Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 13, No. 4, 2025, 489-501.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

GNN-xLSTM: Integrating Graph Convolutional Network and Extended Long Short-Term Memory for Alzheimer's Disease Classification

Fatemeh Iranmanesh, Najme Mansouri* and Behnam Mohammad Hasani Zade

Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran.

Article Info

Article History:
Received 24 March 2025
Revised 18 May 2025
Accepted 06 July 2025

DOI:10.22044/jadm.2025.15978.2714

Keywords:
Alzheimer's Disease, Magnetic Resonance Imaging, Brain Connectivity Graphs, Graph Convolutional Network, Convolutional Neural Network, Extended Long Short-Term Memory.

**Corresponding Author:*
n.mansouri@uk.ac.ir (N. Mansouri)

Abstract

Accurate and efficient diagnosis of Alzheimer's Disease (AD) remains a significant challenge in medical research. To address the limitations of static models in capturing dynamic brain changes, this paper proposes a novel GNN-xLSTM model that integrates Graph Neural Networks (GNN) with an extended Long Short-Term Memory (xLSTM) architecture. The key innovation lies in combining GNN's ability to model spatial relationships in brain imaging data with xLSTM's enhanced sequential learning via matrix-based memory representation and exponential gate stabilization. In the proposed approach, brain images are divided into regions, each represented as a graph node connected in a grid structure, and feature vectors are extracted for each node. The proposed architecture incorporates Graph Convolutional Network (GCN) layers, xLSTM cells, residual connections, batch normalization, and dropout to jointly capture global, local, and temporal dependencies. Evaluated on the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, the GNN-xLSTM model outperforms baseline models in terms of accuracy, precision, recall, and F1-score. For example, GNN-xLSTM achieves an average improvement of 10.4% in accuracy, 11.5% in precision, 12.4% in recall, and 20.5% in F1-score over baselines. These results demonstrate the model's effectiveness in identifying critical brain regions and improving AD classification performance.

1. Introduction

AD affects the brain cells, leading to memory loss and cognitive decline. There are millions of people around the world who suffer from AD [1, 2]. Genetic mutations can cause Alzheimer's syndrome to manifest in individuals as young as 30–50 years old. Memory loss and cognitive impairment are among the symptoms of AD, followed by communication difficulties, spatial difficulties, and motor difficulties. The disease is one of the leading causes of death among the elderly [3, 4]. It is not possible to cure AD with any intervention at all. There have been a number of treatments developed, however, that provide temporary relief from symptoms or alter the progression of the disease [5]. Clinical symptoms tend to appear 15 years after the onset of the pathology, making early diagnosis difficult [6, 7]. After a clinical diagnosis of the disease, it may be too late to return individuals to their pre-morbid state. When transitioning from Normal Cognitive (NC) to AD, Mild Cognitive Impairment (MCI) is an important factor to consider (Figure 1).

The identification of patients with MCI who may progress to AD is essential for clinical practice and clinical trials. In order to prevent the development of AD and ease the progression of dementia, MCI must be detected early. It facilitates the observation of delicate brain tissue details and minor changes in Alzheimer's research and clinical diagnosis using high-contrast, high-resolution MRI images. It has been proposed to use MRI scanning for the diagnosis of AD and MCI.

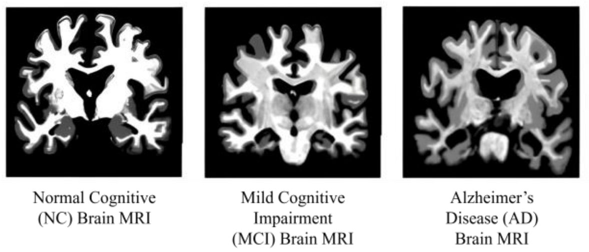
Normal Cognitive (NC) Brain MRI
Mild Cognitive Impairment (MCI) Brain MRI
Alzheimer's Disease (AD) Brain MRI

Figure 1. Sample brain Magnetic Resonance Imagings (MRIs) of NC, MCI and AD [8].

Traditional methods, such as Convolutional Neural Networks (CNNs) and Vanilla GNNs, have shown promise in image analysis but exhibit notable limitations when applied to brain MRI interpretation. CNNs, while effective in capturing local spatial features, struggle with representing non-Euclidean relationships inherent in brain connectivity [9]. On the other hand, Vanilla GNNs, which excel in handling graph-structured data, often fail to capture long-range dependencies and intricate spatial-temporal patterns crucial for comprehensive brain analysis [10].

In order to overcome these limitations, there is a pressing need for models that can simultaneously capture spatial relationships and complex dependencies within brain MRI data. Incorporating advanced architectures such as xLSTM networks can significantly enhance the capacity to model temporal sequences and long-range dependencies [11]. While GNNs efficiently capture spatial relationships, integrating xLSTM allows for modeling complex, sequential dependencies across brain regions, which is particularly important for understanding dynamic neural interactions.

Furthermore, matrix memory in xLSTM enhances feature representation by replacing scalar memory cells with matrices, increasing storage capacity and enabling richer feature learning [11]. This enhanced feature representation is critical for preserving the intricate patterns in brain MRI data, ensuring more accurate and nuanced analysis. By combining GNNs for spatial relationship modeling, xLSTM for capturing long-range dependencies, and matrix memory for enriched feature storage, a more holistic and effective approach to brain MRI analysis can be achieved.

In this paper, we propose a brain GCN based on MRI image data for AD classification and demonstrate significant improvements over traditional methods. The approach consists of three main modules.

Essentially, this paper contributes the following:

- The first step is to divide images into regions, define nodes for each region, and calculate feature vectors for each region. A grid structure is then used to create graph connectivity (edge).
- In the proposed model, multiple computational units are integrated: GCN layers, xLSTM cells, residual connections, batch normalization, and dropout. These units work together to capture global, local, and topological features of the nodes, while the xLSTM cells focus on modeling sequential dependencies. This integration enhances the model's ability to identify important regions in the graph for more accurate predictions.

The following parts of the paper are structured as follows: Section 2 discusses some basic concepts. Section 3 provides an overview of related researches. Section 4 explains in more detail how to use a GNN integrated with xLSTM cells. In section 5, the evaluation results are presented and discussed. Future research directions are described in section 6.

2. Background

2.1. CNN

In CNNs, also called ConvNets, convolutional kernels extract image features. The CNN can learn highly abstract features and identify objects efficiently. It consists of five layers: input layer, convolutional layer, pooling layer, fully connected layer, and output layer. Figure 2 illustrates the basic process. As a result of CNN's convolution algorithm, a kernel is overlaid onto an image grid and slid across it to extract local pixels. In this process, a dot product between the filter and image grid elements is applied, creating a matrix called a feature map or convolved feature. Increasing network depth transforms local pixel details into broader abstract concepts. The result of convolution and pooling layers is a vector that is then used for tasks such as classification, segmentation, and feature localization [12].

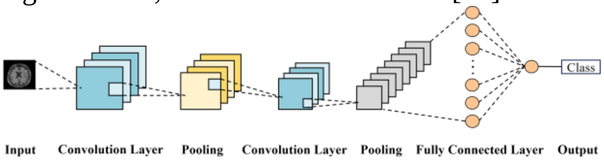

Figure 2. CNN structure [13].

2.2. GNN

GNNs apply convolutional operations to graph-based, non-Euclidean data in a manner similar to traditional CNNs [12]. GNNs are ideal for non-Euclidean data, such as brain networks, because they are efficient models for graph-structured data [14]. Unlike traditional neural networks, GNNs are better at capturing relationships and dependencies within graphs. Consequently, they generate expressive node representations by propagating messages and aggregating edges and nodes [15].

