2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) | 979-8-3503-7149-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/EMBC53108.2024.10781976

A Hybrid GCN-LSTM Model for Ventricular Arrhythmia Classification Based on ECG Pattern Similarity

Qing Lin[1], Dino Oglić[2], Hak-Keung Lam[1], Michael J. Curtis[3] and Zoran Cvetkovic[1]

Abstract— **Accurate differentiation between Ventricular Tachycardia (VT) and Ventricular Fibrillation (VF) is essential in the field of cardiology. Recent advancements in deep learning have facilitated automated arrhythmia recognition, surpassing traditional electrocardiogram (ECG) methods that depend on manual feature extraction. Building on our previous work, which emphasized the importance of identifying patterns of regularity, we have developed a model that merges Graph Convolutional Networks (GCN) with Long Short-Term Memory (LSTM) networks. This GCN-LSTM model employs a trainable weighted ϵ-neighborhood graph to capture the similarity among time series within ECG segments. This approach has demonstrated substantial improvement in the classification of VT, VF, and non-ventricular rhythms.**

I. INTRODUCTION

In clinical cardiology, accurately differentiating between VT and VF is essential for delivering immediate and effective treatment. Whilst traditional ECG diagnostic methods primarily rely on expert manual evaluation, with the advent of deep learning, more sophisticated techniques have been developed that can automatically recognize and differentiate various cardiac arrhythmias.

Recent studies have demonstrated promising results by applying GCNs to model ECG signals for improved arrhythmia detection and classification. For example, He et al. [1] proposed an unsupervised domain adaptation framework utilizing a GCN module to enable robust cross-subject ECG arrhythmia detection. The GCN helps align data distributions across subjects by extracting graph-structured feature representations at multiple levels while also enabling label and semantic alignment. Other studies [2], [3], [4], [5] have similarly utilized GCN architectures to capture spatial and temporal relationships in multi-lead ECG data, incorporate domain knowledge of 12-lead dependencies, model interrelationships between different cardiac disorders for multi-label classification, and exploit inter-lead correlations without handcrafted features. These GCN-based approaches have demonstrated an ability to effectively leverage the intrinsic graph structure of multi-lead ECG signals to advance automated analysis, classification, and diagnostic capabilities beyond current state-of-the-art methods. Most recently, Ma et al. [6] proposed an AF detection algorithm based on GCN, which constructs heartbeat graphs from ECG data and applies GCNs to jointly model ECG morphological patterns and rhythmic dependencies with predefined edge weight, outperforming current methods on atrial fibrillation detection. One possible limitation of these methods is that they predefine the edge weights when constructing the graph structure, without learning from the available data. Hard-coding the edge weights fails to take full advantage of the patterns and relationships in the data. To address this limitation, we proposed a graph structure with trainable edge weights for GCN to enhance the classification of VT and VF from ECG data with the GCN-LSTM network.

Our previous study [7] made progress by mapping ECG feature similarity, demonstrating the importance of capturing regularity for detecting dangerous rhythms. Building on this, our approach leverages GCN to capture the similarities within ECG segments, with the innovative addition of trainable edge weights in the graph-based structure of ECG segment. In this graph structure, each node represents unique morphological features derived from ECG segments. To our knowledge, this is the first application of GCNs for ECG-based VT and VF classification. Our GCN-LSTM framework improves its ability to differentiate between VT and VF, achieving an average sensitivity of 85.86%

The remainder of the paper is organised as follows. Section II presents details of used methodologies. Section III discusses the dataset setting, model structure and experimental evaluation method. Section IV summarises the experimental outcomes. In the end, conclusions were drawn in Section V.

II. METHODS

A. ECG Graph Construction

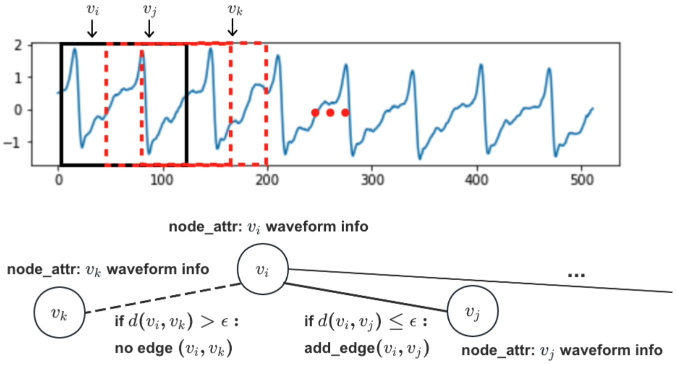
v_i
v_j
v_k
2
1
0
−1
0
100
200
300
400
500
node_attr: v_i waveform info
node_attr: v_k waveform info
v_i
...
v_k
v_j
if $d(v_i, v_k) > \epsilon$:
no edge (v_i, v_k)
if $d(v_i, v_j) \leq \epsilon$:
add_edge(v_i, v_j)
node_attr: v_j waveform info

Fig. 1. An example of an ϵ-neighborhood graph.

