Deep Graph Convolutional Autoencoder with Conditional Normalizing Flow for Power Distribution Systems Fault Classification and Location

Mohsen Saffari, *Member, IEEE*, Mahdi Khodayar, *Member, IEEE*, Mohammad E. Khodayar, *Senior Member, IEEE*, Seyed Saeed Fazlhashemi, *Student Member, IEEE*

***Abstract*—Accurate fault classification and location are critical to ensure the reliability and resilience of large-scale power distribution systems (PDSs). The existing data-driven works in this area struggle to capture essential space-time correlations of PDS measurements and often rely on deterministic and shallow neural architectures. Furthermore, they encounter challenges like over-smoothing and the inability to capture deep correlations. To overcome these limitations, a novel deep space-time generative graph convolutional autoencoder (SGGCA) is proposed. First, the PDS is modeled as a space-time graph where the nodes and edges show the bus measurements and line impedance values, respectively. The proposed SGGCA's encoder captures deep correlations of the space-time graph using a new graph convolution with early connections and identity transformations to mitigate the over-smoothing. Our encoder encompasses a new recurrent method to adjust graph convolution parameters without relying on node embeddings on the temporal dimension. Additionally, it incorporates generative modeling by capturing the probability distribution function of the latent representation through a conditional normalizing flow model. The extracted generative space-time features are enhanced by a multi-head attention mechanism to better capture task-relevant characteristics of the PDS measurements. The extracted features are fed to sparse decoders to classify and locate the faults in the PDS. The feature sparsity of decoders ensures a high generalization capacity and avoids overfitting. The proposed method is evaluated on the IEEE 69-bus and 123-bus systems. It achieves substantial improvements in fault classification accuracy by 3.33% and 6.26% and enhances fault location accuracy by 6.33% and 5.73% for the respective PDSs compared to state-of-the-art models.**

***Index Terms*—Fault Classification, Fault location, Deep Convolution Network, Conditional Normalizing Flow, Deep Spars Architectures**

***Impact Statement*—Fault classification and location are crucial to ensure the resilience of PDSs. Current data-driven models in this field have difficulty capturing space-time correlations of PDS measurements and frequently depend on deterministic and shallow neural networks. Moreover, they face issues including over-smoothing and the failure to capture deep relationships. A new deep space-time generative graph convolutional autoencoder is introduced to address these shortcomings. The proposed framework captures both spatial and temporal characteristics of the PDS using graph learning. Also, a novel generative model is devised to learn the unsupervised features of the input PDS. To mitigate the over-smoothing issue and enhance the generalization capacity of graph learning, a formulation for graph convolution is devised that automatically adjusts the convolution kernels using a recurrent neural architecture. The framework is evaluated on the IEEE 123-bus and IEEE 69-bus systems and has shown significant improvements in fault classification and location compared to the state-of-the-art benchmarks.**

This research is supported by the National Science Foundation under grants ECCS-2223628 and ECCS-2223629.

M.Saffari is with the Department of Electrical and Computer Engineering, Purdue University Northwest, Hammond, IN, 46323, USA (email: msaffari@pnw.edu).

M. Khodayar is with the Department of Computer Science of the University of Tulsa, Tulsa, OK, 74104, USA (email: mahdi-khodayar@utulsa.edu).

M. E. Khodayar and S. S. Fazlhashemi are with the Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX, 75205, USA (emails: mkhodayar@smu.edu and sfazlhashemi@smu.edu).

NOMENCLATURE

Abbreviations

μPMU	Micro Phasor Measurement Unit
ACC	Accuracy
ACNN-LSTM	Attention-based CNN-LSTM
CNF	Conditional Normalizing Flow
CNN	Convolutional Neural Network
D-CNN	Double CNN
DNN	Deep Neural Network
GCN	Graph Convolution Network
GGNN	Gated Graph Neural Network
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MHSA	Multi-Head Self-Attention
PDF	Probability Distribution Function
PDS	Power Distribution Systems
PLSTM	Peephole Long Short-Term Memory
RCIT	Residual Connections and Identity Transformation
Real-NVP	Real-Valued Non-Volume Preserving
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RT	Reparameterization Trick
SGD	Stochastic Gradient Descent
SGGCA	Space-time Generative Graph Convolutional Autoencoder
SGGCE	Space-time Generative Graph Convolutional Encoder
W-CNN	Wavelet CNN
W-SVM	Wavelet Support Vector Machine

Symbols

$\alpha^{(l)}, \beta^{(l)}$	Hyperparameters of l-th GCN layer in Ω
$\bar{N}$	Number of bus
$\hat{\mathcal{Y}}_l^{dist}$	Estimated value of $\mathcal{Y}_l^{dist}$
$\hat{\mathcal{Y}}_l^{type}$	Estimated value of $\mathcal{Y}_l^{type}$
$\hat{A}_{l,t}$	Reconstructed adjacency matrix of $\mathcal{G}_{l,t}$
$\hat{X}_{l,t}$	Reconstructed nodal features of $\mathcal{G}_{l,t}$

κ	Maximum number of $f^{(j)}$
λ_i	i-th eigenvalue of L
$\mathcal{G}_{l,\bar{t}}$	Space-time graph for l-th sample at time time step $\bar{t}$
$\mathcal{G}_l$	Space-time graph for l-th sample during fault time
$\mathcal{L}_{CE}$	Cross entropy loss function
$\mathcal{P}_\eta$	PDF of η
$\mathcal{Q}, \mathcal{K}, \mathcal{V}$	Query, Key, and Value matrix of MHSA
$\mathcal{Y}_l^{dist}$	Fault distance matrix for l-th sample
$\mathcal{Y}_l^{type}$	Fault type matrix for l-th sample
$\odot$	Hadamard product
ϕ_j	Tunable parameters of $f^{(j)}$
Ψ_l	Averaged latent features for l-th sample
$\tilde{C}_t$	Layerwised average of $C_t^{(l)}$ in ξ
ξ	PLSTM neural architecture
$A^{i,j}$	Average activation of i-th neuron at j-th layer
A_t	Adjacency matrix of $\mathcal{G}_t$
$C_t^{(l)}$	Depth control gate of l-th layer of ξ at time t
D^{dist}	Sparse decoder for fault location
D^{type}	Sparse decoder for fault classification
D_E	Sparse decoder for nodal feature reconstruction
D_F	Sparse decoder for edge feature reconstruction
$e_{l,t}^{i,j}$	Connecting edge between i and j nodes of $\mathcal{G}_{l,t}$
$f^{(j)}$	j-th differentiable bijective transformation function
$f_\mu^{(j)}$	Scaling and Translation function in j-th coupling layer
$f_t^{(l)}$	Forget gate of l-th layer of ξ at time t
$H_t^{(l)}$	Hidden representation of l-th GCN layer
H_t	Spatiotemporal features at time step t
$i_t^{(l)}$	Input gate of l-th layer of ξ at time t
J_{η_t}	CNF loss function
J_C	Fault classification loss function
J_G, J_F	Jacobian matrix of transformation G and F
J_L	Fault location loss function
$J_S^{D^{dist}}$	Sparsity loss function for decoder D^{type}
$J_S^{D^{type}}$	Sparsity loss function for decoder D^{type}
$J_S^{D_E}$	Sparsity loss function for decoder D_E
$J_S^{D_F}$	Sparsity loss function for decoder D_F
L_t	Normalized Laplacian matrix
N	Number of bus-phase
N_A	MHSA number of heads
$O_t^{(l)}$	Output gate of l-th layer of ξ at time t
$T_t^{(l)}$	Maximum singular value of $W_t^{(l)}$
U_t	Orthogonal eigenvalue matrix
$v_{l,t}^i$	i-th bus-phase node of the graph $\mathcal{G}_{l,t}$
$W^{(l)}$	Weight matrix of l-th GCN layer
$W_t^{(l)}$	Time-dependent weight matrix of l-th GCN layer
$W^{Q_i}, W^{K_i}, W^{V_i}$	Tunable parameters of i-th head of MHSA
X_t	Nodal features of $\mathcal{G}_t$
$y_{l,i,j}^{dist}$	Entry i and j of the matrix $\mathcal{Y}_l^{dist}$
$y_{l,i,j}^{type}$	Entry i and j of the matrix $\mathcal{Y}_l^{type}$
$Z^{i,j}$	Impedance between bus-phase i and j
Z_t	Attention-enhanced generative spatiotemporal features at time t
η_t	Attention-enhanced spatiotemporal features at time t
Λ_t	Diagonal eigenvalue matrix
Ω	RCIT graph convolution operation
F	Invertible transformation function composed of $f^{(j)}$
G	Inverse of F

I. Introduction

FAULT location and classification play a pivotal role in an efficient restoration scheme for the power distribution system (PDS) that aims to minimize the adverse consequences of outages [1]. PDS suffers from four primary fault types, namely single line-to-ground fault, line-to-line fault, double line-to-ground fault, and three-phase fault [2]. Identifying the precise location of the fault expedites the diagnostic and repair process and reduces the disruption duration and the number of affected customers. Nevertheless, fault location in distribution feeders presents formidable challenges due to the non-homogeneous nature of feeders, the radial topology of the network, the existence of laterals to distribute the power, and the constraints imposed by available measurement assets [3]. This complex landscape has prompted extensive research efforts in recent years, aimed at innovative approaches for fault location and classification in PDS. These approaches are categorized into three forms namely impedance-based techniques, traveling wave-based methods, and data-driven methodologies.

Impedance-based fault location models utilize voltage and current waveforms recorded by intelligent electronic devices during faults to estimate the impedance to the fault. This allows accurate determination of the fault's distance from the device that makes them a practical and effective methodology for fault location [4]. A noise-robust impedance-based method was devised in [5], which needs recorded voltage and current measurements of the feeder and photovoltaic distributed generation buses. This model determines the fault location using a π model of the line. Using Thevenin equivalents and quantifying the contribution of distributed generations to the fault current, Yang et al. [6] proposed an impedance-based model that determined both the fault line section and fault distance without needing to distinguish the fault type in advance. Keshavarz et al. [7] proposed a methodology where the impedance-based technique was used to locate possible fault locations using the recorded voltages and currents. Although the impedance-based methods have found application in pinpointing faults within transmission networks, their application in power distribution networks presents unique challenges due to the heterogeneity of distribution feeders with varying branch types, unbalanced network structure, and load connections and fluctuations. The effectiveness of these approaches is contingent on having precise system parameter measurements, which makes them susceptible to inaccuracies arising from measurement errors. Moreover, these methods rely on intricate computations for fault location, which can be error-prone when dealing with complicated network setups and unbalanced fault scenarios.

Traveling wave-based methods utilize the propagation and reflection characteristics of high-frequency transient signals generated by faults to determine the fault location. These methods primarily focus on time-domain arrival times and in some cases, the frequency-domain analysis of the signals to

locate the fault accurately. Naidu et al. [8] introduced a method for locating faults in two-terminal transmission lines that operates without requiring synchronized current data. By analyzing the initial arrival times of the traveling waves at both ends of the line, their approach determined two candidate locations from which the actual fault point was identified. Next, the correct fault location was identified by comparing the rise time of the first traveling wave recorded at both terminals of the line. A traveling wave-based fault locator methodology was proposed in [1] that adopted a 35 kV single-ended radial distribution network as a model and located faulted buses using the wavelet conversion method. The study in [9] introduces a novel electromagnetic time reversal fault analysis method that leverages the characteristic frequency of traveling waves to improve accuracy, especially under high-impedance fault conditions. The proposed model improved the accuracy of fault location by analyzing the transient signal spectrum and evaluating fault currents based on characteristic frequency energy. Compared to the impedance-based approaches, one key benefit of traveling wave-based methods lies in their immunity to variations in loading, high ground resistance, fault resistance, fault types, and the interconnection of generation resources. Nevertheless, this technique exhibits sensitivity to branch parameters and requires the use of expensive sensors and GPS devices. Additionally, calibration of measurement equipment is necessary to guarantee their accuracy [10].

Due to the recent advances in computational and data acquisition techniques, data-driven and machine-learning approaches are increasingly applied for fault classification and location in power systems. In this context, Rafinia and Moshtagh [11] employed advanced signal processing methods that leverage wavelet analysis to extract meaningful features from raw signal data. They utilized an artificial neural network in conjunction with a fuzzy logic system to identify the fault type and determine its location. Similarly, Ahmed et al. [12] proposed a discrete wavelet transform method and utilized the time-frequency location properties to effectively analyze fault signals in transmission lines under various conditions. More recently, Tunio et al. [13] employed a discrete wavelet transform for extracting the features and a temporal convolutional network for fault classification in a 500 kV transmission line. The work presented in [14] employed a decision tree algorithm to detect single phase-to-ground faults in the PDS by analyzing the collected voltage and current waveforms. In [15], the authors utilized a Random Forest algorithm to perform both fault location and classification. This approach was based on the input data derived from voltage and current information that has been processed utilizing a mathematical morphology technique.