2.3. GCNs

GCNs were developed after CNNs achieved significant success, redefining convolution on graphs [16]. By generalizing convolution operations to graph-structured data, GCNs aggregate and transform information from a node's neighbors to capture local and global graph properties. This allows for the processing of non-Euclidean data, as seen in social network analysis, neuroscience, and computer vision applications involving meshes. The GCN module [2] captures contextual relationships between pixels. By representing image block features as a graph, the GCN module extracts image block features [17]. Figure 3 shows the architecture of a GCN. Stacking layers and updating node representations multiple times enables the gradual capture of higher-order structural information.

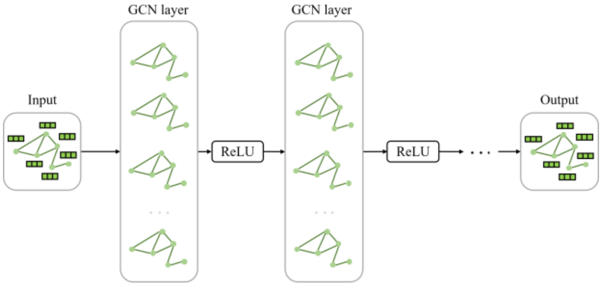

Figure 3. GCN structure diagram.

2.4. xLSTM

The xLSTM network [11] is an advanced variant of the traditional LSTM architecture designed to enhance the modeling of complex temporal and sequential data. While traditional LSTMs effectively capture long-range dependencies by mitigating the vanishing gradient problem, the xLSTM introduces additional mechanisms to improve feature representation, training stability, and computational efficiency. A key feature is its matrix-based memory representation, which allows the model to store and process multi-dimensional dependencies, leading to a better capture of complex patterns in sequential data. This richer representation is particularly useful in applications like brain MRI analysis, where both spatial and temporal relationships must be encoded effectively. Another important innovation is the use of exponential gate stabilization, which helps regulate information flow through memory cells. This mechanism improves training stability, prevents overfitting, and ensures robust performance even when handling noisy or high-dimensional data. Additionally, xLSTM provides enhanced feature extraction by combining matrix-based memory with advanced gating mechanisms, allowing the model to capture more informative features. This capability is advantageous for tasks requiring the modeling of both spatial and temporal dynamics. Furthermore, improved training efficiency is achieved through architectural modifications that lead to faster convergence and more stable gradients. This makes xLSTM more computationally efficient than traditional recurrent architectures, especially when working with large-scale datasets.

3. Related Works

3.1. Deep learning in the diagnosis of brain disorder

Human brain connectome abnormalities have been explored using deep learning algorithms [18,19]. CNNs and deep neural networks (DNNs) are the most commonly used deep learning architectures for AD classification. Although DNNs perform well in many applications, they often suffer from high computational complexity and overfitting on limited data. To accurately diagnose AD, DNNs must be used with reduced input data dimensions and decreased computation times. Benyoussef et al. [20] analyzed principal components to extract features and combined DNNs and K-nearest neighbors (KNN) to classify AD stages. The proposed model utilized KNN to distinguish between easy-to-diagnose and hard-to-diagnose subjects. The DNN contained two hidden layers with 100 nodes each. Experimentally, the proposed model classified AD into different stages. However, their approach is limited by its static feature selection and lacks spatial-temporal modeling. The proposed method addresses this by learning dynamic node features and their temporal dependencies through the integration GNNs and xLSTM.

Park et al. [21] proposed a prospective classification framework to predict AD conversion from MCI using baseline MRI scans. The method generated future MRI representations by simulating brain atrophy progression, aiming to amplify structural differences between converters and non-converters. MRI data from the ADNI dataset were used for training and evaluation, achieving an accuracy of 88.1% for AD conversion prediction. Compared to conventional classifiers trained on only baseline images, the proposed model achieved higher accuracy and better separation of MCI subgroups. The framework also identified discriminative brain regions associated with AD progression, enhancing both predictive performance and interpretability. Although effective, it depends heavily on accurate simulation models and cannot generalize well across datasets. Here, a graph-based approach is used instead, which captures topological changes in real MRI segmentations, improving generalizability.

Ho et al. [22] proposed BiPro, a bidirectional recurrent neural network model with an integrative imputation module for predicting AD progression. The model processes longitudinal data in both forward and backward directions to capture comprehensive temporal dynamics. BiPro introduces a progression module to compute biomarker progression scores and employs a multi-task learning approach to jointly handle missing value imputation, future biomarker forecasting, and clinical status prediction. Using data from the TADPOLE challenge cohort, BiPro achieved a mean area under the curve (AUC) of 78% for clinical status classification and mean absolute errors of 3.5 ml and 6.9 ml for MRI biomarker forecasting and imputation, respectively. The proposed model captures both spatial structure (via GNN) and temporal dynamics (via xLSTM) in an integrated fashion.

Table 1. A comparison of related works.

References	Year	Model	Results	Advantage	Disadvantage
Benyoussef et al. [20]	2019	KNN and a DNN combined model	An accuracy of 86.6%	Balancing simplicity and complexity.	Suboptimal performance due to KNN’s reliance on distance metrics.
Park et al. [21]	2023	Prospective Classification Framework	Accuracy of 88.1% for AD conversion prediction	Simulates future progression to enhance discriminative power; highlights disease-relevant regions	Relies on accurate simulation of brain atrophy
Ho et al. [22]	2022	BiPro (Forward-to-Backward Bi-Directional Network with Integrative Imputation)	mAUC of 78% for clinical status, MAE of 3.5 ml (forecasting), and 6.9 ml (imputation)	Captures bidirectional temporal dynamics; jointly performs imputation, forecasting, and classification	Requires extensive longitudinal data; complex architecture may limit interpretability
Liu et al. [23]	2023	MCENN	an accuracy of 88.7% for AD/MCI/NC classification	Combines 2D-slice efficiency with 3D structural awareness; robust and requires minimal preprocessing	May increase computational load due to ensemble sampling
Kam et al. [24]	2019	sdMB-CNN	The accuracy of 76.07%, sensitivity of 76.27% and specificity of 75.87%	Integrates static and dynamic functional networks for early MCI detection.	Requires pre-processed functional data.
Chang et al. [25]	2023	Using CNN for T1-weighted voxel-based quantification	An accuracy of 90.45%, precision of 86%, recall of 86%, and F1-score of 85%	Discriminates between multiple conditions (e.g., TLE, AD, and controls).	May not capture subtle differences within each condition.
Yu et al. [26]	2024	LSTM-Fuzzy Logic hybrid model	Forecast accuracy >90%, improved robustness under noise	Combines temporal modeling with uncertainty handling via fuzzy logic	Needs careful fuzzy rule design and may not generalize across all domains
Ebrahimi Mood et al. [27]	2025	Evolutionary RNN using EO	An accuracy of 87.20%, precision of 86.20%, recall of 86%, and F1-score of 86.10%	Evolutionary optimization improves convergence and temporal learning in constrained systems	Focused on resource management; not directly tailored for clinical applications
Dong [28]	2025	GNN-PSO hybrid model	An accuracy of 92.5%	PSO fine-tunes GNN parameters, boosting generalization and convergence	Application-specific design; extension to neuroimaging requires customization
Guo et al. [31]	2019	graph-based CNN architecture (PETNet)	An accuracy of 77% for 3-classes images	Uses hierarchical GCNs, enhancing classification accuracy.	Dependence on PET imaging, which is expensive and less accessible.
Song et al. [32]	2019	a multi-class GCN classifier	Average accuracy of 89%	This method demonstrated the capability of GCN over SVM for AD classification.	Simplistic graph representations may overlook complex interactions.
Yao et al. [33]	2021	a mutual multi-scale triplet GCN (MMTGCN)	The accuracy of NC vs. MCI was 86.6% and the AUC was 90.3%	Introduces a multi-scale triplet GCN, enhancing feature granularity.	Computationally intensive due to multi-scale processing.

