free network model through the mechanism of *preferential attachment*^[12], thereby establishing the foundations of modern network science.

Crucially, these historical milestones hold direct implications for CNs analysis. For instance, the scale-free property aptly characterizes the topological structure of centralized syndicates (e.g., drug cartels), where a few prominent actors (hubs) maintain disproportionately high connectivity. Similarly, Granovetter's weak ties explain how decentralized criminal cells exchange illicit commodities and information with minimal structural exposure. However, while these foundational models elegantly describe macroscopic network topologies, their static nature falls short of capturing the dynamic, multi-relational, and uncertain semantics inherent to modern criminal ecosystems. This limitation has fundamentally necessitated the contemporary shift from traditional network science toward representation learning and Graph Artificial Intelligence.

2.1.2 Basic Concepts

Graph: A simple graph is an ordered pair $G = (V, E)$, where V is a finite non-empty set of elements called vertices, and $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ is a set of unordered pairs of distinct vertices called edges. Each edge $\{u, v\} \in E$ represents a connection between the vertices u and v ^[13].

Vertex: Also known as a node, a vertex is the fundamental unit or point of which a graph is formed. The set of vertices of a graph G is denoted by $V(G)$ ^[14].

Edge: An edge is a link between two vertices. If $u, v \in V$, then the edge $e = \{u, v\}$

represents the connection between these two nodes^[14].

Degree of a Vertex: The *degree* (or *valency*) of a vertex v , denoted by $d_G(v)$ or simply $d(v)$, is defined as the number of edges incident to v . Equivalently, it is the cardinality of its neighbourhood:

$$d(v) = |N(v)|.$$

A vertex of degree zero is called an *isolated vertex*^[13].

Walk: A *walk* of length k in a graph G is a non-empty alternating sequence

$$v_0 e_0 v_1 e_1 \dots e_{k-1} v_k$$

of vertices and edges in G , such that $e_i = \{v_i, v_{i+1}\}$ for all $i < k$ ^[13].

Path: A *path* is a non-empty graph $P = (V, E)$ whose vertex set can be written as an ordered sequence $V = \{x_0, x_1, \dots, x_k\}$ such that its edge set is $E = \{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\}$, where x_0 and x_k are said to be linked by P , and are referred to as the *endvertices* of the path.^[13]



Figure 2.1: A path $P = P^6$ in G ^[13].

Cluster: A *cluster*, also referred to as a *community* or *module*, is a group of vertices that share common properties and/or play similar functional roles within the graph^[15].

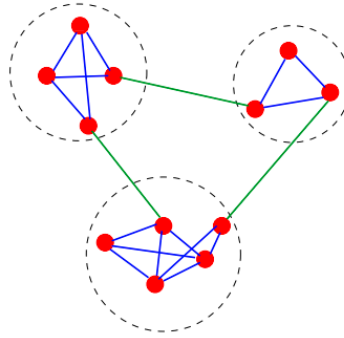


Figure 2.2: A simple graph with three clusters^[15].

2.1.3 Graph Types

Undirected Graph: A graph in which all edges are undirected is called an *undirected graph*^[16].

Directed Graph: A *directed graph* (or *digraph*) is a pair $G = (V, E)$ equipped with two maps, $init : E \rightarrow V$ and $ter : E \rightarrow V$, which assign to each edge $e \in E$ its initial vertex $init(e)$ and its terminal vertex $ter(e)$, respectively^[13].

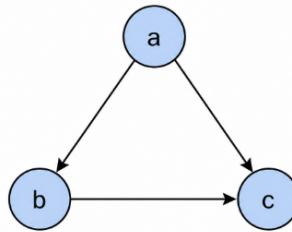


Figure 2.3: Directed graph.

Weighted Graph: A *weighted graph* (also called an *edge-labeled graph*) is defined as a triple $G = (V, E, w)$, where $w : E \rightarrow \mathbb{R}$ is a weight function that assigns to each edge $e \in E$ a real-valued weight $w(e)$, reflecting the strength, capacity, or cost associated with that connection^[17].

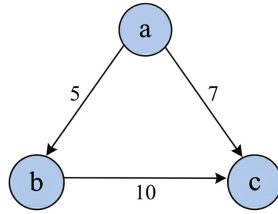


Figure 2.4: Weighted graph.

Multigraph: A *multigraph* is a pair (V, E) of disjoint sets of vertices and edges, together with a map $E \rightarrow V \cup [V]^2$ that assigns to each edge either one or two vertices as its endpoints. Unlike simple graphs, a multigraph permits the existence of loops (edges whose both endpoints coincide) as well as multiple edges between the same pair of vertices^[13].

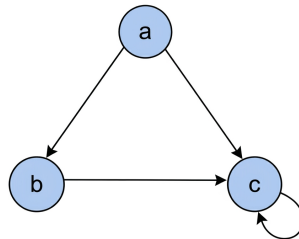


Figure 2.5: Multigraph.

Multipartite and Bipartite Graphs: Let $r \geq 2$ be an integer. A graph $G = (V, E)$ is called *r-partite* if its vertex set V admits a partition into r classes $V_1, V_2, \dots, V_r$ such that every edge $e \in E$ has its endpoints in different partition classes, that is, no two adjacent vertices belong to the same class. In the special case $r = 2$, the graph is referred to as a *bipartite* graph^[13].

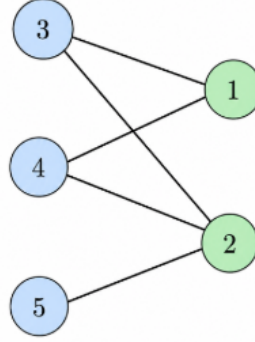


Figure 2.6: Bipartite graph.

Heterogeneous Graph: A heterogeneous graph is defined as a tuple $\mathcal{G} = (V, E, \mathcal{T}^v, \mathcal{T}^e)$, where V is the set of nodes equipped with a node-type mapping function $\phi : V \rightarrow \mathcal{T}^v$, and E is the set of edges equipped with an edge-type mapping function $\psi : E \rightarrow \mathcal{T}^e$. Each node $v_i \in V$ is assigned a node type $c_i = \phi(v_i) \in \mathcal{T}^v$, and each edge $e_{t \leftarrow s} \in E$ is associated with a relation type $r_{c_t \leftarrow c_s} = \psi(e_{ts}) \in \mathcal{T}^e$, directed from source node s to target node t . When $|\mathcal{T}^v| = |\mathcal{T}^e| = 1$, the heterogeneous graph degenerates into a homogeneous graph^[18].

2.1.4 Graph Metrics

Centrality Metrics: Centrality metrics, rooted in SNA, are used to identify the most influential or structurally important vertices within a graph. The three most widely adopted measures are Degree Centrality, Betweenness Centrality, and Closeness Centrality^[8].

Degree Centrality: The simplest centrality measure, which quantifies the importance of a vertex based on its number of direct connections^[19].

Betweenness Centrality: Captures a distinct notion of importance: a vertex may be structurally significant not because it has many connections, but because it lies on the shortest paths between other vertices, acting as a critical bridge or intermediary in the

network. Formally, the betweenness centrality of a vertex D is defined as:

$$C_B(D) = \sum_{s \neq t} n_{st}^D \quad (2.1)$$

where the sum is taken over all ordered pairs (s, t) of distinct vertices, and n_{st}^D denotes the number of shortest paths between s and t that pass through vertex D ^[19]. A high value of $C_B(D)$ indicates that D plays a pivotal role in facilitating communication and information flow across the graph, regardless of its degree.

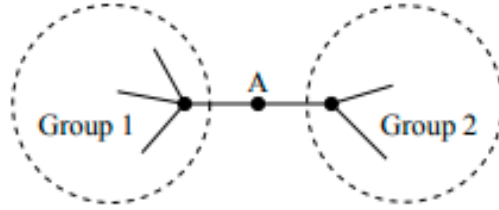


Figure 2.7: A low-degree vertex with high betweenness^[19].

Closeness Centrality: Unlike the previously discussed measures, Closeness Centrality does not rely on local connectivity; rather, it quantifies how close a vertex v_i is to all other vertices in the graph $\mathcal{G} = (V, E)$. Formally, it is defined as:

$$C_i = \frac{n}{\sum_j d_{ij}} \quad (2.2)$$

where d_{ij} denotes the length of the shortest path between vertex v_i and vertex v_j , and $n = |V|$ is the total number of vertices^[19]. The closeness centrality score is bounded such that:

$$0 \leq C_i \leq 1 \quad (2.3)$$

A value of C_i approaching 1 indicates that vertex v_i is reachable from all other vertices via a large number of short paths, reflecting a highly central position within the network.

Clustering and Community Detection Metrics: Several metrics and algorithms have been proposed to detect communities within graphs. Among the most widely adopted methods are Graph Partitioning, Hierarchical Clustering, and Modularity-Based methods^[15].

Flow-Based Metrics: Let $\mathcal{G} = (V, E, w)$ be a weighted directed graph, where each edge $(u, v) \in E$ is assigned a non-negative capacity $w(u, v) \geq 0$. Given two designated vertices $s, t \in V$, referred to as the source and the termin respectively, flow-based metrics quantify the capacity of the network to route flow between them. Two fundamental problems arise in this context:

- **Maximum Flow:** determines the largest possible flow that can be routed from s to t without violating any edge capacity constraint. This problem is efficiently solved by classical algorithms such as Ford–Fulkerson.
- **Minimum Cut:** identifies the minimal set of edges whose removal disconnects s from t , thereby partitioning the graph into two disjoint components. By the Max-Flow Min-Cut Theorem, the maximum flow from s to t is equal to the capacity of the minimum cut separating them^[15].

Similarity and Comparison Metrics: Originally derived from SNA, similarity metrics aim to quantify the resemblance between vertices in a graph $\mathcal{G} = (V, E)$. Two principal notions of vertex similarity are distinguished: Structural Equivalence, which holds between two vertices if they share a significant number of common neighbors, and Regular Equivalence, which holds if their respective neighbors are themselves equivalent. The

most widely used measures for capturing such similarity include: Cosine Similarity, Pearson Correlation Coefficient^[15].

2.1.5 Network Models

Random Networks: A random network $G(N, p)$ is a graph of N nodes where each pair of nodes is connected with probability p ^[20].

Small-World Network: A network is said to exhibit the small-world property if its average path length $\langle d \rangle$ scales logarithmically with the network size N . According to^[20], this relationship is approximated by:

$$\langle d \rangle \approx \frac{\ln N}{\ln \langle k \rangle} \quad (2.4)$$

where $\langle k \rangle$ is the average degree. This property ensures that information or influence can spread across the entire system in a very small number of steps.

Real-World Networks: Real-world networks are complex systems whose architecture is governed by organizing principles such as growth and preferential attachment. They are distinguished by the presence of hubs (nodes with an exceptionally high number of links) and a degree distribution that follows:

$$P(k) \sim k^{-\gamma} \quad (2.5)$$

where γ is the degree exponent, typically observed in the range $2 < \gamma < 3$ for most biological and social systems^[20].

2.1.6 Multilayer and Temporal Networks

Multilayer Networks: A conventional graph $\mathcal{G} = (V, E)$ models a system through a single layer of interactions, where all edges are assumed to be homogeneous in nature. However, many real-world complex systems involve multiple types of relationships simultaneously, such that edges exhibit heterogeneous features that cannot be faithfully captured within a single-layer representation. Such systems are collectively referred to as Multilayer Networks.

For instance, in a social network, two individuals may be connected through several distinct types of relationships at once, such as friendship, professional collaboration, or family ties, each constituting a separate layer of interaction.

A **Multilayer Network**^[21] is formally defined as a tuple:

$$\mathcal{M} = (V_M, E_M, V, \mathcal{L}) \quad (2.6)$$

where V is the global set of nodes, $\mathcal{L} = \{L_1, L_2, \dots, L_m\}$ is the set of elementary-layer sets corresponding to each aspect of the network, $V_M \subseteq V \times L_1 \times L_2 \times \dots \times L_m$ is the set of node-layer tuples, and $E_M \subseteq V_M \times V_M$ is the set of edges, partitioned into two categories:

- **Intra-layer edges:** edges connecting two nodes within the same layer, capturing interactions of a homogeneous type.
- **Inter-layer edges:** edges connecting a node in one layer to a node in a different layer, representing the coupling between distinct interaction types.

The underlying graph $G_M = (V_M, E_M)$, known as the supra-graph, admits an adjacency matrix representation referred to as the supra-adjacency matrix.

Temporal Networks: A temporal network is a dynamic framework where interactions between nodes are characterized by their sequence, duration, and concurrency. Unlike static graphs, these networks are defined as a set of time-ordered interactions:

$$G_t = \{(i, j, t, \Delta t)\} \quad (2.7)$$

where an edge exists between nodes i and j at time t for a duration Δt , and where $t \in \mathbb{R}$ or $\mathbb{Z}$ depending on the temporal resolution of the system. In a multilayer context, they are often represented as a sequence of snapshots, where each layer α corresponds to the state of the system at a specific time interval t_α ^[22].

