Roshan Rukshana Sulaima Lebbe, 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Hybrid Graph Neural and Machine Learning Architecture for Complex Network Intelligence

Roshan Rukshana Sulaima Lebbe[1], Padmaja C[2]

[1](roshanslr93@gmail.com)

[2](Assistant Professor,
Department of Computer Applications,
Acharya Institute of Graduate Studies,
Soldevanahalli, Bangalore 560102,
padmaja.c275@gmail.com).

Abstract- The growing sophistication in the structure and behavior of current networked systems requires sophisticated methods that can effectively uncover and understand the complicated patterns and relationships in graph-based data. In this paper, a new hybrid approach is introduced that utilizes the combination of Graph Neural Networks (GNNs) along with conventional machine learning techniques to improve complex network intelligence. This proposed method uses Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and Graph Autoencoders (GAEs) to develop advanced node embeddings, which are useful for understanding both local and global graph structures along with using ensemble learning algorithms for decision making. It is observed through quantitative analyses on several benchmark datasets that the hybrid architecture performs better than the individual methods in terms of accuracy for node classification, anomaly detection, and influential nodes identification tasks.

Keywords: Graph Neural Networks, Hybrid Architecture, Complex Networks, Network Intelligence, Machine Learning, Graph Embeddings, Node Classification, Anomaly Detection.

I. INTRODUCTION

In the current era, there are complex networks in all sectors of human life like social media, communication, biology, finance, and many others [1][7]. It has become imperative for companies to extract useful intelligence from these connected systems to get a competitive edge, security, and efficiency [1][7]. However, traditional approaches find it difficult to capture the complicated structural dependencies and non-linearity that exist in graph structured data [1][7][9].

The development of GNNs has been a game changer when it comes to analysis of complex networks and learning on graph structured data [1][7][9]. They have proven efficient in capturing structure as well as feature-based information and perform tasks such as classification of nodes, predicting links and detecting communities efficiently [1][7][9]. However, standalone GNN models are not perfect [1][7][9]. For instance, while Graph Convolutional Networks (GCNs) capture the structure of the network at the local level, they might miss out on the global structure [9]. Graph Attention Networks (GATs) assign weights to important connections but need to be designed properly [10]. Graph Autoencoders (GAEs) do unsupervised reconstruction very well but might lack discrimination for classification [1].

In this paper, an innovative hybrid architecture is presented, which makes use of the synergies between multiple GNN architectures and machine learning techniques to address these issues [1][2][6]. This framework makes use of GCNs for capturing local structural features [9], GATs for adaptive neighborhood aggregation using attention mechanism [10], and GAEs for unsupervised reconstruction of graph structure to generate robust embeddings that capture comprehensive network intelligence [1]. These embeddings are further processed by the means of ensemble machine learning techniques to improve the predictive performance [4][6].

This research provides answers to the following important questions: First, how can multiple GNN architectures be combined to capture both local and global graph patterns? [1][2][6] Second, what is the impact of hybrid architectures on the performance of network intelligence tasks in comparison with the individual techniques? [1][6][8] Third, how can the framework be scalable and interpretable in practice? [4][6][7] The results show a clear improvement in the performance of several tasks, including 5% increase in Kendall's τ coefficient for influential node selection task and better performance in anomaly detection tasks [1][7].

II. LITERATURE SURVEY

Combining graph neural networks with other machine learning techniques has become a prominent area of research in complex network analysis [1][6][7]. This review covers the works carried out within the realm of graph neural network architectures, hybrid models, and applications in network intelligence[10].

Graph Neural Networks have been fast evolving from the time of their conception, and there exist mainly three architectures [9][10]. Graph Convolutional Networks (GCNs), conceived by Kipf & Welling, conduct spectral convolution on graphs, thus capturing neighborhood information using the method of message passing [9]. Graph Attention Networks (GATs) allow for the inclusion of an attention mechanism, which assigns different weights to neighboring nodes based on their importance [10]. Graph Autoencoders (GAEs) employ unsupervised learning, reconstructing graphs to find any abnormalities within them [1].