Liu et al. [23] proposed a Monte Carlo Ensemble Neural Network (MCENN) for the diagnosis of AD using 2D-slice MRI data. The model integrates Monte Carlo sampling with ResNet50 to generate a large number of classification decisions, effectively capturing 3D anatomical information across multiple slices. MRI scans from the ADNI, Open Access Series of Imaging Studies (OASIS)-3, and a clinical dataset were utilized for training and evaluation. MCENN achieved an accuracy of 88.7% for AD/MCI/NC classification with random feature sampling set to 32. In contrast, here the entire image is displayed as a graph, and regional relationships are preserved across the entire brain.

According to Kam et al. [24], CNN frameworks based on whole-brain connectivity can help diagnose MCI. Resting-state functional MRI (RS-fMRI) can be used to provide better personalized diagnoses for neurological diseases using this framework. Dynamic connectivity provides a comprehensive picture of brain activity in addition to static networks. The proposed graph representation naturally handles non-Euclidean structures and encodes both dynamic and structural connectivity.

According to Chang et al. [25], the distinction between AD and Temporal Lobe Epilepsy (TLE) was made using MRI T1-weighted scans and a CNN model. The CNN areas differentiated by illness type were identified using feature visualization. The CNN correctly identified TLE with a high degree of accuracy. Furthermore, the model outperformed random permutation classifications in terms of age effects. However, CNNs are limited by their inability to model relational brain topology. In the proposed method, this problem is overcome by constructing brain graphs that better reflect connectivity and function.

To address complexity and generalization challenges, soft computing techniques have been increasingly incorporated into deep learning frameworks in neuroscience and adjacent domains. For instance, Yu et al. [26] proposed an LSTM-Fuzzy Logic system for fault forecasting in propulsion systems. In this approach, LSTM captures temporal trends while fuzzy logic handles uncertainty, resulting in robust predictions under noisy conditions. Such hybrid logic-driven models can enhance diagnosis robustness in clinical applications as well.

In a related development, Ebrahimi Mood et al. [27] introduced an evolutionary Recurrent Neural Network (RNN) optimized using the Equilibrium Optimization (EO) algorithm for cloud-edge resource allocation in IoT environments. The evolutionary optimizer enhances convergence and adaptability in dynamic resource-constrained systems. This suggests potential applications in adaptive and efficient training of temporal models for biomedical scenarios.

Additionally, Dong [28] proposed a GNN-Particle Swarm Optimization (PSO) hybrid framework for classifying and integrating complex educational data. In their method, PSO is employed to tune the parameters of a GNN, achieving better classification and clustering accuracy. This integration shows the promise of metaheuristic techniques in enhancing graph-based learning and can be extended to brain connectivity classification tasks as in the proposed approach.

Inspired by these recent advances, our proposed GNN-xLSTM method combines spatio-temporal modeling with neural architecture simplification to achieve better generalizability, accuracy, and computational efficiency.

3.2. GCN for brain connectivity analysis

GCN embedding [29] is a system for combining structural data with node features proposed by several researchers. The connections within the brain represent information interactions and can be seen as graph architectures in the brain. Graph learning is better described by network topology. Recent studies have shown that GCN is more efficient than other methods in learning representations [30].

Guo et al. [31] used GCN to classify AD into two and three classes using positron emission tomography images. To validate and evaluate PET images from the ADNI 2, PETNet was used. The model was trained with pretrained weights using ResNet-50. Compared to other machine learning models, PETNet was more effective and computationally efficient at analyzing medical images and detecting Alzheimer's early on. Support vector machines (SVM) and ResNet architectures performed similarly for two classification classes (MCI/NC). Compared to ResNet and SVM, the proposed GCN achieved 65% and 57% in classifying three classes (EMCI/LMCI/NC).

Song et al. [32] classified AD into four categories based on a GCN model. The proposed network consisted of eleven layers: nine convolutional layers and two fully connected (FC) layers. Each layer exploits the Rectified Linear Unit (ReLU) activation. Class probabilities are computed using Softmax as the final layer. The proposed scheme was tested using the ADNI dataset. The original dataset contained 12 images per class. As a result of the small data volume, various data augmentations were applied. The previous dataset had 12 images per class, now there are 132. GCN is capable of classifying AD better than SVM (which relies on predefined input features). The average accuracy of GCN is 89%, while SVM is about 65%.

According to Yao et al. [33], MMTGCN is a mutual multiscale triplet GCN suitable for detecting brain disorders using fMRI and diffusion tensor imaging (DTI). The first step was to generate three sets of functional and structural connectivity networks using three different brain segmentation templates. Also, Triplet GCN modules (TGCN) are proposed as a method for learning relationships between GCNs. Comparatively, NC is 86.6% accurate and its AUC is 90.3%. However, KNN is used to ignore global connectivity. Although effective, the model is complex and computationally intensive. In the proposed method, the model is simplified using xLSTM-enhanced GNN blocks that balance complexity and accuracy. Table 1 compares the methods described above.

4. Proposed Method

4.1. Overview

This section presents a method for predicting AD using a combination of a GNN and xLSTM cells. This method consists of three main components: graph construction, GNN-xLSTM processing, and classification. Firstly, brain MRI images are transformed into graph representations by extracting feature vectors from individual brain regions and constructing graph connectivity. Secondly, each brain graph is processed through multiple GNN-xLSTM blocks, which capture both spatial relationships using GCN layers and complex dependencies using xLSTM cells. Finally, the model applies global pooling and classification layers to generate predictions. Figure 4 illustrates the architecture of the GNN-xLSTM framework.

4.2. Graph Construction

To effectively process brain MRI images using the GNN-xLSTM model, each MRI scan is transformed into a graph structure. This transformation captures both the spatial relationships between different brain regions and the feature information of each region. The graph representation involves three main steps: superpixel segmentation, node feature extraction, and edge construction.

4.2.1. Superpixel Segmentation

The first step in constructing the graph is to divide the brain MRI image into smaller, meaningful regions using superpixel segmentation. This method partitions the image into coherent regions that preserve local features and anatomical structures. Each superpixel is treated as a node in the graph. The Simple Linear Iterative Clustering (SLIC) algorithm is commonly used for superpixel segmentation, which adaptively groups pixels based on spatial and intensity proximity. It minimizes a combined distance measure to cluster pixels:

$$D = d_C + m \times d_S \tag{1}$$

where D is the combined distance measure, d_c represents the color (intensity) distance between pixels i and j, d_s represents the spatial distance between pixels, m is a compactness factor controlling the trade-off between spatial and intensity proximity. Each superpixel represents a localized region of the brain, preserving spatial structure while reducing the dimensionality of the input image.

4.2.2. Node Feature Extraction

Each node in the graph corresponds to a superpixel region and is characterized by a feature vector that encapsulates both intensity-based and shape-based information. The intensity-based features of each superpixel include the mean and standard deviation of voxel intensities within the region, which provide a general understanding of the intensity distribution. Additionally, histogram-based texture features such as skewness are computed to capture fine-grained variations in intensity. For shape-based features, the superpixel's area and perimeter are calculated to describe the size and boundary complexity of the region. Compactness and eccentricity are used to capture the geometric form of the superpixel, providing insights into its shape and elongation. These node features provide a comprehensive representation of both the appearance and structural properties of each brain region.

