Spatial-Temporal Graph Convolutional Network for Fault Diagnosis in Weak Electrical Systems

Jing Shen[1,*] , Chi Wu [2] , Yizong Dai [1]
[1]School of Electrical Engineering and Automotive Engineering, Yangzhou Polytechnic University, Yangzhou, Jiangsu, 225009, China
[2]Yangzhou Optoelectronic Product Testning Co., Ltd, Yangzhou, Jiangsu, 225009, China
E-mail: shenjing1994@hotmail.com

[*]Corresponding author

Keywords: graph neural network, failure diagnosis, weak electrical system, fault detection, smart grid

Received: July 9, 2025

Weak electrical systems, such as low-voltage distribution grids and embedded sensor networks, are highly susceptible to faults due to their complex topology and limited fault tolerance. Accurate failure analysis and diagnosis in such systems are essential for maintaining operational reliability and safety. However, traditional diagnostic methods—such as rule-based systems or shallow machine learning—struggle with non-linear relationships, dynamic system behavior, and distributed component interactions. These limitations result in delayed or inaccurate fault detection, particularly in noisy or rapidly changing environments. To overcome these challenges, this paper proposes a failure diagnosis framework based on the Spatial-Temporal Graph Convolutional Network (ST-GCN), comprising a multi-channel CNN feature extractor for spatial pattern learning and an attention-guided temporal module for capturing temporal dependencies across system nodes. The architecture allows the model to learn complex spatiotemporal interactions and adapt to multi-modal sensor inputs effectively. The proposed ST-GCN achieves 96.7% accuracy, 0.95 F1-score, and 93.6% fault localization accuracy, significantly outperforming traditional methods. It also demonstrates sub-10 ms detection latency, 95.4% actual positive rate in the confusion matrix, and a Precision–Recall AUC of 0.96, while converging within 25 epochs and showing only 1.4% accuracy drop when scaled from a 33-bus to a 123-bus system. These results highlight the robustness, real-time applicability, and methodological effectiveness of ST-GCN for fault diagnosis in weakly meshed and low-voltage distribution grids.

Povzetek: Članek predlaga diagnostični okvir ST-GCN, ki z učenjem prostorsko-časovnih odnosov iz večmodalnih senzorskih podatkov omogoča hitro (pod 10 ms) in natančno (≈96,7%) zaznavanje ter lokalizacijo okvar v šibkih nizkonapetostnih omrežjih, pri čemer ostane robusten tudi pri večjem obsegu sistema.

1 Introduction

1.1 Overview to weak electrical systems

Weak electrical systems refer to power distribution infrastructures with little fault tolerance, low-voltage-operated systems, and are highly vulnerable to disturbances. Such systems are commonly found in distributed systems, such as microgrids, smart buildings, and industrial sensor arrays [1]. Weak distribution grids have complex topologies and dynamic behavior, making conventional fault detection unreliable. This paper uses a heterogeneous multi-task GNN to capture spatial dependencies and perform fault detection, location, classification, and parameter estimation. Results show robustness to measurement errors, topology changes, and variable fault conditions, highlighting the need for advanced graph-based fault analysis [2].

1.2 Importance of fault analysis in smart grids

With smart grids, advancements in architecture, and fault analysis have come to the forefront of everything. Proper diagnosis of faults leads to uninterrupted equipment operation, reduced maintenance expenses, and energy security [3]. Distributed energy resources introduce subtle fault signatures and noise in weak grids, challenging traditional detection methods. A GNN-based model encodes both topology and electrical-physical information to enable robust fault diagnosis under weak or incomplete measurements. Experiments demonstrate superior accuracy over classical approaches, emphasizing the importance of GNNs for reliable smart grid fault analysis [4].

1.3 Graph-centric perspective on electrical topologies

Graphs are effective models of electrical systems, in which nodes typically represent physical components, and edges represent either electrical connections or logical connections [5]. This type of graph-based abstraction considers both spatial layout and signal variant dependencies, facilitating an intelligent analysis. Using Graph Neural Networks (GNNs), it can be expected that the network's topology can be influenced by the learning models, allowing localized faults to be detected and contextual predictions to be made in conjunction with real-time monitoring of the related components [6].

