International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Inter-Scale Amalgam Cluster Dense Graph Convolutional Network for Hepatocellular Carcinoma Detection

Kowsalya Selvaraj[1]* **Saraswathi Subbian[2]**

[1]*Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore-641008, Tamilnadu, India*
[2]*Dean, Academic Affairs, Nehru Arts and Science College, Coimbatore- 641105, Tamilnadu, India*
* Corresponding author's Email: kowsalya.selvarajphd@gmail.com

Abstract: Accurate early detection and diagnosis of hepatocellular carcinoma (HCC) need a patient similarity network (PSN) to increase the patient's survival rate. For this reason, the PSN with a dense graph convolutional neural network (pDenseGCN) model has been developed for liver cancer diagnosis. But an over-smoothing problem occurred in the GCN when increasing the network depth, resulting in poor classification performance. Therefore, this article proposes a novel inter-scale amalgam cluster DenseGCN (IACDGCN) model to tackle the over-smoothing problem and increase the accuracy of HCC detection. The major processes in this model are the following: (i) first, the similarity network fusion (SNF) and denoising auto-encoder (DAE) are used to create PSN and extract a latent embedding representation of multi-omics data, respectively; (ii) then, the obtained PSN and extracted feature matrix are given to the IACDGCN model for classification. This IACDGCN is built by comprising a cluster DenseGCN unit and a spatiotemporal attention unit with a larger receptive field (RF) to dynamically strengthen the intermediary feature maps. Also, an inter-scale combined temporal convolution unit is used to learn discriminable temporal feature maps via the mixture of various scale convolution kernels. Moreover, those feature maps are passed to the softmax classifier to get the probability of each class such as healthy and HCC patients. Finally, the experimental results show that the IACDGCN model realizes 96.6% accuracy which is higher than the random forest, artificial neural network (ANN), deep neural network (DNN) and DenseNet models on the liver hepatocellular carcinoma (LIHC) omics dataset.

Keywords: Hepatocellular carcinoma, Patient similarity network, pDenseGCN, Over-smoothing, Cluster GCN, Spatiotemporal attention, Inter-scale combined temporal convolution.

1. Introduction

Liver cancer also known as HCC is the foremost prevalent type of cancer globally, and its incidence is growing every year [1]. According to global cancer statistics 2020, HCC kills around 830,000 people each year, making it the third largest cause of tumor-related mortality in 2020 [2]. Early cancer detection has been proven in studies to enhance survival chances. However, early-stage liver cancer signs are rarely visible, most patients have been diagnosed in the middle or final stages when they are discovered, and diagnosis actions are restricted [3]. These aspects contribute to the poor diagnosis of HCC. As a result, it is critical to develop a system, which can execute early detection and enhance the diagnosis solutions of HCC [4].

In recent decades, the analysis of omics data has become more popular for tumor diagnosis due to the progress of gene sequencing methods, which increases the number of genetic information and allows scientists to use a wide range of omics data from various traits like proteomics, transcriptomics, epigenomics, and genomics [5]. Cancer detection and diagnosis models are often divided into machine learning and deep learning models that learn relevant genetic factors as biomarkers for HCC identification [6]. With the max-relevance and min-redundancy (mRMR), incremental feature selection (IFS), and support vector machine (SVM) classifier, a hybrid tumor diagnosis model [7] has been presented to satisfy the success rate of HCC diagnosis. On the other hand, machine learning models have the complexity to process raw data directly so that they convert the raw data into suitable feature vectors. This results in a high computational cost.

To combat this problem, deep learning models have widely emerged, which can learn complex patterns from original information to draw interest in bioinformatics [8-9]. Though these models outperform machine learning models like SVM, etc., they mostly utilize single omics data that limits the representation of each cancer feature and deep learning efficiency in tumor detection. Therefore, tumor diagnosis models based on several omics data are progressively developed [10]. They often relied solely on genetic information. Principally, interpretability is essential to comprehend the biological causes of cancer and establish specific treatments due to the fairly limited number of samples in genome sequencing [11]. This issue can be resolved by PSN and created interpretable models by combining multi-omics data [12]. But the standard deep learning frameworks such as convolutional neural networks (CNNs) were difficult to handle the PSN because it was non-Euclidean data.

To alleviate these problems, Zhang et al. [13] developed the pDenseGCN based on the PSN and DenseGCN. Also, a vanishing gradient problem in GCN was resolved by densely connecting different layers that enhance the data flow in the network and process non-Euclidean data. The SNF was utilized to merge three distinct omics data and create the PSN. The DAE was utilized to capture a latent embedding interpretation of omics data. The DenseGCN was employed as a classifier based on the PSN and latent interpretation of omics data for HCC diagnosis. Conversely, its performance increases progressively with increasing the number of layers in GCN. By increasing the graph convolution (GConv) layers, the exploration for an accurate framework for the graph pattern may generate recurrent nodes in the novel embedding for a new deep layer. It is known as over-smoothing, which degrades the DenseGCN's accuracy while increasing the depth of GCN.

Hence, the IACDGCN model is proposed in this study to resolve the above-mentioned problems in liver cancer detection. First, the PSN and latent representation of omics data are fed to the IACDGCN model, which comprises a Cluster DenseGCN to enhance memory and computational efficiency. Also, a spatiotemporal attention module with a larger RF is applied to dynamically enhance the discriminatory characteristics and the intermediary feature maps. Additionally, an inter-scale combined temporal convolution module is designed to create an adaptive temporal graph via the mixture of various scale convolution kernels. Furthermore, the obtained feature maps are learned by the Softmax function to classify healthy and HCC patients. Thus, the accuracy of HCC diagnosis can be increased by preventing over-smoothing problems using the IACDGCN model.