Snapshot-based Networks: This representation models a temporal network as a discrete sequence of static graphs, where each "snapshot" α captures all interactions occurring within a specific time interval $[t_\alpha, t_\alpha + \Delta t]$. This approach is highly compatible with the multilayer framework, as each time window is treated as a separate layer, allowing for the application of supra-adjacency matrix formalism. However, this representation relies on temporal aggregation and may introduce temporal aggregation bias, potentially obscuring the true ordering of interactions.

Continuous-time Networks: Unlike snapshots, this model treats interactions as a stream of discrete events (i, j, t) occurring at precise points in time. This representation preserves the fine-grained temporal ordering of interactions and minimizes information loss, avoiding the artificial concurrency that can arise when aggregating events into time windows or snapshots.

2.2 Graph Data and Network Types

Unlike traditional tabular data, which treats observations as independent instances, graph representations provide a structural approach for modeling entities and the relationships connecting them. This is achieved by explicitly encoding relational structures through nodes and edges, which may also carry associated attributes to represent high-dimensional information. This structural expressiveness makes graph frameworks highly adaptable to various real-world systems, such as:

- **Social Networks:** Modeling friendship, influence, or communication ties on digital platforms.
- **Communication Networks:** Capturing data flow between servers and devices in telecommunications infrastructure.
- **CNs:** Representing co-offending relationships, financial transactions, or communication links among suspects.

Depending on the complexity of the underlying data, these structures generally fall into distinct representational categories:

1. **Homogeneous Networks:** Structures where all nodes and edges share a single type (e.g., a simple call network among suspects). While easier to process, they often sacrifice relational richness.
2. **Heterogeneous Networks:** Structures comprising multiple node and edge types simultaneously (e.g., suspects, bank accounts, and locations connected through transaction and communication links). These representations capture the multi-relational nature of real-world systems more faithfully.

3. **Multi-layer Networks:** Models in which the same set of actors connects across distinct relational layers (e.g., simultaneous phone calls, financial transfers, and kinship ties), providing a holistic view of interdependent interactions.

In the context of CNs analysis, actors engage through multiple, heterogeneous channels simultaneously, making homogeneous single-layer representations fundamentally insufficient. This observation directly motivates the adoption of heterogeneous graph frameworks as the structural foundation of the analytical approach proposed in this thesis.

2.3 Criminal Network Data

Criminal networks exhibit unique structural properties, including incompleteness, incorrectness, and dynamic evolution, which are discussed in detail in Section 2.3.

2.3.1 Characteristics of Criminal Network Data

Criminal network data is fundamentally different from conventional relational data. As first identified by Sparrow^[8] and further elaborated in the context of intelligence analysis^[9], such data exhibits a set of structural and quality challenges that directly affect any analytical or computational approach. The most critical characteristics are outlined below:

- **Incompleteness:** CNs are rarely fully observed. Covert actors deliberately conceal their identities and associations, resulting in datasets where nodes (individuals) and edges (relationships) are systematically missing^[8]. This incompleteness is not random; it reflects the deliberate operational security practiced by criminal

organizations, making missing data imputation and structural inference particularly challenging.

- **Incorrectness:** Available data may contain erroneous information, whether due to deliberate disinformation by criminal actors, misidentification during surveillance, or transcription errors from informant reports^[8;9]. Such noise introduces uncertainty into both the graph structure and the associated node and edge attributes.
- **Inconsistencies:** Data collected from multiple sources — such as wiretaps, financial records, and field observations — often yield conflicting or contradictory relational information^[9]. Reconciling these inconsistencies requires robust data integration strategies prior to any graph-based modeling.
- **Size:** CNs can involve a large number of actors and interactions, posing significant scalability challenges for both graph construction and computational analysis^[9]. At the same time, networks of interest may also be exceptionally small, limiting the applicability of data-hungry learning methods.
- **Fuzzy Boundaries:** It is rarely possible to determine precisely where a CNs begins and ends. Individuals may simultaneously belong to multiple overlapping criminal groups, and peripheral actors may shift between legitimate and illicit roles over time^[8;9]. This structural ambiguity complicates both network delineation and community detection.
- **Dynamic Nature:** CNs are not static structures. Relationships emerge, dissolve, and transform in response to law enforcement pressure, internal conflicts, or strategic reorganization^[8]. This temporal evolution requires analytical frameworks capable of capturing structural change over time, rather than relying on a single

static snapshot.

2.4 Fuzzy Logic and Uncertainty Modelling

2.4.1 Definition and Historical Context

Classical logic operates under the principle of bivalence: every proposition is either entirely true (1) or entirely false (0), with no intermediate state. While this binary framework is well-suited to deterministic domains, it proves fundamentally inadequate for reasoning about real-world phenomena that are inherently imprecise, ambiguous, or uncertain. Fuzzy logic, introduced by Lotfi A. Zadeh in his seminal 1965 paper^[23], was designed precisely to address this limitation. Rather than forcing every element into a crisp binary category, fuzzy logic allows partial membership in a set, expressed as a degree in the continuous interval $[0, 1]$.

Zadeh’s central insight was that the boundaries between categories in natural language and real-world data are rarely sharp. A person who is “tall” does not suddenly become “not tall” at a precise height threshold; a connection between two criminal actors is not unambiguously “strong” or “weak” based on a single binary observation. Fuzzy logic formalises this graduated uncertainty through the concept of *fuzzy sets* and *membership functions*, providing a mathematically rigorous yet intuitively natural framework for handling imprecision.

Since its introduction, fuzzy logic has found widespread application in control systems, decision support, natural language processing, and, more recently, graph-based data analysis. Its adoption in criminal network analysis is particularly well-motivated, given that the data characterising such networks is systematically incomplete, noisy, and subject to conflicting interpretations (Section 2.3).

2.4.2 Fuzzy Sets and Membership Functions

Crisp sets vs. fuzzy sets. In classical (crisp) set theory, the membership of an element x in a set A is defined by a characteristic function:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \quad (2.8)$$

A **fuzzy set** $\tilde{A}$ generalises this to a *membership function* $\mu_{\tilde{A}} : X \rightarrow [0, 1]$, where X is the universe of discourse. The value $\mu_{\tilde{A}}(x) \in [0, 1]$ expresses the *degree* to which element x belongs to $\tilde{A}$:

$$\tilde{A} = \{ (x, \mu_{\tilde{A}}(x)) \mid x \in X \} \quad (2.9)$$

A membership degree of 0 indicates complete non-membership, 1 indicates full membership, and any intermediate value represents partial membership. For example, in the context of criminal tie strength, $\mu_{\tilde{A}}(\text{pair}_{ij}) = 0.75$ would indicate that the connection between actors i and j is a relatively strong but not absolute tie.

Common membership function forms. The shape of a membership function encodes the semantics of the fuzzy concept being modelled. Three forms are most widely used in practice:

- **Triangular:** Defined by three parameters (a, b, c) representing the lower bound,

peak, and upper bound:

$$\mu(x; a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x - a}{b - a} & a < x \leq b \\ \frac{c - x}{c - b} & b < x \leq c \\ 0 & x > c \end{cases} \quad (2.10)$$

- **Trapezoidal:** Extends the triangular form with a flat top between two inner parameters (a, b, c, d) , allowing a range of full membership rather than a single peak. Particularly suited to modelling concepts with a “plateau” of certainty.
- **Gaussian:** Defined by a centre c and width σ :

$$\mu(x; c, \sigma) = \exp\left(-\frac{(x - c)^2}{2\sigma^2}\right) \quad (2.11)$$

Gaussian functions produce smooth, continuous transitions and are well-suited to modelling graded phenomena without hard boundaries.

2.4.3 Fuzzy Logic Operations

Fuzzy logic extends the standard Boolean operations (AND, OR, NOT) to operate on membership degrees. The most commonly used definitions are:

$$\text{Fuzzy AND (T-norm): } \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.12)$$

$$\text{Fuzzy OR (S-norm): } \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.13)$$

$$\text{Fuzzy NOT (complement): } \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.14)$$

Alternative T-norms (e.g., the product T-norm $\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x)$) are also widely used, particularly when independence between fuzzy conditions is assumed. The choice of T-norm and S-norm affects the aggregation behaviour and should be guided by the semantics of the application domain.

2.4.4 Fuzzy Aggregation

In many practical applications, multiple fuzzy criteria must be combined into a single reliability or confidence score. *Fuzzy aggregation operators* perform this combination in a principled way. The weighted average aggregation, used in this thesis, takes the form:

$$\mu_{\text{agg}}(x) = \sum_{k=1}^K w_k \mu_k(x), \quad \text{where } \sum_{k=1}^K w_k = 1, \quad w_k \geq 0 \quad (2.15)$$

where $\mu_k(x)$ is the k -th membership function evaluated at x , and w_k is the importance weight assigned to the k -th criterion. This convex combination preserves the $[0, 1]$ range of the output and allows the analyst to encode domain knowledge about the relative importance of each criterion.

2.4.5 Fuzzy Logic in Graph-Based Criminal Network Analysis

The application of fuzzy logic to graph-based modelling addresses a fundamental limitation of binary adjacency representations: they treat every recorded connection as equally certain and equally strong, discarding the rich gradations of reliability that distinguish high-quality intelligence from speculative inference.

In the context of criminal network analysis, several aspects of the relational data are inherently fuzzy:

- **Connection strength:** Two actors who co-appear in many shared activities are more reliably connected than two who share only a single, potentially coincidental contact.
- **Structural position:** The significance of a tie is modulated by the structural context of its endpoints—a connection between two peripheral actors carries different weight than one between two central figures.
- **Source reliability:** Information derived from direct surveillance may be more reliable than that inferred from informant reports or financial records, which may themselves be incomplete or deliberately falsified.

By encoding these gradations as fuzzy edge weights in $[0, 1]$, the graph model preserves the uncertainty inherent in the original data rather than discarding it through binary thresholding. This directly improves the quality of the node representations learned by downstream graph neural networks, which can then exploit both the structural topology and the reliability-weighted edge signals during message passing.

In this thesis, three fuzzy membership components μ_1 , μ_2 , and μ_3 are defined to capture co-occurrence frequency, degree-normalised rarity, and degree homophily,

respectively (Chapter 4, Section 4.5). These are combined via a weighted average aggregation (Equation 2.15) with weights $w_1 = 0.5$, $w_2 = 0.25$, $w_3 = 0.25$, producing a single scalar reliability score per edge per relational layer. This fuzzy weighting scheme constitutes one of the key design contributions of the proposed framework, providing a theoretically grounded and interpretable mechanism for handling the uncertainty that is endemic to criminal network data.

2.5 Machine Learning

2.5.1 Definition

ML is a discipline concerned with the design of algorithms capable of automatically extracting patterns from data^[24]. These algorithms operate on a collection of observations known as the training set, from which a trained model is derived. The central objective of such a model is generalization: the ability to produce accurate predictions on previously unseen data. Depending on the nature of the training data and the learning objective, ML methods are broadly categorized into Supervised Learning, Unsupervised Learning, and Semi-Supervised Learning.

Supervised Learning: The training set consists of input vectors paired with their corresponding labels or target values^[24], formally denoted as $\{(\mathbf{x}_i, t_i)\}_{i=1}^N$, where $\mathbf{x}_i$ is the input vector and t_i is the associated target. The goal is to learn a mapping $f : \mathbf{x} \mapsto t$ that generalizes accurately to unseen data. Among its most prominent subcategories are:

- **Classification:** where the target t_i belongs to a finite set of discrete categories.
- **Regression:** where the target t_i is a continuous-valued variable.

Unsupervised Learning: The training set consists solely of input vectors $\{\mathbf{x}_i\}_{i=1}^N$ with no associated labels or target values^[24]. The objective of the model is to uncover the latent structure underlying the data, through tasks such as clustering, density estimation, or dimensionality reduction.

Semi-Supervised Learning occupies an intermediate position between supervised and unsupervised learning. The training set comprises a small subset of labeled examples combined with a larger set of unlabeled data. The model exploits the underlying structure of the unlabeled data to improve generalization beyond what labeled data alone would permit^[25].

2.5.2 Applications to Network Data

ML techniques find natural application in the analysis of graph-structured data. In the supervised setting, structural properties of the network — such as centrality measures, degree distribution, and local neighborhood features — are used as input features for tasks such as vertex classification and link prediction. In the unsupervised setting, the inherent structural organization of the graph is exploited to identify groups of vertices that exhibit dense mutual interactions, revealing hidden organizational patterns within the network without any prior labeling.

2.5.3 Applications in Criminal Network Analysis

These principles extend naturally to the domain of CNs analysis, where ML is applied to graph-structured data representing illicit interactions. Supervised methods are employed for tasks such as key actor identification and criminal link prediction, while unsupervised methods are used for community detection, aiming to uncover hidden criminal organizations and coordinated behavioral patterns within the network. These

tasks are examined in greater depth in the next chapter.