1.4 Motivation

With the global tendency to decentralise and digitalise power systems, weak electrical systems (microgrids, distribution feeders, sensor-fueled energy networks) become critical in fulfilling the requirements of localised control and in-time responsiveness [7]. They are implanted in essential infrastructure and smart environments and may be involved in quite complex interactions with hundreds or thousands of low-power components. Yet, their weak properties and low redundancy predispose them to disruptions, which is why smart diagnostic measures are required that would assure safety, stability, and self-healing [8].

1.5 Challenges in weak electrical systems

- Granted, most traditional methods of diagnosis in a weak electrical system have several limitations:
- Non-static and dynamic nature: Random noise in the signal, dynamic loads, and external disturbances degrade the performance of any static, threshold-based approach.
- Distributed fault propagation: The faults may be initiated in a single node and then propagated in a way that makes it hard to localise [9].
- Data sparsity and imbalance: Few fault samples are used, and class imbalance is detrimental to a typical machine learning approach.
- Topology dependency: Many current algorithms overlook the electrical structure and thus treat sensor data as flat vectors, thereby eliminating all significant spatial associations [10].

1.6 Contributions of this paper

In response to these shortcomings, the paper proposes a new failure analysis and diagnosis model for the weak electrical system, starting with the ST-GCN. The essential contributions are:

· Graphical representation of electrical systems, using topological and signal relations.

· ST-GCN architecture activated with the joint learning of spatial and temporal relations of fault signatures.

· It was experimentally validated on both simulated and real-world datasets, achieving higher accuracy, localization precision, and real-time diagnostic power than the baseline ML methods.

The following are the key contributions of this paper

- Graphical modeling of weak electrical systems, capturing both topological connections and signal relationships.
- Development of a Spatial-Temporal Graph Convolutional Network (ST-GCN) that jointly learns spatial and temporal fault patterns.
- Comprehensive fault diagnosis including detection, localization, classification, and parameter estimation.
- Experimental validation demonstrating high accuracy, precise localization, and real-time performance on simulated and real-world datasets.
- Robustness to noisy, sparse, and dynamic data, ensuring scalability across varying grid topologies and sizes.

2 Related work

2.1 Conventional diagnostic techniques

Traditional fault diagnosis methods in electrical systems are centered on machine learning models trained on statistical or frequency features. GBT, MLP, and SVM are among the models that have gained widespread use due to their high fault-detection and classification properties. Nonetheless, they tend to perform worse when they are in the form of real-time, high-dimensional, or topologically complicated grids.

2.1.1 Gradient Boosting Tree (GBT) Model

In this research, Sapountzoglou et al [11] examined the use of the GBT algorithm to detect and localize single-phase-to-ground and three-phase faults in low-voltage (LV) Smart Grids. This model is trained on branch-free features, making it generalizable across various grid topologies. Tested on a simulated Portuguese LV network subject to different fault resistances, times, and locations, the procedure realized a peak error in the fault detection of only 0.72%, indicating a robust cross-topology flexibility and a high diagnostic precision.

2.1.2 Multi-Layer Perceptron (MLP) model

The paper presented by Yan et al [12] here is dedicated to an MLP-based diagnostic model for low-voltage circuit breakers, featuring type-specific entropy-based features. Other entropies are combined using PCA to create a robust feature space. An optimized MLP model, along with well-established hyperparameters and sensor input, is recorded to achieve a diagnostic accuracy of over 98%. The method can be effectively used to detect faults such as false closing and jamming, with a high potential for practical breaker health monitoring.

2.1.3 Support Vector Machine (SVM)

Liu et al [13] state that to address issues with diagnosing internal faults in distribution transformers, the work aims to implement support vector machines (SVMs) using frequency response data collected during impulse tests. Using the oscillation patterns of the winding throughout the end of the winding, the method identifies the location of insulation collapse with an accuracy of more than 80%. It shuns intrusive examination and proves that the features of external electric waveforms may be utilized to diagnose internal errors in the transformer, and is one factor that advances safer and more effective service to transformers that do not require mechanical breakdown.