The residual sections are planned as the following: Section 2 covers earlier research for HCC identification. Section 3 presents the IACDGCN model's methodology and section 4 portrays its efficiency. Section 5 concludes the study.

2. Literature survey

A new ensemble framework was developed [14] by stacking learning and evolutionary computation schemes for automatically detecting HCC. Various machine learning classifiers were integrated to create stacking learning with a genetic optimizer to choose the gene features for all classifiers and detect the HCC precisely. But they have less accuracy on large-scale multi-omics datasets. The DNN model [15] was presented for classifying liver cancer with microRNA data. Data normalization was done using various activation functions to improve DNN learning. But its accuracy was impacted by the vanishing gradient problems due to changes in parameters.

The ANN [16] model was trained by the modified objective function to recognize HCC patient subgroups, which are biologically homogeneous and similar in survival when eliminating noise from the data. However, its sensitivity was poor since it needs more robust features to differentiate prognostic groups for HCC. A multi-task deep learning network (MTnet) [17] was designed to predict future macrovascular invasion in HCC. But its precision was less due to the limited number of samples.

Cascaded fully CNN (CFCNN) [18] was developed for recognizing and segmenting liver cancer. Initially, the pre-processing and segmentation of CT scans were performed to segment liver lesion regions. Then, the graph cut method was used to extract features from the segmented region of lesions. Moreover, two CFCNNs were applied to classify the features into different kinds of liver cancers. But its accuracy was still not high while using multi-omics data.

An automated recognition of liver cancer using hybrid pre-trained CNN models [19] was presented. The structure of the model was characterized by the transfer learning strategy. The best model was chosen, which can learn other pre-trained CNNs, and the results of the final layer of all these networks were fused for liver cancer recognition. But its precision and sensitivity were not high. Data mining investigation of the appearance and controlling part

Received: August 2, 2023. Revised: August 29, 2023.

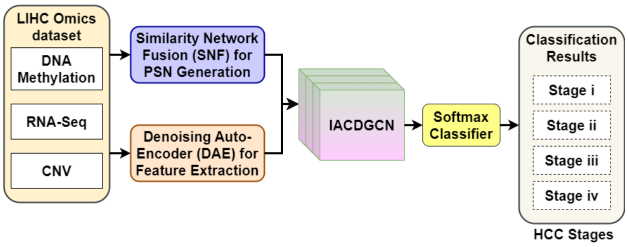

Figure. 1 Entire pipeline of the proposed study

of major genetic factors [20] in HCC was presented to expose new possible biomarkers of identification. But the accuracy was not efficient and a huge quantity of patients was needed to validate the model's efficiency. A novel method using machine learning algorithms [21] was developed to monitor major biomarkers that differentiated HCC cancer lesions from the healthy liver. But its accuracy was not high for large-scale datasets.

Differential expression genes (DEGs) [22] related to the prognosis and diagnosis of HCC were identified by integrated bioinformatics analysis. But the sensitivity was less while considering large-scale datasets. Wang et al. [23] developed a survival prognosis prediction model using random forest integrated with DEGs for estimating relevant DEGs in HCC. However, accuracy was less because the open-source dataset may have missing data and the data are all retrospective.

From the literature, it is observed that the previous studies mostly focused on statistical analysis and machine learning algorithms for HCC detection using different data modalities. But those studies infrequently consider patient similarity, which also supports physicians to diagnose liver cancer efficiently. In contrast with previous studies, the IACDGCN model is a new HCC detection system that learns the patient similarity from different HCC omics data and spatiotemporal features for accurately recognizing HCC patients. It can reduce the model complexity and training efficiency while using large-scale gene expression datasets.

3. Proposed methodology

This section briefly explains the proposed IACDGCN model for HCC detection. An entire pipeline of the proposed study is illustrated in Fig. 1. It encompasses four modules. The PSN is created by an omics dataset based on the SNF. The low-dimensional features are extracted from the omics dataset using DAE [13]. Then, the PSN and feature embedding matrix are passed to the proposed IACDGCN followed by a softmax classifier for HCC detection (i.e., four different stages of liver tumor). Thus, the HCC detection model is developed with better accuracy.

Table 1. Lists of notations

Notations	Description
G	Graph
$N = \|\mathcal{V}\|$	Number of vertices
$\mathcal{E}$	Edge
A	Resultant adjacency matrix
$N \times N$	Sparse matrix
a and b	Two vertices
X	Feature matrix
F	Feature vector dimension
L	Number of Conv layers
$Z^{(q+1)}$	Embedding of the node's adjacent in G from earlier layer
$X^{(q)}$	Embedding at q^{th} layer for N nodes
A'	Normalized adjacency matrix
$W^{(q)}$	Feature conversion matrix
$\sigma(\cdot)$	Activation function
B, k	Number of batches and their dimension
$A_{B,B}$	Subgraph
c	Number of clusters
$\mathcal{V}_t$	Number of nodes in t^{th} partition
A_{tt}	Adjacency matrix having the relations in G_t
$\bar{A}$	Adjacency matrix for graph $\bar{G}$
A_{st}	Relations between $\mathcal{V}_s$ and $\mathcal{V}_t$
Δ	Matrix having each off-diagonal block of A
Y	Learning tags
$\bar{A}'$	Regularized form of $\bar{A}$
$Z^{(Q)}$	Absolute embedding matrix
$\bar{A}'_{tt}$	Corresponding diagonal block of $\bar{A}'$
$\mathcal{L}_{\bar{A}'}$	Loss factor
$z_a^{(L)}$	a^{th} row of $Z^{(Q)}$ with ground truth tag to be y_a
ϵ_{max}	Maximum iteration
$\bar{X}$	Node representation
g	Gradient approximators
M_t	Temporal attention map
M_s	Spatial attention map
M_{st}	Spatiotemporal attention map
$X^{(q+1)}$	Dense GConv function
$\mathcal{F}$	Residual mapping function
$\mathcal{G}$	Inter-scale GConv function
$\|$	Concatenation