Moloi et al. [16] have developed a framework that comprises a wavelet packet decomposition for signal processing and feature extraction, and a support vector machine for fault classification and location. Similarly, the authors of [17] used wavelet packet decomposition as a feature extractor; however, they employed a multi-layer perceptron neural network to identify the location and type of faults. In [18], a convolutional neural network (CNN) was employed for fault classification in a PDS equipped with distributed generations. Additionally, Zou et al. [19] developed a 1-D convolutional autoencoder to extract crucial features from the PDS transient data, which was followed by a double CNN for fault data identification. Similarly, a 1-D convolutional neural network was proposed in [20] for classifying and locating faults in PDS. Furthermore, Zhao and Barati [21] exploited a CNN to classify and localize fault using the feature vectors extracted from PDS measurements. Thomas et al. [22] introduced a deep framework by integrating 1-D deep CNNs for feature extraction and transformer encoders for sequence learning to identify fault types, phases, and locations under high-impedance fault conditions. To capture temporal correlations in the PDS data, the authors of [23] employed an LSTM with the regression window technique and showed the effectiveness of deep architectures over traditional neural networks. The work presented in [24] utilized the micro phasor measurement unit (μPMU) data for fault location within the PDS. The fault section was identified by analyzing zero sequence current data from both sides of the faulted section. Subsequently, a stacked autoencoder approach was employed to pinpoint the fault location within the identified fault section, utilizing the voltage and current data.

In [25], the authors represented the PDS as a graph, with feeder topology identifying the edges and measurements and electrical characteristics as nodes of the graph. In their study, a gated graph neural network was modeled to estimate fault locations. Similarly, Chen et al. [26] utilized a graph convolutional network for fault location in a PDS. In [27], a robust deep unsupervised framework for fault detection and classification in transmission lines was proposed, employing the discrete wavelet transform to extract temporal features from the power system measurements and a capsule network with sparse filtering to learn and classify fault characteristics. The work in [28] introduced a spatiotemporal graph learning approach, incorporating PDS topological attributes for fault location and classification. It used a graph convolutional network to extract fault-related features from voltage and current measurements within the network and determined both the fault location and type using a multi-task learning framework. Rizeakos et al. [2] developed a deep learning method using continuous wavelet transformation and CNNs to detect and locate faults in active PDS. In their model, the CNN received voltage phasor measurements collected over extended periods ranging from 5 to 20 seconds. These measurements were pre-processed using dynamic mode decomposition and continuous wavelet transform before being input into the model.

Current state-of-the-art data-driven methods for fault location and classification have the following limitations that restrict their applicability in real-world scenarios:

1) The majority of existing studies [11], [15]–[17] fail to capture the spatiotemporal characteristics of PDS measurements, instead relying on merely time-dependent measurement features. A few studies considered capturing space-dependent features such as the CNN [19], [21] and graph convolution network (GCN) [25], [28]. These studies primarily employed tunable spatial filters within the local neighborhoods of system buses to obtain spatial features for each bus. Consequently, these models do not consider the global characteristics existing among the buses. However, this valuable information holds the potential to

enhance the model's ability to provide more precise spatial node embeddings, offering a better representation of the spatial relationships between the adjacent buses.

2) The current spatial and temporal methods such as [26], [28] are characterized by shallow computational architectures and tend to achieve their optimal performance with only a limited number of non-linear convolutional layers in their hidden structure. This shallowness in their design restricts their capability to extract deep-seated information within the input data. In simpler terms, when more non-linear layers are stacked and additional non-linearity is introduced, the performance of these models tends to deteriorate. This phenomenon is commonly known as the over-smoothing [29].
3) The existing methodologies employed deterministic models [21], [25], [28] where a direct mapping was formulated from the PDS measurements to the fault locations and types. Such deterministic techniques do not learn the significant unsupervised generative features from their input data. Therefore, they fail to provide a powerful distinctive set of features for accurate fault identification.

Inspired by these limitations, a novel deep learning framework is proposed for spatiotemporal fault detection and location in the PDS. The PDS is initially modeled as a space-time graph, where the nodes represent the voltage sequence components of the network buses. Each graph edge encapsulates the impedance associated with the connecting lines between the respective buses. The Space-Time Generative Graph Convolutional Autoencoder (SGGCA) framework is introduced to capture robust spatiotemporal features embedded within the space-time graph. First, the proposed SGGCA receives the space-time PDS graph and processes it via novel graph convolution layers meticulously designed to capture deep latent features embedded in the input PDS graph. The proposed GCN architecture enhances the generalization capacity of the model by combining the early connection and identity transformation techniques to effectively extract the deep task-relevant representation of the input dynamic graph. In addition, a recurrent neural architecture is devised to learn the temporal dynamics of the space-time graph and dynamically adjust the parameters of graph convolution layers using the input graph dynamics and structure. Furthermore, a new conditional normalizing flow (CNF) is introduced to enhance the model's generalization capacity for effective PDS representation learning and to accommodate the modeling of latent attention-enhanced deep spatiotemporal representations. Finally, sparse decoders are defined to accurately estimate the type and location of the faults. The contributions of this research work are as follows:

1) This research introduces a novel deep neural architecture capable of simultaneously capturing both temporal and spatial characteristics of PDS measurements. This is achieved through a graph-based representation learning approach equipped with a multihead attention mechanism, which provides task-relevant space-time features for accurate fault classification and location.
2) A new CNF framework is introduced for extracting generative features from PDS measurements to identify faults. The generated features capture semantic characteristics of PDS measurements in an unsupervised probabilistic manner. In contrast to the existing research that primarily employs discriminative architectures (e.g., [19], [21]), our framework utilizes a generative architecture, offering improved generalization capability and reduced risk of overfitting.
3) A novel formulation for the graph convolutional operation within the SGGCA encoder is presented that effectively mitigates over-smoothing via early connections and identity transformation techniques. Addressing this issue enhances the computational power of the proposed framework and enables the use of deeper graph neural architectures compared to current graph learning models (e.g., [25]).
4) A recurrent neural network-based model is devised to adjust the parameters of the deep graph convolution layers of the SGGCA model to capture the dynamics of the space-time PDS graph. The proposed RNN method effectively fine-tunes the parameters of graph convolution layers by combining the node embedding and the dynamic graph structure along with the temporal dimensions.

This paper is organized as follows: Section II outlines the spatiotemporal fault location and classification problem addressed in this study. The details of the proposed SGGCA model are provided in Section III. Section IV shows the experimental results of the proposed framework using the modified IEEE 123-bus and IEEE 69-bus systems. Finally, Section V presents the conclusions of this research.

II. Problem Setup

Consider a dataset for fault location and classification in a PDS denoted by $\mathcal{D} = \{\mathcal{G}_l, \mathcal{Y}_l^{dist}, \mathcal{Y}_l^{type}\}_{l=1}^{M}$ where l shows the index of samples. This dataset comprises M space-time graphs $\{\mathcal{G}_l\}_{l=1}^{M}$ where each space-time graph $\mathcal{G}_l$ is formulated by $\mathcal{G}_l =< \mathcal{G}_{l,\bar{t}}, \mathcal{G}_{l,\bar{t}+1}, ..., \mathcal{G}_{l,\bar{t}+\tau} >$ representing the measurements taken at $\tau + 1$ time steps after a fault incident. Each graph snapshot $\mathcal{G}_{l,t} = (\mathcal{V}_{l,t}, \mathcal{E}_{l,t}),\ t \in [\bar{t}, \bar{t} + \tau]$ contains bus and line features characterizing the PDS at time t. Specifically, a node $v_{l,t}^i \in \mathcal{V}_{l,t}\ i \in [1, \bar{N}]$ encodes the features of the i-th bus-phase node in the graph which shows one phase of one bus of the PDS, while each edge $e_{l,t}^{i,j} \in \mathcal{E}_{l,t}$ represents the impedance of the line connecting nodes i and j. Here, N and $\bar{N} = 3N$ denote the number of buses and bus-phase nodes in the PDS, respectively. The nodal feature vector v_t^i is defined as a 6-dimensional vector, capturing the magnitude and angles of symmetrical components of the voltage at the i-th bus-phase node within the PDS. Additionally, the edge between the i-th and j-th nodes is defined as $e_{l,t}^{i,j} = \exp(-Z^{i,j}) \in \mathcal{E}_{l,t}$ where $Z^{i,j}$ represents the impedance between the i-th and j-th bus-phase nodes. In this context, $\mathcal{Y}_l^{type} \in \mathbb{R}^{N \times N}$ is an N-dimensional square matrix where each (i,j)-th entry, denoted by $y_{l,i,j}^{type} \in \{0, 1, ..., F\}$, indicates the class of the fault between buses i and j. Notably, class 0 signifies a normal situation (i.e., no-fault) between buses i and j. Furthermore, $\mathcal{Y}_l^{dist} \in \mathbb{R}^{N \times N}$ represents the fault distance matrix, where the (i,j)-th entry, denoted as $y_{l,i,j}^{dist} \in [0, 1)$ shows the normalized distance of the fault from the node with the smaller index (i.e., $\min(i,j)$). When $y_{l,i,j}^{dist} = 0$, the line connecting bus i to j does not have any faults. The primary objective of the proposed model is to learn a mapping $\mathcal{F} : \mathcal{G}_l \rightarrow \{\mathcal{Y}_l^{dist}, \mathcal{Y}_l^{type}\}$

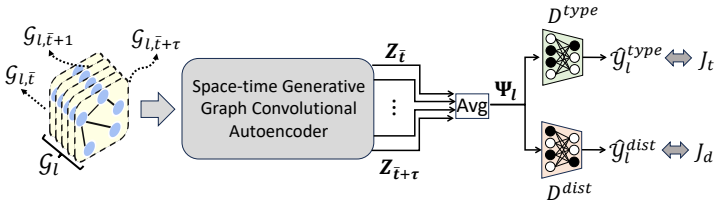


Fig. 1: The proposed deep generative graph convolutional framework for PDS fault classification and location.

that accurately classifies and locates the faults in $\mathcal{G}_l$.

III. Proposed Framework

Fig. 1 shows the structure of the proposed framework for fault classification and location. As shown in this figure, the input PDS space-time graph $\mathcal{G}_l$ is fed to the SGGCA to capture the attention-enhanced spatiotemporal generative feature $Z_t,\ t \in [\bar{t}, \bar{t}+\tau]$ for each snapshot $\mathcal{G}_{l,t}$ and obtain the average latent feature $\Psi_l = \frac{1}{\tau+1}\sum_{t=\bar{t}}^{\bar{t}+\tau} Z_t$. The computed latent feature is then used by sparse decoders D^{type} and D^{dist} to estimate the type and location matrices $\mathcal{Y}_l^{type}$ and $\mathcal{Y}_l^{dist}$, respectively.

Fig. 2 depicts the overall architecture of the proposed SGGCA. As shown in this figure, each snapshot $\mathcal{G}_{l,t}$ of $\mathcal{G}_l$ is processed by the SGGCA to extract the most informative space-time feature H_t from the PDS. The process begins with a novel space-time generative graph convolutional encoder (SGGCE) equipped with residual connections and an identity transformation (RCIT) graph convolution operator. This encoder, further enhanced with recurrent layer-wise weights, is designed to extract complex task-relevant representations of $\mathcal{G}_{l,t}$ at each time step t. Additionally, a multi-head self-attention (MHSA) mechanism is employed that computes an attention-enhanced feature η_t from H_t to capture both global and local dependencies in $\mathcal{G}_{l,t}$. Here, our objective is to learn the generative spatiotemporal features of $\mathcal{G}_l$. Hence, the probability distribution function (PDF) of the attention-enhanced space-time feature $\mathcal{P}_\eta$ of the PDS space-time graph is modeled and learned. To this end, a novel generative CNF is devised that effectively maps the intricate PDF $\mathcal{P}_\eta$ to a predefined Gaussian distribution through a series of adjustable and transformable functions. The reparameterization trick (RT) [30], [31] is employed to sample a generative feature vector Z_t from the newly obtained PDF derived from the CNF. This sample is then fed to the sparse decoders D_E and D_F to accurately reconstruct the input nodal features X_t and adjacency matrix A_t of the input snapshot $\mathcal{G}_{l,t}$ at each time step t. The following subsections provide a detailed explanation of these modules.