4.2.3. Edge Construction Process

The edge construction process is designed to capture spatial relationships between adjacent superpixels in the brain MRI image. For each pixel at position (r,c) within the image dimensions, the algorithm iterates through all its neighboring pixels (r',c'). If the neighboring pixel belongs to a different superpixel (i.e., the segment labels of (r,c) and (r',c') are not equal), an edge is created between the corresponding superpixels. This ensures that only adjacent superpixels are connected, preserving the local spatial structure of the MRI image. Formally, for any two adjacent pixels (r,c) and (r',c'), an edge is added if:

$$\text{if } segments(r,c) \neq segments(r',c'), \quad \text{then } (v_i, v_j) \in E \tag{2}$$

where v_i and v_j represent the superpixels corresponding to (r,c) and (r',c') respectively, and E is the set of graph edges. This approach accurately models the spatial adjacency between superpixels, ensuring that the resulting graph reflects the anatomical layout of the brain MRI. By connecting neighboring regions, the graph captures local dependencies and facilitates effective message passing during the GNN-xLSTM model's learning process.

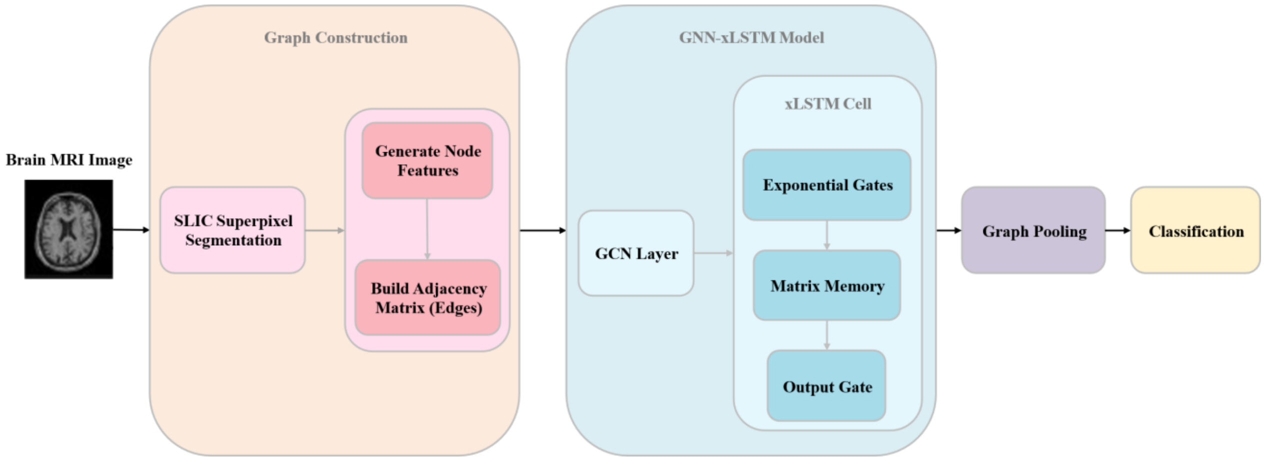

Figure 4. The architecture of GNN-xLSTM.

4.3. The Proposed GNN-xLSTM Architecture

The GNN-xLSTM model integrates GCN and an xLSTM cell to effectively capture spatial and sequential dependencies in brain MRI data. This hybrid architecture allows for comprehensive modeling of both local anatomical relationships and long-term contextual information.

4.3.1. GCN Component

The GCN in the GNN-xLSTM architecture is responsible for extracting spatial relationships between brain regions. Each brain MRI image is represented as a graph $G=(V,E)$, where V represents the set of superpixels (nodes), and E represents the edges connecting adjacent regions. The GCN processes the node features $X \in \square^{N \times F}$, where N is the number of nodes and F is the feature dimension.

Each GCN layer aggregates information from neighboring nodes using the following rule:

$$H^{(l+1)} = \sigma(\hat{A} H^{(l)} W^{(l)}) \tag{3}$$

where $H^{(l)}$ is the node feature matrix at layer l, $W^{(l)}$ is the weight matrix for layer l, σ is the activation function and $\hat{A}$ is the normalized adjacency matrix [34]. By stacking multiple GCN layers, the model captures multi-hop spatial dependencies, allowing the extraction of local anatomical relationships from the brain's structural information.

4.3.2. xLSTM

The xLSTM cell enhances the traditional LSTM by incorporating exponential gating mechanisms and a matrix-based memory cell, which improves the model's ability to capture complex spatial and temporal relationships while ensuring numerical stability during training. Unlike the traditional LSTM that uses sigmoid activation for the input and forget gates, the xLSTM applies an exponential function to these gates. Specifically, given the input x_t and the previous hidden state h_{t-1}, the input gate i_t and forget gate f_t are computed using the following equations [11]:

$$i_t = \exp(W_i^T x_t + r_i h_{t-1} + b_i) \tag{4}$$

$$f_t = \exp(W_f^T x_t + r_f h_{t-1} + b_f) \tag{5}$$

Here, W_i and W_f represent learnable weight matrices, and ε is a small constant ensuring numerical stability. This exponential gating mechanism allows the model to dynamically reweight information and increases its sensitivity to small variations in the input, enhancing fine-grained temporal modeling.

A distinctive feature of the xLSTM is the use of a matrix memory cell $C_t \in \square^{d \times d}$ instead of a scalar memory. This matrix structure allows the model to capture higher-order interactions and maintain richer contextual information across multiple dimensions. The memory cell is updated by combining the input and forget gates [11]:

$$C_t = f_t \square C_{t-1} + i_t \square \tanh(W_z^T x_t + r_z h_{t-1} + b_z) \tag{6}$$

This formulation enables the model to store spatial dependencies more effectively and track the long-term evolution of patterns within brain MRI data.

To prevent numerical instability caused by large exponential values, the xLSTM normalizes the input and forget gates by their combined sum. This stabilization mechanism ensures that the gate activations remain bounded between 0 and 1, preventing issues such as exploding gradients and improving overall training stability.

Additionally, the xLSTM integrates a query-key-value mechanism to dynamically access and update the matrix memory. At each time step, the query vector q_t, key vector k_t, and value vector v_t are generated from the current input and the previous hidden state. The interaction between the query, key, and value vectors allows the xLSTM to selectively retrieve and update information from the matrix memory.

By combining exponential gating and matrix-based memory, the xLSTM effectively captures complex spatial and temporal dependencies, making it particularly suited for modeling brain dynamics in MRI data. Algorithm 1 shows the proposed model.

Algorithm 1 The Proposed Model

Input: Brain MRI images, Graph parameters (n_segments, compactness)
Output: Classification predictions (e.g., AD or NC)

```
function ConstructGraph(image)
    segments ← SLIC (image, n_segments, compactness)
    nodes ← {ExtractFeatures(region) for each region in segments}
    edges ← {(segments [$r,c$], segments [$r',c'$]) for adjacent pixels ($r,c$) and ($r',c'$) if segments [$r,c$] ≠ segments [$r',c'$]}
    return Graph (nodes, edges)
end function

function xLSTMCell($x$ , hidden_state)
    $h$ , $c$ ← hidden_state if hidden_state else (0, 0)
    combined ← concat($x$ , $h$ )
    gates ← exp ($W_i$ · combined), exp ($W_f$ · combined)
    $i_t$ , $f_t$ ← gates / (sum(gates) + $\varepsilon$ )
    $c_{next}$ ← $f_t$ * $c$ + $i_t$ * tanh ($W_c$ · combined)
    $h_{next}$ ← sigmoid ($W_o$ · combined) * tanh( $c_{next}$ )
    return $h_{next}$ , ( $h_{next}$ , $c_{next}$ )
end function

function GNNxLSTMBlock( $x$ , edge_index, hidden_state)
    $gnn_{out}$ ← GCNConv( $x$ , edge_index)
    $h_{next}$ , new_hidden ← xLSTMCell( $gnn_{out}$ , hidden_state)
    return Dropout (ReLU(BatchNorm( $h_{next}$ + Residual( $x$ )))), new_hidden
end function

function GNN-xLSTM(brain_mri_images)
    graphs ← {ConstructGraph(image) for image in brain_mri_images}
    hidden_states ← [null] * len(graphs)
    for graph in graphs
        $x$ ← BatchNorm(graph.node_features)
        for $i$ in range(num_blocks)
            $x$ , hidden_states[ $i$ ] ← GNNxLSTMBlock( $x$ , graph.edges, hidden_states[ $i$ ])
        pooled ← GlobalMeanPool( $x$ , graph.batch)
        logits ← Linear_2(Dropout (ReLU(Linear_1(pooled))))
        predictions ← Softmax(logits)
        Update model parameters using loss (predictions, true labels)
    return trained_model
end function

function Classify (image, trained_model)
    return trained_model(ConstructGraph(image))
end function
```