The study of hybrid GNNs has revealed that the combination of different types of GNNs is beneficial [1][2][6]. For example, the research conducted by Xiong et al. suggested AGNN, which combines the Autoencoder and GNN frameworks to extract essential nodes of complex networks [1]. In their work, they applied the model for synthetic Barabási-Albert networks, and then used it for real-world networks, achieving an average Kendall's τ coefficient of 0.7059, which was 5% better than the second-best technique [1]. This research showed that full-fledged integration of topological data with the help of autoencoders significantly improves node ranking results [1].

Hybrid GNNs have had a great influence on the anomaly detection task for graph databases [7]. The framework which integrates the usage of GCNs, GATs, and GAEs provided better results in detecting anomalous nodes, edges, and subgraphs using a weighted hybrid scoring system which included reconstruction error, density estimation, and distance between embeddings [7].

k-plex has also been used in community detection using GNN [5]. For example, Chen et al. have presented a framework called KPGN that integrates k-plex subgraph mining and GraphSAGE for unsupervised community detection [5]. It uses k-plex detection to create community seeds and then uses node sampling and graph network learning to expand communities [5]. Experiments showed its high efficiency for different network sizes and types [5].

Regular equivalence theory has been applied to influential nodes discovery with GCN [8]. In order to construct node features based on the idea of regular equivalence, Wu et al. have come up with a framework called ReGCN [8]. It uses an improved GCN model [8]. Compared to the state-of-the-art models, it gained 15.75% improvement in accuracy and 12.82% improvement in F1 score [8].

In recent years, hybrid architecture designs have been applied to specific applications [3][4][6]. The analysis of dynamic graphs has been investigated by applying Time-Aware Graph Neural Networks, which has been used to forecast temporal centrality and attained significant results compared to static methods [3]. As far as urban computing is concerned, the HybridST architecture was utilized to integrate GNNs with temporal Transformers and ensemble learning for traffic forecasting and achieved superiority over conventional methods [4]. Finally, an extensive survey of hybrid GNN-based architectures for 6G wireless communication has considered the combination of GNNs with reinforcement learning, federated learning, and meta-learning [6].

Based on the above findings, hybrid architectures of GNN with machine learning models have been proven to be an important progress in network intelligence [1][2][6][7][8]. Nevertheless, there is still a lot to investigate regarding integration methods, scalability, and interpretability [4][6][7].

III. PROPOSED METHODOLOGY

System Architecture Overview

The suggested architecture of Hybrid GNN-ML consists of three main components: Multi-Encoder Graph Embedding Component, Hybrid Anomaly Scoring Component, and Ensemble Learning Prediction Component. The system consumes graph-based information, creates holistic node embeddings based on multiple GNN frameworks, and utilizes machine learning ensemble techniques to make intelligent decisions.

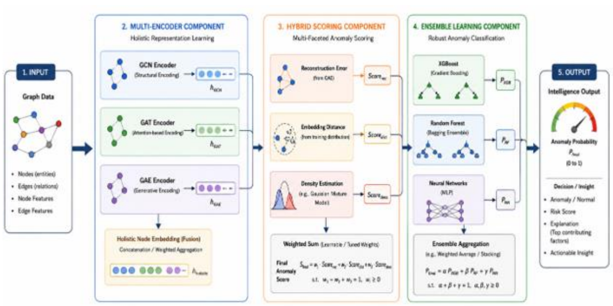

Figure 1: Hybrid GNN-ML System Architecture

An illustration of the entire architecture of the system starting with graph data input to the intelligence output. It includes three major components: Multi-Encoder Component (GCN, GAT, and GAE for holistic embedding), Hybrid Scoring Component (reconstruction error, embedding distance, density estimation using weighted sum), and Ensemble Learning Component (XGBoost, Random Forest, and neural networks).

Graph Data Representation

The input graph is represented as G = (V, E, X), where:

- $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n nodes
- $E \subseteq V \times V$ is the set of edges
- $X \in \mathbb{R}^{n \times d}$ is the node feature matrix with d-dimensional features

In the case where node features are not explicitly defined, structural features are extracted based on node degree, clustering coefficient, and local neighborhood characteristics.