A. Space-time Generative Graph Convolution Encoder

The devised SGGCE observes each space-time graph snapshot $\mathcal{G}_{l,t},\ t \in [\bar{t}, \bar{t}+\tau]$ and outputs a generative feature Z_t that best describes the unsupervised features of the snapshot. This feature vector is fed to sparse decoders D_E and D_F to reconstruct the input nodal features X_t and adjacency matrix A_t, respectively. Within this encoder, a new deep graph convolutional network is introduced, meticulously tailored for temporal analysis without relying on node embedding. The proposed framework excels in capturing the dynamics inherent in the input graph sequence. It achieves this through the strategic implementation of a Peephole Long Short-Term Memory (PLSTM) mechanism, which iteratively fine-tunes and optimizes the parameters of the SGGCE.

Let us consider a PDS snapshot $\mathcal{G}_{l,t}$ in $\mathcal{G}_l$ with an adjacency matrix A_t and a diagonal degree matrix D_t. The node feature matrix is defined as $X_t \in \mathbb{R}^{n\times p}$, where each row $X_{t,i},\ 1 \le i \le n$ is a p-dimensional feature vector that contains the magnitudes and angles of symmetrical components of the voltage for the i-th bus-phase node. The normalized Laplacian matrix $L_t \in \mathbb{R}^{n\times n}$ is defined as $L_t = U_t\Lambda_t U_t^\top = I_n - D_t^{-\frac{1}{2}} A_t D_t^{-\frac{1}{2}}$. Here $U_t \in \mathbb{R}^{n\times n}$ comprises orthonormal eigenvectors and $\Lambda_t = diag(\lambda_1, ..., \lambda_n)$ is a diagonal matrix of eigenvalues. To extract the spatial features of $\mathcal{G}_{l,t}$, the K-layer vanilla GCN [32] is defined by:

$$\begin{aligned} H_t^{(l+1)} &= \text{ReLU}(\bar{P}_t H_t^{(l)} W^{(l)}),\ l \in [1, K-1] \\ \bar{P}_t &= (D_t + I_n)^{-\frac{1}{2}}(A_t + I_n)(D_t + I_n)^{-\frac{1}{2}} \end{aligned} \tag{1}$$

where $H_t^{(l)}$ denotes the l-th hidden layer parameterized by the weight matrix $W^{(l)}$ with the rectified linear unit (ReLU) nonlinear function. Here, the input layer is $H_t^{(1)} = X_t$ and the extracted spatial feature is $H_t^{(K)} \in \mathbb{R}^{n\times d}$. To characterize the over-smoothing problem [33] in GCNs for a graph snapshot $\mathcal{G}_{l,t} = (\mathcal{V}_{l,t}, \mathcal{E}_{l,t})$, a node-similarity metric, $\mu\left(H_t^{(K)}\right) = \frac{1}{\sum_{1\le i\le j\le n}\|H_t^{(K)}(:,i), H_t^{(K)}(:,j)\|_F^2}$ is defined that quantifies the similarity between the node representations at layer K. Here, $H_t^{(K)}(:,i)$ and $H_t^{(K)}(:,j)$ denote representation of nodes $i \in V_{l,t}$ and $j \in V_{l,t}$ in $\mathcal{G}_{l,t}$, respectively. Also, $\| x \|_F$ is the Frobenius norm of vector x. The over-smoothing phenomenon can be expressed as $\mu\left(H_t^{(K)}\right) \le C_1 e^{-C_2 K}$ for $K = 0, 1, \ldots, l$, where $C_1, C_2 > 0$ are positive constants. This formulation implies that as the depth K of a vanilla GCN increases, the node similarity metric μ converges to zero (i.e., $\mu \to 0$). In other words, the node features become increasingly indistinguishable, eventually converging to a constant vector. This convergence significantly reduces the expressive power of multi-layer GCN models, as it hinders their ability to capture meaningful differences between nodes.

In the introduced graph encoder architecture, SGGCE, to extend GCN into a genuinely deep model and mitigate the over-smoothing challenge, the K-order filter is represented with arbitrary adaptive coefficients, using the early connection and identity transformation techniques. Here, the graph embedding of the l-th layer of the encoder is defined as:

$$\begin{aligned} H_t^{(l+1)} &= \Omega(A_t, H_t^{(l)}, W_t^{(l)}) \\ &= \sigma\left((1-\alpha^{(l)})\bar{P}_t H_t^{(l)} + \alpha^l H_t^{(0)}\right)\left((1-\beta^{(l)})I_n + \beta^{(l)} W_t^{(l)}\right) \end{aligned} \tag{2}$$

where Ω is the proposed RCIT which is an adaptive graph convolution layer that extracts the space-time feature $H_t^{(l)}$ for each snapshot $\mathcal{G}_{l,t}$. Also, $W_t^{(l)}$ is a time-dependent adaptive weight matrix of layer l at time step t with hyperparameters $\alpha^{(l)}$ and $\beta^{(l)}$. The following theorem proves that the formulation in (2) can represent an L-order polynomial filter or, equivalently, model an L-layer convolution operation.

Theorem 1: Let $\mathcal{G}$ be a graph with node features x. The application L layers of convolution operation in (2) can represent L order polynomial filter $\left(\sum_{l=0}^{L-1}\theta_l L_{norm}^l\right)x$ with arbitrary coefficients θ.

Proof: Without loss of generality, let us consider the signal vector x to be non-negative. Note that x can be converted to the initial representation $H^{(0)}$ using a linear transformation. For the sake of simplicity, let us consider $\alpha^{(l)} = 0.5$ and replace $(1-\beta^{(l)})I_n + \beta^{(l)}W_t^{(l)}$ with $\pi^{(l)}I_n$ where $\pi^{(l)}$ is a learnable coefficient. This substitution represents a weaker form of the convolution operation in (2). Consequently, we have:

$$H^{(l+1)} = \sigma\left((D+I)^{-1/2}(A+I)(D+I)^{-1/2}(H^{(l)}+x)\pi^{(l)}I_n\right) \tag{3}$$

As the input signal is non-negative, the nonlinear function $\sigma(.) = ReLU(.)$ can be discarded and (3) can be rewritten as:

$$\begin{aligned} H^{(l+1)} &= \pi^{(l)}(D+I)^{-1/2}(A+I)(D+I)^{-1/2}(H^{(l)}+x)I_n \\ &= \pi^{(l)}\left((I_n - L).(H^{(l)}+x)\right) \end{aligned} \tag{4}$$

Hence, we can express the final representation of $H^{(L-1)}$ as:

$$H^{(L-1)} = \left(\sum_{l=0}^{L-1}\left(\prod_{i=L-l-1}^{L-1}\pi^{(i)}\right)(I_n - L)^l\right)x \tag{5}$$

The polynomial filter of graph convolution can be represented as:

$$\begin{aligned} \left(\sum_{l=0}^{L-1}\theta_l L_{norm}^l\right)x &= \left(\sum_{l=0}^{L-1}\theta_l(I_n-(I_n-L))\right)x \\ &= \left(\sum_{l=0}^{L-1}\theta_l\left(\sum_{i=0}^{l}(-1)^i\binom{l}{i}(I_n-L)^i\right)\right)x \end{aligned} \tag{6}$$

By switching the order of summations, (6) can be written as:

$$\left(\sum_{l=0}^{L-1}\theta_l L_{norm}^l\right)x = \left(\sum_{i=0}^{L-1}\left(\sum_{l=i}^{L-1}\theta_l(-1)^i\binom{l}{i}(I_n-L)^i\right)\right)x \tag{7}$$

To establish the validity of this theorem, it must be demonstrated that there exists a solution $\pi^{(l)}, l = 0,\ldots,L-1$ such that the coefficients of $(I_n - L)$ in (5) and (7) are equivalent. Mathematically, it needs to be shown that there exists a solution for the following equation system:

$$\prod_{i=L-l-1}^{L-1}\pi^{(i)} = \sum_{l=i}^{L-1}\theta_l(-1)^i\binom{l}{i}, \quad l = 0,...,L-1 \tag{8}$$

The solution for (8) can be obtained by:

$$\pi_{L-l-1} = \frac{\sum_{l=i}^{L-1}\theta_l(-1)^i\binom{l}{i}}{\sum_{i=l-1}^{L-1}\theta_i(-1)^{l-1}\binom{i}{l-1}}, \quad l = 1,...,L-1 \tag{9}$$

Here, $\pi_{L-1} = \sum_{l=0}^{L-1}\theta_l$. Note that the denominator in (9) does not approach zero, as this would imply that the L-order filter ignores all features from the l-order neighbours in graph $\mathcal{G}$. Therefore, this demonstrates that the proposed formulation in (2) can express an L-order polynomial filter $\left(\sum_{l=0}^{L-1}\theta_l L_{norm}^l\right)$ which represent an L-layers GCN with arbitrary coefficients. ∎

In contrast to the approach in [32], which advocates that the early connection should combine the smoothed representation $\bar{P}_t H_t^{(l)}$ with $H_t^{(l)}$; our method, as described in (2), establishes an early connection to the input representation $H_t^{(0)}$. This novel early connection mechanism guarantees that, even with multiple stacked layers, the final representation of each node preserves at least a portion of $\alpha^{(l)}$ from the input layer. Furthermore, the convolution layer proposed in (2) introduces the identity transformation technique that involves adding an identity matrix to $W_t^{(l)}$. This modification guarantees that the proposed convolutional layer can be utilized in a deep graph network and achieves higher generalization capacity compared to GCNs [25], [26], [28] that can merely have a low number of layers as they encounter over-smoothing. Furthermore, theoretical analysis has shown that the node features in a K-layer GCN eventually converge to a subspace, leading to information loss [34]. The speed of this convergence is determined by $T^{(K)}$, where $T^{(K)}$ denotes the maximum singular value of the weight matrices $W_t^{(l)}$, $l = 1,2,...,K$. In the proposed RCIT graph convolution operation, replacing the weight matrix in (1) with $(1-\beta^{(l)})I_n + \beta^{(l)}W_t^{(l)}$ effectively constrains the norm of $W_t^{(l)}$ to be small. As a result, the singular values of $(1-\beta^{(l)})I_n + \beta^{(l)}W_t^{(l)}$ tend to cluster around 1, suggesting that $T^{(K)}$ remains relatively large. Consequently, this helps alleviate information loss.

A critical aspect of the proposed model lies in the dynamic adjustment of the RCIT graph convolution weight matrix $W_t^{(l)}$ using a PLSTM denoted by ξ with the following recurrent definition:

$$\begin{aligned} W_t^{(l)} &= \xi(H_t^{(l)}, W_{t-1}^{(l)}) \\ i_t^{(l)} &= \sigma(\omega_{ih}W_{t-1}^{(l)} + \omega_{ix}H_t^{(l)} + p_i \odot C_{t-1}^{(l)} + b_i) \\ f_t^{(l)} &= \sigma(\omega_{fh}W_{t-1}^{(l)} + \omega_{fx}H_t^{(l)} + p_f \odot C_{t-1}^{(l)} + b_f) \\ d_t^{(l)} &= \sigma(\omega_{dh}W_{t-1}^{(l)} + \omega_{dx}H_t^{(l)} + b_d) \\ C_t^{(l)} &= f_t \odot C_{t-1}^{(l)} + d_t \odot i_t^{(l)} \\ O_t^{(l)} &= \sigma(\omega_{Oh}W_{t-1}^{(l)} + \omega_{Ox}H_t^{(l)} + p_O \odot C_t^{(l)} + b_O) \\ W_t^{(l)} &= O_t^{(l)} \odot tanh(C_t^{(l)}) \end{aligned} \tag{10}$$

where $\odot$ denotes the Hadamard product. Here, the variables $i_t^{(l)}$, $f_t^{(l)}$, $O_t^{(l)}$, and $W_t^{(l)}$ represent the input gate, forget gate, output gate, and the hidden state of the PLSTM in layer l at time step t, respectively. Furthermore, $C_t^{(l)}$ is the depth-control gate, which establishes a connection between the memory cell of the upper layer, denoted as $C_t^{(l)}$, and the memory cell of the lower layer, denoted as $C_t^{(l-1)}$. This gate plays a pivotal role in determining the extent to which information from the lower memory cell is directly transmitted to the upper memory cell.