5. Experiments

5.1. Data preprocessing

All data analyzed in this study were obtained from the ADNI repository (adni.loni.usc.edu). The dataset comprises a total of 14,966 T1-weighted structural MRI samples, including 7,536 AD, 922 MCI, and 7,430 NC cases. This distribution presents a class imbalance, particularly due to the relatively low number of MCI samples. To mitigate this, the proposed method employs random under sampling of the majority classes (AD and NC) to balance the dataset prior to training.

Initially, these original images underwent augmentation, involving steps like random rotations, flips, and noise addition. Upon obtaining the preprocessed MRI images, they were divided into regions, thereby delineating each image into 16 patches. Classification performance is then evaluated across the classification task (AD vs. MCI vs. NC).

5.2. Experimental setup

The proposed approach was implemented using PyTorch and PyTorch Geometric frameworks, and training was conducted using a single NVIDIA RTX 4080 GPU (13 GB RAM). The Adaptive Moment Estimation with Weight Decay (AdamW) optimizer [35] was used for network optimization, with a learning rate and weight decay set to 1e-4 and 1e-3, respectively. A dropout rate of 0.5 was applied to the hidden layers in all models. The maximum number of epochs was set to 50. The use of ReLU activation functions ensured non-linearity and improved the model's capacity to capture complex relationships within the graph structure. The negative log-likelihood loss function was used to compute the loss between predicted and true labels. Max-pooling layers were also applied to reduce spatial dimensions and computational complexity. The final layer was a fully connected output layer for classification. Various performance metrics, including accuracy, F1-score, precision, and recall, were considered to evaluate classification performance.

5.3. Comparison experiment

The performance of the proposed GNN-xLSTM model was evaluated against three baseline models: Vanilla GNN [36], CNN [37], and BiPro [22]. The models were evaluated on a brain MRI classification task, and the aforementioned performance metrics were considered to determine the effectiveness of each model. The results are summarized below.

From Table 2, it is evident that the proposed method shows superior performance.

Table 2. Classification results of different models on the AD vs. NC vs. MCI.

Metric	Vanilla GNN [36]	CNN [37]	BiPro [22]	GNN-xLSTM
Accuracy	0.5400	0.8000	0.8110	**0.8200**
F1 Score	0.5300	0.7900	0.8080	**0.8100**
Precision	0.5500	0.7800	0.7900	**0.8000**
Recall	0.5400	0.8000	0.8000	**0.8200**
Execution time (s)	**210**	445	475	821

In the AD vs. NC vs. MCI classification task, the proposed GNN-xLSTM model achieves the highest accuracy of 0.8200, outperforming BiPro (0.8110), CNN (0.8000), and significantly surpassing the Vanilla GNN baseline (0.5400). This demonstrates the effectiveness of combining graph-based spatial modeling with xLSTM's ability to capture long-range dependencies in brain MRI data. Additionally, we further demonstrated the performance of our method for the three-classification task by generating a confusion matrix, as illustrated in Figure 5. A comparative analysis across the four models highlights the superiority of the proposed approach in distinguishing between AD, NC, and MCI. The Vanilla GNN model shows moderate classification capability, with 65% accuracy for AD, but struggles with MCI, achieving only 45%. Misclassifications between NC and MCI are particularly evident, indicating a limited capacity for temporal modeling. The CNN model performs better for MCI (80%) but misclassifies NC and AD more frequently, likely due to the absence of explicit graph or temporal modeling, which is crucial for subtle transitions between classes. The BiPro model improves upon both previous baselines, reaching 80% for AD and 81% for NC, but its performance on MCI (80%) is slightly less consistent, with minor confusion between classes. This demonstrates the benefits of temporal modeling and imputation used in BiPro, but also suggests room for improvement in integrating spatial and sequential information. In contrast, the proposed GNN-xLSTM achieves the best balance across all three categories. It classifies AD with 82% accuracy, NC with 83%, and MCI with 82%, demonstrating a notable reduction in off-diagonal errors. These results affirm the effectiveness of combining GNN for spatial feature learning with xLSTM for enhanced temporal dependency modeling. The GNN-xLSTM model not only outperforms in raw accuracy but also shows a more consistent and reliable classification behavior across the three classes, reflecting its robustness and applicability for real-world diagnostic support in Alzheimer's progression analysis.

From an execution time perspective, where we measured the time to complete one epoch, the proposed GNN-xLSTM model achieves the best overall classification performance, at the expense of a longer runtime. As Table 2 shows, GNN-xLSTM attains an accuracy of 82.0% (vs. 80.0% for CNN and 54.0% for Vanilla GNN), an F1-score of 0.810 (vs. 0.790 and 0.530), and the highest recall of 0.820, thanks to its ability to jointly exploit neighborhood structure in the brain network and forward-and backward-looking patient trajectories. This richer representational backbone comes at a computational cost—GNN-xLSTM requires 821 s of execution time, less than double that of BiPro (475 s) and less than twice that of the CNN (445 s)—but the trade-off is justified by its superior predictive power in distinguishing AD, MCI, and NC.

To provide a more intuitive representation of the classification accuracy in various comparative experiments, we employed a box plot of accuracy distribution over 4 runs, depicted in Figure 6, to directly showcase the differences in their performance. As shown, the GNN-xLSTM model consistently outperforms the other approaches, achieving the highest mean accuracy of 0.8219, with a narrow distribution range, indicating both superior performance and high stability across multiple runs. The BiPro model follows with a mean accuracy of 0.8110, showing strong but slightly less consistent performance. Though effective in capturing temporal patterns, BiPro lacks the enhanced sequential modeling offered by the xLSTM component in the proposed architecture. The CNN model shows a mean accuracy of 0.8006, performing relatively well due to its deep learning structure, but limited by its inability to explicitly capture spatial graph structures or model temporal dependencies. Notably, CNN's performance exhibits greater variance and outliers, which suggests susceptibility to data fluctuations.

In contrast, the Vanilla GNN achieves the lowest mean accuracy of 0.5349, with a wider interquartile range and noticeable outliers, reflecting both poor generalization and unstable performance. This result highlights the inadequacy of static GNNs in handling longitudinal AD data without temporal modeling mechanisms.

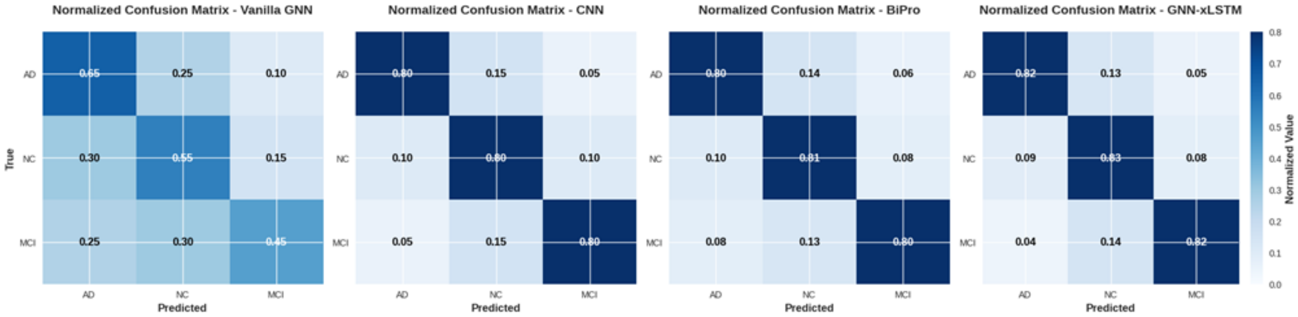

Figure 5. Confusion matrix of different models for AD vs. NC vs. MCI classifications.