The notations used in this study are presented in Table 1.

3.1 Inter-scale amalgam cluster DenseGCN model

Fig. 2 depicts an overview of IACDGCN model for HCC detection and diagnosis, which includes the cluster DenseGCN, spatiotemporal attention mask,

Received: August 2, 2023. Revised: August 29, 2023.

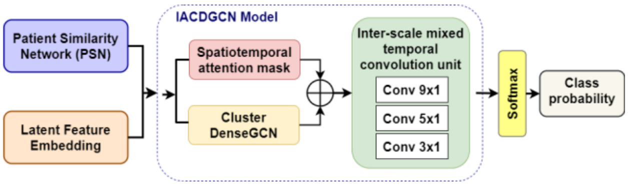
Patient Similarity Network (PSN)
Latent Feature Embedding
IACDGCN Model
Spatiotemporal attention mask
Cluster DenseGCN
Inter-scale mixed temporal convolution unit
Conv 9x1
Conv 5x1
Conv 3x1
Softmax
Class probability

Figure. 2 Overview of proposed IACDGCN model for HCC detection and diagnosis

and inter-scale combined temporal convolution units. This model can learn spatiotemporal features at multiple scales and enhance the low-dimensional intermediate features.

3.1.1. Cluster DenseGCN

Assume a graph $G = (\mathcal{V}, \mathcal{E}, A)$ containing $N = |\mathcal{V}|$ vertices and $|\mathcal{E}|$ edges such that an edge between two vertices a and b define their relationship. The resultant adjacency matrix A is an $N \times N$ sparse matrix with (a, b) data equivalent to one when an edge between a and b is exist and zero or else. As well, all nodes are related to an F-dimensional feature vector and $X \in \mathbb{R}^{N \times F}$ indicates the feature matrix for N nodes. An Q-layer DenseGCN has L GConv layers and each create embedding for all nodes via combining the embedding of the node's adjacent in G from earlier layer:

$$Z^{(q+1)} = A' X^{(q)} W^{(q)}, X^{(q+1)} = \sigma\left(Z^{(q+1)}\right) \quad (1)$$

In Eq. (1), $X^{(q)} \in \mathbb{R}^{N \times F_q}$ refers to the embedding at q^{th} layer for N nodes and $X^{(0)} = X$; A' is the normalized adjacency matrix and $W^{(q)} \in \mathbb{R}^{F_q \times F_{q+1}}$ denotes the feature conversion matrix that can be learned for the downstream processes. Notice that for easiness, consider the feature sizes are equal for each layer $\left(F_1 = \cdots = F_Q = F\right)$. The activation function $\sigma(\cdot)$ is assigned to be an element-wise rectified linear unit (ReLU).

In the DenseGCN, a grid search algorithm is applied for GCN learning. However, it influences from high computation and memory expense when exponentially increases the number of GCN layers. To avoid this issue, a new DenseGCN training algorithm called cluster DenseGCN is developed by exploiting graph clustering structure. This can enhance the convergence speed and reduce memory space of GCN using several clusters as one batch based on the mini-batch Stochastic gradient descent (SGD). It utilizes $B \subseteq [N]$ with dimension $k = |B|$ to represent a batch of node indices, and all SGD steps can determine the gradient prediction as follows to execute an update in Eq. (2).

$$\frac{1}{|B|} \sum_{a \in B} \nabla loss\left(y_a, z_a^{(Q)}\right) \quad (2)$$

The cluster DenseGCN performs as follows: at every step, it samples a block of nodes that relate to a dense subgraph recognized through a graph clustering technique, and limits the neighborhood exploration in this subgraph. Consider the case that in all batches, then the embedding for a group of B from layer 1 to Q can be calculated. Because a similar subgraph $A_{B,B}$ (relations within B) is utilized for all layers of calculation, it can be observed that the embedding usage means the amount of edges in this B, i.e., $\left\|A_{B,B}\right\|_0$.

