Harnessing the Power of Ensembled Deep Learning and Graph Neural Networks for Multidimensional Insights

Paramjot Kaur Sarao[1][a], Manish Sharma[1][b] and Anupriya[2][c]
[1]*Institute of Engineering, Chandigarh University, Mohali, Punjab, India*
[2]*Department of Computer Science and Engineering, Chandigarh University, Mohali, India*

Keywords: Ensemble Methods, Deep Learning, Graph Neural Networks, Link Prediction, Bagging, Stacking.

Abstract: Ensemble methods have long been recognized for their ability to enhance the performance and robustness of machine learning models. With the advent of deep learning and Graph Neural Networks (GNNs), the integration of ensemble techniques has opened new avenues for research and application. This paper explores the synergistic potential of combining deep learning ensembles with graph neural networks (GNNs) to enhance performance on complex graph-structured data tasks. The paper first examines traditional ensembling methods adapted for deep learning, including bagging, boosting, and stacking approaches tailored to neural architectures. We then delve into novel ensemble techniques specifically designed for GNNs, addressing the unique challenges posed by graph-structured data. This covers diverse applications, from computer vision and natural language processing to recommendation systems and bio-informatics. It concludes by identifying open challenges, promising research directions, and potential real-world impacts of ensembling deep learning and GNN models, providing a roadmap for future work in this rapidly evolving field.

1 INTRODUCTION

In recent years, deep learning and graph neural networks (GNNs) have revolutionized the field of artificial intelligence, achieving unprecedented performance across a wide range of tasks. Concurrently, ensemble methods, which combine multiple models to improve overall predictive performance, have proven to be powerful techniques in machine learning.Deep learning has demonstrated remarkable success in areas such as computer vision, natural language processing, and speech recognition. However, challenges persist in terms of model uncertainty, overfitting, and the need for large amounts of labeled data. Graph Neural Networks, on the other hand, have emerged as a promising approach for learning on graph-structured data, with applications ranging from social network analysis to molecular property prediction. Despite their success, GNNs face unique challenges related to scalability, heterogeneity, and the dynamic nature of real-world graphs.

Ensemble methods(Dietterich, 2000) offer a potential solution to many of these challenges by leveraging the power of multiple diverse models. By combining predictions from different models, ensembles can reduce overfitting, improve generalization, and provide more robust predictions. In the context of deep learning and GNNs, ensemble techniques can be adapted and extended to address domain-specific issues and exploit the unique structures of neural networks and graph data.This paper aims to provide a comprehensive overview of the current state of research in ensembling deep learning models and GNNs. We will explore various ensemble strategies, including bagging, boosting, and stacking, as well as more recent innovations tailored to neural architectures and graph-structured data. The paper will cover theoretical foundations, practical implementations, and empirical results across different application domains. Furthermore, we will discuss the challenges and open questions in this field, such as balancing model diversity and computational efficiency, adapting ensemble methods to dynamic and heterogeneous graphs, and developing interpretable ensemble models. By synthesizing recent advancements and identifying future research directions, this paper aims to serve as a valuable resource for researchers and practitioners working at the intersection of ensemble learning, deep neu-

[a] https://orcid.org/0009-0008-4219-7362
[b] https://orcid.org/0000-0002-0129-400X
[c] https://orcid.org/0000-0002-2245-4092

Sarao, P. K., Sharma, M. and Anupriya,
Harnessing the Power of Ensembled Deep Learning and Graph Neural Networks for Multidimensional Insights.
DOI: 10.5220/0013589400004664

In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 2*, pages 203-211
ISBN: 978-989-758-763-4

ral networks, and graph-based machine learning.