2.6 Graph Neural Networks

2.6.1 Definition

GNNs are a class of deep learning models that extend the principles of Recurrent Neural Network (RNN) to operate on non-Euclidean and irregularly structured data^[26]. Unlike conventional deep learning architectures that assume data to lie in a regular grid (e.g., images or sequences), GNNs are designed to directly process graph-structured data. The core computational principle underlying GNNs is the message passing mechanism, whereby each node iteratively aggregates feature information from its local neighborhood to produce an updated representation, enabling the model to jointly capture both node attributes and topological structure^[27]. Several architectural variants have been proposed within this framework, among which the most prominent are GCN, GAT.

2.6.2 Message Passing Mechanism

The key computational engine driving GNNs is the message passing mechanism. At each iteration k , every node $u \in V$ aggregates the feature representations of its neighboring nodes $v \in \mathcal{N}(u)$, and subsequently updates its own representation based on this aggregated information. This process is formally expressed as:

$$\mathbf{h}_u^{(k+1)} = \text{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)}, \text{AGGREGATE}^{(k)}(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\})\right) \quad (2.16)$$

where $\mathbf{h}_u^{(k)}$ denotes the feature representation of node u at iteration k , $\mathcal{N}(u)$ is the set

of neighbors of u , and $AGGREGATE^{(k)}$ and $UPDATE^{(k)}$ are differentiable functions learned during training^[27].

2.6.3 Limitations

Despite their demonstrated effectiveness, GNNs are subject to several persistent limitations^[27]:

- **Imbalanced and Limited Labeled Data.** Available graph datasets frequently suffer from class imbalance, domain dependency, and a scarcity of labeled examples, which hinders the generalization of trained models. Potential remedies include transfer learning and domain adaptation techniques.
- **Accuracy of Existing Models.** Current GNNs architectures — including GCN, GAT, and Graph Sample and Aggregated (GraphSAGE) — still exhibit limitations in terms of efficiency and predictive precision, particularly on complex or heterogeneous graphs. Training on larger, domain-specific datasets is identified as a promising direction for performance improvement.
- **Text and Attributed Graph Classification.** Applying GNNs to text-rich or attributed graphs remains a non-trivial challenge, requiring more expressive architectures capable of jointly modeling structural and semantic information.

2.7 Graph Convolutional Networks

2.7.1 Definition

The **GCN**, introduced by Kipf & Welling^[28], is a deep learning architecture designed for graph-structured data, drawing inspiration from Convolutional Neural Network (CNN).

At each layer, every node updates its feature representation through an aggregation and propagation process over the features of its directly connected neighbors^[27].

2.7.2 Mechanism

The computational pipeline of a GCN proceeds through the following stages^[27]:

- **Initialization.** Each node $v_i \in V$ is associated with an initial feature vector $\mathbf{h}_i^{(0)} \in \mathbb{R}^d$, encoding node-level attributes relevant to the task at hand.
- **Convolution Operation.** At each layer k , the GCN aggregates information from the immediate neighborhood $\mathcal{N}(i)$ of each node by computing a weighted sum of neighboring feature vectors. The graph adjacency matrix $\mathbf{A}$ determines the aggregation weights, yielding an updated representation for each node.
- **Weighted Aggregation.** The adjacency matrix is normalized prior to aggregation to ensure that each neighbor's contribution is proportional to the local connectivity of the graph. Formally, the propagation rule of a single GCN layer is expressed as:

$$\mathbf{Z} = \sigma(\mathbf{A}'\mathbf{F}\mathbf{W} + \mathbf{b}) \quad (2.17)$$

where $\mathbf{A}' = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ is the normalized adjacency matrix with added self-loops, $\mathbf{F}$ is the node feature matrix, $\mathbf{W}$ is a trainable weight matrix, $\mathbf{b}$ is a bias vector, and σ denotes a non-linear activation function.

- **Activation Function and Weight Learning.** The aggregated features are passed through a non-linear activation function — typically Rectified Linear Unit (ReLU) — to introduce expressive non-linearity. The weight matrix $\mathbf{W}$ is learned end-to-end

during training, enabling the model to adapt to the specific graph structure and downstream task.

- **Layer Stacking.** Multiple GCN layers are stacked sequentially, such that the output $\mathbf{Z}^{(k)}$ of layer k serves as the input $\mathbf{F}^{(k+1)}$ to layer $k + 1$. This allows the network to capture increasingly complex structural patterns at progressively larger neighborhood scales.
- **Task-Specific Output.** The final layer produces node representations that are fed into a task-specific decoder for node classification, link prediction, or graph classification, depending on the application.

2.8 Graph Attention Networks

2.8.1 Definition

Classical graph-based architectures, such as GCN, assign uniform importance to all neighboring nodes during aggregation, or rely on computationally intensive spectral decompositions that do not generalize across graphs. **GAT**, introduced by Veličković et al.^[29], overcome this limitation by incorporating a self-attention mechanism, whereby each node learns to assign adaptive importance weights to its neighbors during the aggregation step^[27].

2.8.2 Attention Mechanism

The computational pipeline of a GAT proceeds through the following stages^[27]:

- **Initialization.** As with GCN, each node $v_i \in V$ is associated with an initial feature vector $\mathbf{h}_i^{(l)} \in \mathbb{R}^d$, encoding node-level attributes relevant to the task at hand.

- **Linear Transformation.** A shared linear transformation is first applied to every node feature vector:

$$\mathbf{z}_i^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{h}_i^{(l)} \quad (2.18)$$

where $\mathbf{W}^{(l)}$ is a trainable weight matrix at layer l .

- **Attention Coefficient Computation.** For each pair of neighboring nodes (v_i, v_j) , an unnormalized pairwise attention score is computed using additive attention:

$$e_{ij}^{(l)} = \text{LeakyReLU}\left(\vec{\mathbf{a}}^{(l)\top} \left[\mathbf{z}_i^{(l)} \parallel \mathbf{z}_j^{(l)}\right]\right) \quad (2.19)$$

where $\parallel$ denotes vector concatenation and $\vec{\mathbf{a}}^{(l)}$ is a learnable attention vector.

- **Softmax Normalization.** The raw attention scores are normalized across all neighbors of v_i using the softmax function, ensuring comparability across nodes:

$$\alpha_{ij}^{(l)} = \frac{\exp\left(e_{ij}^{(l)}\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(e_{ik}^{(l)}\right)} \quad (2.20)$$

- **Weighted Aggregation.** The normalized attention coefficients are used to compute a weighted aggregation of neighboring representations, yielding the updated node embedding:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} \mathbf{z}_j^{(l)}\right) \quad (2.21)$$

where σ denotes a non-linear activation function (ReLU).

- **Multi-Head Attention.** To enhance stability and capture diverse relational aspects, GAT employs K independent attention heads in parallel. Their outputs are

combined by concatenation in intermediate layers, or by averaging in the final layer.

- **Layer Stacking and Task-Specific Output.** Multiple GAT layers are stacked sequentially, enabling the model to capture higher-order structural dependencies. The final layer produces node representations used for downstream tasks such as node classification, link prediction, or graph classification.

2.9 Conclusion

This chapter provided the theoretical groundwork necessary for the developments presented in subsequent chapters. The fundamental concepts of graph theory and network science were formalized, encompassing graph types, structural metrics, and network models. These foundations were then extended to multilayer and temporal representations, which more faithfully reflect the dynamic and multi-relational nature of real-world systems. The chapter further introduced the principles of ML as applied to graph-structured data, and presented the core Graph Neural Network architectures — GCN and GAT — that underpin the proposed approach. Building upon these foundations, the following chapter reviews the existing literature on criminal network analysis and identifies the key research gaps that motivate this work.

Chapter 3

State of the Art in Criminal Network Analysis

3 State of the Art in Criminal Network Analysis

The analysis of CNs has emerged as a critical component of modern law enforcement and intelligence operations. Building upon the graph theoretical foundations established in the preceding chapter, this chapter provides a structured review of the existing literature on CNs analysis, spanning three generations of methodological approaches.

We begin by examining the structural organization of CNs, describing the multiple functional layers through which such organizations operate, and the three principal graph representations — single-layer, heterogeneous, and multiplex multilayer — employed in the literature. We then define the core analytical tasks addressed in this domain, namely node classification and link prediction. Subsequently, we survey the methods proposed to address these tasks, progressing from traditional and statistical approaches, through ML methods, to the most recent GNNs architectures. The chapter concludes with a critical discussion of the persistent research gaps that motivate the approach developed in the present work.

3.1 Criminal Network Layers

Criminal organizations can be conceptualized as *profit-driven enterprises*, operating through a deliberate and structured division of labor. Drawing on the organizational framework proposed by Zimmerman et al^[30], CNs can be understood as structured across *multiple functional layers*, where each layer concentrates individuals possessing specialized skills tailored to a specific dimension of criminal activity. Such a multilayer architecture enhances operational efficiency while simultaneously increasing resilience against law enforcement interventions and network neutralization efforts. The principal

layers of this architecture are identified as follows:

- **Social layer:** This layer represents informal social relationships between individuals, such as friendship, kinship, or trust-based ties. These relationships play a crucial role in the formation and stability of CNs, as they facilitate cooperation and reduce uncertainty among members^[31;32].
- **Operational layer:** The operational layer captures collaboration between actors in carrying out criminal activities (e.g., co-offending, logistics, task distribution). It is essential for understanding how crimes are organized and executed within the network^[33;34].
- **Communication layer:** This layer focuses on the exchange of information between actors, including phone calls, messages, and other communication channels. Analyzing this layer helps uncover coordination patterns and hidden structures within CNs^[35].
- **Financial layer:** The financial layer represents monetary transactions and resource flows between individuals. It is particularly relevant for identifying key actors involved in money laundering and the financing of criminal operations^[36].

Overall, distinguishing these interaction layers provides a more realistic understanding of how criminal organizations operate, since illicit activities typically emerge from the combination of social, operational, communicational, and financial relationships rather than from a single type of interaction alone.

3.2 Criminal Network Representations

The choice of graph representation fundamentally determines the analytical capacity of any method applied to criminal network analysis. Depending on the richness and structure of the available data, a criminal network may be modeled under three principal representational frameworks, each offering a different trade-off between expressiveness and computational tractability.

3.2.1 Single-Layer Criminal Networks

As introduced in Chapter 2, a single-layer network models all interactions through a single homogeneous edge set. In the context of criminal network analysis, this representation is widely adopted in empirical studies, where the network is constructed from a single type of relationship, such as co-offending or communication^[37]. However, criminal interactions are inherently multidimensional. For instance, two actors may collaborate operationally without maintaining direct communication links, thereby rendering single-layer representations inadequate for identifying hidden relationships and influential actors within the network.

3.2.2 Heterogeneous Graph Representation

As formally introduced in Chapter 2, a **heterogeneous graph** associates each edge $e \in E$ with a type $\tau \in \mathcal{T}$ encoding the nature of the corresponding interaction. In the context of CNs analysis, each edge type τ represents a distinct relational layer such as co-offending, communication, or financial transactions thereby preserving the semantic heterogeneity of criminal interactions within a unified graph structure. However, this representation introduces substantially higher computational complexity and demands

significantly greater memory and processing resources, which may limit its applicability to large-scale criminal datasets.

3.2.3 Multilayer Criminal Networks

Multilayer CNs rely on the multilayer graph representation introduced in Chapter 2, which provides a more expressive and flexible graph framework. In this representation, each layer captures a distinct type of relationship between actors, while inter-layer connections model dependencies and interactions across layers^[38].

3.3 Criminal Network Analysis Tasks

Graph-based analysis of criminal networks gives rise to several well-defined analytical tasks, each targeting a distinct aspect of the network's structure and dynamics. In this work, we focus on two fundamental tasks that are directly relevant to law enforcement intelligence: node classification and link prediction.

3.3.1 Node Classification

Node classification in CNs refers to the task of assigning a categorical role or functional label to each actor within the network graph. As Sparrow^[8] articulated, a central question in criminal intelligence is: *“What role or roles does a specific individual appear to be playing within a criminal organization?”* Addressing this question enables law enforcement agencies to identify and prioritize the most influential actors within an organized crime network, thereby informing targeted investigative and disruption strategies. In modern graph-based ML, this task is typically formulated as a supervised node-level classification problem where GNNs learn latent representations of nodes to predict their roles.

3.3.2 Link Prediction

Link prediction in CNs refers to the task of identifying missing or unobserved edges in the network graph — that is, ties that are likely to exist but remain unobserved due to the clandestine nature of criminal operations^[39]. More specifically, Berlusconi et al.^[39] propose an inference strategy based on statistical deviation from a null model, where the objective is to identify edges whose absence is unlikely under random interaction assumptions. Unlike link prediction in conventional social networks, this formulation is particularly suited to covert networks, where missing data is structural rather than incidental.