To enhance the computed space-time feature $H_t^{(K)}$ for $\mathcal{G}_{l,t}$, an MHSA block is incorporated into the SGGCE. This addition empowers our SGGCE to effectively focus on the most significant short-term and long-term dynamics of the $H_t^{(K)}$. As illustrated in Fig. 2, at each time step t, the latent space-time representation $H_t^{(K)}$ is fed into the MHSA block, which computes a set of N_A attention heads $\{\eta_{i,t}\}_{i=1}^{N_A}$. The i-th attention head $\eta_{i,t} \in \mathbb{R}^{n\times\bar{d}}$ is defined as:

$$\begin{aligned} \zeta(\mathcal{Q},\mathcal{K},\mathcal{V}) &= \text{softmax}\left(\frac{\mathcal{Q}\mathcal{K}^\top}{\sqrt{\bar{d}}}\right)\mathcal{V} \\ \eta_{i,t} &= \zeta(H_t^{(K)}W^{\mathcal{Q}_i}, H_t^{(K)}W^{\mathcal{K}_i}, H_t^{(K)}W^{\mathcal{V}_i}) \end{aligned} \tag{11}$$

where the matrices $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ are the query, key, and value matrices. Here, $W^{\mathcal{Q}_i}, W^{\mathcal{K}_i}, W^{\mathcal{V}_i} \in \mathbb{R}^{d\times\bar{d}}$ are the tunable parameters of the i-th head. Also, $\bar{d} = \frac{d}{N_A}$ is the dimension of the attention-enhanced space-time features. The MHSA block concatenates the elements in $\{\eta_{i,t}\}_{i=1}^{N_A}$ to output the attention-enhanced feature η_t corresponding to $H_t^{(K)}$.

B. Generative Conditional Normalizing Flow Graph Decoder

The CNF aims to learn the PDF $\mathcal{P}_\eta$ to capture the generative space-time feature Z_t for $\mathcal{G}_{l,t}$. The representation η_t originates from an unknown distribution $\mathcal{P}_\eta(\eta_t|\tilde{C}_t)$, where $\tilde{C}_t$ represents the average of $C_t^{(l)}$ in (10) for $l = 1, 2, ..., K$. Using a set of differentiable bijective transformation functions $\{f^{(j)},\ j = 1, ..., \kappa\}$ where each function $f^{(j)}$ is parameterized by a tunable parameter ϕ_j, $\eta_t \sim \mathcal{P}_\eta(\eta_t|\tilde{C}_t)$ is mapped to the generative space-time feature vector $Z_t = F(\eta_t, \tilde{C}_t) = F(G(Z_t, \tilde{C}_t), \tilde{C}_t)$, where the distribution $\mathcal{P}_Z(Z_t|\tilde{C}_t)$ is a known PDF. Consider a bijective invertible transformation F and its inverse G, which are composed of κ mapping functions defined as:

$$F = f^{(1)} \circ \dots f^{(\kappa-1)} \circ f^{(\kappa)} = \left(g^{(1)} \circ \dots g^{(\kappa-1)} \circ g^{(\kappa)}\right)^{-1} = G^{-1} \tag{12}$$

where each $g^{(j)},\ j = 1, ..., \kappa$ is parameterized by ψ_j. The forward evaluation of the $\eta_t \to Z_t$ transformation is expressed as a recurssive relationship $Z_t^{(j)} = f^{(j)}(Z_t^{(j-1)}),\ j = 1, ..., \kappa$ with the base case $Z_t^{(0)} = \eta_t$ and final iteration $Z_t^{(\kappa)} = Z_t$. Note that the prior distribution $\mathcal{P}_Z(Z_t|\tilde{C}_t) = \mathcal{N}(Z_t|\mu(\tilde{C}_t), \sigma(\tilde{C}_t))$ takes the form of a diagonal Gaussian distribution with mean and standard deviation parameters μ and σ, respectively, which are modeled using deep neural architectures. The two distributions, $\mathcal{P}_\eta$ and $\mathcal{P}_Z$, are interconnected through the change of variable rule defined by:

$$\begin{aligned}
\mathcal{P}_\eta(\eta_t|\tilde{C}_t) =& \mathcal{P}_Z(Z_t|\tilde{C}_t)\,|det J_G(Z_t)|^{-1} \\
=& \mathcal{P}_Z(Z_t|\tilde{C}_t)\left|det \frac{\partial G(Z_t, \tilde{C}_t)}{\partial Z_t}\right|^{-1} \\
=& \mathcal{P}_Z(F(\eta_t, \tilde{C}_t)|\tilde{C}_t)\,|det J_F(\eta_t)| \\
=& \mathcal{P}_Z(F(\eta_t, \tilde{C}_t)|\tilde{C}_t)\left|det \frac{\partial F(\eta_t, \tilde{C}_t)}{\partial \eta_t}\right|
\end{aligned} \tag{13}$$

where J_G and J_F denote the Jacobian matrices of G and F, respectively. Motivated by the success of the Real-Valued Non-Volume Preserving (Real-NVP) model [35] in efficient Jacobian matrix computation and scalability, the generative space-time feature $Z_t = F(\eta_t, \tilde{C}_t)$ is computed in our model using the following recursive relationship:

$$\begin{aligned}
&Z_t^{(j),1:r-1} = Z_t^{(j-1),1:r-1} \\
&Z_t^{(j),r:\bar{d}} = Z_t^{(j-1),r:\bar{d}} \odot \exp\left(f_\mu^{(j)}([Z_t^{(j-1),1:r-1}; \tilde{C}_t])\right) \\
&+ f_\alpha^{(j)}([Z_t^{(j-1),1:r-1}; \tilde{C}_t])
\end{aligned} \tag{14}$$

where $Z_t^{(j)} = [Z_t^{(j),1:r-1}; Z_t^{(j),r:\bar{d}}]$ represents the column-wise partitioning of $Z_t^{(j)}$, and $Z_t^{(0)} = \eta_t$ is the base case. Here, $f_\mu^{(j)}$ and $f_\alpha^{(j)}$ are the scaling and translation functions of the j-th coupling layer. These functions are implemented as deep ReLU neural networks. Similarly, the attention-enhanced space-time feature is obtainable by $\eta_t = G(Z_t, \tilde{C}_t)$ using the inverse of (14) computed as:

$$\begin{aligned}
&Z_t^{(j-1),1:r-1} = Z_t^{(j),1:r-1} \\
&Z_t^{(j-1),r:\bar{d}} = \exp\left(-f_\alpha^{(j)}([Z_t^{(j),1:r-1}; \tilde{C}_t])\right) \odot (Z_t^{(j),r:\bar{d}} \\
&- f_\mu^{(j)}([Z_t^{(j),1:r-1}; \tilde{C}_t]))
\end{aligned} \tag{15}$$

Given (13), the CNF loss function J_{η_t} is defined using minimum negative log-likelihood estimation to learn the unknown PDF $\mathcal{P}_\eta(Z_t|\tilde{C}_t)$:

$$J_{\eta_t} = -\log\left(\mathcal{P}_Z(F(\eta_t, \tilde{C}_t)|\tilde{C}_t)\right) - \log\left(|\det J_F(\eta_t)|\right) \tag{16}$$

The stochastic gradient descent (SGD) method with the reparameterization trick is employed to minimize J_{η_t} and optimize the tunable parameters of F.

C. Sparse Edge and Node Decoder

To learn the powerful unsupervised features of the input PDS snapshot $\mathcal{G}_{l,t}$, two deep sparse decoders, D_F and D_E, are devised to reconstruct the input node feature X_t and adjacency matrix A_t, respectively. As shown in Fig. 2, following the application of κ successive CNF blocks and the mapping of the attention-enhance space-time feature η_t to a new generative space-time representation Z_t, the reparametrization trick is employed to sample a random variable $Z_t \sim \mathcal{P}_Z(Z_t|\tilde{C}_t)$. Let us denote μ_t and ϵ_t as the mean and standard deviation of the latent variable Z_t provided by the CNF, respectively. Z_t is defined as $Z_t = \mu_t + \epsilon_t \odot r_t$, where r_t represents a random sample drawn from the normal distribution $\mathcal{N}(0, 1)$. The computed latent variable Z_t is then fed into D_F and D_E that are implemented as deep sparse ReLU neural networks with L_F and L_E hidden layers, respectively. Here, D_F outputs the node feature reconstruction $\hat{X}_{l,t}$, and D_E outputs the adjacency matrix reconstruction $\hat{A}_{l,t}$ for each snapshot $\mathcal{G}_{l,t}$. The reconstruction objective functions for D_F and D_E are defined as:

$$\begin{aligned}
J_{D_F} &= \frac{1}{\tau+1}\frac{1}{M}\sum_{l=1}^{M}\sum_{t=\bar{t}}^{\bar{t}+\tau}\left\|X_{l,t} - \hat{X}_{l,t}\right\|_2^2 \\
J_{D_E} &= \frac{1}{\tau+1}\frac{1}{M}\sum_{l=1}^{M}\sum_{t=\bar{t}}^{\bar{t}+\tau}\left\|A_{l,t} - \hat{A}_{l,t}\right\|_2^2
\end{aligned} \tag{17}$$

where $X_{l,t}$ and $A_{l,t}$ show the actual values of node features and adjacency matrix of $\mathcal{G}_{l,t}$. To promote the sparsity of D_F and D_E, a sparsity constraint is introduced on the activation functions of the hidden units in these decoders. Let $\rho_{l,t}^{i,j}$ represent the activation of the i-th hidden unit in the j-th layer of the decoder for the input graph snapshot $\mathcal{G}_{l,t}$. The sparsity loss function of D_F is defined as:

$$\begin{aligned}
J_S^{D_F} &= \sum_j \sum_i \mathrm{KL}(\nu||A^{i,j}) \\
&= \sum_j \sum_i \nu \log(\frac{\nu}{A^{i,j}}) + (1-\nu)\log\frac{1-\nu}{1-A^{i,j}} \\
A^{i,j} &= \frac{1}{\tau+1}\frac{1}{M}\sum_{l=1}^{M}\sum_{t\in[\bar{t},\bar{t}+\tau]}\rho_{l,t}^{i,j}
\end{aligned} \tag{18}$$

Here, the sparsity hyperparameter ν is a small positive value that indicates the desired average activation level for the hidden units of the decoder D_F. The sparsity loss function of D_E denoted by

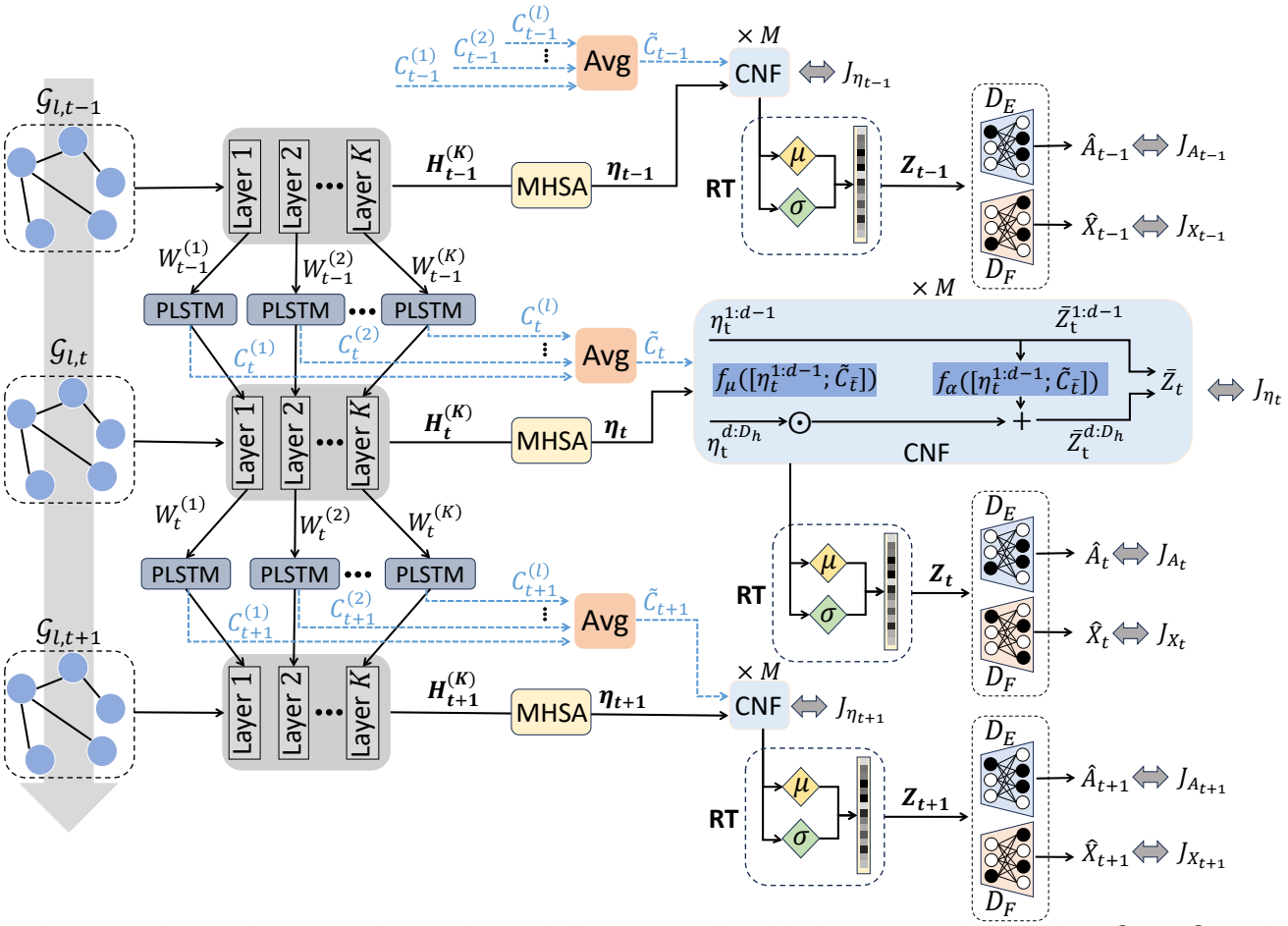


Fig. 2: The proposed space-time generative graph convolution autoencoder with three consecutive snapshots $\mathcal{G}_{l,t-1}, \mathcal{G}_{l,t}$ and $\mathcal{G}_{l,t+1}$.