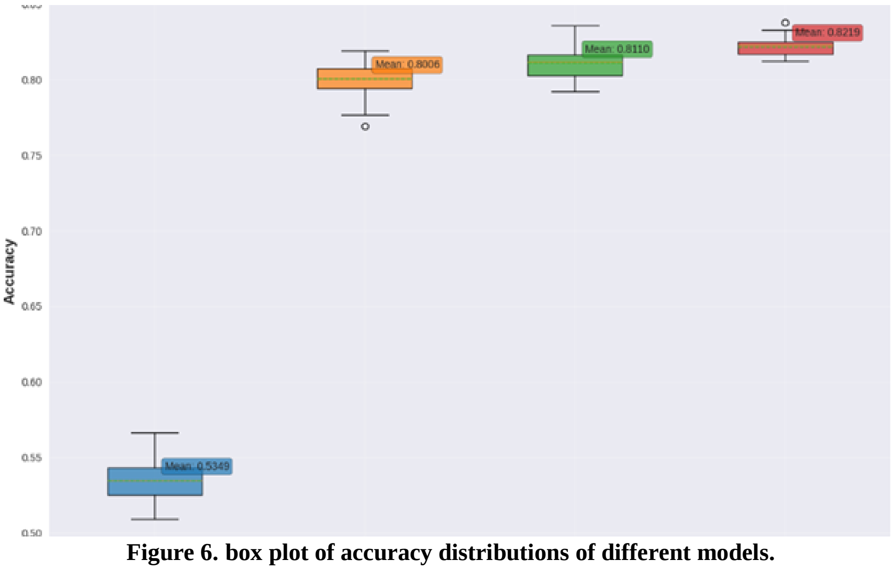

Figure 6. box plot of accuracy distributions of different models.

Figure 7 presents both the overall and per-class F1-score comparisons among Vanilla GNN, CNN, BiPro, and GNN-xLSTM models. The overall F1-score results indicate that GNN-xLSTM achieves the highest score of 0.8230, outperforming BiPro (0.8080), CNN (0.7900), and significantly surpassing Vanilla GNN (0.5300). The per-class F1-score comparison further highlights the superiority of GNN-xLSTM across all three classes. For AD classification, CNN, BiPro, and GNN-xLSTM all achieve strong F1-scores of 0.82, 0.81, and 0.82, respectively, whereas Vanilla GNN lags behind at 0.52. In the NC class, GNN-xLSTM again attains the highest F1-score of 0.83, narrowly outperforming BiPro (0.81), CNN (0.76), and considerably surpassing Vanilla GNN (0.55). Similarly, for MCI classification, GNN-xLSTM leads with an F1-score of 0.82, followed by BiPro (0.80), CNN (0.78), and Vanilla GNN (0.52). These results clearly demonstrate that the proposed GNN-xLSTM model provides more robust and consistent classification performance across all diagnostic categories. This improvement can be attributed to its hybrid architecture, which effectively combines the spatial learning capabilities of GCN layers with the temporal modeling strength of xLSTM cells. The GCN layers capture complex topological and relational features between brain regions, while the xLSTM units enable the model to learn temporal dependencies across imaging sequences or spatial transitions, improving context-awareness.

Moreover, the incorporation of matrix-based memory representation and stabilized exponential gates in xLSTM enhances training stability and information retention. Together, these architectural innovations allow GNN-xLSTM to identify subtle and distributed patterns in neuroimaging data, thereby improving its ability to discriminate between closely related cognitive conditions like MCI, NC, and AD.

To further demonstrate the convergence of the proposed GNN-xLSTM model, error rates from different models were recorded at each epoch, with corresponding curves depicted in Figure 8. The results show that GNN-xLSTM achieves a 22.5% lower final error rate (0.1380) compared to BiPro (0.1780), 35.2% lower than CNN (0.2130), and a remarkable 72.7% reduction versus Vanilla GNN (0.5063). Beyond these substantial improvements in final performance, GNN-xLSTM converges approximately 30% faster than BiPro and nearly twice as fast as CNN, reaching stability within the first 15 epochs. The model's steep initial error decline (60% drop in the first 5 epochs versus BiPro's 40%) indicates its rapid learning capability, while Vanilla GNN maintains a high residual error throughout training, never achieving better than 50% of GNN-xLSTM's convergence quality. These quantitative advantages stem from GNN-xLSTM's matrix memory and stabilized exponential gates, which enable both faster and more precise learning dynamics compared to conventional architectures.

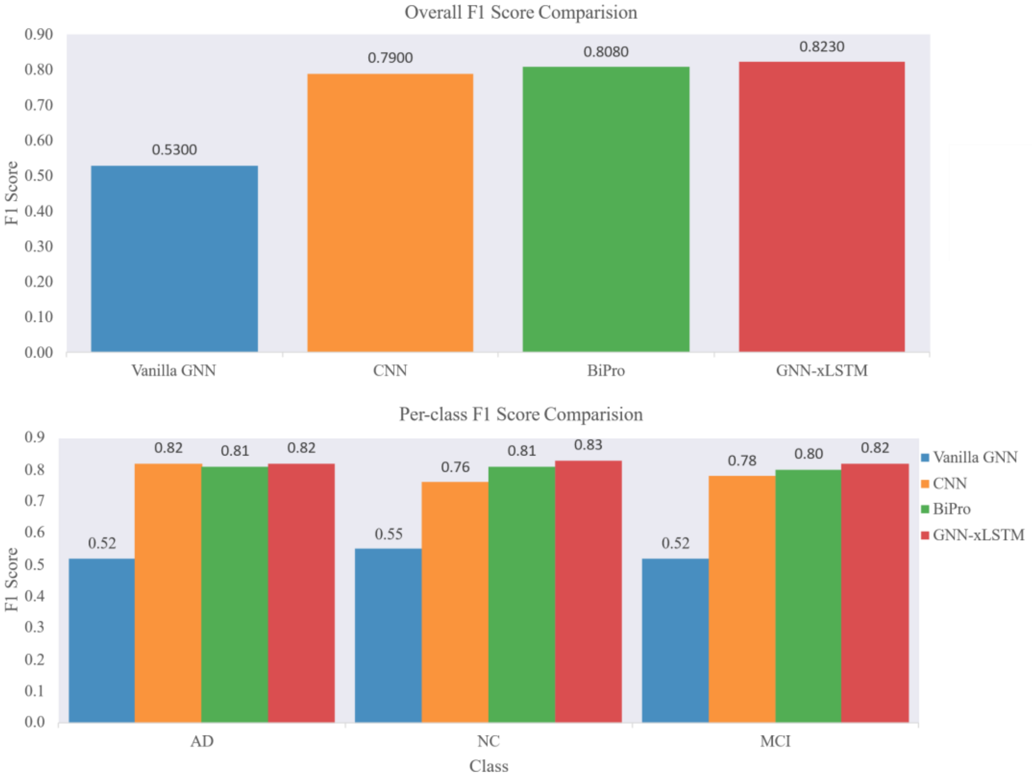

Figure 7. Overall and per-class F1-score comparison across different models.