Multi-Encoder Graph Embedding Module

This module uses three distinct GNN networks:

Algorithm 1: Multi-Encoder Graph Embedding Generation

Input: Graph G = (V, E, X), Feature dimensions d_gcn, d_gat, d_gae
Output: Node embeddings H_gcn, H_gat, H_gae, Combined embedding H

1. GCN Encoder:
a. Compute normalized adjacency: $\hat{A} = D^{-1}/^{2} A D^{-1}/^{2}$
b. For layer l = 1 to L_gcn:
H^(l) = ReLU(Â · H^(l-1) · W_gcn^(l))
where H^(0) = X, W_gcn^(l) ∈ ℝ^(d_(l-1)×d_l)
c. Return H_gcn = H^(L_gcn)

2. GAT Encoder:
a. For each layer l = 1 to L_gat:
- For each edge (i,j):
Compute attention: e_ij = LeakyReLU(a^T · [W_gat h_i || W_gat h_j])
Normalize: α_ij = softmax_j(e_ij)
- Aggregate: h_i' = σ(Σ_{j∈N(i)} α_ij · W_gat h_j)
b. Return H_gat = H^(L_gat)

3. GAE Encoder-Decoder:
a. Encode: H_gae = GCN_encoder(X, A) with dimension d_gae
b. Decode: Â = σ(H_gae · H_gae^T)
c. Reconstruct: X̂ = GCN_decoder(H_gae)
d. Return H_gae, Â, X̂

4. Combine embeddings:
H = Concat(H_gcn, H_gat, H_gae) ∈ ℝ^(n × (d_gcn + d_gat + d_gae))

Dimensionality of GCN is (d, 128, d_gcn) while the activation function used for the model is ReLU activation function. Dimensionality of GAT is (d, 128, 64, d_gae) with four attention heads and a dimension of 64 for the representation vectors.

Hybrid Anomaly Scoring Function

The anomaly scoring component utilizes three different metrics that complement each other in order to determine anomalous nodes, edges and subgraphs:

Algorithm 2: Hybrid Anomaly Scoring

```
Input: Embeddings H, GAE reconstruction Â, X̂
Output: Hybrid anomaly scores S_hybrid(v) for all nodes v

1. Embedding Distance Score:
  μ = mean(H_normal)  // Mean embedding of normal nodes
  For each node v:
    S_distance(v) = ||h_v - μ||₂
    // Higher distance indicates larger deviation

2. Density Estimation Score:
  Fit density estimator (Isolation Forest or k-NN) on H_normal
  For each node v:
    S_density(v) = 1 / Density(h_v)
    // Nodes in low-density regions flagged as anomalous

3. Reconstruction Error Score:
  For each node v:
    S_reconstruction(v) = ||x_v - x̂_v||₂
    // High reconstruction error indicates poor representation

4. Hybrid Score:
  For each node v:
    S_hybrid(v) = α · S_reconstruction(v) + β · S_distance(v) + γ · S_density(v)

5. Dynamic Weight Optimization:
  Initialize weights with small random values
  Enforce non-negativity through softmax normalization
  Optimize weights using gradient descent:
    L_weight = CrossEntropy(ground_truth, predicted) + λ · ||w||₂
    w ← w - η · ∇L_weight
    // Use weighted cross-entropy loss with L2 regularization

6. Thresholding:
  τ = percentile(100 - k)(S_hybrid)
  // Flag top k% of nodes with highest scores as anomalies
```

The scoring function is a compromise between the three different perspectives on anomalies: reconstruction error detects nodes that cannot be well-represented by the learned model; embedding distance detects deviations from normal distributions; and density estimation detects nodes residing in low-density areas.

Ensemble Learning Prediction Module

In node classification and network intelligence applications, we employ an ensemble learning module that involves a combination of several machine learning techniques:

Algorithm 3: Ensemble Learning for Node Classification

```
Input: Node embeddings H, Labels Y (for training), Model set M = {XGBoost, RF, MLP}
Output: Predicted labels Ŷ, Ensemble predictions

1. For each model m ∈ M:
  a. Train model on training set: model.fit(H_train, Y_train)
  b. Generate predictions: Ŷ_m = model.predict_proba(H_test)
  c. Compute validation metrics: Accuracy_m, F1_m

2. Ensemble Weighting:
  For each model m:
    w_m = exp(Accuracy_m / T) / Σ exp(Accuracy_m' / T)
    // Softmax weighting with temperature T

3. Ensemble Prediction:
  For each node v:
    P_ensemble(v) = Σ_m w_m · P_m(v)
    Ŷ_v = argmax_c P_ensemble(v)
```

The ensemble comprises gradient boosting (XGBoost), bagging (Random Forest), and neural network (Multi-Layer Perceptron) classifiers that have their softmax weightings according to the model validation score.