So, to improve embedding usage, a batch B is designed to exploit the within-batch edges, by which the efficacy of SGD for training modifies with graph clustering algorithms. In cluster DenseGCN, for a graph G, its nodes are split into c clusters: $\mathcal{V} = [\mathcal{V}_1, \dots, \mathcal{V}_c]$, wherein $\mathcal{V}_t$ comprises the nodes in t^{th} partition. So, c subgroups are obtained as:

$$G = [G_1, \dots, G_c] = [\{\mathcal{V}_1, \mathcal{E}_1\}, \dots, \{\mathcal{V}_c, \mathcal{E}_c\}] \quad (3)$$

In Eq. (3), all $\mathcal{E}_t$ solely comprises the relations between nodes in $\mathcal{V}_t$. Once nodes are updated, A is split into c^2 submatrices as:

$$A = \bar{A} + \Delta = \begin{bmatrix} A_{11} & \cdots & A_{1c} \\ \vdots & \ddots & \vdots \\ A_{c1} & \cdots & A_{cc} \end{bmatrix} \quad (4)$$

Where

$$\bar{A} = \begin{bmatrix} A_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_{cc} \end{bmatrix}, \Delta = \begin{bmatrix} 0 & \cdots & A_{1c} \\ \vdots & \ddots & \vdots \\ A_{c1} & \cdots & 0 \end{bmatrix} \quad (5)$$

In Eqns. (4)-(5), all diagonal blocks A_{tt} is a $|\mathcal{V}_t| \times |\mathcal{V}_t|$ adjacency matrix having the relations in G_t, $\bar{A}$ denotes the adjacency matrix for graph $\bar{G}$, A_{st} comprises the relations between $\mathcal{V}_s$ and $\mathcal{V}_t$, and Δ denotes the matrix having each off-diagonal block of A. Also, X and learning tags Y are divided based on the partition $[\mathcal{V}_1, \dots, \mathcal{V}_c]$ into $[X_1, \dots, X_c]$ and $[Y_1, \dots, Y_c]$, where X_t and Y_t contain the features and tags for nodes in V_t, correspondingly. The advantage of this block-diagonal estimation $\bar{G}$ is that the objective function of DenseGCN is decomposed into multiple B (i.e., clusters). If $\bar{A}'$ is the regularized form of $\bar{A}$, then an absolute embedding matrix is defined in Eq. (6):

$$Z^{(Q)} = \bar{A}'\sigma\left(\bar{A}'\sigma\left(\cdots\sigma\left(\bar{A}'\mathrm{X}W^{(0)}\right)W^{(1)}\right)\cdots\right)W^{(Q-1)} = \begin{bmatrix} \bar{A}'_{11}\sigma\left(\bar{A}'_{11}\sigma\left(\cdots\sigma\left(\bar{A}'_{11}\mathrm{X}_1W^{(0)}\right)W^{(1)}\right)\cdots\right)W^{(Q-1)} \\ \vdots \\ \bar{A}'_{cc}\sigma\left(\bar{A}'_{cc}\sigma\left(\cdots\sigma\left(\bar{A}'_{cc}\mathrm{X}_cW^{(0)}\right)W^{(1)}\right)\cdots\right)W^{(Q-1)} \end{bmatrix} \tag{6}$$

This is because the block-diagonal form of $\bar{A}'$ (observe that $\bar{A}'_{tt}$ denotes the corresponding diagonal block of $\bar{A}'$). Additionally, the loss factor is split as:

$$\mathcal{L}_{\bar{A}'} = \sum_t \frac{|\mathcal{V}_t|}{N}\mathcal{L}_{\bar{A}'_{tt}},$$

$$\text{Where } \mathcal{L}_{\bar{A}'_{tt}} = \frac{1}{|\mathcal{V}_t|}\sum_{a\in\mathcal{V}_t} loss\left(y_a, z_a^{(Q)}\right) \tag{7}$$

In Eq. (7), $z_a^{(L)}$ is a^{th} row of $Z^{(Q)}$ with ground truth tag to be y_a defining the last layer projection of node a. After that, the Cluster DenseGCN is trained according to the decomposition form in Eqns. (6) and (7). At all steps, a cluster $\mathcal{V}_t$ is sampled and the SGD is performed to modify according to the gradient of $\mathcal{L}_{\bar{A}'_{tt}}$, and this only needs the sub-graph A_{tt}, X_t, Y_t on the present B and the models $\left\{W^{(q)}\right\}_{q=1}^{Q}$.

The execution merely needs forward and backward propagation of matrix products (single block of (6)), which is simpler to execute compared to the neighborhood exploration process in the grid search-based learning schemes. The graph clustering technique, namely Metis [24] intends to create the partitions over $\mathcal{V}$ in G such that within-cluster relations are greater than between-cluster relations to achieve the grouping pattern of G. Conversely, the entropy values of maximum clusters are reduced, defining that the tag distributions of clusters are biased towards a few particular tags. This results in high difference across multiple batches and may influence the SGD convergence. As a result, a stochastic multi-clustering technique is applied to integrate between-cluster relations and minimize difference across batches.

Initially, the graph is split into p clusters $\mathcal{V}_1, \dots, \mathcal{V}_p$ with a quite large p. While creating B for an SGD update, rather than using simply single cluster, u clusters are arbitrarily chosen, indicated by $t_1, \dots, t_u$, and comprise their nodes $\left\{\mathcal{V}_{t_1} \cup \dots \cup \mathcal{V}_{t_u}\right\}$ into B. Moreover, the relations between the selected clusters, i.e., $\{A_{ab} | a, b \in t_1, \dots, t_u\}$ are included. In this manner, such between-cluster relations are reintegrated and the mixtures of clusters create the difference across B lesser. Therefore, the convergence of SGD for GCN training is enhanced by considering many clusters as one batch. The cluster DenseGCN's architecture is portrayed in Fig. 3. The pseudocode for the cluster DenseGCN is provided in Algorithm 1.