2 BACKGROUND

2.1 Deep Learning

Solely working on the concept or architecture of Artificial Neural Networks(Wilamowski, 2009), deep learning is a specialized form of machine learning. It employs multi-layered neural networks to progressively learn and represent data at increasing levels of abstraction. This approach allows the model to grasp intricate patterns by building upon simpler concepts learned in earlier layers.The architecture of deep learning models consists of numerous computational layers between the input and output, each performing various linear and non-linear transformations. These layers work in a hierarchical, sequential, or recurrent manner to extract features from raw data at multiple levels of complexity. In essence, a deep learning model can be viewed as a series of inter-connected, continuous transformations that map input data to output predictions. This mapping is achieved by learning from a comprehensive set of input-output pairs, known as training data. The learning process involves iteratively adjusting the parameters of each transformation in the network using optimization algorithms, which fine-tune the model based on its performance. This layered approach enables deep learning models to automatically discover and engineer relevant features from raw data, eliminating the need for manual feature extraction. As a result, deep learning has demonstrated remarkable capabilities in handling complex, high-dimensional data across various domains, including computer vision, natural language processing, and speech recognition.

2.1.1 Key Architectures Include

Convolutional Neural Networks (CNNs: Convolutional Neural Networks (CNNs) are specialized deep learning architectures designed primarily for processing grid-like data, especially images (Li et al., 2021).They consist of convolutional layers that apply filters to detect features, pooling layers that reduce spatial dimensions, and fully connected layers for final output as shown in figure 1. CNNs leverage local connectivity, parameter sharing, and translation invariance to efficiently extract hierarchical features from input data. This architecture significantly reduces the number of parameters compared to fully connected networks, making them highly effective for tasks like image classification, object detection, and facial recognition. CNNs have revolutionized computer vision and have also been adapted for other domains, including natural language processing and signal analysis.

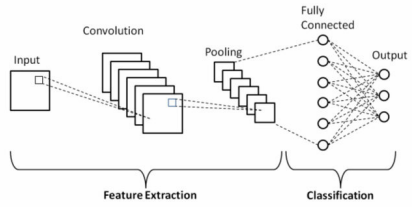

Figure 1: Convolutional Neural Network (CNN) Architecture (Phung and Rhee, 2019).

Recurrent Neural Networks (RNNs: Recurrent Neural Networks (RNNs) are a class of deep learning models designed to process sequential data as shown in figure 2. They feature loops that allow information to persist, enabling them to maintain a ”memory” of previous inputs. This architecture makes RNNs particularly suited for tasks involving time series, natural language, or any data with temporal dependencies. Key features include a hidden state that updates with each input, the Ability to handle variable-length sequences and shared parameters across time steps.
However, basic RNNs struggle with long-term dependencies due to vanishing/exploding gradients. This led to the development of more advanced variants like LSTM (Long Short-Term Memory) (Scher and Messori, 2021) and GRU (Gated Recurrent Unit) networks (Salem and Salem, 2022), which better capture long-range dependencies in sequences.

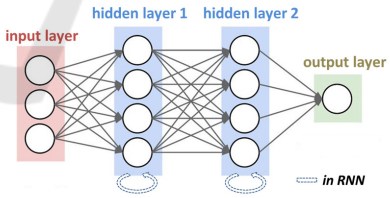

Figure 2: Recurrent Neural Networks (RNNs) Architecture (Ma et al., 2019).

Transformers: Transformers are a powerful deep learning architecture designed for sequence-to-sequence tasks, particularly in natural language processing. Key features include: Self-attention mechanism: Allows the model to weigh the importance of different parts of the input. Positional encoding: Maintains sequence order information. Multi-head attention: Enables learning from multiple representation subspaces. Feed-forward networks: Process the attention output. Layer normalization and residual connections: Improve training stability. Trans-

formers excel in tasks like machine translation, text summarization, and question-answering. They've led to breakthrough models like BERT and GPT. Unlike recurrent neural networks, Transformers process entire sequences in parallel, allowing for more efficient training on large datasets.

2.2 Graph Neural Networks (GNNs)

Graph Neural Networks (GNNs) (Wu et al., 2020) are a class of neural networks designed to operate on graph-structured data. Unlike traditional neural networks, which typically handle data in grid formats such as images or sequences, GNNs are tailored to manage data where entities (nodes) and their relationships (edges) form a graph. The key features of GNN includes-Node and Edge Features which represent entities relationships between entities in the graph respectively.Both nodes and edges can have associated features that provide additional context or information.Message Passing Mechanism-GNNs update node representations by aggregating information from their neighbors. This process involves two main steps: message passing (or aggregation) and node update.
Message Passing means nodes receive messages from their neighbors, which are functions of the neighbors' features while node update means updating the features by nodes based on the aggregated messages.