2.1.4 Variational Mode Decomposition (VMD) + Ensemble Bagged Trees Model (EBTM)

In this study, VMD is integrated with the EBTM, which is used to detect faults in photovoltaic-rich distribution networks by Nsaif et al [14]. Dynamic protection and high-impedance faults are resolved using a method that decomposes voltage sequences and classifies the faults. It has a better result compared to SVM and other ML models, and under the radial and SNOP topologies, it gets a diagnostic accuracy of 100% within 1.25 ms. The model is locally dependent and communication-free; therefore, it is suitable for real-time and distributed smart grid protection.

2.1.5 Random Forest (RF)

This paper applies RF in grids of limited observability to estimate the voltages of buses using CATV sensor measurements, as described by Markovic et al. [15]. The process harnesses the spatio-temporal measurements of the popular CATV sensor to deduce the voltage values of unmonitored buses. The model demonstrated high accuracy on a 1572-bus SMART-DS, including both passive and PV-active networks. The method enhances the observability of systems without adding extra infrastructure to increase the hosting capacity of distributed renewable energy resources.

2.2 Machine learning approaches

The more advanced machine learning methods, including NCFS, XGBoost-SVM, GANs, and Neural Architecture Search, have been recently studied to enhance the automation, accuracy, and flexibility of fault diagnosis. Such models offer improvements in terms of data selection, feature learning, and robustness, yet struggle with issues of interpretability, scalability, and deployment at the edge.

2.2.1 Neighborhood Component Feature Selection (NCFS) and SVM

This paper is presented by Mirshekali et al [16] a fault localization model utilizing Neighborhood Component Feature Selection (NCFS) and Support Vector Machine (SVM) techniques. Voltage measurements taken by the micro-PMU at fault conditions are converted into frequency-domain characteristics and narrowed down with NCFS such that only the most significant elements are preserved. The SVM classifier is applied to find the faulty part on a DG-equipped IEEE 11-node feeder. The model remains effective even in cases where the behavior of the DG and the randomness of faults are unknown, with a high score in fault section classification.

2.2.2 Combined Extreme Gradient Boosting and Support Vector Machine model (XGBoost+SVM)

According to Liu et al. [17], a hybrid approach combining Extreme Gradient Boosting (XGBoost) and SVM is proposed in the current paper for fault location in distribution networks. The model has been tested using an IEEE 34-bus power system with single-phase-to-ground faults using node voltages to determine fault location. This technique is more accurate and precise, and, in terms of F1-score and time complexity, it outperforms KNN and MLP, which suggests its viability as a tool in real-time fault diagnosis applications for power systems.

2.2.3 Decision Tree-based fault detection and classification (DT-FDC)

The research paper proposes a Decision Tree (DT) based model for intelligent Fault Detection and Classification (FDC) in transmission networks, as stated by Venkatachalam et al. [18]. It utilizes the PMU data of the WSCC 9-bus test system to perform simulations on various types of faults by adjusting parameters such as resistance, location, and angle. The DT receives feeds of post-fault bus voltages, current, and angles to classify them. The methodology guarantees proper and effective FDC operation and optimal positioning of PMU in a wide-area monitoring system.

2.2.4 Generative Adversarial Network (GAN)

This paper aims to address the issue of class imbalance in fault diagnosis by proposing a Generative Adversarial Network (GAN) coupled with Feature Matching (FM) by Zareapoor et al [19]. The GAN uses a mixture of both normal and faulty data as a distribution to create synthetic samples of the faults, and the discriminator classifies both real and synthetic ones. The approach outperforms oversampling techniques in effectively increasing detection accuracy and robustness against outliers and overfitting in industrial fault diagnosis.

2.2.5 Neural Architecture Search (NAS)

According to Li et al [20], to streamline the deep model design due to faults, this study utilizes Neural Architecture Search (NAS) and Reinforcement Learning (RL). A recurrent network serves as the controller, with no restrictions on its output, and the validation accuracy of the

model architectures is used as feedback. When using the NAS-designed model on the PHM 2009 gearbox dataset, the results yield state-of-the-art performance, outperforming manually built networks. This is among the earliest applications of NAS in a fault diagnosis environment.