Algorithm 1 Cluster DenseGCN

```
Input: Graph $G$, feature matrix $X$, tag $Y$, and maximum iteration ($\epsilon_{max}$)
Result: Node representation $\bar{X}$
1. Begin
2. Split the graph nodes into $c$ clusters $\mathcal{V}_1, \dots, \mathcal{V}_c$ by Metis;
3. for($\epsilon \le \epsilon_{max}$)
4.         Select $u$ clusters $t_1, \dots, t_u$ from $\mathcal{V}$ arbitrarily, with no substitution;
5.         Create a subgraph $\bar{G}$ with nodes $\bar{\mathcal{V}} = \left[\mathcal{V}_{t_1}, \dots, \mathcal{V}_{t_u}\right]$ and relations $A_{\bar{\mathcal{V}},\bar{\mathcal{V}}}$;
6.         Calculate $g \leftarrow \nabla\mathcal{L}_{A_{\bar{\mathcal{V}},\bar{\mathcal{V}}}}$ (loss on the subgraph $A_{\bar{\mathcal{V}},\bar{\mathcal{V}}}$);
7.         Perform Adam update using gradient approximators $g$;
8. end for
9. Return $\left\{W_q\right\}_{q=1}^{Q}$
```

3.1.2. Spatiotemporal attention mask

Typically, the DenseGCN utilizes spatial GConv functions to capture features in the first-order neighborhood of convolution nodes and combine multi-layer GConv functions to increase the node's RFs for concatenating global features regarding HCC. Though the dense GConv function decreases the error of initial characteristics in the procedure of GConv, this dense link provides unwanted features.

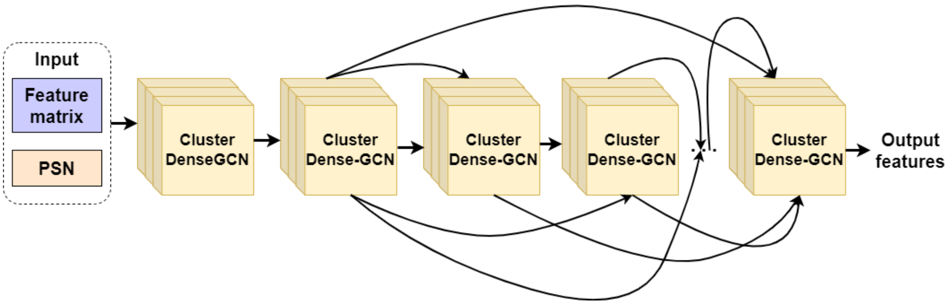

Figure. 3 Architecture of cluster DenseGCN

Since not each omics data is considered to create an equal contribution to a detection process, the most diverse features to discriminate among complex biological interactions occurs in a limited temporal omics data. Thus, a spatiotemporal attention mask is incorporated in the residual part of dense GConv link to enrich the intermediate features. After that, an adaptive weight merging scheme is utilized to implement an inter-scale merging of the spatial features of a GConv concatenation, which decreases the redundant features and allows the dense GConv function to dynamically capture combined spatial traits for effective HCC detection.

For an intermediate feature map $X \in \mathbb{R}^{N \times F}$ as input, such informative features are reutilized in the residual GConv function. So, the spatiotemporal attention mask is integrated to the residual unit to mutually concentrate on meaningful spatiotemporal features.

An 1D mean pooling is used to combine the spatial features along a spatial size and passed to the neural network (NN) to capture the temporal correlation, resulting in a temporal attention map $M_t \in \mathbb{R}^{1 \times F}$. As well, the spatial attention map $M_s \in \mathbb{R}^{1 \times F}$ is created via 1D mean pooling function and activation function. Afterward, the product of spatial and temporal attention maps is computed to obtain the spatiotemporal attention maps $M_{st} \in \mathbb{R}^{1 \times F}$ as follows:

$$M_t(X) = \sigma\left(NN\big(MeanPool(F)\big)\right) \tag{8}$$

$$M_{st}(X) = M_t^N \otimes M_s \tag{9}$$

In Eq. (8), $\sigma(\cdot)$ is the sigmoid activation. The attention mask $M_{st} \in \mathbb{R}^{1 \times F}$ in Eq. (9) is multiplied by the given feature map in a residual way for dynamic feature enhancement. Such attention units enlarge the node's RF via pooling processes, allowing the system to capture the latent relationships among omics data.

Accordingly, spatiotemporal features of multiple scales are merged by including an attention mask on the residual unit. The dense GConv function is defined in Eq. (10),

$$X^{(q+1)} = \sigma\left(\sum_{i \in u} A' X^{(q)} W^{(q)} \left(M^{(i)} \otimes \widetilde{\mathcal{A}^{(i)}} + M_{st} \otimes X^q\right)\right) \tag{10}$$

In the cluster DenseGCN, such dense relations support to reutilize intermediate features among various layers to efficiently discover the internal dependencies among multi-omics data in G. After that, Eq. (10) is converted to

$$X^{(q+1)} = \mathcal{F}\big(\mathcal{G}\big(A', X^{(q)}, W^{(q)}\big), X^q\big) = \mathcal{F}\big(\mathcal{G}\big(A', X^{(q)}, W^{(q)}\big), \ldots, \mathcal{G}\big(A', X^{(0)}, W^{(0)}\big), X^0\big) \tag{11}$$

In Eq. (11), $\mathcal{F}$ denotes the residual mapping function, and $\mathcal{G}$ is the inter-scale GConv function.

3.1.3. Inter-scale combined temporal convolutional unit

For a temporal graph, the temporal convolution unit extracts the temporal characteristics of the multi-omics data via 1D convolution. The temporal convolution kernel dimension is fixed, and the temporal characteristics are more common because of the large local temporal neighborhood. It is complex to explore for each mixture of time convolution kernels of multiple dimensions.