[1] Qing Lin, Hak-Keung Lam and Zoran Cvetkovic are with Department of Engineering, King's College London, The strand, London, WC2B 4BG, UK. qing.lin@kcl.ac.uk, hak-keung.lam@kcl.ac.uk, zoran.cvetkovic@kcl.ac.uk

[2] Dino Oglic is with Center for AI, Data Sciences & AI, BioPharmaceuticals R&D, AstraZeneca, Cambridge, UK. dino.oglic@astrazeneca.com

[3] Michael Curtis is with School of Cardiovascular Medicine & Sciences, King's College London. michael.curtis@kcl.ac.uk

1) Unweighted ϵ-neighborhood graph: For each ECG segment, we construct a graph $G = (V, E)$ by dividing the 5s ECG segment into overlapping sub-segments of equal length using a sliding window approach. Specifically, we use a window size of 0.5 seconds and a stride of 0.06 seconds. Each of these sub-segments is considered as a node in the graph, resulting in a set of nodes denoted as $V = \{v_1, v_2, ..., v_n\}$. The node feature matrix $V \in \mathbb{R}^{n \times m}$ represents the attributes of the nodes, where n represents the number of nodes and m represents the dimension of input features for each node ($m = 128$). In this experiment, we assign the raw ECG sub-segments as the node attributes.

The edge set E represents the connectivity information between the nodes in the graph. To establish connections between two nodes, a distance metric, dynamic time warping, is used to calculate the distance between each node pair. If the calculated distance is smaller than a specified threshold value ϵ, a non-negative scalar, the nodes are considered connected and a connection is established between them in the graph. If the distance exceeds ϵ, the two nodes are not connected. In this study, ϵ is 5.

The node-wise formulation is given by:

$$\mathbf{v}'_i = \left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{\hat{A}_{ij}}{\sqrt{\hat{D}_{ii} \hat{D}_{jj}}} \mathbf{v}_j \right) \mathbf{W} \tag{1}$$

where A_{ij} are entries of the adjacency matrix $\hat{\mathbf{A}}$ with inserted self-loops, such that $\hat{A}_{ij}$ is 1 if there is a connection between node j and node i, $\hat{D}_{ii} = 1 + \sum_{j=0}^{n-1} \hat{A}_{ij}$, where both adjacency matrix $\hat{A}$ and degree matrix $\hat{D}$ are constants, and $\mathbf{W}$ is a trainable weights matrix and its dimension is the size of the input sample × size of the output sample.

2) Weighted ϵ-neighborhood graph: Our previous study [7] has demonstrated the importance of capturing regularity in ECG data for detecting abnormal heart rhythms. To better model these regularities, we extended the ϵ-neighbourhood graph framework by introducing edge weights, which were determined by the similarity between each pair of ECG sub-segments.

The edge weight between each node pair is calculated from the pairwise distance between the two sub-segments – smaller distances indicating higher similarity result in larger edge weights. The rationale is that sub-segments with higher similarity (smaller distances) likely share common features and rhythmic patterns. The distance-based edges allow the GCN to propagate relevant signals between similar sub-segments. This preserves local regularities critical for accurate rhythm classification while leveraging correlations between related sub-segments across the ECG time series. The graph connectivity and weighting schemes enable the GCN to jointly model local patterns within each sub-segment and global correlations between similar sub-segments.

To calculate the weights of the edges, the process begins with determining the distance $d(v_i, v_j)$ between the feature representations of segments v_i and v_j through the application of dynamic time warping. The edge weight between v_i and v_j is then assigned as the exponential decay of this distance, $\alpha_{ij} = e^{-d(v_i, v_j)}$.

3) Weighted ϵ-neighborhood graph with trainable edge weights: In addition to using a predefined edge weight, we also explore the adaptation of edge weights during the training process. These weights are initially defined according to the similarity among subsegments of ECG data and updated as part of the training process of the GCN model. Tuning edge weights during training aims to capture the varying importance of relationships within the graph data in specific tasks that enhance the model's ability to classify accurately.