2.2.1 Basic GNN Layer

A typical GNN layer can be described by the following operations: Message Function: Computes messages between nodes based on their features and the features of the connecting edge.

$$m_{vu} = Message(h_v, h_u, e_{vu}) \tag{1}$$

where m_{vu} is the message from node u to node v,
h_v, h_u are the feature vectors of the respective nodes
e_{vu} is the feature edge between these nodes.
Aggregation Function: Aggregates messages from all neighboring nodes.

$$m_v = Aggregate(m_{vu} : u \in N(v)) \tag{2}$$

where $N(v)$ denotes the set of neighbors of node v.
Update Function: Updates the node's feature vector based on the aggregated message.

$$h'_v = Update(h_v, m_v) \tag{3}$$

where h'_v is the updated feature vector of node v. There are many architectures that work on the concept of GNNs namely Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), Graph Isomorphism Networks (GIN), Spatial-Temporal Graph Neural Networks (ST-GNN), Message Passing Neural Networks (MPNN), Graph Recurrent Neural Networks (GraphRNN), Graph Autoencoders (GAE), Relational Graph Convolutional Networks (R-GCN), GraphSAGE, ChebNet (Spectral-Based GNN), Dynamic Graph Neural Networks (DGNN), Hypergraph Neural Networks (HGNN), Graph Transformers, Diffusion Convolutional Neural Networks (DCNN) and Topology-Based GNNs.But,

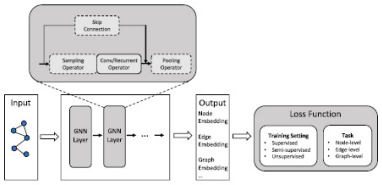

Figure 3: General Design of GNN model(Zhou et al., 2020).

2.2.2 Key Architectures Include

Graph Convolutional Networks (GCNs) Graph Convolutional Networks are a type of neural network specifically designed to operate on graph-structured data (Jin et al., 2021). They generalize the convolution operation commonly used in Convolutional Neural Networks (CNNs) to graphs, allowing for effective feature learning and representation in non-Euclidean domains such as social networks, molecular structures, and knowledge graphs. Basic Architecture: A typical GCN consists of multiple layers, each performing the graph convolution operation. The architecture includes:
Input Layer: Initial node features $H(0)$, which could be raw features or embeddings.
Hidden Layers: Multiple GCN layers that iteratively update the node features based on their neighbors.
Output Layer: Produces the final node representations used for downstream tasks like node classification, link prediction, or graph classification.
GCNs generalize the convolution operation to graphs. They aggregate information from a node's neighbors to update its representation. The basic layer of a GCN can be defined as:

$$h_v^{k+1} = \sigma \tag{4}$$

Graph Attention Networks (GATs) Graph Attention Networks are a type of Graph Neural Network (GNN) that leverages attention mechanisms to address the limitations of traditional GNNs like Graph Convolutional Networks (GCNs) (Vrahatis et al., 2024).Traditional GNNs, such as GCNs, use fixed or predefined weights for aggregating information from neighbors.GAT addresses these issues by introducing an attention mechanism to dynamically assign importance (weights) to each neighbor, allowing the

model to learn which neighbors are most relevant for the task.Key Concepts of GAT Node-Level Attention: GAT computes attention scores between a node and each of its neighbors to decide how much information to aggregate from each neighbor. Attention scores are learnable and depend on the node's and neighbor's features. Learnable Weights: The attention mechanism introduces trainable parameters that adaptively learn the importance of neighbors during training. Self-Attention Mechanism: Inspired by the success of attention mechanisms in sequence models (e.g., Transformers), GAT employs a similar idea tailored for graphs. Parallel Multi-Head Attention: GAT often uses multi-head attention to stabilize learning and capture diverse patterns by attending to multiple aspects of the data.
Architecture of GAT: 1.Input: Graph with nodes V, edges E, and node features $X \in R^{N*F}$,where N is the number of nodes, and F is the feature dimension.
2. Attention Mechanism: For a node i and its neighbor j, the attention score $\propto_{ij}$ is computed as:

$$\propto_{ij} = softmax_j(LeakyReLU(\overrightarrow{a}[Wh_i \parallel Wh_j])) \quad (5)$$