6. Conclusion

This study proposed a hybrid GNN-xLSTM model to enhance AD classification by integrating graph-based spatial feature modeling (via GCN layers) with temporal dependency learning (via xLSTM cells). The xLSTM component introduces matrix-based memory representation and exponential gate stabilization, resulting in richer feature representations and improved training stability compared to traditional models.

Experimental results on the publicly available ADNI dataset confirm that the proposed GNN-xLSTM model outperforms baseline models across all performance metrics. Specifically, GNN-xLSTM achieves an average accuracy of 82.19%, representing a 28.70% improvement over Vanilla GNN (53.49%), a 1.89% improvement over CNN (80.06%), and a 1.09% improvement over BiPro (81.10%). Furthermore, it obtains the highest overall F1-score of 0.8230, exceeding both BiPro (0.8080) and CNN (0.7900), and significantly outperforming Vanilla GNN (0.5300).

Per-class F1-score analysis further demonstrates that GNN-xLSTM delivers consistently high performance across all diagnostic categories, including AD, NC, and MCI. These results validate the effectiveness of combining spatial and temporal modeling, particularly the ability of xLSTM cells to capture sequential transitions and subtle spatial dependencies in neuroimaging data. This makes GNN-xLSTM a promising and interpretable tool for MRI-based neurological analysis.

Future work will aim to extend the dataset, optimize the model's computational complexity, and explore more advanced memory architectures to further enhance performance and generalization capabilities across diverse clinical scenarios.

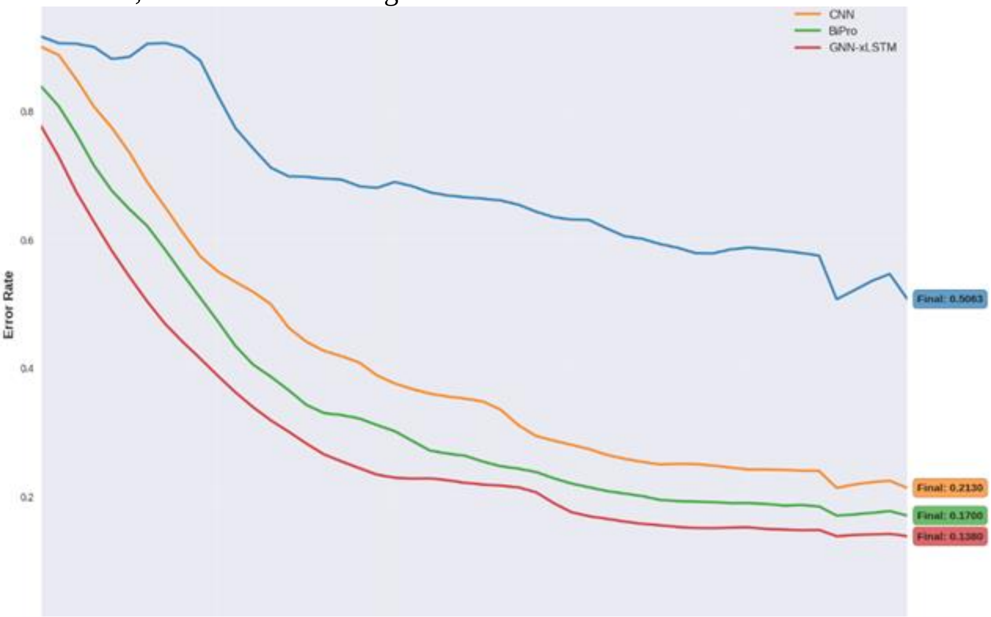

Figure 8. The convergence curve of different models over 150 iterations.

References

[1] A. K. Mishra, M. K. Tripathi, D. Kumar, and S. P. Gupta, "Neurons specialize in presynaptic autophagy: A perspective to ameliorate neurodegeneration," Molecular Neurobiology, 2024, pp. 1–15.

[2] P. Scheltens, B. De Strooper, M. Kivipelto, H. Holstege, G. Chételat, C. E. Teunissen, J. Cummings, and W. M. Van der Flier, "Alzheimer’s disease," The Lancet, vol. 397, no. 10284, pp. 1577–1590, 2021.

[3] G. B. Frisoni, N. C. Fox, C. R. Jack Jr, P. Scheltens, and P. M. Thompson, "The clinical use of structural MRI in Alzheimer disease," Nature Reviews Neurology, vol. 6, no. 2, pp. 67–77, 2010.[4] N. Habib, C. McCabe, S. Medina, M. Varshavsky, D. Kitsberg, R. Dvir-Szternfeld, G. Green, D. Dionne, L. Nguyen, J. L. Marshall, et al., "Disease-associated astrocytes in Alzheimer’s disease and aging," Nature Neuroscience, vol. 23, no. 6, pp. 701–706, 2020.

[5] O. O. Olatunde, K. S. Oyetunde, J. Han, M. T. Khasawneh, H. Yoon, and A. D. Initiative, "Multiclass classification of Alzheimer’s disease prodromal stages using sequential feature embeddings and regularized multikernel support vector machine," NeuroImage, 2024, p. 120929.

[6] R. J. Bateman, C. Xiong, T. L. Benzinger, A. M. Fagan, A. Goate, N. C. Fox, D. S. Marcus, N. J. Cairns, X. Xie, T. M. Blazey, et al., "Clinical and biomarker changes in dominantly inherited Alzheimer’s disease," New England Journal of Medicine, vol. 367, no. 9, pp. 795–804, Sept. 2012.

[7] M. W. Weiner, D. P. Veitch, P. S. Aisen, L. A. Beckett, N. J. Cairns, R. C. Green, D. Harvey, C. R. Jack, W. Jagust, E. Liu, *et al.*, "The Alzheimer’s disease neuroimaging initiative: A review of papers published since its inception," *Alzheimer’s & Dementia*, vol. 9, no. 5, pp. e111–e194, 2013.

[8] G. Mukhtar and S. Farhan, "Convolutional neural network based prediction of conversion from mild cognitive impairment to Alzheimer's disease: A technique using hippocampus extracted from MRI," *Advances in Electrical and Computer Engineering*, vol. 20, no. 2, pp. 113–122, 2020.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[11] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. K. Kopp, et al., "XLSTM: Extended long short-term memory," in *First Workshop on Long-Context Foundation Models @ ICML 2024*, 2024.

[12] A. Singh, P. van de Ven, C. Eising, and P. Denny, "Dynamic filter application in graph convolutional networks for enhanced spectral feature analysis and class discrimination in medical imaging," *IEEE Access*, 2024.

[13] M. Wei, Y. Li, M. Liang, M. Xi, and H. Tian, "Artificial intelligence approaches for early detection and diagnosis of Alzheimer's disease: A review," *Academic Journal of Science and Technology*, vol. 5, no. 3, pp. 215–221, 2023.

[14] G. Liu, Y. Yan, J. Cai, E. Q. Wu, S. Fang, A. D. Cheok, and A. G. Song, "GCD: Graph contrastive denoising module for GNNs in EEG classification," *Expert Systems with Applications*, p. 126013, 2024.

[15] J. Tang, Y. Yang, W. Wei, L. Shi, S. Su, S. Cheng, D. Yin, and C. Huang, "GraphGPT: Graph instruction tuning for large language models," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 491–500.

[16] M. Ghorbani, A. Kazi, M. S. Baghshah, H. R. Rabiee, and N. Navab, "RA-GCN: Graph convolutional network for disease prediction problems with imbalanced data," *Medical Image Analysis*, vol. 75, p. 102272, 2022.

[17] H. Song, C. Liu, S. Li, and P. Zhang, "TS-GCN: A novel tumor segmentation method integrating transformer and GCN," *Mathematical Biosciences and Engineering*, vol. 20, no. 10, pp. 18173–18190, 2023.