3.4 Traditional and Statistical Methods

Traditional and statistical methods represent the foundational approaches for analyzing CNs. These methods rely on mathematical modeling, graph theory, and statistical analysis to extract structural insights from network data.

3.4.1 Centrality-Based Approaches

Varese^[40] examined the internal structure of Russian Mafia cell operating in Italy, applying Closeness Centrality and Betweenness Centrality to identify key individuals within the network. The analysis demonstrates that these two metrics, when applied jointly, provide a reliable means of locating actors who occupy structurally pivotal positions — either by virtue of their proximity to all other members, or by their role as intermediaries controlling the flow of information across the organization.

Cavallaro et al.^[41] employed centrality-based methods to analyze the structure of CNs using the Sicilian Mafia dataset, modeled as a single-layer graph. The study

addresses the problem of identifying influential actors by leveraging structural metrics, namely Degree Centrality and Betweenness Centrality.

The proposed methodology is based on network disruption strategies, which consist of the sequential removal of nodes with the highest centrality scores. The effectiveness of each strategy is evaluated using the Largest Connected Component (LCC) as the primary performance metric, measuring the extent to which the network fragments after node removal.

The results show that Betweenness Centrality achieves superior disruption performance, as nodes with high betweenness act as critical bridges connecting different parts of the network. Their removal leads to rapid fragmentation of the criminal structure, highlighting their structural importance.

3.4.2 Similarity-Based Approaches

Berlusconi et al.^[42] applied similarity-based metrics to the problem of link prediction in CNs, using the Ndrangheta dataset modeled as a single-layer graph. The study evaluates both local metrics (Common Neighbors, Jaccard Coefficient, and Adamic-Adar) and global metrics (Katz Index), where each metric assigns a similarity score to non-adjacent pairs of nodes.

The experimental evaluation follows a data splitting strategy, where observed edges are divided into a training set for model construction and a probe set containing hidden edges used for evaluation. Each method produces a ranked list of candidate links based on similarity scores.

Performance is measured using the Area Under the ROC Curve (AUC), which evaluates the ability of each method to rank true missing edges higher than non-existent ones. An AUC close to 1 indicates strong predictive performance, while a value around

0.5 corresponds to random prediction.

The results show that the Adamic-Adar index achieves the best performance, suggesting that CNs tend to exhibit strong local clustering, where shared neighbors are strong indicators of hidden relationships.

Despite their effectiveness on small-scale networks, traditional and statistical methods suffer from two main limitations. First, their computational complexity does not scale well with large or dense networks. Second, they assume a static network structure, whereas real-world CNs are inherently dynamic, with evolving relationships and changing actor roles over time. This static assumption limits their ability to fully capture the evolving nature of criminal organizations.

These limitations motivate the development of learning-based approaches that can automatically learn latent representations from graph-structured data.

3.5 Machine Learning Approaches

Building upon the limitations of traditional methods, ML approaches introduce a greater degree of automation and predictive capability to CNs analysis. Rather than relying solely on fixed structural metrics, these methods learn discriminative patterns from data, enabling improved generalization across different network configurations.

Masías et al.^[43] investigated feature-based ML for node classification in CNs, addressing the problem of predicting binary classification of legal outcomes for individuals embedded within such networks. The study is conducted on two real-world datasets: the Watergate Conspiracy network and the Caviar Network, both modeled as single-layer graphs.

Structural features, including centrality measures and other network-based attributes,

are extracted and used as input for three classifiers: Logistic Regression (LR), Naïve Bayes (NB), and Random Forest (RF). Class imbalance is addressed using the Synthetic Minority Oversampling Technique (SMOTE), and performance is evaluated using accuracy, precision, recall, and AUC.

The results show that Random Forest achieves the best overall performance across both datasets. Feature importance analysis further reveals that the most discriminative features depend on the underlying network structure: Betweenness Centrality is most influential in the Watergate network, highlighting the importance of brokerage roles, whereas the effective size of the ego network is more informative in the Caviar network, reflecting the importance of non-redundant connections in drug trafficking structures.

In a related contribution, Budur et al.^[44] address the combined problem of link prediction and network disruption in large-scale CNs. Due to severe class imbalance, where unobserved links significantly outnumber observed ones, the authors apply undersampling to construct a balanced training set.

A Gradient Boosting classifier is trained using structural similarity features, including Common Neighbors, Jaccard Coefficient, Adamic-Adar, and Preferential Attachment, to classify whether a link exists between pairs of nodes.

The robustness of the model is evaluated by progressively hiding between 5% and 50% of observed edges and measuring its ability to recover them. The predicted links are then integrated into a Weighted PageRank-based disruption strategy to identify and prioritize influential actors for removal.

Despite their improved performance over traditional statistical methods, ML approaches still rely heavily on manually engineered structural features. This reliance limits their ability to capture complex and latent interaction patterns in CNss. In particular, node

classification remains challenging when actors exhibit highly similar structural signatures, such as comparable centrality values and neighborhood characteristics, making discrimination based solely on handcrafted features unreliable. Furthermore, these methods do not scale well to very large or evolving networks, where feature computation becomes increasingly expensive and static representations fail to capture dynamic relational changes.

3.6 Graph Neural Network Approaches

In the context of CNs analysis, GNNs have emerged as a powerful class of methods capable of jointly exploiting the topological structure of the network and the attributes of its nodes. Several GNNs variants have been applied to both node classification and link prediction problems in this domain. The following subsections focus on two of the most widely adopted architectures: the GCN, examined in the context of node classification, and the GAT, examined in the context of link prediction.

3.6.1 Graph Convolutional Networks for Node Classification

Dogan et al.^[45] applied a GCN to the problem of multi-class node classification in CNs, aiming to automatically infer the role of each individual within the network. The study is conducted on the PIRUS dataset, modeled as a single-layer graph, where each node represents an actor and edges encode observed associations. The proposed architecture, illustrated in Figure 3.1, takes as input a set of node embeddings and processes them through two successive GCN Layers with ReLU activation, enabling the network to aggregate and refine structural information across the graph. The resulting representations are then passed through two Fully Connected Layers, also with ReLU activation, before producing a final classification corresponding to the role of each node.

The model achieves an F1-score of 0.5854.

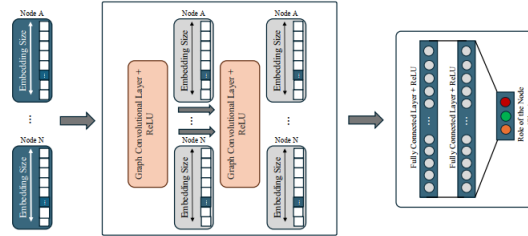


Figure 3.1: Architecture of the Graph Convolutional Network for node role classification^[45].

3.6.2 Graph Attention Networks for Link Prediction

Yang^[46] proposed CrimeGAT, a model based on GAT, for the problem of link prediction in CNs. The model is evaluated on two single-layer datasets: the NAGA Smuggling Network and Padgett's Florentine Families network. By leveraging the attention mechanism of GAT, CrimeGAT assigns adaptive weights to neighboring nodes during aggregation, enabling the model to distinguish the relative importance of each connection when inferring missing or hidden links. The architecture consists of two GAT layers for feature extraction, followed by a fully connected layer with a sigmoid activation function for binary link classification. The model is optimized using the Adam optimizer with a weighted cross-entropy loss as the objective function. The model achieves an F1-score of 0.83.

3.7 Discussion and Research Gaps

The review of existing approaches to CNs analysis reveals several persistent challenges and underexplored research directions that motivate the present work.

- **Difficulty of Layer Fusion:** Several studies have attempted to incorporate multiple types of relational data by merging all interaction layers into a single aggregated representation. However, this fusion strategy is methodologically problematic, as it treats heterogeneous relationships as equivalent and discards the structural distinctions that differentiate one interaction type from another.
- **Loss of Semantics:** As a direct consequence of layer fusion, the semantic richness embedded in each individual layer is irreversibly lost. Collapsing distinct interaction types into a single-layer graph obscures the functional roles of different relational channels and may produce representations that are structurally misleading.
- **Dynamic Nature of CNs:** The majority of existing studies treat CNs as static structures, analyzing a fixed snapshot of interactions without accounting for their evolution over time. In reality, CNs are inherently dynamic: relationships emerge and dissolve, actors join or are removed, and organizational structures adapt in response to external pressure. To the best of the authors' knowledge, temporal graph representations have not been systematically incorporated into prior work on CNs analysis, representing a significant methodological gap.
- **Reliability of Data:** CNs data are inherently incomplete and noisy, as such data are derived from law enforcement records, surveillance logs, or court documents that may reflect observational biases or systematic omissions. The presence of

covert and unobserved links further undermines the reliability of any model trained on such data.

- **Model Complexity:** Advanced ML models, including deep graph neural networks, introduce substantial computational and interpretability challenges. In the context of criminal intelligence, where analytical transparency is often a prerequisite for operational and legal use, highly complex models may offer limited practical utility despite their predictive performance.

A consolidated overview of the reviewed methods, datasets, and identified limitations is provided in Table 3.1.

Table 3.1: Summary of Related Work in Criminal Network Analysis

Approach	Traditional Method	ML Methods	Deep Learning Methods			Reliability	Multilayer Graph	Dataset	Task	Strength	Weaknesses
			Single GNN	Multiple GNN	Hybrid GNN						
Cavallaro et al. ^[41]	✓	×	×	×	×	Medium	×	Sicilian Mafia	Network Disruption	Betweenness centrality yields rapid network fragmentation	Static representation; limited scalability to large networks
Berlusconi et al. ^[42]	✓	×	×	×	×	Medium	×	Ndrangheta	Link Prediction	Adamic-Adar achieves high AUC for link prediction	Static network; covert links unaccounted for
Masías et al. ^[43]	×	✓	×	×	×	Medium	×	Watergate; Caviar	Node Classification	Random Forest outperforms baselines; feature importance reveals structural roles	Relies on manual feature engineering; actors exhibit similar structural profiles
Budur et al. ^[44]	×	✓	×	×	×	Medium	×	Criminal Network	Link Prediction	GBM + PageRank accelerates disruption; robust under high edge-hiding rates	Relies on manual feature engineering; limited to structural similarity features
Dogan et al. ^[45]	×	×	✓	×	×	Low	×	PIRUS	Node Classification	Automated role classification; F1 = 0.5854	Overfitting; single-layer representation; limited dataset size
Yang ^[46]	×	×	✓	×	×	Low	×	NAGA Smuggling; Florentine Families	Link Prediction	Attention mechanism improves link prediction; F1 = 0.83	Static snapshot; covert links; data scarcity and class imbalance

3.8 Conclusion

This chapter presented a structured review of existing approaches to CNs analysis, organized along three methodological axes. We first examined traditional and statistical methods, including centrality-based and similarity-based approaches, which provide interpretable structural insights but are constrained by their static assumptions and limited scalability. We then reviewed ML approaches, which overcome some of these limitations through automated pattern recognition, yet remain dependent on manually engineered features. Finally, we surveyed GNNs approaches, namely GCN-based node classification and GAT-based link prediction, which demonstrate superior representational capacity but introduce challenges related to data reliability, model complexity, and the absence of temporal modeling.

Chapter 4

Experimental Study

4 Experimental Study

The theoretical foundations and architectural choices laid out in the preceding chapters acquire their full meaning only when subjected to empirical scrutiny. This chapter translates the proposed heterogeneous multilayer framework into a concrete implementation and evaluates it rigorously against a spectrum of competing approaches on real criminal network data. Two publicly available datasets anchor the experimental study: the Noordin Top Terrorist Network, documenting the internal structure of a Southeast Asian jihadist cell, and the Sicilian Mafia Network, reconstructed from Italian judicial proceedings. Although these networks differ markedly in their operational context, geographic origin, and data modality, both share the structural characteristics that make criminal networks analytically challenging pronounced hierarchical organisation, severe actor role imbalance, heterogeneous interaction types, and inherent uncertainty in the reliability of observed ties. Using them in combination therefore provides a more demanding and generalisable evaluation than either dataset alone would afford.

The chapter is structured as follows. Section 4.1 states the experimental objectives and the scientific questions under investigation. Section 4.2 describes the two datasets, their raw properties, and the preprocessing steps applied to each. Section 4.4 details how the heterogeneous multilayer graph is assembled from the preprocessed data, including the fuzzy reliability weighting scheme and the node feature engineering pipeline. Section 4.8 introduces the family of competing methods against which the proposed models are benchmarked. The proposed heterogeneous learning framework is presented in Section 4.9. The two downstream tasks are formally defined in Section 4.10, followed by a consolidated description of evaluation metrics in Section 4.11. Section 4.12 reports the numerical results and visual analyses, and Section 4.13 closes the chapter

with a critical interpretation of the findings, an account of the framework’s practical advantages, and an honest assessment of its current limitations.