We define the edge weights α_{ij} between nodes v_i and v_j using an exponential decay on their dynamic time warping distance $d(v_i, v_j)$:

$$\alpha_{ij} = b^{-d(v_i, v_j)} \tag{2}$$

The decay rate, denoted by b, is a parameter that can be learned during training. It is bounded within the interval from 1 to 5 and is initially set to a random value within this range. By learning b, the weights can be tuned flexibly during training. The parameter b controls how rapidly the edge weights decay exponentially with increasing distance between nodes. Larger values of b, approaching the upper limit of 5, cause the weights to decay more sharply as distance grows. This makes the graph structure emphasize stronger connections between more similar nodes. In contrast, smaller b values produce a more gradual decay, allowing nodes farther apart to still have a significant influence on each other.

We incorporate these trainable weights into GCN propagation:

$$\mathbf{v}'_i = \left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{\alpha_{ij}}{\sqrt{\hat{D}_{ii} \hat{D}_{jj}}} \mathbf{v}_j \right) \mathbf{W} \tag{3}$$

where degree matrix $\hat{D}_{ii} = \alpha_{ii} + \sum_{j=0}^{n-1} \alpha_{ij}$ sums the node's self-weight and adjacent edge weights. Both adjacency matrix $\hat{A}$ and degree matrix $\hat{D}$ are trainable.

B. Graph Neural Network Architectures

1) Graph convolutional network (GCN): For each graph representation G, node features are propagated using a GCN. For the k-th layer in GCN, the GCN takes graph G's adjacent matrix $\hat{\mathbf{A}}$ and hidden representation matrix $\mathbf{H}_i$ as input, and then the next layer's output will be generated as follows:

$$\mathbf{H}_{\mathrm{k}+1} = \sigma \left(\hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}_{\mathrm{k}} \mathbf{W}_{\mathrm{k}} \right) \tag{4}$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix of graph G with self-connections, where $\mathbf{I}$ is the identity matrix. $\hat{\mathbf{D}}$ is the diagonal matrix of $\hat{\mathbf{A}}$, with $\hat{\mathbf{D}}_{ii}$ giving the degree of node i by counting edges in $\hat{\mathbf{A}}$ connected to node i, including the self-loop. $\sigma(.)$ is the activation function, $\mathbf{W}_k$ is a trainable weight matrix, $\mathbf{H}_k$ is the output of the k-th layer, and $\mathbf{H}_0 = \mathbf{X}$ is the input feature matrix.

This equation represents a single layer of a GCN, where the input $\mathbf{H}_k$ is transformed using a weight matrix $\mathbf{W}_k$ and the adjacency matrix $\hat{\mathbf{A}}$, which is normalized using the diagonal degree matrix $\hat{\mathbf{D}}$. The result is then passed through an activation function $\sigma(.)$ and used as the output $\mathbf{H}_{k+1}$ of the layer.

2) GCN-LSTM: Pooling operations in GCN may result in the loss of critical sequential or positional data, which is problematic when node order matters for understanding the context or behavior. So we applied GCN to extract spatial features from graph-based data, and then these features were input into LSTM to capture their sequential dependencies.

3) Graph Attention Networks (GAT): Considering the need to assign varying levels of importance to each edge and node when generating node features for the next layer, we are considering the use of a GAT that selectively focuses on the most critical patterns from an unweighted ϵ-neighborhood graph.

The node-wise formulation is given by:

$$\mathbf{v}'_i = \alpha_{ii}\mathbf{W}\mathbf{v}_i + \sum_{j\in\mathcal{N}i} \alpha_{ij}\mathbf{W}\mathbf{v}_j \qquad (5)$$

The attention coefficients α_{ij} are computed as:

$$\alpha_{ij} = \frac{exp(LeakyReLU(\mathbf{a}^\top[\mathbf{W}v_i||\mathbf{W}v_j]))}{\sum_{k\in\mathcal{N}i} exp(LeakyReLU(\mathbf{a}^\top[\mathbf{W}v_i||\mathbf{W}v_k]))} \qquad (6)$$

III. EXPERIMENTAL SETUP

A. Data and Preprocessing

ECG signals were obtained from five publicly available arrhythmia databases: MIT-BIH Arrhythmia Database (MITDB) [8], the MIT-BIH Malignant Ventricular Arrhythmia Database (VFDB) [9], the European ST-T Database (EDB) [10], the Creighton University Ventricular Tachyarrhythmia Database (CUDB) [11] and the extended American Heart Association Database (AHADB). In this study, only records containing VT or VF episodes are kept, excluding those with ventricular flutter due to inconsistent labeling. As a result, 91 records were used in our experiments. In the context of ventricular arrhythmias diagnostics, the rhythms within the selected recordings are labeled as VT, VF, and NVR (all other rhythms that are not ventricular arrhythmias). Mislabeling is commonplace in public databases because of a lack of consistency in the use of diagnostic criteria for distinguishing between VF and VT. Consequently, we invited an expert from King's College London to carefully relabel the collected data according to Lambeth conventions [12]. After relabeling, only 78 of 91 records carried ventricular arrhythmia episodes. Table I shows the total duration of each category present across used records before and after the relabeling. The relabeled data set will be made publicly available.