2.2.6 GNN-Based Fault Diagnosis in PV Networks

Liu et al. [21] proposed a DACDFE-GNN model to diagnose faults in distribution networks with integrated photovoltaic systems. The model encodes both network topology and electrical measurements, using a dynamic graph aggregation mechanism to handle weak and noisy fault features. The study addresses challenges arising from bidirectional power flows, low sample rates, and measurement noise, which make traditional methods unreliable. Limitations include reliance on simulation data and increased computational complexity for larger networks. Results demonstrated superior fault detection, localization, and classification accuracy under noisy conditions and variable PV output compared to baseline methods.

2.2.7Multi-Task GNN for Grid Fault Detection

Chanda and Soltani [22] introduced a heterogeneous multi-task GNN (MTL-GNN) that performs simultaneous fault detection, localization, classification, resistance, and current estimation. The model also includes an explainability module to identify key nodes, supporting sparse measurement strategies. The work addresses limitations of previous approaches that required multiple separate models and struggled with noisy or sparse measurements. Limitations include evaluation on simulation-based test feeders rather than real-world data and the need for validation in highly dynamic grids. Results showed high accuracy across all tasks and effective identification of key nodes for efficient monitoring.

Table 1: Comparison of fault diagnosis techniques in electrical systems

Technique	Key Features	Limitations	How ST-GCN Addresses	Reference
GBT	Gradient boosting on statistical features	Not real-time, limited adaptation to unseen faults	Real-time, handles high-dimensional data, adaptable	[11]
MLP	Multi-layer perceptron on engineered features	Sensitive to noise, needs manual feature engineering	Learns spatio-temporal features automatically, robust to noise	[12]
SVM	Kernel-based classification	Poor scalability, limited to specific boundaries	Scales to large, sparse data; captures complex dependencies	[13]
VMD+EBTM	Signal decomposition + ensemble bagged trees	High computational load, topology-specific	Lower computational overhead, topology-agnostic	[14]
RF + CATV	Random forest on sensor measurements	Depends on sensor placement, limited in dynamic scenarios	Works with sparse/noisy measurements, scalable	[15]
NCFS+SVM	Feature selection + SVM	Requires labeled data, degrades with changing conditions	Handles sparse and unknown conditions, adaptive	[16]
XGBoost+SVM	Hybrid gradient	Needs full voltage	Works with partial	[17]

	boosting + SVM	coverage, limited generalization	measurements, generalizes across topologies	
DT-FDC	Decision tree for FDC	Overfitting, limited interpretability	Robust and interpretable, scalable for high-dimensional grids	[18]
GAN + FM	Synthetic data generation	Computationally expensive, data quality issues	Avoids synthetic data reliance, reduces training overhead	[19]
NAS	Neural architecture search	High training overhead, not edge-suitable	Efficient architecture for edge deployment, real-time ready	[20]
DACDFE GNN	GNN for PV networks	Relies on simulation, high computational cost	Robust to sparse/noisy data, scalable to weak grids	[21]
Multi-task GNN	Heterogeneous multi-task GNN	Simulation-based evaluation, complex	Real-time, handles sparse measurements, interpretable	[22]
ST-GCN (proposed)	Spatial-temporal graph convolution	Needs topology info	Real-time, scalable, interpretable, robust to sparse/noisy data	This work

Table 1 summarizes conventional, advanced, and GNN-based fault diagnosis methods, highlighting their key features and limitations. It also illustrates how the proposed ST-GCN approach overcomes these challenges through real-time detection, spatio-temporal feature learning, and scalability. The comparison emphasizes the research gap and the motivation for adopting ST-GCN in weak electrical systems.

2.3 Research gap

The shortcomings of relevant research are shown in Table 2.