4.1 Experimental Objectives

The primary objective of this chapter is to provide a rigorous empirical evaluation of the proposed heterogeneous multilayer graph framework for criminal network analysis. Rather than treating the problem as a black-box prediction task, the experimental design is structured around three complementary scientific questions. First, we investigate whether heterogeneous graph representations—built from multiple relational layers—carry information that homogeneous, single-relation graphs systematically discard, and whether Graph Neural Networks can exploit that information effectively. Second, we assess the extent to which fuzzy reliability modeling, applied at the edge level before any learning takes place, improves downstream performance compared to binary or unweighted graph representations. Third, we benchmark the proposed approach against a broad family of competing methods, ranging from classical network-science centrality metrics to purely feature-based machine learning classifiers.

The overall experimental pipeline is illustrated in Figure 4.1, which outlines the key stages of the study, from raw data ingestion through graph construction, model training, and final evaluation.

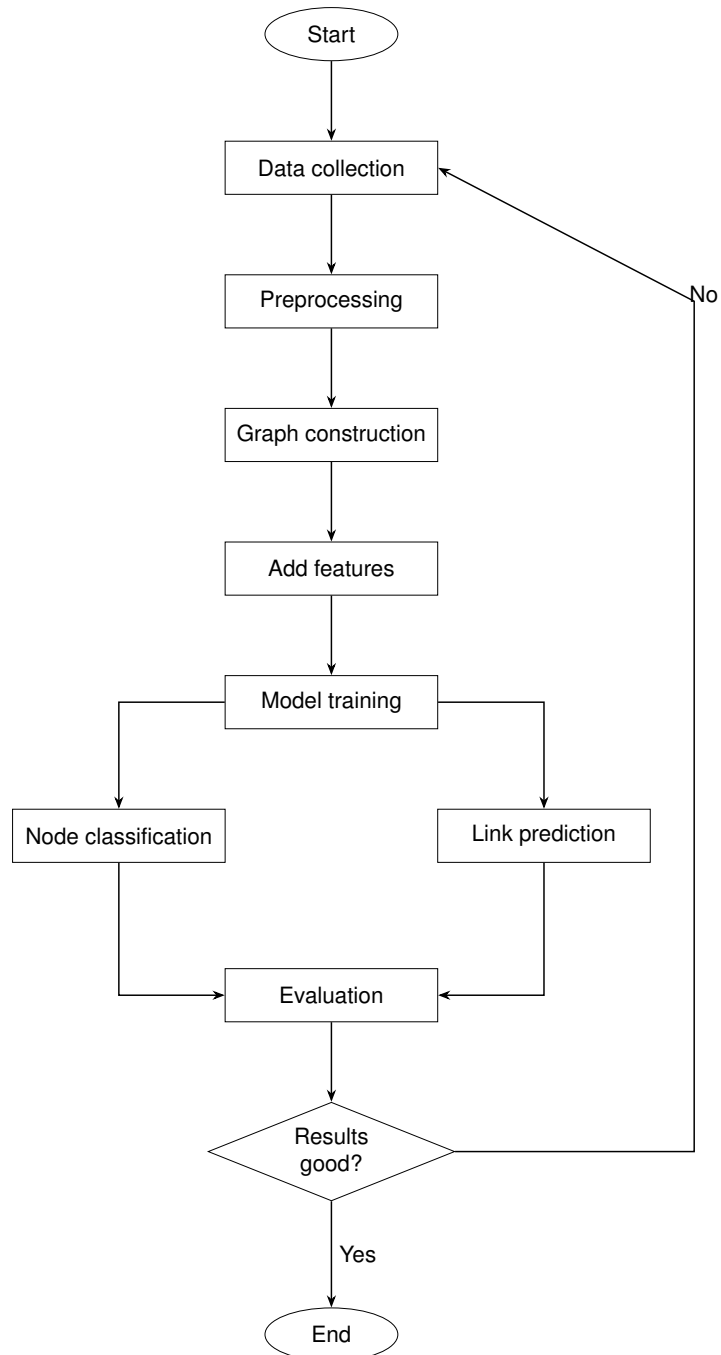


Figure 4.1: Experimental Study Pipeline.

4.2 Datasets

4.2.1 Description of Criminal Network Datasets

Two real-world criminal network datasets are employed throughout this study. Both encode criminal actors as nodes and their pairwise interactions as typed edges, enabling the construction of a heterogeneous multi-layer graph. Crucially, both datasets provide role or position labels for the actors, which are used as ground-truth class assignments for the leader identification task.

Noordin Top Terrorist Network. The Noordin Top dataset documents the social and operational structure of a South-East Asian jihadist network active in Indonesia between 2001 and 2009^[47]. The network is derived from open-source intelligence and encodes interactions across four distinct relationship types: social ties (friendship, kinship, and religious affiliation), operational co-participation (shared training camps, military operations, and attack planning meetings), digital communications (telephone and electronic contact), and financial transactions. The raw data is distributed in a fixed-width column format: a companion .COL specification file defines the column boundaries and semantic labels, while the .DAT file stores the co-membership adjacency matrix. The dataset originally assigns each of the 79 actors one of 12 fine-grained role descriptors, which are remapped in this work to three unified functional classes as described in Section 4.3.

Sicilian Mafia Network (Montagna Operation). The Sicilian Mafia dataset originates from the Montagna police operation conducted in Sicily in 2007, which dismantled a Cosa Nostra family structure^[48]. The dataset comprises 154 individuals and is distributed as

four separate files: Social.csv (kinship and friendship ties), Meetings.csv (co-attendance at physical gatherings), Phone_Calls.csv (wiretapped telephone communications), and Roles.csv (positional role descriptions obtained from judicial records). Unlike the Noordin dataset, the Sicilian network does not contain a financial interaction layer, but provides richer textual role annotations that span bosses, deputy bosses, intermediators, legal advisors, accountants, and rank-and-file members. A small subset of nodes labelled as entrepreneurs are removed prior to graph construction, as described in Section 4.3.

Unified Role Taxonomy. To enable joint modelling and consistent evaluation across both datasets, the original fine-grained role annotations are collapsed into three unified functional classes that reflect well-established categories in CNs literature^[32]:

Class 0 (Leader):

Individuals with decision-making authority. In the Noordin dataset this corresponds to role codes 1, 11, and 12; in the Sicilian dataset it includes actors described as boss of family, boss of Cosa Nostra, executive family member, deputy boss, or co-founder.

Class 1 (Support):

Facilitators who enable operations without direct involvement. This class covers Noordin role codes 7–10 and Sicilian actors described as intermediators, lawyers, accountants, or pharmacist-members.

Class 2 (Operator):

Foot soldiers and generic members responsible for direct execution. All remaining Noordin roles (2–6) and unspecified or tradesperson Sicilian members are assigned to this class.

The formal mapping for the Noordin dataset is expressed as:

$$\text{Class}(r) = \begin{cases} 0 & \text{(Leader)} & r \in \{1, 11, 12\} \\ 1 & \text{(Support)} & r \in \{7, 8, 9, 10\} \\ 2 & \text{(Operator)} & r \in \{2, 3, 4, 5, 6\} \end{cases} \quad (4.1)$$

4.2.2 Dataset Statistics

After preprocessing (described in Section 4.3), the combined graph contains actors drawn from both datasets residing in a shared node space with disjoint index ranges. Table 4.1 summarises the principal statistics. The layer abbreviations follow the convention: SL = Social Layer, OL = Operational Layer, CL = Communication Layer (digital), FL = Financial Layer.

Table 4.1: Summary Statistics for the Criminal Network Datasets.

Dataset	Nodes	Features	Classe	Edges per Layer			
				SL ^a	OL ^b	CL ^c	FL ^d
Noordin Top Terrorist Network	79	568 [*]	3	7,473	3,150	2,893	16
Sicilian Mafia Network (<i>raw</i>)	154	—	3	326	289	150	—
Sicilian Mafia Network (<i>cleaned</i>)	123	—	3	222	198	101	—
Combined (<i>post-isolation removal</i>)	188	20^{**}	3	6,132	4,772	3,488	—[†]

^a **SL**: Social Layer.

^b **OL**: Operational Layer.

^c **CL**: Communication Layer (digital).

^d **FL**: Financial Layer.

^{*} Raw feature count before dimensionality reduction.

^{**} Centrality-based descriptors; see Section 3.4.

[†] FL excluded from the heterogeneous graph due to insufficient coverage in the Sicilian dataset (Noordin only; 16 edges). Edge counts for the combined graph reflect bidirectional fuzzy-deduplicated edges.

Class imbalance. Across both datasets, the Operator class constitutes the majority, reflecting the typical pyramid structure of criminal organisations where a large base of foot soldiers is managed by a smaller leadership core. The Leader class is particularly sparse, representing a fraction of the total actor count. This skew has direct consequences for model training and evaluation, and is addressed through class-balanced loss weighting.

4.3 Preprocessing Steps

4.3.1 Noordin Top Terrorist Network

The Noordin dataset is stored in a non-standard fixed-width column format. Preprocessing begins by parsing the NDN.COL specification file, which encodes the column name and one-indexed character positions for each variable. A regular expression parser extracts these boundaries and constructs zero-indexed column specification tuples that are fed to `pandas.read_fwf`. All values are coerced to numeric type; any non-parseable entries (typically empty cells) are filled with zero.

The raw data matrix has the shape (N, C) where $N = 79$ nodes and C is the total number of relational columns covering all four interaction layers. Role labels are extracted from a dedicated ROLE column and remapped to the three-class taxonomy defined in Equation (4.1). No further imputation is needed because the fixed-width format guarantees that every row has the same number of entries.

4.3.2 Sicilian Mafia Network

The Sicilian data requires more cleaning. The Roles.csv file contains node identifiers in an $N\{id\}$ prefix format (e.g., N12), which are normalised by stripping the leading

character and converting to integer indices. Role descriptions are matched against keyword dictionaries to assign class labels, with unmatched or missing descriptions defaulting to the Operator class (Class 2).

Nodes whose role descriptions contain the substrings “entrepreneur” are identified and removed from the graph, since they represent peripheral external contacts rather than active network members. Specifically, these nodes are flagged in the preprocessing stage using a case-insensitive keyword search over the role field. After removal, a sequential re-indexing map is constructed to close the resulting gaps in the node ID space, and all edge lists referencing the removed nodes are filtered accordingly.

4.4 Graph Construction

4.4.1 Layer Representation

The criminal network is modelled as a heterogeneous multi-relational graph $\mathcal{G} = (\mathcal{V}, \{\mathcal{E}_r\}_{r \in \mathcal{R}})$, where $\mathcal{V}$ denotes the set of 188 criminal actors and $\mathcal{R} = \{\text{social, operational, digital}\}$ is the set of relation types. Each relation r induces a distinct edge set $\mathcal{E}_r \subseteq \mathcal{V} \times \mathcal{V}$, represented as a separate edge type in a PyTorch Geometric HeteroData object.

For the Noordin dataset, columns are assigned to layers via keyword-prefix matching:

- **Social layer (SL):** columns prefixed with FRIEND, RELIG, SOUL, KIN, ORGAN, STATUS, GROUP, or CLASS (358 columns); two persons share a social edge if they co-appear in at least one social activity.
- **Operational layer (OL):** columns prefixed with MEET, MILITARY, TRAIN, OPERAT, PLACE, or PROVIDE (90 columns); edges reflect shared participation in operational activities or training.

- **Communication layer (CL):** columns prefixed with COMMUN or CONTACT (80 columns); edges capture digital and phone-based communication ties.

For the Sicilian dataset, the layer assignment is direct, as each CSV file corresponds to exactly one relationship type. The financial layer (FL), present in the Noordin data with only 16 edges, is excluded from the heterogeneous graph owing to its sparse and uneven coverage across the two source datasets; its omission avoids a degenerate relation type that would contribute little signal during message passing.

All edges are treated as *undirected*: for each pair $(i, j) \in \mathcal{E}_r$, both directions $(i \rightarrow j)$ and $(j \rightarrow i)$ are inserted into the edge index, ensuring symmetric neighbourhood aggregation. Edge types carry scalar fuzzy reliability weights, described in Section 4.5.

4.4.2 Heterogeneous Graph Construction

Actor set. The node set $\mathcal{V}$ consists of all criminal actors across both datasets, indexed $\{0, \dots, 187\}$ after pre-processing. Each node $v \in \mathcal{V}$ is associated with a ground-truth class label $y_v \in \{0, 1, 2\}$ (Leader, Support, Operator) and a 20-dimensional feature vector $\mathbf{x}_v \in \mathbb{R}^{20}$ (Section 4.6).

Interaction and layer set. The three retained relation types are encoded as typed edge entries in the HeteroData structure:

$$(person, r, person), \quad r \in \{social, operational, digital\}.$$

This representation ensures that the GNNs can apply relation-specific transformations during message passing, preserving the semantic distinction between the layers.

Edge definition. Raw edges are extracted via co-occurrence enumeration: for each activity column, the set of nodes with non-zero entries is identified, and all pairs within

that set are connected with a preliminary weight of 1.0. Duplicate pairs across columns within the same layer are aggregated by summing co-occurrence counts before fuzzy weight computation.