The omics dataset might need many temporal RFs that controls the generalizability of the temporal convolution unit. To combat these challenges, an inter-scale combined temporal convolution scheme is developed. Rather than a 9×1 large convolution kernel utilized by the earlier temporal convolution unit, three large, medium, and small convolution kernels of 9×1, 5×1, and 3×1 are applied to collect temporal characteristics at multiple scales.

Those are merged in the kernel size, and dynamically choose the temporal characteristics at multiple scales based on the layer weight; thus, the system can adaptively merge the ideal temporal RF value. The inter-scale combined temporal convolution function is defined by

$$X^{(q+1)} = A' W^{(q)} \left(X_t^{(q)}(k_t = 9) \,\Big\|\, X_t^{(q)}(k_t = 5) \,\Big\|\, X_t^{(q)}(k_t = 3)\right) \tag{12}$$

In Eq. (12), ‖ denotes the concatenation, and $W^{(q)}$ can alter the role of temporal characteristics at multiple scales. Thus, this IACDGCN will dynamically capture distinct temporal characteristics, and adaptively fine-tune the temporal convolution RF to particular extent. After learning spatiotemporal features, the obtained features are fed to the softmax classifier to find the final class probability and detect HCC patients precisely.

4. Experimental result

This section presents the IACDGCN model's effectiveness compared to the existing models (such as pDenseGCN [13], DNN [15], ANN [16], MTnet [17], and random forest [23]) by executing them in

Table 2. Parameter settings for existing and proposed IACDGCN model

Model	Parameters	Range
Random forest [23]	Number of trees in the forest	40
	Maximum tree depth	6
ANN [16]	Number of hidden layer	1
	Number of hidden units	62
	Loss function	Mean squared error
	Activation function	ReLU
	Optimizer	Adam
	Training rate	0.002
DNN [15]	Number of hidden layers	3
	Number of neurons at k^{th} hidden layer	32
	Number of neurons at output layer	4
	Learning rate	0.001
	Activation function	ReLU
	Optimizer	Adam
	Loss function	Mean squared error
MTnet [17]	Training rate	0.001
	Batch size	64
	Loss function	Cross-entropy
pDenseGCN [13]	Learning rate	0.01
	Batch size	64
	Loss function	Mean squared error
Proposed IACDGCN	Training rate	0.01
	Dropout rate	15%
	Weight decay	0
	Number of epochs	600
	Number of hidden units	256
	Number of partitions	100
	Number of clusters per batch	2
	SGD with Nesterov momentum	0.9
	Loss	Cross-entropy

MATLAB 2019b.

To measure the performance of the proposed IACDGCN model, the considered existing models are also implemented and tested on the TCGA-LIHC omics dataset. Table 2 lists parameter settings for the IACDGCN and the existing models for HCC detection.

In this experiment, the LIHC omics datasets are collected from TCGA open source [25]. The TCGA-assembler is utilized to acquire DNA methylation, RNA-Seq, and CNV data of LIHC in different stages: (a) stage i, (b) stage ii, (c) stage iii, and (d) stage iv. Table 3 lists the statistics of these datasets. The collected datasets are pre-processed to remove missing values using the data imputation. The residual samples are normalized and a total of 364 samples are considered from the pre-processed dataset, which is divided into learning and test sets. The learning set comprises 219 samples, of which 92 are stage i, 56 are stage ii, 64 are stage iii, and 7 are stage iv samples. The test set covers 145 samples, of which 63 are stage i, 33 are stage ii, 47 are stage iii, and 2 are stage iv samples.

Table 3. Statistics of different omics datasets

Omics category	No. of samples	No. of features
DNA methylation	429	20421
RNA-Seq	424	20530
CNV	760	24924

Table 4. Confusion matrix results of IACDGCN model for testing

Classified/Actual	Stage i	Stage ii	Stage iii	Stage iv
Stage i	61	0	0	0
Stage ii	0	32	2	0
Stage iii	1	1	45	0
Stage iv	1	0	0	2

Table 4 presents the confusion matrix results of IACDGCN model on the TCGA-LIHC omics dataset for classifying four different stages of HCC.

With the confusion matrix results in Table 3, the below metrics are calculated to measure the performance of HCC diagnosis models.

- Accuracy: It is the percentage of accurate classifications.

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{False\ Positive\ (FP) + False\ Negative\ (FN) + TP + TN} \tag{13}$$

For example of stage i class, in Eq. (13), TP is the quantity of stage i instances categorized as themselves, TN is the quantity of other stage instances accurately categorized as themselves, FP is the quantity of stage i instances inaccurately categorized as other stages, and FN is the quantity of other stage instances inaccurately categorized as stage i.