All collected records were resampled to 250Hz and processed by a high-pass Butterworth filter with a cutoff of 0.5 Hz to remove the ECG baseline wander. Each record was normalized to zero mean and a standard deviation of 1.

TABLE I
THE AMOUNT OF DATA BEFORE AND AFTER RELABELING.

Class	Original(s) - 91 records	Relabeled(s) - 78 records
NVR	157366	122405
VT	5348	10021
VF	16229	3175
Total	178943	135601

ECG records were split into non-overlapping segments for training and testing. To deal with the class imbalance problem, we randomly subsampled the largest class (NVR class) to the size of the second-largest class.

B. Model Structure

1) GCN: The graph network constructed in this study consisted of three GCN layers that encoded node attributes while incorporating graph structure, with each of the three GCN layers containing 64 hidden units. Node representations were then propagated through the graph to a readout layer, which produced a graph-level representation by averaging the features across all nodes. Dropout with a probability of 0.5 was applied to the pooled features to prevent overfitting during training. Finally, a fully connected layer with a softmax function was applied to produce the output.

2) GCN-LSTM: The architecture of the GCN-LSTM model comprises three GCN layers succeeded by two LSTM layers, where the number of hidden units is 20 for the LSTM. A dropout layer with a probability of 0.5 is applied thereafter. The model concludes with a fully connected layer and a softmax function for the classification.

3) GAT: We explore two versions of the Graph Attention Network (GAT [13] and GATv2 [14]). Our model, comprising three GAT layers, encodes node attributes and graph structure. Node representations flow to a readout layer for averaging into a graph-level output. Dropout (p=0.5) combats overfitting, and a fully connected layer with softmax activation produces predictions.

C. Training

Classification experiments were performed using 10 bootstraps resamples, with 80% of collected ECG records as training set and the rest held for testing. To have a fair comparison between methods, each experiment used the same hyperparameters including learning rate and batch size, etc. Adam optimizer with initialized learning rate $lr = 5 \times 10^{-4}$ was applied to update the weights. The models were trained with a maximum of 50 epochs and mini-batches of size 32.

D. Evaluation Method

Due to the class imbalance, we report the sensitivity of each given category, which measures the proportion of correctly classified samples:

$$sn_s = \frac{TP_s}{TP_s + FN_s}, s \in S \qquad (7)$$

where the TP_s is the number of examples in class s correctly assigned to class s, and the NF_s is the number of examples in class s incorrectly assigned to other classes.

Another metric to determine the performance of a classifier is the unweighted sensitivity, or average sensitivity, which can be calculated as:

$$SEN_{avg} = \frac{1}{|S|} \sum_{s \in S} sn_s \tag{8}$$

IV. Experimental Results

TABLE II
Average Confusion Matrices for Graph Neural Network Methods in ECG Arrhythmia Classification

Method		Diagnosed as			
	SEN_{avg}	Truth	VT(%)	VF(%)	NVR(%)
ResNet34 [15]	75.90	VT	78.68	16.85	4.47
		VF	25.22	55.01	19.77
		NVR	2.83	3.00	94.17
GCN with unweighted ϵ-neighborhood graph	79.15	VT	75.08	24.64	0.28
		VF	36.86	62.89	0.25
		NVR	0.05	0.47	99.48
GAT [13] with unweighted ϵ-neighborhood graph	80.85	VT	83.34	16.66	0.00
		VF	40.09	59.22	0.69
		NVR	0.00	0.01	99.99
GATv2 [14] with unweighted ϵ-neighborhood graph	81.76	VT	85.32	14.65	0.03
		VF	37.82	59.97	2.21
		NVR	0.00	0.01	99.99
GCN with weighted ϵ-neighborhood graph	82.53	VT	71.05	28.95	0.00
		VF	22.98	76.62	0.40
		NVR	0.00	0.11	99.89
GCN with weighted ϵ-neighborhood graph and trainable edge weights	84.11	VT	77.12	22.88	0.00
		VF	24.58	75.33	0.09
		NVR	0.00	0.15	99.85
GCN-LSTM with weighted ϵ-neighborhood graph and trainable edge weights	85.87	VT	82.01	17.99	0.00
		VF	23.98	75.95	0.07
		NVR	0.00	0.37	99.63