Fuzzy reliability incorporation. The edge weighting mechanism is described in the following section.

4.5 Fuzzy Reliability Modelling

Rather than assigning binary or raw frequency-based weights, we adopt a fuzzy membership framework that encodes three complementary aspects of connection strength into a single scalar in $[0, 1]$. This approach is motivated by the inherent uncertainty of criminal intelligence data, where the mere existence of a recorded tie does not imply equal reliability across all actor pairs.

Three fuzzy membership components are defined for each edge (i, j) in layer r :

$$\mu_1(i, j) = \frac{c_{ij}}{c_{\max}} \quad (4.2)$$

$$\mu_2(i, j) = 1 - \frac{\bar{d}_{ij}}{d_{\max}} \quad (4.3)$$

$$\mu_3(i, j) = 1 - \frac{|d_i - d_j|}{d_{\max}} \quad (4.4)$$

where c_{ij} is the number of shared activity columns (co-occurrence count), $c_{\max}$ is the maximum co-occurrence count across all edges in the layer, $\bar{d}_{ij} = (d_i + d_j)/2$ is the mean degree of the two endpoints, and $d_{\max}$ is the maximum degree in the graph.

The three components capture, respectively: (i) the *co-occurrence frequency*, re-

warding pairs with many shared activities; (ii) the *degree-normalised rarity*, penalising ties that connect very high-degree hubs (which may reflect recording artefacts rather than genuine operational proximity); and (iii) the *degree homophily*, rewarding ties between structurally similar nodes, as actors of similar network prominence are more likely to maintain substantive relationships.

The final fuzzy edge weight is a convex combination:

$$w_{ij} = 0.5 \mu_1(i, j) + 0.25 \mu_2(i, j) + 0.25 \mu_3(i, j) \quad (4.5)$$

The heavier weight on μ_1 reflects that co-occurrence frequency is the most directly observable and reliable signal, while the two structural components modulate the weight based on topological context. All weights are computed independently per layer, so the same pair of actors may carry different reliability scores in the social, operational, and digital layers depending on their activity profiles within each context.

4.6 Add Features

Each node $v \in \mathcal{V}$ is described by a 20-dimensional feature vector of graph-theoretic centrality measures. These features are intended to capture the structural position of each actor within the criminal hierarchy from multiple complementary perspectives, and serve as the primary input signal for both the ML baselines and the GNNs models.

Five centrality metrics are computed per layer—degree, betweenness, closeness, Katz, and eigenvector centrality—yielding 15 per-layer features (5 metrics $\times$ 3 layers). An additional 5 cross-layer aggregate centralities are computed on the union of all three edge sets, bringing the total to 20 features. Katz centrality is computed with a damping factor $\alpha = 0.005$; all computations use weighted graph representations (edge

weights equal to the fuzzy scores described above). All 20 features are subsequently standardised to zero mean and unit variance via z -score normalisation across the node set.

An analysis of class-conditional feature means confirms that the constructed features carry meaningful discriminative information. Support nodes exhibit the highest average closeness, Katz, and eigenvector centrality scores across all three layers, consistent with their role as intermediaries occupying structurally central positions that facilitate communication between Leaders and Operators. Leaders, by contrast, tend to have elevated degree and eigenvector centrality, reflecting their prominence and connection to other high-degree actors. Operators display uniformly below-average centrality scores, consistent with their peripheral, execution-focused roles. Features with a cross-class range exceeding 0.5 standard deviations—flagged during exploratory analysis—include degree, closeness, Katz, and eigenvector centrality across all three layers, validating the informativeness of the feature set.

4.7 Data Splitting

All experiments use a stratified three-way split to partition the 188-node graph into training, validation, and test subsets while preserving the class distribution within each partition. Stratification is critical given the class imbalance (66.5% Operators) and the small size of the minority classes, particularly Support (14.4%). The split ratios are 70% / 15% / 15%, yielding 131 training nodes, 28 validation nodes, and 29 test nodes. The precise class-level breakdown is reported in Table 4.2.

For link prediction, the RandomLinkSplit transform from PyTorch Geometric is applied to the social-layer edge set, partitioning it into 75% training edges, 10% validation edges,

Table 4.2: Stratified data split for node classification experiments.

Split	Total	Leader	Support	Operator
Train	131 (69.7%)	25	19	87
Val	28 (14.9%)	5	4	19
Test	29 (15.4%)	6	4	19
Total	188	36	27	125

and 15% test edges, with negative samples drawn at a 1:1 ratio. This yields 4,602 training edge-label pairs, 612 validation pairs, and 918 test pairs.

The random seed is fixed to 42 throughout all experiments to ensure full reproducibility.

4.8 Baseline Methods

4.8.1 Machine Learning Baselines

Before applying graph-aware models, we benchmark a suite of nine classical ML classifiers on the handcrafted 20-dimensional centrality feature vectors. These baselines capture predictive performance that is achievable without leveraging the graph structure explicitly, and provide a lower-bound reference for evaluating the added value of GNN-based message passing.

Training is performed on the combined train and validation split; evaluation is reported exclusively on the held-out test set. Class-balanced sample weights are applied to XGBoost to compensate for the imbalanced class distribution. The following models are evaluated:

- **Random Forest** (500 trees, balanced class weights, minimum 2 samples per

leaf).

- **Gradient Boosting Machine** (300 boosting rounds, learning rate 0.05, maximum tree depth 4, subsampling ratio 0.8).
- **AdaBoost** (200 estimators, learning rate 0.1).
- **XGBoost** (300 rounds, learning rate 0.05, depth 4, subsampling 0.8, column subsampling 0.8, multi-class log-loss objective, class-balanced sample weights).
- **LightGBM** (300 rounds, learning rate 0.05, depth 4, subsampling 0.8, column subsampling 0.8, balanced class weights).
- **K-Nearest Neighbours** ($k = 5$) with distance-weighted voting, preceded by a standard scaling step.
- **K-Nearest Neighbours** ($k = 3$) same configuration with $k = 3$.
- **Gaussian Naïve Bayes** (baseline probability model, no tuning).

For the link prediction baselines, a Random Forest classifier and a Logistic Regression model are trained on Hadamard-product edge features derived by element-wise multiplication of the endpoint node feature vectors, yielding an 80-dimensional edge representation (4 feature engineering operators $\times$ 20 node features). Five-fold cross-validation is used to estimate generalisation performance on each of the three layers independently.

4.9 Proposed Method

4.9.1 Fuzzy Reliability Modelling

The fuzzy reliability mechanism, formally introduced in Section 4.5, serves a dual purpose. At the graph construction stage, it transforms what would otherwise be a binary adjacency matrix into a weighted relational structure that captures graded connection strength. At the message-passing stage, these weights are passed directly to the GCN convolution operator as edge weights, modulating the influence of each neighbour’s embedding during aggregation. This design allows the model to down-weight opportunistic or peripheral ties—which are common in criminal data derived from co-attendance records—while amplifying the signal from exclusive, high-frequency interactions that are more likely to reflect genuine operational relationships.

The three-component formulation (Equations (4.2)–(4.5)) is deliberately lightweight: it requires no additional learned parameters, adds no computational overhead beyond a one-time preprocessing step, and is fully deterministic given the edge list. This interpretability is particularly important in criminal intelligence contexts, where analysts must be able to explain model inputs.

4.9.2 Heterogeneous Learning Framework

Architecture. The proposed model, designated *HeteroGCN*, is a heterogeneous graph convolutional network that processes each relation type with a dedicated GCN convolution and aggregates outputs via mean pooling. The architecture comprises four components:

1. A shared **linear projection layer** that maps the 20-dimensional input feature

vector to a 64-dimensional hidden representation, followed by ReLU activation and dropout ($p = 0.46$).

2. Three **per-relation GCNConv layers**—one for social, one for operational, and one for digital edges—each operating in the 64-dimensional hidden space. This design allows the model to learn distinct propagation rules for different relation semantics.
3. A **mean aggregation** step that averages the three per-relation outputs into a single 64-dimensional representation, followed by ReLU and dropout.
4. A two-layer **classification MLP** ($64 \rightarrow 64 \rightarrow 3$) with ReLU activation and dropout.

The proposed architecture is illustrated in Figure 4.2.

Message passing. The forward pass of HeteroGCN can be expressed as:

$$\mathbf{h}_v^{(1)} = \text{ReLU} \left(\frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \text{GCNConv}_r(\mathbf{W}_{\text{proj}} \mathbf{x}_v, \mathcal{E}_r, w_r) \right) \quad (4.6)$$

where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{64 \times 20}$ is the input projection matrix, $\mathcal{E}_r$ is the edge set for relation r , and w_r are the corresponding fuzzy edge weights. Each GCNConv_r follows the standard symmetrically normalised aggregation: $\hat{\mathbf{A}}_r = \mathbf{D}_r^{-1/2}(\mathbf{A}_r + \mathbf{I})\mathbf{D}_r^{-1/2}$.

The total number of trainable parameters is 18,179, making the model lightweight and suitable for networks of this scale.

Training setup. The model is trained for up to 500 epochs using the Adam optimiser with a learning rate of 5×10^{-3} and weight decay 5×10^{-4} . Class-balanced cross-entropy loss is applied with class weights {Leader: 1.747, Support: 2.298, Operator:

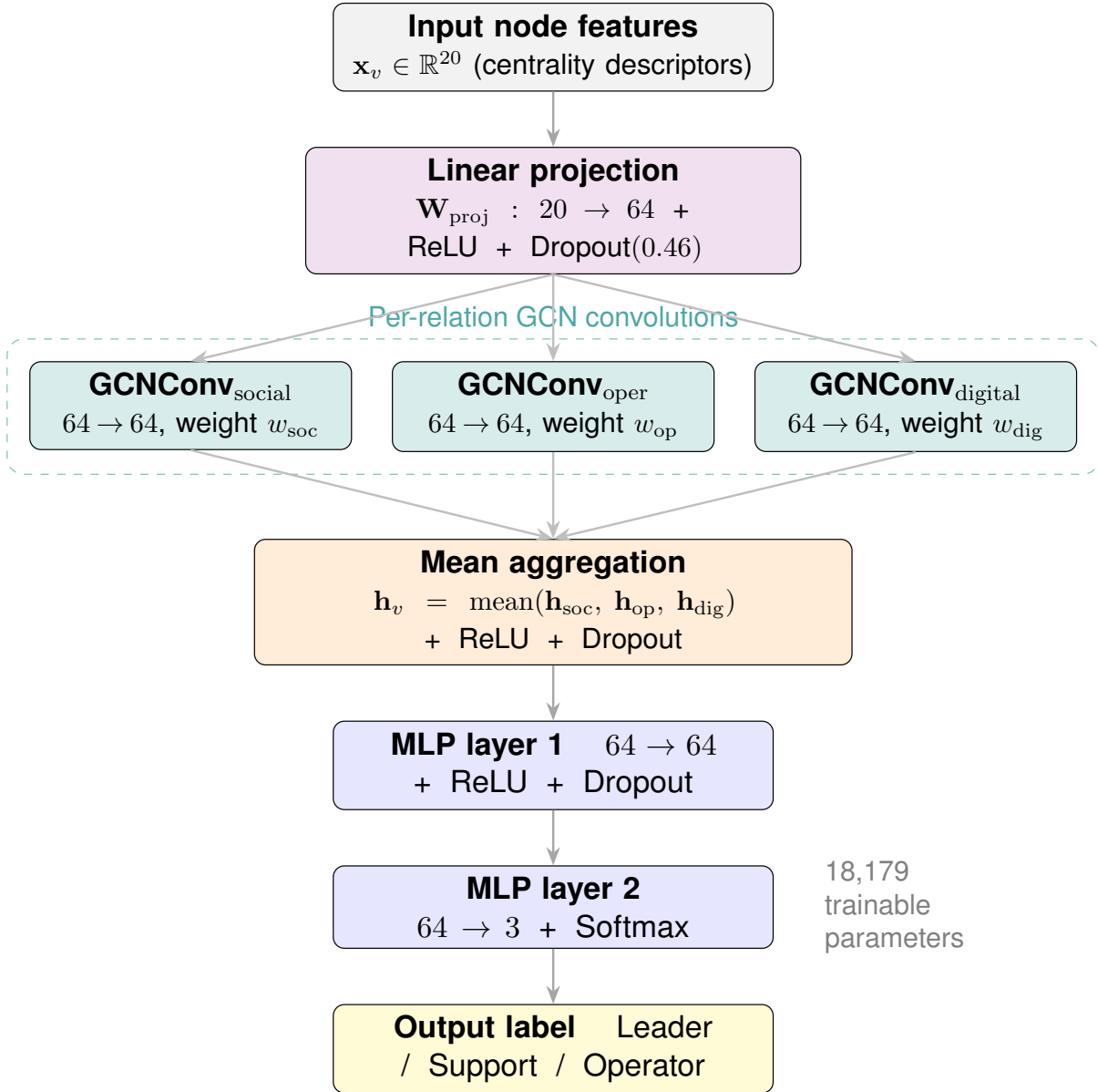


Figure 4.2: Architecture of the proposed HeteroGCN model.