$J_S^{D_E}$ is similarly defined using (18).

D. Fault Type Classification and Location Estimation

Once the generative space-time feature Z_t corresponding to $\mathcal{G}_{l,t}$ is computed for all time steps $t \in [\bar{t}, \bar{t}+\tau]$, the hidden representation of $\mathcal{G}_l$ is obtained by computing the average of its generative space-time features defined by $\Psi_l = \frac{1}{\tau+1}\sum_{t=\bar{t}}^{\bar{t}+\tau} Z_t$. Subsequently, the resulting Ψ_l is fed into the deep sparse decoders D^{type} and D^{dist} to classify and locate the faults in $\mathcal{G}_l$, respectively. The fault classification and location loss functions for these two neural architectures are defined as:

$$\begin{aligned} J_C &= \frac{1}{M}\sum_{l=1}^{M} \mathcal{L}_{CE}(\mathcal{Y}_l^{type}, \hat{\mathcal{Y}}_l^{type}) \\ J_L &= \frac{1}{M}\sum_{l=1}^{M} \left\| \mathcal{Y}_l^{dist} - \hat{\mathcal{Y}}_l^{dist} \right\|_2^2 \end{aligned} \tag{19}$$

where $\mathcal{L}_{CE}$ denotes the multi-class classification cross entropy loss function [36]. Here, $\hat{\mathcal{Y}}_l^{type}$ and $\hat{\mathcal{Y}}_l^{dist}$ denote the estimated values of $\mathcal{Y}_l^{type}$ and $\mathcal{Y}_l^{dist}$, respectively. Note that, the sparsity loss functions for D^{type} and D^{dist} that are respectively denoted by $J_S^{D^{type}}$ and $J_S^{D^{dist}}$ are defined similar to $J_S^{D_F}$ in (18).

E. Learning and Optimization

This section discusses the training process of the proposed model by minimizing the loss functions defined in (16)-(19) as well as $J_S^{D_E}$, $J_S^{D^{dist}}$, and $J_S^{D^{type}}$. To this end, the total loss function is defined as:

$$\begin{aligned} J_{Total} &= \lambda_R J_R + \lambda_{NF} J_{NF} + \lambda_S J_S + \lambda_F J_F \\ J_R &= J_{D_F} + J_{D_E} \\ J_S &= J_S^{D_F} + J_S^{D_E} + J_S^{D^{type}} + J_S^{D^{dist}} \\ J_F &= J_C + J_L \\ J_{NF} &= \frac{1}{\tau+1}\frac{1}{M}\sum_{l=1}^{M}\sum_{t=\bar{t}}^{\bar{t}+\tau} J_{\eta_{l,t}} \end{aligned} \tag{20}$$

where $J_{\eta_{l,t}}$ is the CNF loss function corresponding to each snapshot $\mathcal{G}_{l,t}$. Here, λ_R, λ_{NF}, λ_S and λ_F are positive constant hyperparameters. To minimize J_{Total}, the gradients of the total loss with respect to the model's parameters are computed, and the SGD method is employed to tune the parameters in an end-to-end fashion.

IV. Numerical Results

A. Data Description and Evaluation Criteria

The performance of the proposed model is evaluated on two datasets generated using electromechanical transient simulations in the modified IEEE 123-bus and IEEE 69-bus systems using DIgSILENT PowerFactory [37] with the sampling interval of 0.01 seconds. In this study, 10 classes associated with 10 fault types are considered, including line-to-ground faults (AG, BG, and CG), line-to-line faults (AB, AC, and BC), a three-phase fault (ABC), and double-line-to-ground faults (ABG, ACG, and BCG).

In the IEEE 69-bus system, the capacity of the DG1 to DG6

are 100, 150, 150, 150, 100, and 100 KVA, respectively. In the IEEE 123-bus system, the electrical loads (three-phase, two-phase, and single-phase) are modeled using ZIP load models (constant impedance, current, and voltage models). Each load is varied using a scale factor derived from a truncated Gaussian density function with a mean of 1 and a standard deviation of 0.05, bounded by 0.90 and 1.10. This network is also equipped with 8 DGs with the capacities of 0.05-0.35 MVA. The faults are applied at 0.1 sec at 0%, 20%, 40%, 60%, and 80% of the line length and cleared after 0.105 seconds. For the IEEE 69-bus system, faults are applied at 0.1 seconds and cleared at 0.205 seconds, for every 10% of the line length from 10%-90%. A total of 26,200 samples for the IEEE 123-bus system and 13,800 samples for the IEEE 69-bus system are collected. To ensure a reliable evaluation of our proposed model against baseline methods, we employ 10-fold cross-validation [38] in this study and report the average performance across all 10 folds as the final performance of the model.

A diverse set of classification metrics [39], including confusion matrix, class-wise accuracy, total accuracy, and macro F1 (ma-F1) score are employed to rigorously evaluate the performance of our model for fault classification compared to the recent benchmarks. The confusion matrix summarizes the model's performance by comparing the predicted and actual labels for each fault type. Each row represents the actual fault type, and each column represents the predicted fault type. For class i, True Positive (TP) samples are the diagonal element M_{ii}, False Positive (FP) samples are the sum of the i-th column excluding M_{ii}, False Negative (FN) samples are the sum of the i-th row excluding M_{ii}, and True Negative (TN) samples are all other elements in the matrix. Class-wise accuracy refers to the accuracy of the model for each individual class. The total accuracy is the average of the model's performance across all fault classes. These two metrics can be computed by:

$$\begin{aligned} \text{Total Acc} &= \frac{1}{C}\sum_{i=1}^{C} ACC_i \\ \text{ACC}_i &= \frac{TP_i}{TP_i + FP_i} \end{aligned} \tag{21}$$

where $C = 10$ is the number of fault classes. The F1 score represents the harmonic mean of precision and recall obtained from the confusion matrix. The ma-F1 score computes the F1 score for each class individually and then averages the results. The ma-F1 is computed by:

$$\begin{aligned} \text{ma-F1} &= \frac{1}{C}\sum_{i=1}^{C} F1_i \\ \text{F1}_i &= 2 \times \frac{P_i \times R_i}{P_i + R_i} \end{aligned} \tag{22}$$

with $P_i = \frac{TP_i}{TP_i+FP_i}$ and $R_i = \frac{TP_i}{TP_i+FN_i}$ denote the precision and recall for ith fault class.

Furthermore, to demonstrate fault location accuracy, a set of regression metrics [40] are employed, including root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) that are computed by:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{M \times N^2}\sum_{l=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(y_{l,i,j}^{dist} - \hat{y}_{l,i,j}^{dist}\right)^2} \\ \text{MAE} &= \frac{1}{M \times N^2}\sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|y_{l,i,j}^{dist} - \hat{y}_{l,i,j}^{dist}\right| \\ \text{MAPE} &= \frac{1}{M \times N^2}\sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{i=1}^{N}\left|\frac{y_{l,i,j}^{dist} - \hat{y}_{l,i,j}^{dist}}{y_{l,i,j}^{dist}}\right| \times 100 \end{aligned} \tag{23}$$

where $y_{l,i,j}^{dist}$ denotes the ith and jth element of the location matrix $\mathcal{Y}_l^{dist}$ corresponding to lth fault sample. Also, M is the number of samples in the test dataset.

B. Hyperparameter Selection and Experimental Settings

In this work, grid search [41] is employed to systematically evaluate the impact of different hyperparameter values on the proposed model's performance. The hyperparameters are varied within predefined ranges, and the configuration yielding the highest F1 score in the validation is considered to be the optimal one. Here, the hyperparameters are loss functions coefficients $\lambda_R, \lambda_S, \lambda_F, \lambda_{NF} \in [0, 1]$, GCN coefficients $\alpha^{(l)} \in [0.1, 0.5]$, and $\beta^{(l)} = \frac{\lambda_\beta^{(l)}}{l}$, with $\lambda_\beta^{(l)} \in [0.5, 1, 1.5, 2]$, number of mapping function in the CNF block $\kappa \in \{3, 5, 7, 9, 11\}$, number of graph convolution layers in encoder $l \in \{4, 8, 14, 32, 64\}$, number of decoders latent layers $N_{D_E}, N_{D_F}, N_{D^{type}}, N_{D^{dist}} \in [2, 6]$, and the decoders' sparsity coefficient $\rho \in [3, 6] \times 10^{-2}$. Fig. 3 illustrates the optimal hyperparameter values $\lambda_F = 0.8$, $\lambda_R = 0.7$, $\lambda_S = 0.3$, $\lambda_{NF} = 0.4$, $\lambda_\beta = 1.5$, $\alpha = 0.2$, $l = 32$, $\kappa = 7$, $N_{D_E,D_F} = 4$, $N_{D^{type},D^{dist}} = 3$ and $\rho = 0.05$. Similar to decoders D_E and D_F, the scaling and translation functions in our CNF are implemented by 4-layer deep ReLU neural networks.

To optimize the defined loss function in (20), the SGD algorithm is employed, incorporating a momentum of 0.95, weight decay of 10^{-4}, and an adaptive learning rate $q_{e+1} = q_e \times (1 - \frac{e}{E})^{0.9}$, where e represents the training iteration number and $E = 3 \times 10^4$ denotes the total training epochs. The initial learning rate is 2×10^{-4}. All experiments were conducted using Python 3.8 with the PyTorch framework [42] on a PC equipped with an Intel Core-i7 CPU, an NVidia Quadro RTX 6000 GPU, and 256 GB of memory.

C. Results

The devised SGGCA is compared with recent data-driven fault detection methods, including wavelet support vector machine (W-SVM) [16], CNN [21], LSTM [43], wavelet CNN (W-CNN) [2], double CNN (DCNN) [19], capsule neural network (Capsule NN) [27], and attention-based CNN-LSTM (ACNN-LSTM) [44]. In addition, recent spatiotemporal and graph-based methods, including GCN [45] and gated graph neural network (GGNN) [25], are considered. We discuss the performance evaluation of the proposed model in both fault classification and location tasks, compared with state-of-the-art methods.

1) Fault Classification Results

Tables I and II present the F1 score and accuracy (ACC) results for the considered baselines and our proposed model across 10

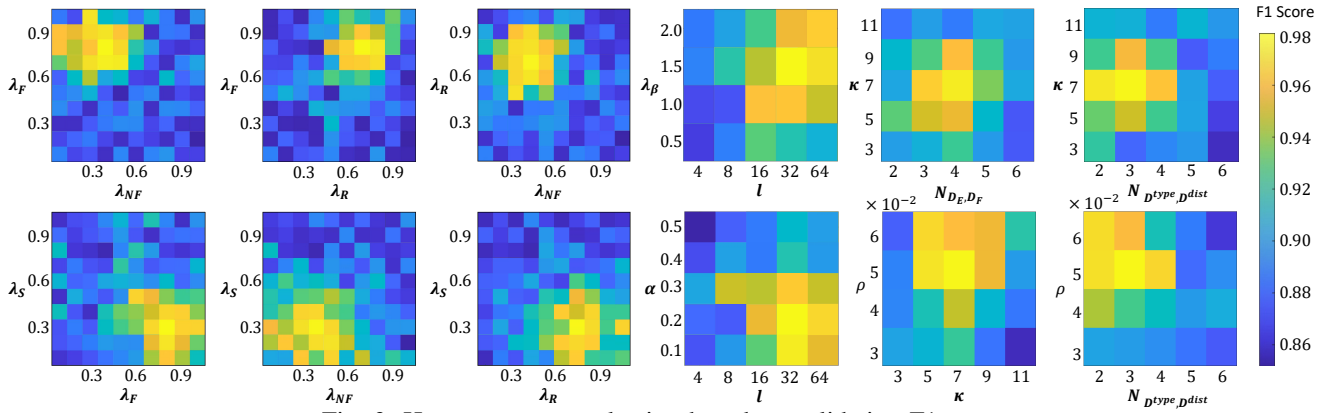


Fig. 3: Hyperparameter selection based on validation F1 score.