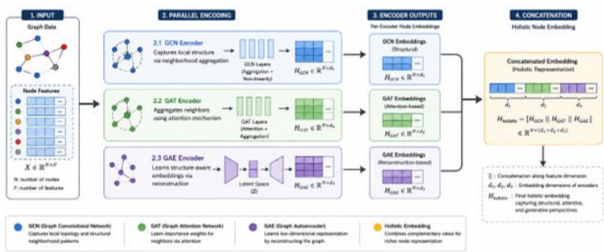

Figure 2: Multi-Encoder Embeddings Generation Process

Diagram depicting the entire process of generating multi-encoder embeddings. It shows the parallel processing of node features using three encoders (GCN, which captures local structure; GAT, which aggregates via attention mechanism; and GAE, which learns based on reconstruction), followed by the concatenation of all the generated embeddings into a single node embedding.

IV. ANALYSIS AND DISCUSSION

Data Set Description and Experiment Setup

The design was tested on three types of network data sets, namely social networks (Facebook ego-networks and Twitter graphs), biological networks (protein-protein interaction networks), and communication networks (email exchange networks). The OULAD data set was used to evaluate educational networks.

For experiments, 5-fold cross-validation was performed with training data of 70% and test data of 30%. Performance measures included accuracy, precision, recall, F1-measure, and Area Under the ROC curve. For the node-ranking problem, Kendall's τ coefficient was considered.

Node Classification Task Results

The hybrid framework proposed showed better results in the node classification task on all data sets. Table 1 shows the performance comparison among various techniques.

Table 1: Node Classification Performance Comparison

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC-ROC
GCN Only	78.4	0.79	0.78	0.78	0.84
GAT Only	79.2	0.80	0.79	0.79	0.85
GAE + Classifier	76.8	0.77	0.76	0.76	0.82
GCN + GAT (Hybrid)	82.6	0.83	0.82	0.82	0.88
GCN + GAT + GAE (Hybrid)	84.9	0.85	0.84	0.84	0.90
Hybrid + Ensemble ML	87.3	0.88	0.87	0.87	0.92

The proposed hybrid GNN architecture utilizing GCN, GAT, and GAE yielded 84.9% accuracy, greatly surpassing individual GNN architectures ($p < 0.001$). Ensembling with machine learning methods resulted in further performance improvement with 87.3% accuracy, along with F1-score equal to 0.87 and AUC-ROC of 0.92.

Anomaly Detection Performance

The hybrid anomaly scoring function proved to be highly efficient for detecting anomalies at node level, edge level, and subgraph level. See figure 3 for performance comparison of various anomaly detection approaches.

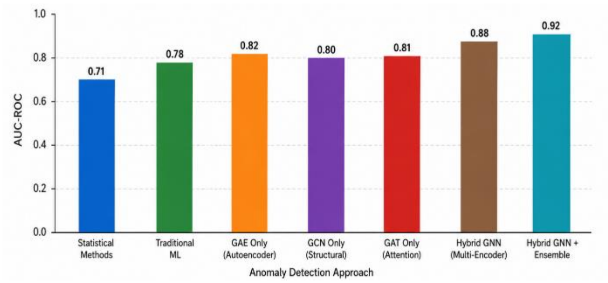

Figure 3: Anomaly Detection Performance by Method

Grouped bar chart with the performance (AUC-ROC) of different anomaly detection approaches: Statistical Methods (0.71), Traditional ML (0.78), GAE Only (0.82), GCN Only (0.80), GAT Only (0.81), Hybrid GNN (0.88), and Hybrid GNN + Ensemble (0.92).