[18] F. Behrad and M. S. Abadeh, "An overview of deep learning methods for multimodal medical data mining," *Expert Systems with Applications*, vol. 200, p. 117006, 2022.

[19] A. De and A. S. Chowdhury, "DTI based Alzheimer's disease classification with rank modulated fusion of CNNs and random forest," *Expert Systems with Applications*, vol. 169, p. 114338, 2021.

[20] E. M. Benyoussef, A. Elbyed, and H. El Hadiri, "3D MRI classification using KNN and deep neural network for Alzheimer's disease diagnosis," in *Advanced Intelligent Systems for Sustainable Development (AI2SD'2018) Vol 4: Advanced Intelligent Systems Applied to Health*, Springer, 2019, pp. 154–158.

[21] S. Park *et al.*, "Prospective classification of Alzheimer's disease conversion from mild cognitive impairment," *Neural Networks*, vol. 164, pp. 335–344, 2023.

[22] N.-H. Ho *et al.*, "Predicting progression of Alzheimer's disease using forward-to-backward bi-directional network with integrative imputation," *Neural Networks*, vol. 150, pp. 422–439, 2022.

[23] C. Liu *et al.*, "Monte Carlo ensemble neural network for the diagnosis of Alzheimer's disease," *Neural Networks*, vol. 159, pp. 14–24, 2023.

[24] T.-E. Kam, H. Zhang, Z. Jiao, and D. Shen, "Deep learning of static and dynamic brain functional networks for early MCI detection," *IEEE Transactions on Medical Imaging*, vol. 39, no. 2, pp. 478–487, 2019.

[25] A. J. Chang, R. Roth, E. Bougioukli, T. Ruber, S. S. Keller, D. L. Drane, R. E. Gross, J. Welsh, A. Abrol, V. Calhoun, et al., "MRI-based deep learning can discriminate between temporal lobe epilepsy, Alzheimer's disease, and healthy controls," *Communications Medicine*, vol. 3, no. 1, p. 33, 2023.

[26] B. Yu, Y. Li, C. Wang, H. Zhang, and J. Chen, "LSTM-Fuzzy Logic for Fault Forecasting and Mitigation in Propulsion System," *IEEE Transactions on Aerospace and Electronic Systems*, 2024.

[27] S. Ebrahimi Mood, A. Rouhbakhsh, and A. Souri, "Evolutionary recurrent neural network based on equilibrium optimization method for cloud-edge resource management in internet of things," *Neural Computing and Applications*, vol. 37, no. 6, pp. 4957–4969, 2025.

[28] Y. Dong, "Application Research on Classification and Integration Model of Innovation and Entrepreneurship Education Resources Based on GNN-PSO Algorithm," *Systems and Soft Computing*, p. 200326, 2025.

[29] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "Accurate, efficient and scalable graph embedding," in *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2019, pp. 462–471.

[30] G. Ma, N. K. Ahmed, T. L. Willke, D. Sengupta, M. W. Cole, N. B. Turk-Browne, and P. S. Yu, "Deep graph similarity learning for brain data analysis," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2743–2751.

[31] J. Guo, W. Qiu, X. Li, X. Zhao, N. Guo, and Q. Li, "Predicting Alzheimer's disease by hierarchical graph convolution from positron emission tomography imaging," in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 5359–5363.

[32] T.-A. Song, S. R. Chowdhury, F. Yang, H. Jacobs, G. El Fakhri, Q. Li, K. Johnson, and J. Dutta, "Graph convolutional neural networks for Alzheimer's disease classification," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019, pp. 414–417.

[33] D. Yao, J. Sui, M. Wang, E. Yang, Y. Jiaerken, N. Luo, P.-T. Yap, M. Liu, and D. Shen, "A mutual multi-scale triplet graph convolutional network for classification of brain disorders using functional or structural connectivity," *IEEE Transactions on Medical Imaging*, vol. 40, no. 4, pp. 1279–1289, 2021.

[34] P. Rabiei and N. Ashrafi-Payaman, "Anomaly detection in dynamic graph using machine learning algorithms," *Journal of AI and Data Mining*, vol. 12, no. 3, pp. 359–367, 2024, doi: 10.22044/jadm.2024.14476.2551.

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[36] S. Parisot, S. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker, and D. Rueckert, "Disease prediction using graph convolutional networks: Application to autism spectrum disorder and Alzheimer's disease," Medical Image Analysis, vol. 48, pp. 117-130, 2018.

[37] S. Sarraf and G. Tofighi, "Classification of Alzheimer's disease using fMRI data and deep learning convolutional neural networks," arXiv preprint arXiv:1603.08631, 2016.

GNN-xLSTM: ادغام شبکه پیچشی گراف و حافظه بلندمدت- کوتاه‌مدت توسعه یافته برای طبقه‌بندی بیماری آلزایمر

فاطمه ایرانمنش، نجمه منصوری* و بهنام محمد حسنی زاده

گروه علوم کامپیوتر، دانشگاه شهید باهنر کرمان، کرمان، ایران.

ارسال ۲۰۲۵/۰۳/۲۴؛ بازنگری ۲۰۲۵/۰۵/۱۸؛ پذیرش ۲۰۲۵/۰۷/۰۶

چکیده:

تشخیص دقیق و کارآمد بیماری آلزایمر همچنان یک چالش مهم در تحقیقات پزشکی است. برای رفع محدودیت‌های مدل‌های ایستا در ثبت تغییرات پویای مغز، این مقاله یک مدل جدید GNN-xLSTM را پیشنهاد می‌دهد که شبکه‌های عصبی گراف را با معماری حافظه بلندمدت- کوتاه‌مدت توسعه‌یافته ادغام می‌کند. نوآوری کلیدی در ترکیب توانایی شبکه‌های عصبی گراف برای مدل‌سازی روابط مکانی در داده‌های تصویربرداری مغز با یادگیری متوالی بهبودیافته حافظه بلندمدت- کوتاه‌مدت توسعه‌یافته از طریق نمایش حافظه مبتنی بر ماتریس و تثبیت دروازه نمایی نهفته است. در رویکرد پیشنهادی، تصاویر مغز به مناطقی تقسیم می‌شوند که هر کدام به عنوان یک گره گراف متصل در یک ساختار شبکه نمایش داده می‌شوند و بردارهای ویژگی برای هر گره استخراج می‌شوند. معماری پیشنهادی شامل لایه‌های شبکه پیچشی گراف، سلول‌های حافظه بلندمدت- کوتاه‌مدت توسعه‌یافته، اتصالات باقیمانده، نرمال‌سازی دسته‌ای و dropout برای ثبت مشترک وابستگی‌های جهانی، محلی و زمانی است. مدل پیشنهادی که بر روی مجموعه داده‌های طرح تصویربرداری عصبی بیماری آلزایمر (ADNI) ارزیابی شده است، از نظر دقت، صحت، یادآوری و امتیاز F1 از مدل‌های پایه بهتر عمل می‌کند. برای مثال، GNN-xLSTM به طور متوسط بهبود ۱۰,۴٪ در دقت، ۱۱,۵٪ در صحت، ۱۲,۴٪ در یادآوری و ۲۰,۵٪ در امتیاز F1 نسبت به مقادیر پایه را به دست می‌آورد. این نتایج، اثربخشی مدل را در شناسایی مناطق حیاتی مغز و بهبود عملکرد طبقه‌بندی بیماری آلزایمر نشان می‌دهد.

کلمات کلیدی: بیماری آلزایمر، تصویربرداری تشدید مغناطیسی، گراف های اتصالات مغزی، شبکه پیچشی گراف، شبکه عصبی پیچشی، حافظه بلندمدت- کوتاه‌مدت توسعه یافته.