Table 2: Shortcomings in related research

Technique	Limitation
GBT	Lacks real-time processing; trained on static grid simulations; limited adaptation to unseen faults [11].
MLP	Requires manual feature engineering; sensitive to noise; high dependency on sensor accuracy [12].
SVM	Poor scalability to large datasets; limited to linear or kernel-specific boundaries [13].
VMD +EBTM	High computational load in signal decomposition; topology-specific tuning needed [14].
CATV + Random Forest	It depends on sensor placement and resolution; therefore, it is not suitable for rapid dynamic fault scenarios [15].
NCFS + SVM	Relies on labeled fault data; performance degrades with changes in DG behavior or noise [16].
XGBoost + SVM	Requires comprehensive voltage coverage; may not generalize well across topologies [17].
DT-FDC	Prone to overfitting; limited interpretability in large, high-dimensional systems [18].

GAN	Synthetic data quality is challenging to control and computationally expensive for edge deployment [19].
NAS	High training overhead; unsuitable for low-power or embedded system applications [20].

Explicitly highlight how the proposed ST-GCN model addresses the limitations of prior methods and positions it within the research gap. Specifically:

- Critically analyze the shortcomings of conventional and advanced diagnostic techniques (e.g., GBT, MLP, SVM, VMD+EBTM, RF, NCFS+SVM, XGBoost+SVM, DT-FDC, GAN, NAS) in terms of real-time deployment, scalability, adaptability, interpretability, and edge applicability.
-
- Clearly state how ST-GCN overcomes these issues, e.g., by providing real-time fault detection, handling high-dimensional and sparse data, capturing spatio-temporal dependencies, reducing computational overhead, and enabling scalable deployment.
- Link the discussion directly to the research gap, showing the motivation for introducing ST-GCN for weak electrical systems.

3 Graph-theoretic system representation

3.1 Electrical system as a graph: node and edge semantics

Electrical power systems can be easily modelled as graphs with each system component abstracted as a node in the graph and the electrical or logical connection between them abstracted as edges. In this context, the nodes can be buses, transformers, circuit breakers, or smart sensors, while the edges represent transmission lines, feeder sections, or logical signal paths. It is a graph abstraction that enables the system to maintain both topological integrity and structural dependencies; thus, it can be utilized in a graph-based learning framework. Putting spatial structures directly into the learning model, the global pattern can be used to infer localized behavior.

3.2 Failure signatures and graph encodings

Malfunctions on weak electrical systems, e.g., short circuits, load imbalances, or breaker faults, would indicate faults in observations of sensors, currents, or voltages. The dynamics of these disruptions propagate across the network and can be recorded as signatures of failure. With the graph encoding, the features representing the node (e.g., voltage, current, frequency) and edge are embedded into a matrix feature. In this form, this encoded graph is an input to a Graph Neural Network, which learns the spatial correlations and is able to identify normal and faulty states thanks to message passing and aggregation layers.

3.3 Temporal dynamics in sensor-driven environments

The environment in question is sensor-rich and constantly generates time-series data regarding the system's state. Fault events are not only localized in space but also transitive in time, evolving over milliseconds to minutes. It is essential to incorporate the time dimension sequences in the endeavor to learn about fault progression as well as early warning. Regarding this, every graph at a moment snapshot t is connected to its previous and successor states to create a spatial-temporal graph sequence. It is this time relationship that enables the suggested model, e.g., ST-GCN, to detect not only the instantaneous faults, but also their transitions, leading to better detection rates and predictive maintenance.

4 The proposed intelligence framework

4.1 ST-GCN for spatio-temporal pattern learning

The essence of the model suggested consists of ST-GCN, which competently embraces the depictions of both spatial interactions (network topology) and the dynamics of the evolution. ST-GCN generalizes standard GCNs by adding temporal convolution layers that operate on graph state sequences. Such a combination of convolutional mechanisms enables the model to learn the variations of electrical signals in both nodes and timesteps, allowing for precise fault diagnosis in dynamic yet noisy operating conditions. ST-GCN learns the causal dynamics of failure from patterns in sensor data sequences; this is why it can perform this task more effectively compared to static classifiers.