TABLE I: Fault classification results of different baselines in the IEEE 123-bus system. The best results are shown in **bold**, and the second best results are underlined.

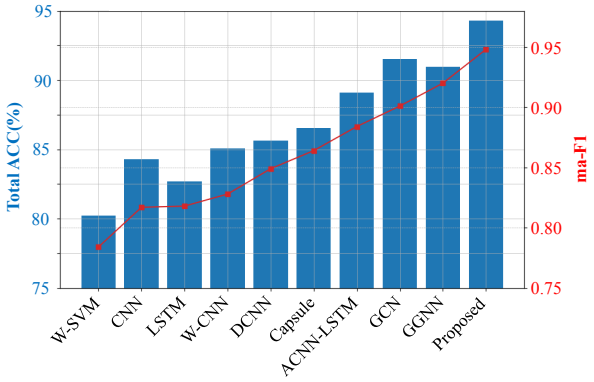
Method	AG		BG		CG		AB		AC		BC		ABG		BCG		ACG		ABC	
	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)
W-SVM	0.765	83.89	0.725	81.56	0.778	82.79	0.689	81.32	0.812	80.43	0.774	81.80	0.761	75.23	0.801	76.39	0.774	80.76	0.772	80.23
CNN	0.783	85.48	0.736	82.33	0.807	86.27	0.861	84.67	0.826	82.67	0.793	86.67	0.818	80.14	0.797	78.34	0.806	79.66	0.801	80.88
LSTM	0.808	88.19	0.769	84.71	0.817	83.98	0.829	84.23	0.889	85.12	0.802	80.10	0.837	83.67	0.785	81.32	0.832	81.36	0.801	78.98
W-CNN	0.850	88.98	0.852	81.34	0.801	82.08	0.836	83.98	0.847	84.98	0.827	84.67	0.815	82.65	0.887	81.43	0.824	83.76	0.813	80.54
DCNN	0.921	89.11	0.864	85.23	0.871	85.43	0.839	87.38	0.842	87.98	0.807	83.12	0.806	84.46	0.808	84.13	0.804	80.87	0.887	82.02
Capsule NN	0.884	89.98	0.846	84.12	0.896	86.34	0.842	87.23	0.874	89.23	0.813	82.11	0.835	86.54	0.836	83.12	0.876	81.42	0.826	83.78
ACNN-LSTM	0.930	89.27	0.881	86.23	0.926	89.23	0.882	89.12	0.868	87.87	0.852	87.03	0.832	91.76	0.903	82.41	0.889	84.09	0.852	87.24
GCN	0.952	90.77	0.881	87.41	0.908	88.47	0.812	90.33	0.920	88.41	0.919	83.28	0.901	90.09	0.883	87.69	0.908	88.46	0.928	90.11
GGNN	0.925	91.34	0.914	89.47	0.846	90.26	0.911	91.64	0.909	89.36	0.915	85.85	0.928	89.56	0.874	91.32	0.916	89.71	0.934	90.78
Proposed	**0.964**	**95.12**	**0.939**	**94.11**	**0.942**	**95.41**	**0.945**	**93.67**	**0.928**	**93.52**	**0.937**	**92.61**	**0.959**	**96.14**	**0.932**	**96.58**	**0.951**	**95.17**	**0.967**	**95.31**

TABLE II: Fault classification results of different baselines in the IEEE 69-bus system. The best results are shown in **bold**, and the second best results are underlined.

Method	AG		BG		CG		AB		AC		BC		ABG		BCG		ACG		ABC	
	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)	F1	ACC(%)
W-SVM	0.770	81.05	0.751	80.33	0.818	78.65	0.793	84.03	0.809	82.67	0.752	78.54	0.772	85.98	0.783	74.27	0.795	75.33	0.794	81.42
CNN	0.783	82.76	0.753	84.32	0.824	82.06	0.812	86.56	0.853	88.02	0.779	83.78	0.802	86.53	0.880	78.79	0.819	80.42	0.865	89.58
LSTM	0.779	81.43	0.763	82.02	0.834	80.63	0.831	82.68	0.842	83.55	0.806	81.67	0.796	87.12	0.855	81.06	0.867	81.43	0.812	85.47
W-CNN	0.804	84.65	0.801	86.12	0.810	82.11	0.801	84.93	0.862	85.09	0.812	85.54	0.821	89.33	0.865	82.78	0.858	84.01	0.847	86.34
DCNN	0.845	84.98	0.802	84.71	0.832	85.16	0.875	85.23	0.881	87.32	0.870	86.98	0.836	89.76	0.889	81.54	0.807	83.11	0.854	87.46
Capsule NN	0.858	81.98	0.836	85.47	0.881	86.42	0.863	86.18	0.903	88.79	0.831	89.49	0.838	90.12	0.891	84.86	0.842	83.65	0.893	88.55
ACNN-LSTM	0.890	86.21	0.837	86.35	0.886	90.45	0.937	89.78	0.902	90.99	0.894	90.31	0.858	94.72	0.923	85.43	0.862	85.01	0.851	91.65
GCN	0.919	90.82	0.854	90.12	0.882	92.98	0.883	94.02	0.936	91.55	0.923	90.01	0.927	96.01	0.883	89.64	0.887	86.32	0.916	93.65
GGNN	0.893	91.46	0.856	91.31	0.923	89.11	0.910	90.98	0.920	93.45	0.904	92.54	0.953	93.13	0.916	89.54	0.925	85.78	0.947	92.33
Proposed	**0.941**	**93.87**	**0.925**	**95.76**	**0.931**	**92.01**	**0.953**	**95.48**	**0.972**	**96.12**	**0.947**	**94.63**	**0.961**	**96.98**	**0.953**	**91.32**	**0.937**	**91.45**	**0.962**	**95.32**

fault categories in the IEEE 123-bus and IEEE 69-bus systems. Additionally, the ma-F1 score and total accuracy of the proposed model and the baselines for each IEEE system are depicted in Fig. 4. As shown in Table I and Fig 4, the W-SVM that employs the wavelet packet decomposition for signal processing and SVM for fault classification and location exhibits the lowest accuracy compared with the deep neural architectures in both datasets. This outcome is attributed to the limitation of wavelet decomposition in extracting significant correlations embedded in the input data. Furthermore, compared to deep neural networks, the SVM shows a less generalization capacity for classifying the intricate patterns inherent in a multi-class classification problem. In Table I, one can observe that the deep CNN model that applies convolutional kernels to input voltages and currents outperforms the W-SVM in both F1 and ACC metrics by 6.96% and 4.91% for ABG fault, respectively. Similar comparison results are observable for the IEEE 69-bus system.

In Fig. 4-(a), the W-CNN that leverages the combined capabilities of wavelet packets and deep convolutional kernels for the analysis of input voltages demonstrates a notable improvement in F1 compared to both CNN and LSTM with 3.83% and 2.16% improvements, respectively. Similarly, obtained results of the IEEE 69-bus system presented in Table II show that the W-CNN outperforms CNN and LSTM by 3.59% and 2.58% in terms of ACC of ACG fault, respectively. This indicates the effectiveness of incorporating wavelet packet decomposition in conjunction with deep convolutional architectures for improved fault detection performance. The results presented in Table I further underscore the advantages of the DCNN approach. Utilizing two separate 1-D CNNs for feature extraction and fault classification, the DCNN achieves a higher classification performance in terms of both F1 and ACC metrics across all fault classes. For example, in the case of the AC fault, the DCNN improves the accuracy of CNN and LSTM by 5.31% and 2.86%, respectively. Also, for the BC fault in the IEEE 69-bus system, the DCNN outperforms CNN and LSTM by 3.2% and 5.31%, respectively. This improved generalization capacity of the DCNN underscores the benefits of employing



(b)

Fig. 4: Classification ma-F1 score and accuracy of the proposed model and benchmarks on (a) IEEE 123-bus system and (b) IEEE 69-bus system

The results presented in Fig. 4 demonstrate the superiority of ACNN-LSTM compared to Capsule NN and DCNN across all fault classes in both IEEE 123-bus and IEEE 69-bus systems. For instance, in the IEEE 123-bus system, the ACNN-LSTM outperforms DCNN and Capsule NN by 2.45% and 2.03% in terms of total accuracy, respectively. Similarly, the results of Table II show the higher performance of ACNN-LSTM compared to DCNN and Capsule NN in the IEEE 69-bus system for each fault class. As an example, for the ABG fault, the ACNN-LSTM improves the ACC of DCNN and Capsule NN architectures by 4.96% and 4.60%, respectively. This higher accuracy is attributed to the inclusion of an attention mechanism within ACNN-LSTM, compelling the model to concentrate on more task-relevant information within the latent space. For the IEEE 123-bus system, the GCN model, which leverages a combination of 1-D and graph-based convolutions for the extraction of spatiotemporal correlations, outperforms both Capsule NN and ACNN-LSTM by 5.32% and 2.10% in F1

TABLE III: Fault location results of different models in the IEEE 123-bus system. The best results are shown in **bold**, and the second best results are underlined.

Model	MAE ($\times 10^{-2}$)	RMSE ($\times 10^{-2}$)	MAPE (%)
W-SVM	29.976	27.324	27.11
CNN	26.121	24.706	23.78
LSTM	24.254	20.867	22.03
W-CNN	24.132	19.852	21.09
DCNN	22.511	18.031	17.01
Capsule NN	20.672	14.569	14.87
ACNN-LSTM	19.812	14.987	13.98
GCN	16.891	11.421	11.75
GGNN	14.842	10.797	10.09
Proposed	**9.578**	**7.511**	**4.36**

TABLE IV: Fault location results of different models in the IEEE 69-bus system. The best results are shown in **bold**, and the second best results are underlined.

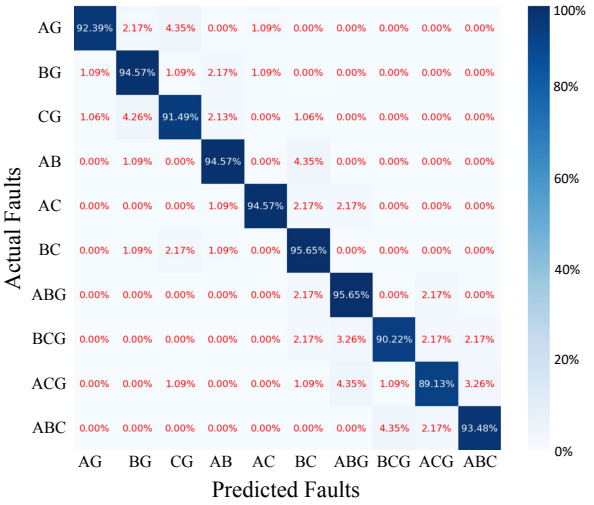
Model	MAE ($\times 10^{-2}$)	RMSE ($\times 10^{-2}$)	MAPE (%)
W-SVM	27.111	26.261	22.32
CNN	22.789	21.812	17.05
LSTM	24.453	22.846	16.78
W-CNN	21.044	19.788	14.89
DCNN	21.862	20.321	14.27
Capsule NN	18.732	16.997	12.22
ACNN-LSTM	13.863	13.583	8.78
GCN	13.082	11.635	9.75
GGNN	10.987	10.763	6.98
Proposed	**7.476**	**6.335**	**3.42**

score, respectively (see Fig 4-(a)). In the IEEE 69-bus system, the GGNN, which integrates graph-based convolution with a gating mechanism using a recurrent neural network, demonstrates superior performance with improvements of 4.41% and 1.87% in total accuracy compared to Capsule NN and ACNN-LSTM, respectively. These improvements are due to the advantages of incorporating both spatial and temporal correlations when analyzing the voltage measurements in PDSs.