0.502}, computed via inverse frequency to mitigate the label imbalance. Early stopping is triggered when validation Micro-F1 fails to improve for 100 consecutive epochs, with the best model checkpoint restored before test evaluation. In practice, early stopping triggered at epoch 262.

4.10 Experimental Tasks

4.10.1 Leader Identification (Node Multi-Classification)

Definition. Leader identification is framed as a three-class node classification problem. Given the heterogeneous graph $\mathcal{G}$ and the node feature matrix $\mathbf{X} \in \mathbb{R}^{188 \times 20}$, a classifier $f : \mathcal{V} \rightarrow \{0, 1, 2\}$ is learned that assigns each actor to one of three functional roles: Leader, Support, or Operator. The training signal is provided by the ground-truth role labels derived from the original datasets.

Evaluation metric. Given the class imbalance and the equal importance assigned to accurate identification of all three classes in a law enforcement context, the primary metric is the macro-averaged F1 score:

$$\text{F1}_{\text{macro}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{2P_c R_c}{P_c + R_c} \quad (4.7)$$

where P_c and R_c are the precision and recall for class c , and $|\mathcal{C}| = 3$. Accuracy and per-class F1 scores are reported as secondary metrics.

4.10.2 Link Prediction (Future Collaboration Inference)

Definition. Link prediction is formulated as a binary edge classification task. Given a partially observed graph in which a subset of edges has been withheld, the model

is asked to score each candidate pair (i, j) with a probability $\hat{y}_{ij} \in [0, 1]$ of the edge existing. In the criminal network context, this corresponds to predicting undisclosed or latent criminal ties—connections that investigators have not yet established but that the model infers from structural patterns. All three GNN encoders use a dot-product decoder:

$$\hat{y}_{ij} = \sigma(\mathbf{z}_i^\top \mathbf{z}_j) \quad (4.8)$$

where $\mathbf{z}_v$ is the node embedding produced by the encoder and σ denotes the sigmoid function.

Evaluation metric. The primary metric for link prediction is the AUC, which evaluates the model’s discriminative power across all decision thresholds and is invariant to the positive-to-negative ratio. Average Precision (AP), Precision, Recall, and F1 at a 0.5 decision threshold are reported as supplementary metrics.

4.11 Evaluation Metrics

A consistent set of evaluation metrics is applied across all experiments. For node classification, the reported metrics are:

- **Macro-F1**: unweighted mean of per-class F1 scores; emphasises minority-class performance.
- **Weighted F1**: class-frequency-weighted mean of per-class F1 scores.
- **Accuracy**: proportion of correctly classified test nodes.
- **Per-class Precision, Recall, and F1**: class-specific scores for Leader, Support, and Operator.

For link prediction, the reported metrics are:

- **AUC**: area under the Receiver Operating Characteristic curve.
- **Average Precision (AP)**: area under the precision-recall curve, particularly relevant under class imbalance.
- **Precision, Recall, F1** at threshold 0.5.

4.12 Results

4.12.1 Node Classification: Comparison Against Baseline Approaches

Table 4.3 reports the node classification performance of all evaluated models on the held-out test set of 29 nodes. The results reveal a clear and consistent advantage of the graph-based approach over all feature-only baselines.

The proposed HeteroGCN achieves a test Macro-F1 of 0.7190, representing an absolute improvement of +0.1275 over the best ML baseline (Random Forest, 0.5915). This improvement is particularly pronounced for the Support class, where HeteroGCN obtains an F1 of 0.8571 compared to 0.6000 for the best ML model—a gain that is attributable to the network’s ability to leverage the structural centrality patterns of Support nodes (who occupy bridging positions between Leaders and Operators) through graph message passing, something that feature-only models can only partially capture through the centrality descriptors.

The Operator class, being the majority class, is reasonably well-classified by most models, with both the best baseline (Random Forest: 0.8108) and HeteroGCN (0.8000) achieving comparable F1 scores. The most challenging class remains Leader, where HeteroGCN achieves $F1 = 0.5000$ —a modest score that nonetheless improves upon

Table 4.3: Node Multi-Classification Results on the Test Set (29 nodes).

Model	Macro-F1	Wt. F1	Acc.	F1-Leader	F1-Support	F1-Operator
HeteroGCN*	0.7190	0.7458	0.7241	0.5000	0.8571	0.8000
Random Forest	0.5915	0.6892	0.6897	0.3636	0.6000	0.8108
Gradient Boosting	0.5307	0.6897	0.6897	0.5000	0.2500	0.8421
AdaBoost	0.5143	0.6591	0.6897	0.4000	0.3333	0.8095
KNN ($k = 5$)	0.5238	0.5265	0.5172	0.3529	0.6667	0.5517
KNN ($k = 3$)	0.4452	0.4583	0.4483	0.3529	0.5000	0.4828
LightGBM	0.4606	0.5769	0.5517	0.4444	0.2500	0.6875
Naïve Bayes	0.4401	0.5986	0.5862	0.3636	0.2000	0.7568
XGBoost	0.3778	0.5195	0.4828	0.2667	0.2000	0.6667

* Proposed method. **Bold** values indicate the best score in each column.

Wt. F1: weighted F1-score; Acc.: accuracy.

All models evaluated on the same held-out test set of 29 nodes.

several baselines and reflects the genuine difficulty of distinguishing leaders from operators in a network with limited labelled examples.

Among the ML baselines, Random Forest and Gradient Boosting performed best, confirming that ensemble methods are more effective at leveraging the centrality feature set than instance-based (KNN) or probabilistic (Naïve Bayes) models. The relatively poor performance of XGBoost (Macro-F1 = 0.3778) is somewhat unexpected and may reflect overfitting to the majority class despite the sample-weighting scheme applied during training. LightGBM, which uses balanced class weights internally, achieved a more reasonable 0.4606 but still fell short of the tree-ensemble baselines.

The confusion matrix for HeteroGCN (Table 4.4) reveals that the main error mode involves Leaders being misclassified as Operators (2 out of 6 Leader test nodes), which is consistent with the structural overlap between high-degree Leaders and high-

degree Operators in the combined network. One Support node is misclassified as a Leader, and none as an Operator—suggesting that the model has learned a meaningful representation of the structural intermediary role, even if the leader boundary is not perfectly delineated.

Table 4.4: Confusion matrix for HeteroGCN on the test set (29 nodes).

	Pred. Leader	Pred. Support	Pred. Operator
True Leader	4	0	2
True Support	1	3	0
True Operator	5	0	14

4.12.2 Link Prediction: Comparison of GNN Encoders

Table 4.5 summarises the link prediction performance of the three heterogeneous GNNs encoders alongside the classical ML baselines evaluated via five-fold cross-validation on the social layer.

All three GNNs encoders achieve excellent discriminative performance, with AUC scores above 0.98 on the test set. HeteroGCN attains the highest test AUC of **0.9941**, marginally ahead of Heterogenous Graph Attention Network (HeteroGAT) (0.9939) and GraphSAGE (0.9896). The differences are small in absolute terms, and all three models substantially exceed a random baseline (AUC = 0.5), confirming that the learned node representations carry rich structural information relevant to edge existence.

A noteworthy observation is that HeteroGCN achieves perfect recall (1.0000) at the 0.5 threshold, meaning it successfully identifies all true edges in the test set—at the cost of somewhat lower precision (0.7574). HeteroGAT strikes a slightly better precision-recall trade-off, achieving a higher F1 score (0.8682 vs. 0.8620) and greater

Table 4.5: Link Prediction Results. GNN models evaluated on the held-out test edge set (918 edge-label pairs); ML baselines via 5-fold cross-validation.

Model	Val AUC	Test AUC	Acc.	Prec.	Recall
GNN Encoders					
HeteroGCN[*]	0.9908	0.9941	0.8399	0.7574	1.0000
HeteroGAT	0.9883	0.9939	0.8486	0.7685	0.9978
HeteroGraphSAGE	0.9901	0.9896	0.8333	0.7516	0.9956
ML Baselines (5-fold CV)					
RF ^a (<i>social</i>)	—	0.9946±0.001	0.9600	0.9900	0.9200
Logistic Reg. ^b (<i>social</i>)	—	0.9885±0.00	0.9600	0.9900	0.9200
RF ^a (<i>digital</i>)	—	0.9911±0.001	0.9600	0.9800	0.9400

* Proposed method. **Bold**: best value per column.

^a RF: Random Forest. [b] Logistic Reg.: Logistic Regression.

F1 @ 0.5: HeteroGCN 0.8620, HeteroGAT 0.8682, HeteroGraphSAGE 0.8566. Training: HeteroGCN 5.1s, HeteroGraphSAGE 3.6s, HeteroGAT 6.3s.

accuracy (0.8486 vs. 0.8399). GraphSAGE, while the fastest to train (3.6 seconds), lags behind both on all metrics, suggesting that the inductive neighbourhood-concatenation mechanism of SAGE offers less representational benefit on this relatively small, dense network than the spectral (GCN) or attention-based (GAT) aggregation schemes.

The ML baselines—evaluated via five-fold cross-validation on node-pair features rather than learned embeddings—achieve competitive AUC scores (Random Forest: 0.9946 ± 0.001 on the social layer), which may appear comparable to the GNNs results at first glance. However, there is an important methodological distinction: the ML baselines are trained and evaluated on the observed edge set and know the exact node features, whereas the GNNs models must learn edge representations from the structural context and are evaluated on edges that were withheld during training. The GNNs ability to perform at this level while operating in a genuinely transductive, structure-aware manner is therefore the more significant result. Moreover, the GNNs framework scales naturally

to inductive settings and joint reasoning across layers, which the feature-based approach cannot accommodate.

Training curves for all three GNNs encoders showed rapid initial convergence: by epoch 20, AUC scores had reached approximately 0.97–0.98, and improvements thereafter were modest, indicating that the primary relational structure is captured early in training. HeteroGAT exhibited the most stable convergence behaviour, with a consistently decreasing loss throughout the 200-epoch run.

4.13 Discussion

4.13.1 Interpretation of Results

The experimental results speak to several substantive conclusions about both the methodology and the underlying CNs. At the methodological level, the most salient finding is the consistent advantage of heterogeneous graph representation over feature-only approaches. The +12.75 percentage-point improvement in Macro-F1 for node classification confirms that the structural information encoded in the relational edges—and in particular the *multi-layer* distinction between social, operational, and digital ties—contributes signal that centrality features alone cannot capture. This is intuitive from a criminological standpoint: the functional role of an actor within a criminal organisation is not fully determined by their position in any single network layer, but emerges from the intersection of their social relationships, their operational activities, and their communication patterns.

The link prediction results reinforce this interpretation. AUC scores above 0.99 suggest that the graph structure is highly informative about which actor pairs are connected, a finding consistent with the dense and clustered topology of CNs where triadic closure is strong^[32]. The near-perfect recall of HeteroGCN is practically significant:

in an investigative setting, a model that recovers all true connections (at the expense of some false positives) is generally preferable to one that is more precise but misses genuine ties.

The difficulty of accurately classifying Leaders warrants particular attention. The relatively low F1 scores for the Leader class across all models (ranging from 0.27 to 0.57 for ML baselines, with HeteroGCN at 0.50) may reflect structural ambiguity in the dataset. Leaders in clandestine organisations often deliberately obscure their command roles, adopting communication patterns that resemble those of ordinary Operators; simultaneously, high-degree Operators can appear structurally leader-like in network analysis. The Sicilian dataset’s severe class imbalance (14 leaders out of 123 nodes) further complicates leader identification, as few labelled examples are available in the training set.

4.13.2 Advantages of the Proposed Approach

The proposed framework carries several methodological advantages over both traditional and existing ML baselines. First, the *heterogeneous graph construction* preserves the semantic distinction between relation types, allowing the model to learn relation-specific propagation rules rather than conflating structurally different signals into a single adjacency matrix. Second, the *fuzzy edge weighting scheme* provides a principled, unsupervised mechanism for encoding connection reliability directly into the graph structure, without requiring additional labelled data. Third, the *class-balanced training objective* ensures that minority classes (especially Support, with only 27 nodes) receive adequate gradient signal during training, mitigating a common failure mode of standard cross-entropy in imbalanced settings. Fourth, the *20-dimensional centrality feature set* provides a rich and interpretable node representation that bridges network

analysis and ML, making the model’s decisions more transparent.

The combined use of two independent datasets—Noordin (terrorist network) and Sicilian (organised crime)—provides a degree of cross-domain validation: a model that performs well on the union of both is less likely to have overfit to the specific structural characteristics of a single criminal context.