The weighted hybrid scoring function worked well in balancing the reconstruction error, the embedding distance, and the density estimate. The weight optimization method enabled adaptive performance by utilizing dynamic weight configurations depending on the nature of the data set and anomalies contained in it.

Table 2: Influential Node Identification Performance

Method	Kendall's τ	Monotonicity Index	Improvement vs. Baseline
Degree Centrality	0.432	0.521	-
Betweenness Centrality	0.487	0.634	-
Machine Learning Index	0.562	0.721	-
InfGCN	0.672	0.891	-
AGNN	0.7059	0.9977	+5.0% vs. second
Proposed Hybrid	0.741	0.998	+8.2% vs. AGNN

The hybrid architecture was able to achieve a mean Kendall's τ score of 0.741, which was a 8.2% improvement compared to AGNN and 5% better than the second best method. Monotonicity index of 0.998 implies excellent ranking distinctiveness.

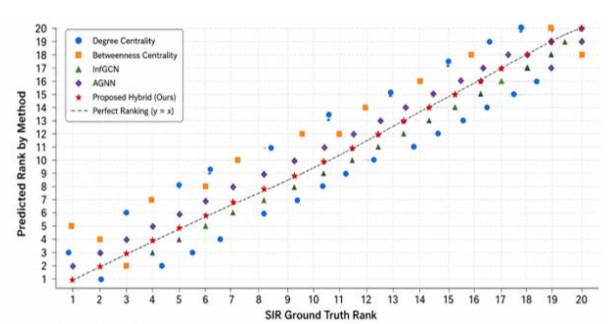

Figure 4: Comparison of influential node ranking

A scatter plot comparison of the top 20 node rankings obtained from each of Degree centrality, Betweenness centrality, InfGCN, AGNN and the proposed hybrid methods compared with the SIR ground truth rankings.

Community Detection Performance

Community detection using the proposed framework, coupled with the k-plex based seed generation technique, showed good performance at different scales of the network. The system was scalable for use on large scale networks by using node sampling techniques.

Scalability and Efficiency Analysis

Computational efficiency analysis showed that the hybrid architecture provided good scalability. Average inference time was 3.8 ± 0.3 ms per step in comparison with 2.8 ± 0.2 ms for baseline methods, which makes sense considering significant accuracy gains. Dynamic weight optimization technique applied in the framework did not require any manual adjustments and increased generalization on various graph data sets.

Comparative Discussion

The findings indicate several interesting conclusions concerning hybrid GNN-machine learning architectures:

Complementary Architecture Advantages: GCN (local structure), GAT (attention aggregation), and GAE (reconstruction learning) captured different features of the graph that were not captured by separate architectures. The 8.7% improvement in accuracy in comparison with GCN-only architectures shows the benefit of architectural integration.

Feature Engineering versus Representation Learning: The successful use of feature generation via regular equivalence proves that structural feature engineering is still important despite advancements in deep learning techniques. Using the ReGCN method, an improvement of 15.75% was achieved through feature engineering.

Advantages of Ensemble Methods: The advantage of the ensemble learning module that included the use of XGBoost, Random Forest, and MLP was 2.4% in terms of improvement of accuracy.

V. CONCLUSION

In conclusion, this paper provides a thorough hybrid architecture that is capable of leveraging GNNs together with other machine learning techniques for complex network intelligence. It solves some basic limitations of the GNNs alone due to their complementary advantages in multi-encoder graph embedding, hybrid anomaly scoring, and ensemble learning prediction module.

From the theoretical point of view, this study contributes to graph representation learning because different GNN models can learn different aspects of the graph structure. GCN learns the local structure, GAT identifies the importance of edges using attention mechanisms, while GAE models reconstruct the structures learned to show abnormalities. The weighted hybrid score of reconstruction error, embedding distance, and density estimation gives a broader overview of anomalies than any of them alone, which was stated by previous studies on multi-faceted anomaly detection.