Fig. 5 shows the confusion matrices that encapsulate the SGGCA performance in classifying various fault types in the IEEE 123-bus and IEEE 69-bus systems. As illustrated in the figure, the proposed model demonstrates remarkable accuracy across various fault classes. For instance, the proposed SGGCA achieves 96.67% and 93.48% accuracy in the ABC fault classification in the IEEE 123-bus and IEEE 69-bus systems, respectively. For the AC fault in the IEEE 123-bus system and the AB fault in the IEEE 69-bus system, the proposed model achieves 93.89% and 94.57% accuracy, respectively. Additionally, as illustrated in Fig. 4 and Tables I, II, the devised model outperforms the GGNN and GCN baselines, by achieving 4.84% and 6.26% classification accuracy improvements in the IEEE 123-bus system, respectively. Similar improvements in the classification accuracy of the devised model are observable in the IEEE 69-bus system. These results affirm the SGGCA model's superior classification accuracy across various fault classes, underscoring its overall superior performance compared to the recent benchmarks.

2) Fault Location Results

Tables III and IV display the performance of the proposed model



(b)

Fig. 5: Confusion matrices of the proposed model for the 10 fault classes of the (a) IEEE 123-bus system and (b) IEEE 69-bus system

and recent baselines in fault location for the IEEE 123-bus and IEEE 69-bus systems, respectively. Notably, the W-SVM model emerges as the weakest model among the considered baselines, achieving MAE and MAPE of 29.976×10^{-2} and 27.11% in the IEEE 123-bus system, respectively. Consistent with fault classification results, convolutional-based approaches exhibit superior performances in capturing spatial correlations within the input data when compared to W-SVM. For instance, in the IEEE 69-bus system, the W-CNN surpasses W-SVM by 6.07×10^{-2} and 7.43% in MAE and MAPE, respectively. As shown in Table IV, leveraging a double convolutional-based encoder for capturing spatial correlations, the DCNN further enhances location accuracy over W-CNN by 3.74% and 2.6% in terms of MAE and RMSE, respectively. Also, in the case of the IEEE 123-bus system, the Capsule NN demonstrates a higher prediction accuracy with a 2.23% improvement in MAPE compared with DCNN, attributed to its capabilities in handling hierarchical relationships and spatial hierarchies in the data.

As shown in tables III and IV, the ACNN-LSTM that leverages attention techniques within its latent space achieves

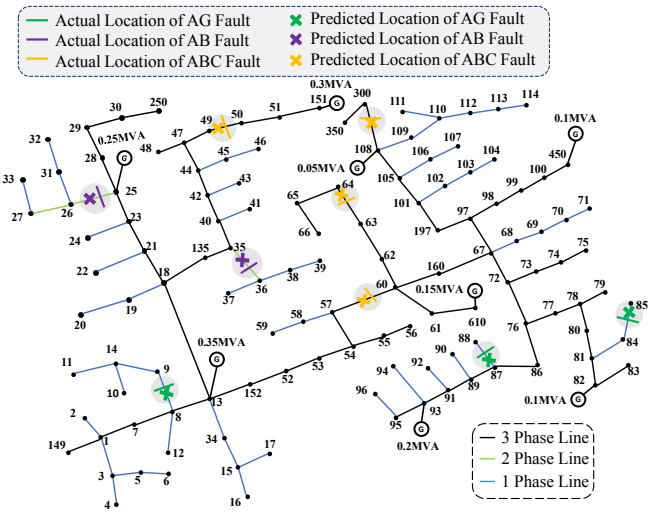


Fig. 6: Fault location results of the proposed model with different faults in the IEEE 123-bus system.

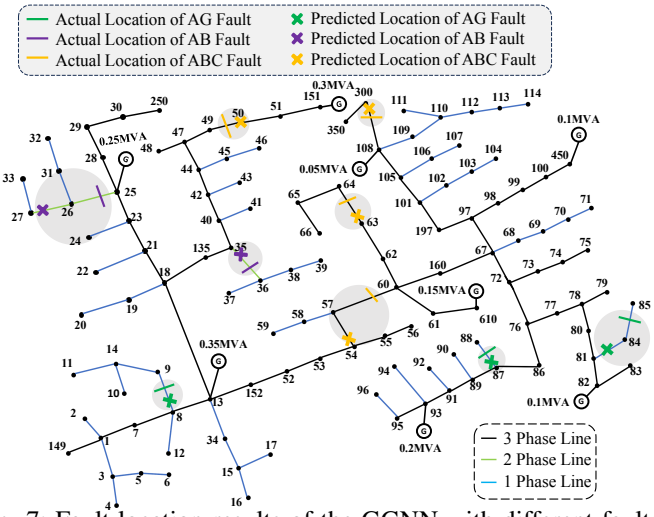


Fig. 7: Fault location results of the GGNN with different faults in the IEEE 123-bus system.

superior fault location performance compared to Capsule NN. For instance, in terms of MAPE in both IEEE 123-bus and IEEE 69-bus systems, the ACNN-LSTM improves the fault location estimation of Capsule NN by 0.89% and 3.44%, respectively. Table III shows the higher performance of the graph-based models compared with recent benchmarks in fault location. As shown in this table, the GCN enhances the MAE of both ACNN-LSTM and Capsule NN by 0.030 and 0.038, respectively. In the IEEE 123-bus system, the GGNN shows a slight 0.020 improvement in the MAE of the GCN, leveraging its gating mechanism to capture long-term dependencies and spatial correlations more effectively. Similar improvements are observable in the fault location estimation for the IEEE 69-bus system presented in Table IV. These enhancements extend to other defined metrics, reinforcing the effectiveness of graph-based approaches in fault location.

As presented in Tables III and IV, the proposed SGGCA outperforms graph-based models (i.e., GCN and GGNN) as well as other data-driven baselines. For instance, with respect to MAPE, the SGGCA outperforms GGNN and ACNN-LSTM by 5.73% and 9.62% in the IEEE 123-bus system, respectively. Also, in the IEEE 69-bus system, the devised SGGCA

TABLE V: Ablation study of the proposed model in the IEEE 123-bus and IEEE 69-bus systems. Here, ACC and RMSE are used to evaluate fault classification and location tasks, respectively.

Architecture	ξ	CNF	GCN	MHSA	IEEE 123-bus System		IEEE 69-bus System	
					Classification	Location	Classification	Location
A0	PLSTM	✓	Eq. (2)	✓	95.11	0.7327	94.78	0.6234
A1	PLSTM	✓	Eq. (2)	✘	94.95	0.7529	94.50	0.6498
A2	PLSTM	✓	Eq. (1)	✓	91.25	0.7983	89.90	0.7001
A3	PLSTM	✓	Eq. (1)	✘	90.95	0.8351	87.62	0.7470
A4	PLSTM	✘	Eq. (2)	✓	92.02	0.7805	90.70	0.7150
A5	PLSTM	✘	Eq. (2)	✘	91.80	0.8020	88.40	0.7312
A6	PLSTM	✘	Eq. (1)	✓	88.18	0.8569	84.85	0.8190
A7	PLSTM	✘	Eq. (1)	✘	83.92	1.0542	80.55	1.1262
A8	LSTM	✓	Eq. (2)	✓	92.10	0.8010	90.75	0.7027
A9	LSTM	✓	Eq. (2)	✘	94.82	0.8315	89.45	0.7585
A10	LSTM	✓	Eq. (1)	✓	90.20	0.8695	86.88	0.7715
A11	LSTM	✓	Eq. (1)	✘	88.93	1.0368	85.58	0.9279
A12	LSTM	✘	Eq. (2)	✓	89.05	0.9922	86.72	0.8042
A13	LSTM	✘	Eq. (2)	✘	88.78	0.9130	85.42	0.8905
A14	LSTM	✘	Eq. (1)	✓	86.15	1.0780	82.87	0.9307
A15	LSTM	✘	Eq. (1)	✘	80.90	1.2355	79.55	1.0667

architecture improves the MAPE of GGNN and GCN by 3.56% and 6.33%, respectively. Similar improvements are observed in comparison with other baselines using MAE and RMSE measures. Figures 6 and 7 visually present the location results achieved by our proposed SGGCA and the GGNN for nine random test samples of AG, AB, and ABC fault classes in the IEEE 123-bus system. Similarly, Figures 8 and 9 illustrate the fault location results for the proposed SGGCA and GGNN in the IEEE 69-bus system. As shown in these figures, the distance between the actual and the estimated locations of the faults obtained by the proposed SGGCA is lower compared to the GGNN model in both systems. The robust performance of the SGGCA model in both fault classification and location tasks can be attributed to its incorporation of deep spatiotemporal features through our novel deep convolutional graph encoder equipped with early connections and identity transformation techniques, enhancing the model's capacity to capture intricate fault patterns. Additionally, the devised encoder is equipped with the multi-head self-attention mechanism and CNF, enabling effective learning of generative correlations within the PDS data. These advancements collectively establish the SGGCA model as a promising and effective approach for fault location.

D. Ablation Study

An ablation study is carried out to evaluate the contribution of each module to the overall performance of the proposed approach. Specifically, certain components are either removed or substituted, and the resulting changes in fault classification and localization accuracy are analyzed using the IEEE 123-bus and IEEE 69-bus systems. We examine four fundamental factors, including the type of recurrent architecture that determines GCN weights (i.e., ξ in (16)), GCN formulation, CNF, and MHSA. Table V shows the validation fault classification and location corresponding to the designed ablation study for the IEEE 123-bus and IEEE 69-bus datasets. In Table V, the symbols ✓ and ✘ are used to denote the presence or absence of the original components within our architecture, respectively. For instance, architecture A1 utilizes PLSTM (as defined in (16)) to determine the weights of the GCN formulation in (2). This model includes the CNF to capture unsupervised generative features and removes the MHSA block. A0, on the other hand, represents the proposed model, which includes all modules.

As shown in the table, employing the deep GCN convolution devised in (2) results in significantly higher accuracy in both fault classification and regression tasks. By comparing A2 and A0 architectures, one can observe that A0, which utilizes GCN with identity transformation and early connection techniques, outperforms A2 by 3.86% and 4.88% in fault classification accuracy in the IEEE 123-bus and IEEE 69-bus systems,

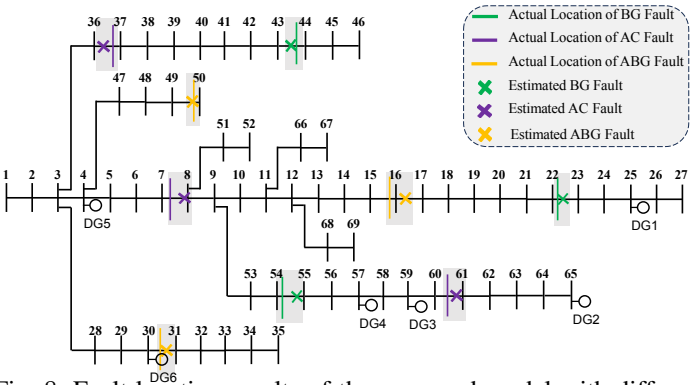


Fig. 8: Fault location results of the proposed model with different faults in the IEEE 69-bus system.

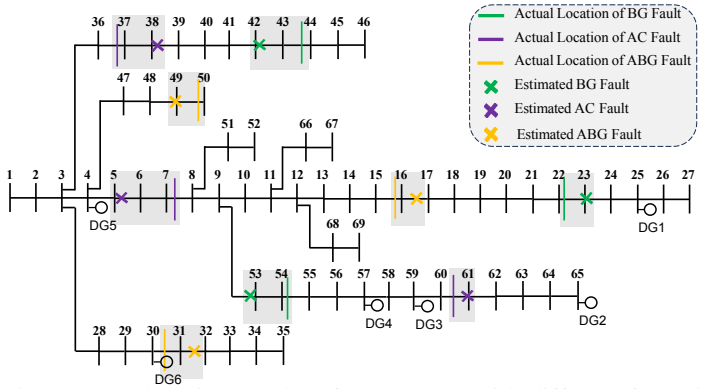


Fig. 9: Fault location results of the GGNN with different faults in the IEEE 69-bus system.

respectively. By cc
architecture A0 and
and its correspondi
loss J_η degrades th
6.12% in fault cla
IEEE 123-bus sy
importance of the
learning module in
is observable in th
PLSTM with LSTM
base model's perfo
and 3.01% in fault
IEEE 123-bus syste
generalization capac
the LSTM model in
snapshots. Addition
shows a 2.75% hig
IEEE 123-bus syste
models that includ
task-relevant feature
which does not in
lowest performance
classification and lo
a 9.30% and 14.0
comparing the clas
A15, respectively. T
importance of the c
classification and lo
the proposed RCI
task-relevant featu
challenges, Fig. 10
latent feature representations. These visualizations correspond to 1,000 randomly sampled observations from various fault types, including AG, CG, AB, BC, and ABC, from the validation set of the IEEE 123-bus system dataset. Fig. 10 (a) and (b) depict t-SNE visualization of the deep latent representations generated by the SGGCE when the RCIT graph convolution is substituted with a vanilla GCN, whereas Fig. 10 (c) and (d) present the latent representation generated by the proposed SGGCE. As shown in the figure, the vanilla GCN suffers from the over-smoothing problem, where increasing the number of convolutional layers l results in an indistinguishable latent feature space. In contrast, the proposed RCIT graph convolution operator, which incorporates residual connections and identity transformation techniques, effectively maps the raw input nodal features into an informative and distinguishable latent representation tailored for the underlying task.