- Precision: It is calculated as:

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

Table 5. Comparison analysis of proposed and existing HCC diagnosis models on TCGA-LIHC omics dataset

Models	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Random forest	79.64	74.39	76.97	79.22
ANN	83.40	77.15	80.15	82.97
DNN	86.81	80.52	83.55	86.41
MTnet	89.25	82.77	85.89	89.11
pDenseGCN	92.89	85.53	89.06	91.03
IACDGCN	**97.38**	**89.13**	**93.08**	**96.60**

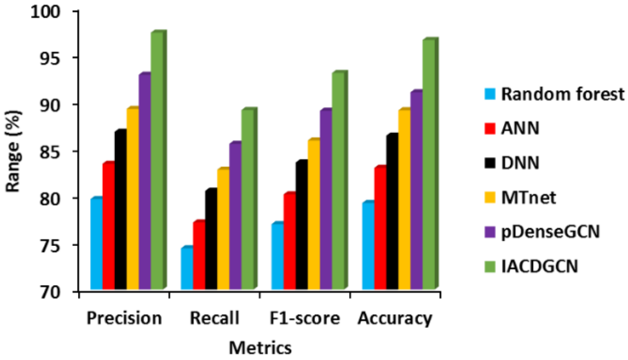

Figure 4. Performance comparison of proposed and existing HCC diagnosis models on TCGA-LIHC omics dataset

- Recall: It is determined by

$$Recall = \frac{TP}{TP+FN} \tag{15}$$

- F1-score: It is calculated as:

$$F1 - score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{16}$$

Table 5 demonstrates the results of proposed and existing models tested on the TCGA-LIHC omics dataset for HCC diagnosis.

As illustrated in Fig. 4, it is noticed that IACDGCN model on the TCGA-LIHC omics dataset achieved an improved efficiency in contrast with the other models in terms of each metric. The IACDGCN model attains 96.6% accuracy, which is increased by 21.94%, 16.43%, 11.79%, 8.41%, and 6.12% compared to the random forest, ANN, DNN, MTnet, and pDenseGCN models, respectively. In terms of precision, the IACDGCN is 22.28%, 16.76%, 12.18%, 9.11%, and 4.83% superior to the random forest, ANN, DNN, MTnet, and pDenseGCN models, respectively. The IACDGCN reaches recall, up to 19.81%, 15.53%, 10.69%, 7.68%, and 4.21% better than the random forest, ANN, DNN, MTnet, and pDenseGCN models, respectively. Also, the f1-score of IACDGCN is improved by 20.93%, 16.13%, 11.41%, 8.37%, and 4.51% compared to the random forest, ANN, DNN, MTnet, and pDenseGCN models, respectively. As a consequence, it demonstrates that the proposed IACDGCN model is highly beneficial for HCC diagnosis by adopting PSN and cluster DenseGCN with attention strategy.

5. Conclusion

In this paper, the IACDGCN model was developed for HCC detection and diagnosis. Initially, the PSN and latent feature embedding representation of multi-omics data were created. Those were given to the IACDGCN model to enhance intermediary traits and learn spatiotemporal features from multi-omics data in the graph. Then, the learned features were classified by the softmax classifier for detecting different HCC stages. At last, extensive experiments proved that the IACDGCN model on the TCGA-LIHC omics dataset has an accuracy of 96.6%, whereas the existing models such as random forest, ANN, DNN, MTnet and pDenseNetGCN on the TCGA-LIHC omics dataset have an accuracy of 79.22%, 82.97%, 86.41%, 89.11% and 91.03%, respectively for HCC diagnosis. Similarly, the IACDGCN model achieves 97.38% precision, 89.13% recall and 93.08% f-measure in contrast with the considered existing models. Thus, it is concluded that the proposed IACDGCN model outperformed other classification models for HCC diagnosis.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, methodology, software, validation, Kowsalya; formal analysis, investigation, Saraswathi; resources, data curation, writing—original draft preparation, Kowsalya; writing—review and editing, Kowsalya; visualization; supervision, Saraswathi;

References

[1] M. Falette Puisieux, A. Pellat, A. Assaf, C. Ginestet, C. Brezault, M. Dhooge, and R. Coriat, "Therapeutic Management of Advanced Hepatocellular Carcinoma: An Updated Review", *Cancers*, Vol. 14, No. 10, pp. 1-14, 2022.

[2] Z. Guizhen, J. Guanchang, L. Liwen, W. Huifen, R. Zhigang, S. Ranran, and Y. Zujiang, "The Tumor Microenvironment of Hepatocellular

Carcinoma and its Targeting Strategy by CAR-T Cell Immunotherapy", *Complexity of Tumor Microenvironment: a Major Culprit in Cancer Development*, Vol. 13, pp. 1-16, 2023.

[3] P. Soldath, A. Lund, and J. Vissing, "Late-Onset MADD: A Rare Cause of Cirrhosis and Acute Liver Failure?", *Acta Myologica*, Vol. 39, No. 1, pp. 19-23, 2020.

[4] C. W. Spearman, G. Dusheiko, E. Jonas, A. Abdo, M. Afihene, L. Cunha, and M. W. Sonderup, "Hepatocellular Carcinoma: Measures to Improve the Outlook in Sub-Saharan Africa", *The Lancet Gastroenterology & Hepatology*, Vol. 7, No. 11, pp. 1036-1048, 2022.

[5] G. L. D. Adamo, J. T. Widdop, and E. M. Giles "The Future is Now? Clinical and Translational Aspects of "Omics" Technologies", *Immunology and Cell Biology*, Vol. 99, No. 2, pp. 168-176, 2021.

[6] A. Dhillon, A. Singh, and V. K. Bhalla, "A Systematic Review on Biomarker Identification for Cancer Diagnosis and Prognosis in Multi-Omics: From Computational Needs to Machine Learning and Deep Learning", *Archives of Computational Methods in Engineering*, Vol. 30, No. 2, pp. 917-949, 2023.

[7] Z. M. Zhang, J. X. Tan, F. Wang, F. Y. Dao, Z. Y. Zhang, and H. Lin, "Early Diagnosis of Hepatocellular Carcinoma Using Machine Learning Method", *Frontiers in Bioengineering and Biotechnology*, Vol. 8, pp. 1-9, 2020.