,where h_i, h_j: Feature vectors of nodes i and j,
W: Weight matrix to transform features,
$\overrightarrow{a}$: Learnable attention vector,
$[\cdot \parallel \cdot]$:Concatenation of feature vectors,
$softmax_j$:Normalization to ensure attention scores across neighbors sum to 1.
3. Feature Aggregation: The updated feature for node i is computed as:

$$h_i{}' = \sigma(\Sigma_{j \in N(i)} \propto_{ij} Wh_j) \quad (6)$$

, where σ : Nonlinear activation function (e.g., ReLU, ELU).
4.Multi-Head Attention: Multiple attention mechanisms are applied in parallel, and the outputs are either concatenated or averaged:

$$h_i{}' = \|_{k=1}^{K} \sigma(\Sigma_{j \in N(i)} \propto_{ij}^{k} W^k h_j) \quad (7)$$

, where K: Number of attention heads.

3 ENSEMBLING TECHNIQUES

Ensembling techniques are a set of methods in machine learning that combine the predictions of multiple models to produce a more robust, accurate, and generalizable final prediction. The core idea is that aggregating the strengths of diverse models can mitigate individual weaknesses, reduce overfitting, and improve predictive performance. These techniques are widely used in both classification and regression tasks, as well as in specialized domains like deep learning and graph-based learning.

3.1 Key Ensembling Techniques

The folloing are the key ensembling techniques (Ganaie et al., 2022).

1. Bagging (Bootstrap Aggregating): Bootstrap aggregating, also known as bagging, is a machine learning technique as shown in figure 4 that improves the accuracy and stability of classification and regression algorithms. It's an ensemble learning method that uses a group of models to work together to produce a better final prediction. The goal is to reduce variance and prevent overfitting.

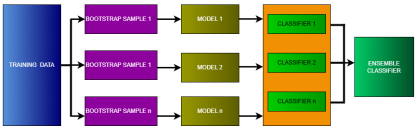

Figure 4: Bagging (Bootstrap Aggregating.)

2. Boosting: The technique involves training the weak learners sequentially, with each predictor trying to correct the errors of the previous one, with greater emphasis on difficult-to-learn examples as shown in figure 5. Boosting techniques help avoid underfitting of the model. Some examples include: AdaBoost, Gradient boosting, LightGBM and CatBoost.

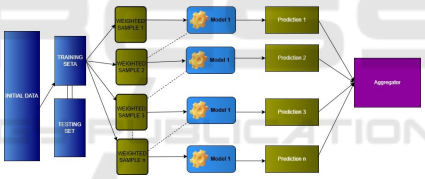

Figure 5: Boosting.

3. Stacking: Stacking, or stacked generalization, is an advanced ensembling technique that combines the predictions of multiple base models (or learners) using a meta-model (or meta-learner). The figure 6 shows that the meta-model learns how to best integrate the predictions of the base models to produce a more accurate final prediction.

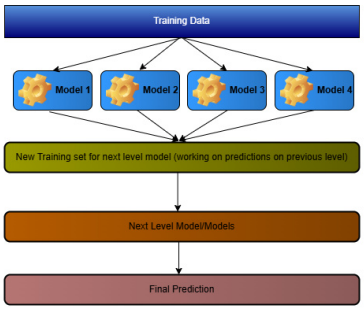

Figure 6: Stacking.