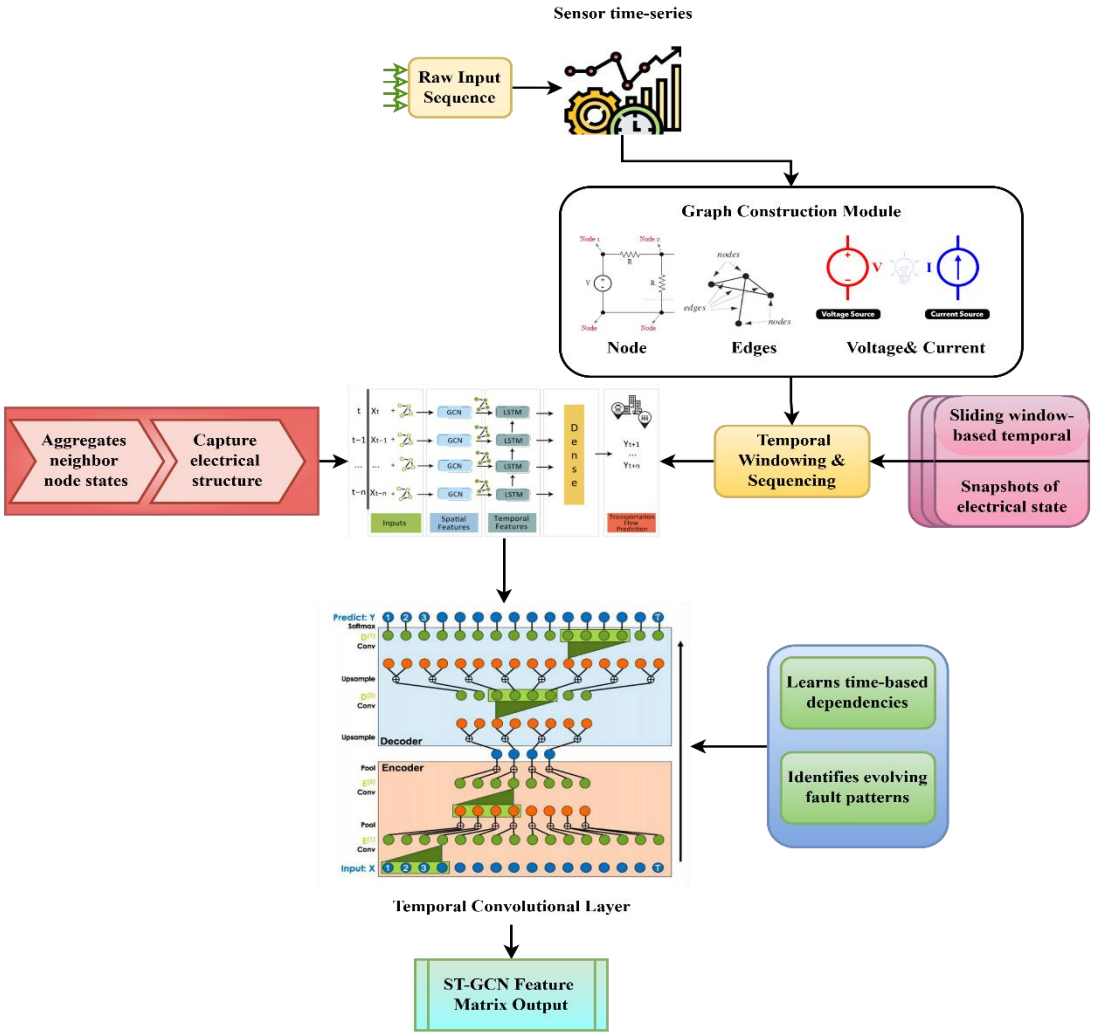

Figure 1: ST-GCN temporal-spatial modeling flow

Figure 1 illustrates the ST-GCN-based fault diagnosis framework for weak electrical systems. It begins with raw time-series input (voltage/current), constructs a graph using electrical topology (nodes and edges), and applies temporal windowing to form sequential snapshots. GCNs units extract spatial and temporal features, respectively. The model encodes neighbor node interactions and temporal dependencies through a spatio-temporal encoder-decoder. The final output is a learned feature matrix capable of detecting and localizing evolving fault patterns with high accuracy in real time. The layout is now linear, showing the main stages from Raw Input, Graph Construction, Temporal Windowing, ST-GCN, Feature Output. Each module is clearly labeled with short descriptions, such as Graph Construction: Nodes = buses, Edges = electrical connections, Temporal Windowing: Sliding window captures temporal snapshots, ST-GCN: Learns spatial and temporal dependencies of faults, and Feature Output: Matrix representing fault patterns for downstream tasks. A glossary box beside the figure explains key terms, including Node, Edge, Temporal Window, and ST-GCN. Consistent colors and symbols are used for nodes, edges, voltage/current, and temporal modules to make the flow intuitive and visually clear.

Evaluation Protocol

The dataset is divided into training, validation, and testing subsets using a 70/15/15 split. Model performance is assessed using accuracy, F1-score, fault localization accuracy, latency analysis, PR curves, loss convergence behavior, and scalability tests. All hyperparameters, thresholds, and window sizes are explicitly documented to ensure full reproducibility.

4.2 Data processing pipeline

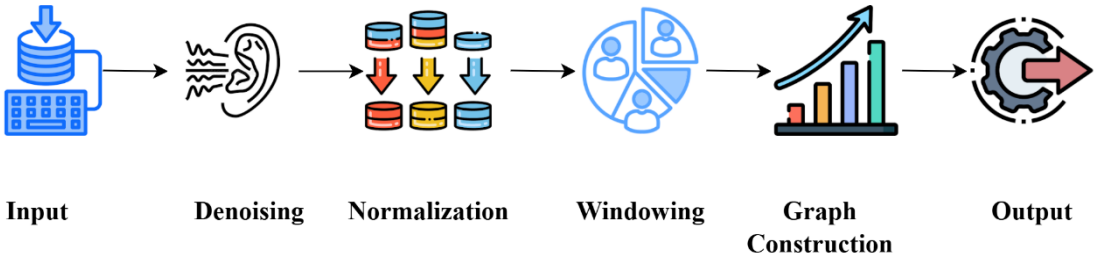

Figure 2: Data processing pipeline for smart grid fault analysis