[8] B. Tang, Z. Pan, K. Yin, and A. Khateeb, "Recent Advances of Deep Learning in Bioinformatics and Computational Biology", *Frontiers in Genetics*, Vol. 10, pp. 1-10, 2019.

[9] X. Yu, S. Zhou, H. Zou, Q. Wang, C. Liu, M. Zang, and T. Liu, "Survey of Deep Learning Techniques for Disease Prediction Based on Omics Data", *Human Gene*, Vol. 35, pp. 1-13, 2022.

[10] K. A. Tran, O. Kondrashova, A. Bradley, E. D. Williams, J. V. Pearson, and N. Waddell, "Deep Learning in Cancer Diagnosis, Prognosis and Treatment Selection", *Genome Medicine*, Vol. 13, No. 1, pp. 1-17, 2021.

[11] A. J. B. Luna, J. P. García, A. Iftene, F. Guadagni, P. Ferroni, N. Scarpato, and H. P. Sánchez, "Towards the Interpretability of Machine Learning Predictions for Medical Applications Targeting Personalised Therapies: A Cancer Case Survey", *International Journal of Molecular Sciences*, Vol. 22, No. 9, pp. 1-31, 2021.

[12] G. Nicora, F. Vitali, A. Dagliati, N. Geifman and R. Bellazzi, "Integrated Multi-Omics Analyses in Oncology: A Review of Machine Learning Methods and Tools", *Frontiers in Oncology*, Vol. 10, pp. 1-11, 2020.

[13] G. Zhang, Z. Peng, C. Yan, J. Wang, J. Luo, and H. Luo, "A Novel Liver Cancer Diagnosis Method Based on Patient Similarity Network and DenseGCN", *Scientific Reports*, Vol. 12, No. 1, pp. 1-10, 2022.

[14] W. Książek, M. Hammad, P. Pławiak, U. R. Acharya, and R. Tadeusiewicz, "Development of Novel Ensemble Model Using Stacking Learning and Evolutionary Computation Techniques for Automated Hepatocellular Carcinoma Detection", *Biocybernetics and Biomedical Engineering*, Vol. 40, No. 4, pp. 1512-1524, 2020.

[15] O. H. Purba, E. A. Sarwoko, and A. Wibowo, "Classification of Liver Cancer with Microrna Data Using the Deep Neural Network (DNN) Method", *Journal of Physics: Conference Series*, Vol. 1524, No. 1, pp. 1-9, 2020.

[16] A. R. Owens, C. E. M. Inerney, K. M. Prise, D. G. M. Art, and A. J. Loughrey, "Novel Deep Learning-Based Solution for Identification of Prognostic Subgroups in Liver Cancer (Hepatocellular Carcinoma)", *BMC Bioinformatics*, Vol. 22, pp. 1-22, 2021.

[17] S. Fu, H. Lai, Q. Li, Y. Liu, J. Zhang, J. Huang, and M. Huang, "Multi-Task Deep Learning Network to Predict Future Macrovascular Invasion in Hepatocellular Carcinoma", *EClinicalMedicine*, Vol. 42, pp. 1-10, 2021.

[18] P. K. Shukla, M. Zakariah, W. A. Hatamleh, H. Tarazi, and B. Tiwari, "AI-DRIVEN Novel Approach for Liver Cancer Screening and Prediction Using Cascaded Fully Convolutional Neural Network", *Journal of Healthcare Engineering*, pp. 1-14, 2022.

[19] E. Othman, M. Mahmoud, H. Dhahri, H. Abdulkader, A. Mahmood, and M. Ibrahim, "Automatic Detection of Liver Cancer Using Hybrid Pre-Trained Models", *Sensors*, Vol. 22, No. 14, pp. 1-20, 2022.

[20] M. Cabiati, M. Gaggini, P. D. Simone, and S. D. Ry, "Data Mining of Key Genes Expression in Hepatocellular Carcinoma: Novel Potential Biomarkers of Diagnosis Prognosis or Progression", *Clinical & Experimental Metastasis*, Vol. 39, No. 4, pp. 589-602, 2022.

[21] W. Li, J. Liu, W. Zhu, X. Jin, Z. Yang, W. Gao, and H. Zhu, "Identification of Biomarkers for Hepatocellular Carcinoma Based on Single Cell Sequencing and Machine Learning Algorithms", *Frontiers in Genetics*, Vol. 13, pp. 1-10, 2022.

[22] M. U. Kakar, M. Z. Mehboob, M. Akram, M. Shah, Y. Shakir, H. W. Ijaz, and Y. Yin, “Identification of Differentially Expressed Genes Associated with the Prognosis and Diagnosis of Hepatocellular Carcinoma by Integrated Bioinformatics Analysis”, *BioMed Research International*, pp. 1-12, 2022.

[23] Y. Wang, L. Ma, P. Xue, B. Qin, T. Wang, B. Li, and X. Liu, “Construction and Analysis of Hepatocellular Carcinoma Prognostic Model Based on Random Forest”, *Canadian Journal of Gastroenterology and Hepatology*, pp. 1-20, 2023.

[24] G. Karypis and V. Kumar, “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs”, *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 359-392, 1998.

[25] The cancer genome atlas program (TCGA). ccg - National Cancer Institute. (n.d.). Retrieved February 1, 2023, from https://www.cancer.gov/ccg/research/genome-sequencing/tcga