A comparative analysis was conducted in Table II evaluating different graph neural network architectures on the task of classifying ECG segments into ventricular tachycardia, ventricular fibrillation, or normal ventricular rhythm. We compared the following architectures: a 3-layer GCN employing unweighted ϵ-neighborhood graph, a 3-layer GCN using weighted ϵ-neighborhood graph, a 3-layer GCN with trainable exponential decay edge weights, a 3-layer GCN followed by 2-layer LSTM using trainable exponential decay edge weights, and a three-layer GAT and GATv2, both using unweighted ϵ-neighborhood graphs.

We are comparing with ResNet34 [15] as a reference because it achieved cardiologist-level accuracy in detecting and classifying arrhythmias in ambulatory ECGs. All GCN-based models outperformed the 34-layer ResNet in terms of average sensitivity, with notable enhancements in the differentiation of ventricular arrhythmias from non-ventricular arrhythmias. Models utilizing weighted graphs demonstrated improved performance compared to those with unweighted graphs. Additionally, models that learn edge weights dynamically showed further improvements over those with fixed weights. This indicates that tuning the graph structure to the specific task allows for more precise node embeddings and predictions. The GCN-LSTM model, which employs a weighted ϵ-neighborhood graph with trainable edge weights achieved the highest average sensitivity of 85.87% over 10 distinct data splits by introducing temporal dependencies within ECG segments,

V. Conclusions

In this study, we employed a GCN-LSTM model to distinguish between VT, VF and NVR using ECG data. The ECG dataset was collected from five public repositories and then re-labeled according to labeling standards established by the Lambeth Conventions (II) [12]. A new method was introduced that utilizes a weighted ϵ-neighborhood graph, designed to account for the similarities among ECG subsegments. This approach includes trainable edge weights to provide a more precise representation of the importance of each connection in the graph for targeted tasks, consequently improving the model's ability to classify data effectively. The results show that GCN-LSTM model, which employs a weighted ϵ-neighborhood graph with trainable edge weights may improve the classification between VT, VF and NVR in ECG signals. The GCN variant explored in this study has certain limitations. Future work will aim to introduce more advanced and expressive versions of GCNs.

References

[1] Z. He et al. A novel unsupervised domain adaptation framework based on graph convolutional network and multi-level feature alignment for inter-subject ecg classification. *Expert Systems with Applications*, 221:119711, 2023.

[2] H. Wang et al. A weighted graph attention network based method for multi-label classification of electrocardiogram abnormalities. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 418–421, 2020.

[3] B. Andayeshgar et al. Developing graph convolutional networks and mutual information for arrhythmic diagnosis based on multichannel ecg signals. *International Journal of Environmental Research and Public Health*, 19(17):10707, 2022.

[4] Z. Jiang et al. Diagnostic of multiple cardiac disorders from 12-lead ecgs using graph convolutional network based multi-label classification. In *2020 Computing in Cardiology*, pages 1–4. IEEE, 2020.

[5] M. Zhong et al. Automatic arrhythmia detection with multi-lead ecg signals based on heterogeneous graph attention networks. *Math. Biosci. Eng*, 19:12448–12471, 2022.

[6] H. Ma and L. Xia. Atrial fibrillation detection algorithm based on graph convolution network. *IEEE Access*, 2023.

[7] Q. Lin et al. Similarity maps for ventricular arrhythmia classification. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1927–1930. IEEE, 2022.

[8] G.B. Moody and R.G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[9] S.D. Greenwald. *The development and analysis of a ventricular fibrillation detector*. PhD thesis, Massachusetts Institute of Technology, 1986.

[10] A. Taddei et al. The european st-t database: standard for evaluating systems for the analysis of st-t changes in ambulatory electrocardiography. *European heart journal*, 13(9):1164–1172, 1992.

[11] F.M. Nolle et al. Crei-gard, a new concept in computerized arrhythmia monitoring systems. *Computers in Cardiology*, 13:515–518, 1986.

[12] M.J. Curtis et al. The lambeth conventions (ii): guidelines for the study of animal and human ventricular and supraventricular arrhythmias. *Pharmacology & therapeutics*, 139(2):213–248, 2013.

[13] P Veličković et al. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[14] S Brody et al. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

[15] A.Y. Hannun et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65, 2019.