4 LITERATURE REVIEW

For some disease categorisation problems, quantum computing offers a more effective model than traditional machine learning techniques. Alzheimer's disease categorisation problems do not fully utilise the capabilities of quantum computing. To categorise Alzheimer's illness, we presented an ensemble deep learning model in this paper that is built on quantum machine learning classifiers. For the classification of AD disease, the datasets from the Alzheimer's Disease Neuro-imaging Initiative I and II are combined. To classify them as non-demented, mildly demented, moderately demented, and very mildly demented, authors integrated significant features that were derived from the merged images using the modified versions of the VGG-16 and ResNet50 models. Then,the authors fed these features into the Quantum Machine Learning classifier and evaluated the performance by using six metrics; accuracy, the area under the curve, F1-score,precision, and recall(Jenber Belay et al., 2024). The authors explored the various types of skin cancer, including squamous cell carcinoma (SCC), basal cell carcinoma (BCC), and melanoma. It has also provided a system for skin cancer detection using convolutional neural network (CNN) techniques, specifically the multi-model ResNet (M-ResNet) architecture. Researchers have provided a ResNet architecture with improved skin cancer detection performance that can handle deep networks. To detect skin cancer, the suggested method employs a comprehensive pipeline. To increase the model's ability to generalise, the dataset first undergoes pre-processing (PP) techniques such image resizing, normalisation, and augmentation methods. Improved accuracy, sensitivity, and specificity are achieved in skin cancer LEARNING Classification SYSTEM (SC-LCS) jobs as a result of the multi-model assembly(Sardar et al., 2024).

In order to better categorise malware variants into their respective families and increase classification accuracy, this study (Adamu et al., 2024) suggests a novel malware ensemble architecture that integrates deep learning methods with a variety of malware features. It increases its sensitivity to related malware families and improves its classification accuracy by extracting visual features from raw bytes (data) and malware's opcode frequency. The Microsoft Malware Classification Challenge benchmark dataset is used to validate the suggested strategy, and its effectiveness in contrast with that of other approaches. The findings demonstrate that the suggested method performs better than current techniques and detects disguised malware with a higher accuracy rate (99.3 perecnt). Additionally, experimental results demonstrate that the suggested method is more accurate in categorising malware variants and more sensitive to comparable malware families.

In cybersecurity, malware data classification is essential for identifying and removing harmful software from computer systems. Because deep learning techniques can automatically learn characteristics from raw data, they have been used to improve data categorisation performance. These methods are prone to overfitting, though, which may reduce their generalisability.

The table 1 describe the literature survey on ensembles in deep learning and graph neural networks, including insights, results, limitations and challenges.

This research addresses the problem by presenting a novel malware ensemble framework that improves the classification accuracy of malware variants by classifying them into their respective families using deep learning methods and different malware attributes. It increases its sensitivity to related malware families and improves its classification accuracy by extracting visual features from raw bytes (data) and malware's opcode frequency. (Adamu et al., 2024). In graph-structured data, graph neural networks, or GNNs, (Wei et al., 2023) have found widespread use. However, annotated data is frequently absent from current graph-based systems. To make inferences on a large amount of test data, GNNs must learn latent patterns from a small amount of training data. Overfitting and suboptimal performance are typically the results of GNNs' greater complexity and single point of model parameter initialisation. Furthermore, it is well known that adversarial attacks can target GNNs. With enhanced accuracy, generalisation, and adversarial robustness, we advance the ensemble learning of GNNs in this study. We present a novel technique, GNN-Ensemble, for building an ensemble of random decision graph neural networks based on the ideas of stochastic modelling.

5 PROPOSED SYSTEM ARCHITECTURE

The proposed system integrates deep learning (DL) models with Graph Neural Networks (GNNs) to leverage both the feature extraction and relational learning. The architecture is designed to ensemble predictions from both approaches, ensuring complementary strengths are utilized.

Input Data Module-the topmost layer in the diagram represents the system's entry point for data. It accepts multiple types of data (structured, unstruc-

Table 1: Summary of Existing work.