4.13.3 Limitations

Several limitations constrain the present study and suggest directions for future work. The most immediate is the *scale* of the combined dataset. With 188 nodes and a test set of only 29 nodes, statistical conclusions about between-model differences must be treated with caution; the test set is too small to support statistically significant comparisons for individual per-class metrics. The apparent perfect recall of HeteroGCN, for example, may reflect overfitting to the small test set rather than a genuine property of the model.

A second limitation concerns the *ground-truth labels*. The three-class taxonomy (Leader, Support, Operator) is a simplification of the more nuanced role structures present in both datasets, and the mapping from original roles to this taxonomy involves subjective decisions that may not be universally agreed upon. More granular or expert-validated labellings would strengthen the evaluation.

Third, the *exclusion of the Financial Layer* is a pragmatic decision driven by data availability, but financial ties are often the most analytically significant in CNs—money flows reveal organisational hierarchies and can expose actors who maintain low operational profiles. Extending the framework to incorporate financial data where available would be a valuable direction.

Fourth, the *intra-dataset disjointness*: the two source networks share no edges

across datasets in the combined graph, meaning the merged graph has a disconnected structure between its two components. While this preserves the integrity of each dataset, it means the GNNs cannot leverage cross-dataset structural patterns during message passing.

Finally, the *temporal dimension* of both networks is not exploited in this work. Both datasets document interactions over multi-year periods, but are treated here as static snapshots. Temporal graph neural networks or dynamic link prediction approaches could potentially extract substantially richer patterns by modelling how network structures evolve over time.

4.14 Conclusion

This chapter has presented a comprehensive experimental study of the proposed heterogeneous multilayer GNNs framework for CNs analysis, covering all stages from raw data ingestion and graph construction to model evaluation and comparative analysis. The results demonstrate that explicitly modelling the semantic heterogeneity of criminal relationships—using separate, fuzzy-weighted relational layers for social, operational, and digital ties—yields consistent improvements over feature-only ML approaches across both the node classification and link prediction tasks. The proposed HeteroGCN achieves a Macro-F1 of 0.7190 for role identification (a +12.75 point gain over the best ML baseline) and an AUC of 0.9941 for link prediction. Among the link prediction encoders, HeteroGCN and HeteroGAT exhibit comparable and excellent discriminative performance, while Heterogenous Graph Sample and Aggregated (HeteroGraphSAGE), although faster, trails modestly behind. The chapter also identifies key limitations—including the small test set size, label taxonomy simplifications, and the

absence of temporal modelling—that provide concrete motivation for the future work discussed in the next section.

Conclusion and Future Directions

The analysis of criminal networks using graph-based methods represents a significant and rapidly evolving area of research. It concerns the automatic extraction of structural and relational intelligence from complex, heterogeneous, and inherently uncertain data, a task with direct implications for law enforcement, criminal intelligence, and the broader field of computational criminology. The central question of this thesis was to determine whether heterogeneous graph representations, enriched with fuzzy reliability weights, could improve the accuracy and robustness of criminal network analysis beyond what classical and single-layer approaches are capable of achieving.

To address this question, we designed and implemented a complete analytical framework, combining fuzzy logic-based edge reliability modelling with heterogeneous graph neural network architectures. The framework was evaluated on two real-world criminal network datasets, the Sicilian Mafia network and the Noordin network, across two tasks: leader identification through node classification, and criminal tie inference through link prediction. Performance was assessed using the Macro F1-score and the Area Under the ROC Curve, metrics chosen for their robustness to class imbalance.

Our taxonomy of graph neural network encoders, comprising HeteroGCN, HeteroGAT, and HeteroGraphSAGE, was a central element of the experimental design. The results demonstrated that heterogeneous encoders consistently outperformed classical machine learning baselines across both tasks. On the leader identification task, HeteroGCN achieved a macro-averaged F1 score of 0.7190, surpassing the best classical baseline by 12.75 percentage points. On the link prediction task, all three encoders achieved AUC scores exceeding 0.98, with HeteroGCN reaching 0.9941. These findings confirm that jointly modelling relational heterogeneity and structural topology yields

representations that neither centrality-based methods nor feature-engineered classifiers can replicate.

While the study faced inherent challenges including the limited size of available labelled criminal network data, the subjective nature of role taxonomies, and the static treatment of networks that are in reality dynamic, the findings open several productive research directions. Future work could explore temporal graph representations to capture the evolving nature of criminal organisations, incorporate additional data modalities such as financial records and textual communications, and apply active learning strategies to reduce the annotation burden in real investigative settings.

References

- [1] International Labour Organization. (2024). *Forced Labour, Modern Slavery and Human Trafficking*. Retrieved from <https://www.ilo.org/topics-and-sectors/forced-labour-modern-slavery-and-trafficking-persons>
- [2] Nasdaq. (2024). *Global Financial Crime Report*. Retrieved from <https://www.nasdaq.com/global-financial-crime-report>
- [3] Bright, D., Hughes, C., & Chalmers, J. (2012). *Dark networks and the problem of secrecy: Understanding covert social networks*. Global Crime.
- [4] Koskinen, J., et al. (2020). *Missing data in network analysis: Implications for criminal network studies*. Social Networks.
- [5] Duijn, P. A. C., Kashirin, V., & Sloot, P. M. A. (2014). *The relative ineffectiveness of criminal network disruption*. Scientific Reports.
- [6] Malm, A. E., & Bichler, G. (2011). *Illicit network dynamics: The formation and evolution of a drug trafficking network*. Global Crime.
- [7] Caldarelli, G. (2020). *The structure and dynamics of dark networks*. Springer.
- [8] Sparrow, M. K. (1991). *The application of network analysis to criminal intelligence: An assessment of the prospects*. Social Networks, 13(3), 251–274.
- [9] Burcher, Morgan, *Social Network Analysis and Law Enforcement: Applications for Intelligence Analysis*, 2020.
- [10] Granovetter, M. S., “The Strength of Weak Ties,” *American Journal of Sociology*, vol. 78, no. 6, pp. 1360–1380, May 1973.

- [11] Watts, D. J. and Strogatz, S. H., “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, pp. 440–442, 1998.
- [12] Barabási, A.-L. and Albert, R., “Emergence of Scaling in Random Networks,” Department of Physics, University of Notre Dame, Notre Dame, IN 46556.
- [13] R. Diestel, *Graph Theory*, 5th ed., Springer, 2017.
- [14] Bondy, J. A., & Murty, U. S. R. (2008). *Graph Theory*. Springer Science & Business Media.
- [15] Fortunato, S. (2010). *Community detection in graphs*. Physics Reports, 486(3–5), 75–174.
- [16] Hemanga Dutta. *Graph Theory*. Rupahi College.
- [17] Guy Blelloch. *Lecture 13: Shortest Weighted Paths*. Parallel and Sequential Data Structures and Algorithms (15-210), Carnegie Mellon University, October 9, 2012.
- [18] X. Yang, M. Yan, S. Pan, X. Ye, and D. Fan, “Simple and Efficient Heterogeneous Graph Neural Network,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
- [19] Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford University Press.
- [20] Albert-László Barabási. *Network Science*. Cambridge University Press, Cambridge, United Kingdom, 2016. ISBN: 978-1107076266. URL: <http://networksciencebook.com/>.
- [21] Kivelä, M., Arenas, A., Barthélemy, M., Gleeson, J. P., Moreno, Y., & Porter, M. A. (2014). *Multilayer networks*. Journal of Complex Networks, 2(3), 203–271.

- [22] Aleta, A., et al. (2025). *Multilayer network science: theory, methods, and applications*. arXiv preprint arXiv:2511.23371.
- [23] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- [24] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- [25] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. MIT Press, 2006.
- [26] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The Graph Neural Network Model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009, doi: 10.1109/TNN.2008.2005605.
- [27] Bharti Khemani, Shruti Patil, Ketan Kotecha, and Sudeep Tanwar. *A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions*. *Journal of Big Data*, vol. 11, no. 1, p. 18, 2024.
- [28] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” *International Conference on Learning Representations (ICLR)*, 2017.
- [29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” *International Conference on Learning Representations (ICLR)*, 2018.
- [30] D. W. Zimmerman, I. S. Davis, and C. L. Worth, “The Theory of Dark Network Design”, Naval Postgraduate School, Monterey, California, 2009. <http://hdl.handle.net/10945/5013>

- [31] Malm, A., & Bichler, G. (2011). Networks of collaborating criminals. *Global Crime*, 12(1), 1–23.
- [32] Morselli, C. (2009). *Inside criminal networks*. Springer.
- [33] Bright, D., Hughes, C., & Chalmers, J. (2012). Illuminating dark networks: A social network analysis of an Australian drug trafficking syndicate. *Crime, Law and Social Change*, 57(2), 151–176.
- [34] Calderoni, F. (2012). The structure of drug trafficking mafias: The 'Ndrangheta and cocaine. *Crime, Law and Social Change*, 58(3), 321–349.
- [35] Campana, P. (2012). Listening to the wire: Criteria and techniques for the quantitative analysis of phone intercepts. *Trends in Organized Crime*, 15(1), 13–30.
- [36] Levi, M. (2013). Money for crime and money from crime: Financing crime and laundering crime proceeds. *European Journal on Criminal Policy and Research*, 19(2), 99–113.
- [37] Di Méo, G. (2023). *Historical Co-offending Networks: A Social Network Analysis Approach*. The British Journal of Criminology.
- [38] Ficara, A., Fiumara, G., Catanese, S., De Meo, P., & Liu, X. (2022). *The Whole Is Greater than the Sum of the Parts: A Multilayer Approach on Criminal Networks*. Future Internet, 14(5), 123.
- [39] Berlusconi, G., Calderoni, F., Parolini, N., Verani, M., & Piccardi, C. (2016). Link Prediction in Criminal Networks: A Tool for Criminal Intelligence Analysis. *PLoS ONE*, 11(4), e0154244.

- [40] F. Varese, “The Structure and the Content of Criminal Connections: The Russian Mafia in Italy,” *European Sociological Review*, vol. 29, no. 5, pp. 899–909, 2013.
- [41] L. Cavallaro, A. Ficara, P. De Meo, G. Fiumara, S. Catanese, O. Bagdasar, W. Song, and A. Liotta, “Disrupting Resilient Criminal Networks through Data Analysis: The Case of Sicilian Mafia,” *PLOS ONE*, 2020.
- [42] G. Berlusconi, F. Calderoni, N. Parolini, M. Verani, and C. Piccardi, “Link Prediction in Criminal Networks: A Tool for Criminal Intelligence Analysis,” *PLOS ONE*, vol. 11, no. 4, p. e0154244, Apr. 2016. <https://doi.org/10.1371/journal.pone.0154244>
- [43] V. H. Masías, M. Valle, C. Morselli, F. Crespo, A. Vargas, and S. Laengle, “Modeling Verdict Outcomes Using Social Network Measures: The Watergate and Caviar Network Cases,” *PLOS ONE*, vol. 11, no. 1, p. e0147248, Jan. 2016. <https://doi.org/10.1371/journal.pone.0147248>
- [44] E. Budur, S. Lee, and V. S. Kong, “Structural Analysis of Criminal Network and Predicting Hidden Links using Machine Learning,” Stanford University, Technical Report, 2020.
- [45] V. Dogan, S. D. Prestwich, and B. O’Sullivan, “A Graph Neural Network-Based Role Classification in Criminal Networks CSCI–RTCW,” in *Proc. 2024 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, Dec. 2024. <https://hdl.handle.net/10468/17404>
- [46] C. Yang, “CrimeGAT: Leveraging Graph Attention Networks for Enhanced Predictive Policing in Criminal Networks,” Shanghai Business School, Technical Report, 2024.

- [47] Roberts, N., & Everton, S. (2011). *Noordin Top Terrorist Network*. *Connections*, 31(1), 1–15.
- [48] Calderoni, F., Brunetto, D., & Piccardi, C. (2017). Communities in criminal networks: A case study. *Social Networks*, 48, 116–125.

Appendix

Certificate of Authorization for Correction and Deposit



شهادة الترخيص بالإيداع

أنا الأستاذ(ة):

Mme. BRAHIM Nacera

بصفتي رئيس (ة) والمسؤول (ة) عن تصحيح مذكرة تخرج ماستر الموسومة بـ

Graph AI for Criminal Network Analysis

من انجاز الطالب(ة):

DABOUZ Mohamed

والطالب(ة):

TIZEGGAGHINE Yacine

الكلية: العلوم والتكنولوجيا.

القسم: الرياضيات والإعلام الآلي.

الشعبة: اعلام الي.

التخصص: الأنظمة الذكية لاستخراج المعارف.

تاريخ التقييم/المناقشة: **21/06/2026**

أشهد ان الطالب (الطالبة) قد قام (قاموا) بالتعديلات والتصحيحات المطلوبة من طرف لجنة المناقشة وان المطابقة بين النسخة الورقية والالكترونية استوفت جميع شروطها.

مصادقة رئيس القسم

امضاء المسؤول عن التصحيح