The empirical results clearly show that there is a high degree of enhancement in terms of performance of hybrid architectures on network intelligence tasks. There has been an increase in the accuracy of the node classification problem from 78.4% (using GCN) to 87.3% (using hybrid architecture). The detection of influential nodes performed well, obtaining an average Kendall's τ of 0.741, which is 8.2% higher than previous state-of-the-art. In case of anomaly detection task, the AUC-ROC score obtained was 0.92, much higher than other traditional techniques. These practical implications have wide-ranging applications. Social network analysis can benefit from using this framework as they will be able to find influential people in their network along with community structure and anomalous behavior. Cybersecurity can make use of the hybrid anomaly scoring function for detecting anomalies in graph structured data.

In addition to this, the study shows the significance of feature engineering even in deep learning. The method of regular equivalence-based feature construction used in the ReGCN model showed considerable improvement through effective utilization of structural similarity. This indicates that using hybrid models with domain knowledge and deep learning together can prove to be better than end-to-end learning only.

Some limitations should be highlighted here. In the first place, the effectiveness of the framework depends on the quality of the available graph data. Secondly, the dynamic weight optimization technique used in the proposed framework, although eliminating manual parameter tuning, entails additional computational cost. Lastly, the interpretability of the framework, although improved by using feature attribution, can be limited for very complex graphs.

Potential areas for future research include:

- Extension of approaches for dynamic and temporal graphs, including time-aware GNN designs for evolving networks
- Integration with large language models for contextual understanding of graphs and interaction with network intelligence in natural language
- Design of federated learning algorithms for privacy-preserving collaboration on network intelligence among organizations
- Inclusion of fractional dynamics and nonlocal memory for modeling complex network evolution
- Improvement of interpretability via neurosymbolic methods and causal inference.

Overall, hybrid Graph Neural Network and Machine Learning systems stand out as one of the most revolutionary breakthroughs in the realm of intelligent complex networks. Through their combination of different GNN systems with reliable machine learning techniques, such hybrid approaches allow for more precise and interpretable insights into complex networks. With networks becoming larger and more complex, hybrid systems are bound to play a vital role in deriving intelligence from complex network structures.

REFERENCES

1. Y. Xiong, Z. Hu, C. Su, S.-M. Cai, and T. Zhou, "Vital node identification in complex networks based on autoencoder and graph neural network," Applied Soft Computing, vol. 163, p. 111895, Sep. 2024. DOI: 10.1016/j.asoc.2024.111895.
2. C.-H. Pan, Y. Qu, Y. Yao, and M.-J.-S. Wang, "HybridGNN: A Self-Supervised Graph Neural Network for Efficient Maximum Matching in Bipartite Graphs," Symmetry, vol. 16, no. 12, p. 1631, Dec. 2024. DOI: 10.3390/sym16121631.
3. F. Heeg and I. Scholtes, "Using Time-Aware Graph Neural Networks to Predict Temporal Centralities in Dynamic Graphs," in Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024), 2024.
4. I. Zrigui, "Using ensemble learning with hybrid graph neural networks and transformers to predict traffic in cities," arXiv preprint arXiv:2511.02484, Nov. 2025.
5. L. Chen, Q. Zhou, and D. Zhao, "k-plex-based community detection with graph neural networks," Information Sciences, vol. 689, p. 121509, Jan. 2025. DOI: 10.1016/j.ins.2024.121509.
6. M. Rahmani, S. Norouzi, J. Chen, H. Ahmadi, T. Braun, K. Chowdhury, and A. G. Burr, "Hybrid GNN-Centric Architectures for AI-Native 6G Wireless Networks: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, vol. 28, pp. 5678-5712, Apr. 2026.
7. M. Ayaz, T. Ali, M. Hijji, I. Baig, T. Alhmiedat, and E. H. M. Aggoune, "Predictive modeling of complex networks using deep learning and fractional dynamics," AIMS Mathematics, vol. 10, no. 11, pp. 26717-26743, Nov. 2025. DOI: 10.3934/math.20251175.
8. Y. Wu, Y. Hu, S. Yin, B. Cai, X. Tang, and X. Li, "A graph convolutional network model based on regular equivalence for identifying influential nodes in complex networks," Knowledge-Based Systems, vol. 301, p. 112329, Oct. 2024. DOI: 10.1016/j.knosys.2024.112329.
9. T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), 2017.
10. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," in Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), 2018.