Title	Insights	Results	Limitations	Challenges
Deep Ensemble learning and quantum machine learning approach for Alzheimer's disease detection (Jenber Belay et al., 2024)	Ensemble deep learning combines multiple models to enhance performance. In the study, a quantum machine learning-based ensemble model achieved high accuracy in Alzheimer's disease classification.	Accuracy of 99.89, F1-score of 98.37 achieved. Outperformed state-of-the-art methods in Alzheimer's disease detection.	Full potential of quantum computing not fully utilized. Disparities in data observed after training for 10 epochs	Full potential of quantum computing not utilized for AD classification. Boosting performance of deep learning models through training epochs.
Ensemble Deep Learning Methods for Detecting Skin Cancer (Sardar et al., 2024)	The paper explores ensemble deep learning methods, particularly the multi-model ResNet architecture, for detecting various types of skin cancer, enhancing accuracy, sensitivity, and specificity in classification tasks.	Improved skin cancer detection using multi-model ResNet architecture. Promising results in accurately identifying different types of skin cancer.	———	Early identification crucial for effective treatment outcomes. Deep learning algorithms show promising results in skin cancer detection
Malware Classification Using Deep Learning and Ensemble Framework (Adamu et al., 2024)	The paper proposes an ensemble framework combining deep learning and multiple malware features to classify malware variants accurately, achieving a high detection rate of 99.3 percent for obfuscated malware.	Achieved 99.3 percent accuracy in detecting obfuscated malware . Outperformed existing methods in malware classification accuracy.	Susceptibility to overfitting Generalisability decrease due to deep learning techniques.	Overfitting in deep learning techniques affecting generalisability. Need for improved accuracy in classifying malware variants.
Super Deep Learning Ensemble Model for Sentiment Analysis (Garg and Subrahmanyam, 2023)	The paper introduces a Super Deep Learning Ensemble Model (SDL-EM) for sentiment analysis, combining various deep learning architectures to enhance accuracy and performance through ensemble learning techniques.	Superiority over state-of-the-art models in accuracy and metrics. Elevated performance and generalization capabilities demonstrated in experiments.	Conventional deep learning models struggle with accuracy and resilience. Inherent deficiencies and biases affect conventional deep learning models.	Conventional deep learning models face accuracy and resilience limitations. Inherent deficiencies and biases hinder conventional deep learning models.
Deep Learning Ensemble Method for Plant Disease Classification (Jain et al., 2023)	The paper introduces a Deep Learning Ensemble Method (NLRSGD-Ensemble) combining CNN, logistic regression, and stochastic gradient descent for accurate plant disease classification, achieving 97.7 percent accuracy.	Achieved 97.7 percent accuracy in plant disease classification experiments. Used CNN (NLRSGD-Ensemble) method for deep image attribute extraction.	———	Crop-borne illnesses impact net productivity. Early detection and warning to farmers can solve the problem.
GNN-Ensemble: Towards Random Decision Graph Neural Networks (Wei et al., 2023)	GNN-Ensemble constructs random decision Graph Neural Networks to improve accuracy, generalization, and adversarial robustness by combining multiple GNNs trained on different substructures and sub-features.	Improved accuracy, generalization, and adversarial robustness in GNNs. GNN-Ensemble reduces overfitting and enhances classification performance.	Overfitting and sub-optimal performance due to model complexity. Vulnerability to adversarial attacks.	Overfitting and sub-optimal performance due to model complexity. Vulnerability to adversarial attacks on GNNs.
Ensemble Learning for Graph Neural Networks (Wong et al., 2023)	Ensemble Learning for Graph Neural Networks (ELGNN) combines multiple GNN models to enhance accuracy, reduce bias and variance, and improve robustness in analyzing graph-structured data.	Ensemble learning improves GNN performance, robustness, accuracy, and reduces bias. ELGNN model combines diverse GNNs to mitigate noisy data impact.	Ensemble learning mitigates impact of noisy data. ELGNN enhances accuracy and reduces bias and variance.	Improve performance and robustness of Graph Neural Networks (GNNs) Mitigate impact of noisy data on GNN capabilities.
Ensemble Methods for Neural Network-Based Weather Forecasts (Scher and Messori, 2021)	The paper discusses ensemble methods for neural network-based weather forecasts, exploring perturbation techniques like random initial perturbations, retraining, random dropout, and singular vector decomposition to improve forecast accuracy.	Ensemble methods improve neural network weather forecasts. Retraining method shows highest improvement in ensemble mean forecasts.	Neural network forecasts have lower skill compared to numerical models. Generating ensemble with good spread-error relationship is challenging.	Generating ensemble with good spread-error relationship. Neural network forecasts skill lower than numerical weather prediction models

tured, and graph-structured). Deep Learning Models - this branch extracts meaningful features from data such as images, text, or tabular formats.Graph Neural Networks- this branch processes graph-structured data, capturing relationships between entities (nodes) and interactions (edges).Intermediate Fusion Layer- it combines feature representations from both the deep learning and GNN branches into a single, unified representation.The diagram shows the convergence of data paths from the DL and GNN blocks:

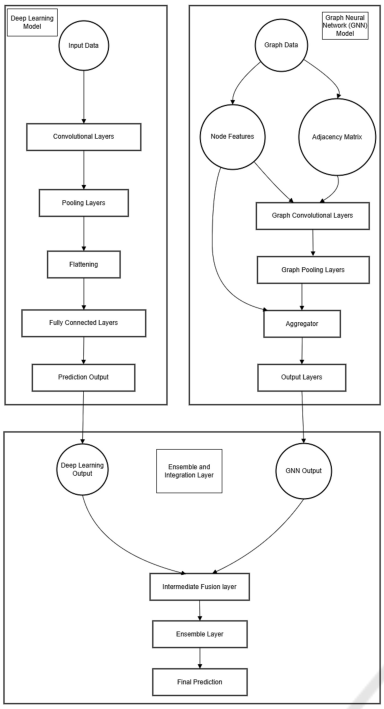

Figure 7: Proposed Architecture.

Concatenation: Directly stack features from both models.

Cross-Attention: Facilitate interactions between DL and GNN features for enhanced contextual understanding.And produces a combined feature vector, ready for prediction or ensembling. Ensembling Layer- it aggregates predictions from the DL and GNN pipelines, improving accuracy and robustness.Output- Final predictions enriched by the combined strengths of DL and GNN.

6 CHALLENGES

Despite their potential, combining ensembled deep learning and Graph Neural Networks (GNNs) presents notable challenges. Future directions should focus on developing lightweight, scalable ensemble frameworks that integrate GNNs and deep learning seamlessly, leveraging techniques like model distillation and federated learning. Advancements in dynamic graph modeling, automated hyperparameter tuning, and transfer learning for cross-domain generalization will further expand applicability. Standardized tools and libraries are also needed to simplify implementation and comparisonas shown in table 2. By addressing these challenges, ensembled approaches can unlock unprecedented capabilities in solving complex relational problems across diverse domains.

Table 2: Challenges and Solutions.

Sno	Challenges Faced	Solution Via ensembling
1	Overfitting	Ensembling multiple models reduces overfitting by averaging out model-specific biases and errors.
2	Scalability to Large Datasets	Ensembles can distribute the computational load by training smaller, specialized models on subsets of the data.
3	Sensitivity to Noisy Data	Ensembles can mitigate noise sensitivity by incorporating diversity in the models.
4	Interpretability	Ensembles provide an opportunity to combine interpretable models (e.g., shallow GNNs or feature-based decision trees) with high-performing
5	Transferability Across Domains	Ensembles can leverage transfer learning techniques by combining models trained on different domains.
6	Lack of Standardized Frameworks	The development of hybrid frameworks that integrate deep learning (e.g., PyTorch, TensorFlow) with GNN-specific libraries (e.g., DGL, PyTorch Geometric) allows seamless combination and experimentation.

7 CONCLUSION

Ensemble methods in deep learning and GNNs offer significant improvements in accuracy, robustness, and generalization. Despite challenges in computational complexity, interpretability, and scalability, the combination of these techniques holds great promise for advancing AI applications across various domains. In drug discovery, they can predict molecular properties by integrating feature-based learning from deep models with relational insights from GNNs. Social network analysis benefits from this combination for tasks like community detection and influence maximization. In fraud detection, financial networks modeled as graphs allow ensembles to identify anomalies by combining structural patterns with feature-based predictions. Recommender systems can improve accuracy by combining user-item interaction graphs processed by GNNs with user feature embeddings learned by deep networks. For cybersecurity, ensembles can enhance intrusion detection by integrating communication patterns from GNNs with temporal trends captured by recurrent deep models. In traffic management, urban graphs with intersections and roads can be analyzed to optimize routes and predict congestion. Supply chain optimization uses ensembles to model complex logistics networks, improving demand forecasting and route planning. In biological research, protein interaction networks can be studied for structure and function prediction, combining GNNs for spatial dependencies and deep models for sequence patterns. Stock market prediction can integrate company relationship graphs with financial trend data for enhanced market movement predictions. Lastly, smart city planning utilizes ensembled methods to optimize urban infrastructure by combining graph-based spatial analysis with deep learning models for sensor data. Together, these approaches create robust and scalable solutions for tackling complex, real-world problems.