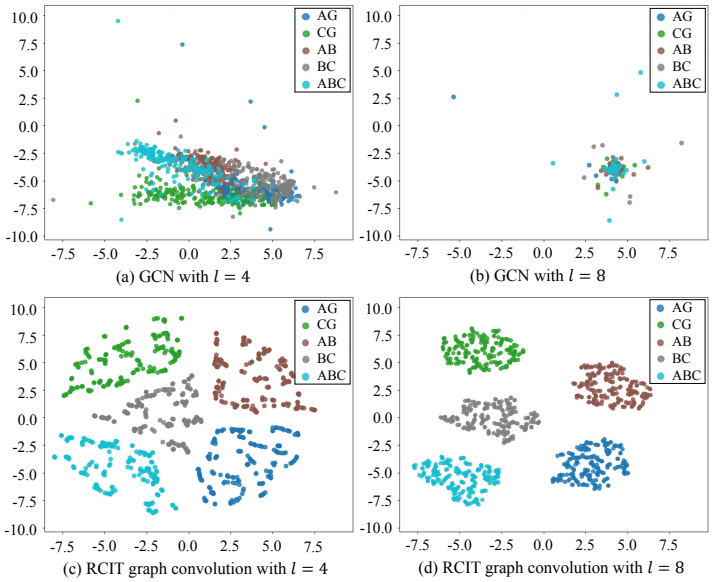


Fig. 10: Visualization of deep latent features obtained from SGGCE with (a) 4-layer vanilla GCN, (b) 8-layer vanilla GCN, (c) 4-layer RCIT convolution operator, and (d) 8-layer RCIT convolution operator.

V. Conclusion

This paper introduces a spatiotemporal approach to enhance fault classification and location in PDSs. In our framework, the PDS measurements are first modeled as a weighted space-time graph over a short time interval following a fault incident, where each graph snapshot includes edge-based and bus-phase-based features of the PDS. Our model introduces novel graph convolution layers that incorporate early connection and identity transformation techniques to effectively uncover deep correlations within the spatiotemporal snapshots of the PDS graph. Significantly, these convolution layers address and reduce the over-smoothing problems typically faced by graph-based models, demonstrating their robustness in capturing complex spatial patterns. Additionally, a recurrent Peephole LSTM architecture is utilized to dynamically adjust the parameters of the convolutional layers, focusing on the evolving structure of the graphs at each time step. A novel CNF architecture is devised to learn the latent PDF of the space-time graph representation to capture unsupervised generative features that further enhance the model's understanding of the complex input measurements. Extensive experiments on the IEEE 123-bus and IEEE 69-bus systems validate the effectiveness of our proposed model, demonstrating superior performance in fault classification and location tasks on both datasets.

This research can be further extended by exploring the application of the proposed model to PDS fault analysis with inverter-based resources (IBRs) such as solar, wind, and energy storage technologies. In these scenarios, the dynamic behavior of IBRs and their low contributions to the fault current introduce additional complexity in fault detection and classification. Furthermore, in more challenging scenarios, it would be beneficial to address fault classification and location in the presence of incomplete or partially observed dynamic graphs of the PDS. This involves designing robust algorithms that can infer missing data and maintain high accuracy in identification despite data sparsity. Moreover, future work could consider the deployment of these models in a real-time operational environment. This requires optimizing the computational efficiency and scalability of the models to ensure that they can handle the large volume of data generated by PDSs. Incorporating edge computing technologies could facilitate rapid data processing and decision-making at the edge of the network, further enhancing the reliability and resilience of PDSs.

REFERENCES

[1] X. Xu, F. Zhou, Y. Nie, W. Xu, K. Wang, J. OuYang, K. Zhou, S. Chen, and Y. Han, "Fault detection and location of 35 kv single-ended radial distribution network based on traveling wave detection method," *Processes*, vol. 11, no. 8, p. 2494, 2023.

[2] V. Rizeakos, A. Bachoumis, N. Andriopoulos, M. Birbas, and A. Birbas, "Deep learning-based application for fault location identification and type classification in active distribution grids," *Applied Energy*, vol. 338, p. 120932, 2023.

[3] S. S. Gururajapathy, H. Mokhlis, and H. A. Illias, "Fault location and detection techniques in power distribution systems with distributed generation: A review," *Renewable and sustainable energy reviews*, vol. 74, pp. 949–958, 2017.

[4] S. Das, S. Santoso, A. Gaikwad, and M. Patel, "Impedance-based fault location in transmission networks: theory and application," *IEEE access*, vol. 2, pp. 537–557, 2014.

[5] R. Dashti, M. Ghasemi, and M. Daisy, "Fault location in power distribution network with presence of distributed generation resources using impedance based method and applying π line model," *Energy*, vol. 159, pp. 344–360, 2018.

[6] W.-J. Yang, X.-Q. Yin, J. Tao, and H.-Y. Zhang, "Fault current constrained impedance-based method for high resistance ground fault location in distribution grid," *Electric Power Systems Research*, vol. 227, p. 109998, 2024.

[7] A. Keshavarz, R. Dashti, M. Deljoo, and H. R. Shaker, "Fault location in distribution networks based on svm and impedance-based method using online databank generation," *Neural Computing and Applications*, pp. 1–17, 2022.

[8] O. Naidu and A. K. Pradhan, "A traveling wave-based fault location method using unsynchronized current measurements," *IEEE Transactions on Power Delivery*, vol. 34, no. 2, pp. 505–513, 2018.

[9] H. Cui, Q. Yang, J. Sun, T. Zhou, and Y. He, "Electromagnetic time reversal fault location method based on characteristic frequency of traveling waves," *IEEE Transactions on Power Delivery*, vol. 38, no. 5, pp. 3033–3044, 2023.

[10] E. R. Sanseverino, V. L. Vigni, A. Di Stefano, and R. Candela, "A two-end traveling wave fault location system for mv cables," *IEEE Transactions on Industry Applications*, vol. 55, no. 2, pp. 1180–1188, 2018.

[11] A. Rafinia and J. Moshtagh, "A new approach to fault location in three-phase underground distribution system using combination of wavelet analysis with ann and fls," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 261–274, 2014.

[12] N. Ahmed, A. A. Hashmani, S. Khokhar, M. A. Tunio, and M. Faheem, "Fault detection through discrete wavelet transform in overhead power transmission lines," *Energy Science & Engineering*, vol. 11, no. 11, pp. 4181–4197, 2023.

[13] N. A. Tunio, A. A. Hashmani, S. Khokhar, M. A. Tunio, and M. Faheem, "Fault detection and classification in overhead transmission lines through comprehensive feature extraction using temporal convolution neural network," *Engineering Reports*, p. e12950, 2024.

[14] A. L. da Silva Pessoa and M. Oleskovicz, "Fault location in radial distribution systems based on decision trees and optimized allocation of power quality meters," in *2017 IEEE Manchester PowerTech*, pp. 1–6, IEEE, 2017.

[15] F. Wilches-Bernal, M. Jiménez-Aparicio, and M. J. Reno, "An algorithm for fast fault location and classification based on mathematical morphology and machine learning," in *2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, IEEE, 2022.

[16] K. Moloi, N. W. Ndlela, and I. E. Davidson, "Fault classification and localization scheme for power distribution network," *Applied Sciences*, vol. 12, no. 23, p. 11903, 2022.

[17] M. Dashtdar, R. Dashti, and H. R. Shaker, "Distribution network fault section identification and fault location using artificial neural network," in *2018 5th international conference on electrical and electronic engineering (ICEEE)*, pp. 273–278, IEEE, 2018.

[18] P. Rai, N. D. Londhe, and R. Raj, "Fault classification in power system distribution network integrated with distributed generators using cnn," *Electric Power Systems Research*, vol. 192, p. 106914, 2021.

[19] M. Zou, Y. Zhao, D. Yan, X. Tang, P. Duan, and S. Liu, "Double convolutional neural network for fault identification of power distribution network," *Electric Power Systems Research*, vol. 210, p. 108085, 2022.

[20] K. D. Khattak, M. A. Choudhry, and A. Feliachi, "Fault classificaton and location in power distribution networks using 1d cnn with residual learning," in *2024 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, IEEE, 2024.

[21] M. Zhao and M. Barati, "A real-time fault localization in power distribution grid for wildfire detection through deep convolutional neural networks," *IEEE Transactions on Industry Applications*, vol. 57, no. 4, pp. 4316–4326, 2021.

[22] J. B. Thomas, S. G. Chaudhari, K. Shihabudheen, and N. K. Verma, "Cnn-based transformer model for fault detection in power system networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023.

[23] P. K. Shukla and K. Deepa, "Deep learning techniques for transmission line fault classification–a comparative study," *Ain Shams Engineering Journal*, vol. 15, no. 2, p. 102427, 2024.

[24] G. Luo, Y. Tan, M. Li, M. Cheng, Y. Liu, and J. He, "Stacked auto-encoder-based fault location in distribution network," *IEEE Access*, vol. 8, pp. 28043–28053, 2020.

[25] J. T. de Freitas and F. G. F. Coelho, "Fault localization method for power distribution systems based on gated graph neural networks," *Electrical Engineering*, vol. 103, no. 5, pp. 2259–2266, 2021.

[26] K. Chen, J. Hu, Y. Zhang, Z. Yu, and J. He, "Fault location in power distribution systems via deep graph convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 1, pp. 119–131, 2019.

[27] S. R. Fahim, S. K. Sarker, S. Muyeen, S. K. Das, and I. Kamwa, "A deep learning based intelligent approach in detection and classification of transmission line faults," *International Journal of Electrical Power & Energy Systems*, vol. 133, p. 107102, 2021.

[28] J. Hu, W. Hu, J. Chen, D. Cao, Z. Zhang, Z. Liu, Z. Chen, and F. Blaabjerg, "Fault location and classification for distribution systems based on deep graph learning methods," *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 1, pp. 35–51, 2022.

[29] F. Chen, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, "Graph representation learning: a survey," *APSIPA Transactions on Signal and Information Processing*, vol. 9, p. e15, 2020.

[30] D. P. Kingma, M. Welling, *et al.*, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[31] M. Saffari, M. Khodayar, S. M. J. Jalali, M. Shafie-khah, and J. P. Catalão, "Deep convolutional graph rough variational auto-encoder for short-term photovoltaic power forecasting," in *2021 International Conference on Smart Energy Systems and Technologies (SEST)*, pp. 1–6, IEEE, 2021.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[33] T. K. Rusch, M. M. Bronstein, and S. Mishra, "A survey on oversmoothing in graph neural networks," *arXiv preprint arXiv:2303.10993*, 2023.

[34] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," *arXiv preprint arXiv:1905.10947*, 2019.

[35] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[36] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[37] DIgSILENT, "Powerfactory - digsilent," 2024.

[38] D. Berrar *et al.*, "Cross-validation.," 2019.

[39] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.

[40] M. Saffari, M. Khodayar, and M. E. Khodayar, "Deep recurrent extreme learning machine for behind-the-meter photovoltaic disaggregation," *The Electricity Journal*, vol. 35, no. 5, p. 107137, 2022.

[41] M. Saffari, M. Khodayar, M. E. Khodayar, and M. Shahidehpour, "Behind-the-meter load and pv disaggregation via deep spatiotemporal graph generative sparse coding with capsule network," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[43] S. Belagoune, N. Bali, A. Bakdi, B. Baadji, and K. Atif, "Deep learning through lstm classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems," *Measurement*, vol. 177, p. 109330, 2021.

[44] S. Zhang, J. Duan, Y. Li, J. Chen, and J. Zhao, "Anomaly detection of power information system based on attention mechanism cnn-lstm," in *2023 2nd International Conference on Big Data, Information and Computer Network (BDICN)*, pp. 154–157, IEEE, 2023.

[45] B. L. Nguyen, T. Vu, T.-T. Nguyen, M. Panwar, and R. Hovsapian, "1-d convolutional graph convolutional networks for fault detection in distributed energy systems," in *2022 IEEE 1st Industrial Electronics Society Annual On-Line Conference (ONCON)*, pp. 1–6, IEEE, 2022.

[46] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.