8 FUTURE SCOPE

Future research should focus on developing efficient, interpretable, and scalable ensemble techniques to fully realize their potential. Innovations in dynamic graph modeling will allow these methods to adapt to real-time changes in data, enabling applications in dynamic social networks, evolving financial systems, and real-time traffic management. Advancements in transfer learning and domain adaptation will make these ensembles applicable across diverse fields, enabling cross-domain insights and improving performance on sparse datasets.With the rise of edge computing and IoT, deploying lightweight, distributed ensembles capable of operating on large-scale, decentralized graph data will become crucial for applications in smart cities, personalized healthcare, and cybersecurity. Techniques such as automated model selection, hyperparameter optimization, and explainability will make ensembled approaches more accessible and interpretable, fostering their adoption in high-stakes domains like medicine and law. Furthermore, leveraging quantum computing for ensemble-based GNNs and deep learning could redefine their computational limits, enabling breakthroughs in areas like quantum chemistry and cryptography.

REFERENCES

Adamu, U., Awan, P. I., and Younas, M. (2024). Malware classification using deep learning and ensemble framework. *Available at SSRN 4705897*.

Dietterich, T. (2000). Ensemble methods in machine learning. *International workshop on multiple classifier systems*.

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., and Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Garg, S. B. and Subrahmanyam, V. (2023). Super deep learning ensemble model for sentiment analysis. In *2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pages 341–346. IEEE.

Jain, S., Jaidka, P., and Jain, V. (2023). Deep learning ensemble method for plant disease classification. In *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)*, pages 383–387. IEEE.

Jenber Belay, A., Walle, Y. M., and Haile, M. B. (2024). Deep ensemble learning and quantum machine learning approach for alzheimer's disease detection. *Scientific Reports*, 14(1):14196.

Jin, D., Yu, Z., Huo, C., Wang, R., Wang, X., He, D., and Han, J. (2021). Universal graph convolutional networks. *Advances in Neural Information Processing Systems*, 34:10654–10664.

Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019.

Ma, S., Xiao, B., Hong, R., Addissie, B., Drikas, Z., Antonsen, T., Ott, E., and Anlage, S. (2019). Classification and prediction of wave chaotic systems with machine learning techniques. *arXiv preprint arXiv:1908.04716*.

Phung, V. H. and Rhee, E. J. (2019). A high-accuracy model average ensemble of convolutional neural net-

works for classification of cloud image patches on small datasets. *Applied Sciences*, 9(21):4500.

Salem, F. M. and Salem, F. M. (2022). Gated rnn: The gated recurrent unit (gru) rnn. *Recurrent Neural Networks: From Simple to Gated Architectures*, pages 85–100.

Sardar, M., Niazi, M. M., Nasim, F., et al. (2024). Ensemble deep learning methods for detecting skin cancer. *Bulletin of Business and Economics (BBE)*, 13(1).

Scher, S. and Messori, G. (2021). Ensemble methods for neural network-based weather forecasts. *Journal of Advances in Modeling Earth Systems*, 13(2).

Vrahatis, A. G., Lazaros, K., and Kotsiantis, S. (2024). Graph attention networks: a comprehensive review of methods and applications. *Future Internet*, 16(9):318.

Wei, W., Qiao, M., and Jadav, D. (2023). Gnn-ensemble: Towards random decision graph neural networks. In *2023 IEEE International Conference on Big Data (BigData)*, pages 956–965. IEEE.

Wilamowski, B. M. (2009). Neural network architectures and learning algorithms. *IEEE Industrial Electronics Magazine*, 3(4):56–63.

Wong, Z. H., Yue, L., and Yao, Q. (2023). Ensemble learning for graph neural networks. *arXiv preprint arXiv:2310.14166*.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.