The raw voltage and current time-series streams from the Smart Grid Monitoring Dataset are first denoised using a moving-average filter to remove high-frequency noise while preserving fault-related variations (Figure 2). The signals are then normalized to a standard range to ensure consistent input scaling across sensors. Next, the time-series data are segmented into fixed-length windows (50–200 ms) to capture local temporal patterns. For each window, graph structures are constructed based on the electrical connectivity matrix, encoding spatial dependencies between sensors as edges in the graph. Missing or corrupted readings are handled through linear interpolation or zero-filling as appropriate, ensuring robust input to the ST-GCN model. This structured pipeline ensures that both temporal dynamics and spatial relationships are preserved, facilitating effective learning of fault patterns across diverse scenarios. Raw voltage and current streams are denoised using a moving-average filter, normalized, and segmented into fixed-length windows (50–200 ms). For each window, graph structures are constructed based on the electrical connectivity matrix to capture spatial dependencies within the weak electrical system.

4.3 Model training and setup

The ST-GCN model is composed of stacked spatial-temporal graph convolutional blocks with residual connections and batch normalization to improve gradient flow and stability. Each block captures local temporal dynamics while propagating spatial information across the graph nodes. After feature extraction, the high-level spatio-temporal embeddings are flattened and fed into a dense classification layer. Training is performed using the Adam optimizer with cross-entropy loss, and dropout regularization is applied to prevent overfitting. Hyperparameters such as learning rate, batch size, number of layers, and hidden units are selected through a combination of grid search and validation performance monitoring. Fine-tuning is conducted iteratively, adjusting layer sizes and regularization coefficients until convergence is achieved on the validation set. Each graph sequence is passed through stacked ST-GCN blocks with residual connections and batch normalization, after which the extracted spatio-temporal features are flattened and fed into a dense classification layer. Training is performed using the Adam optimizer with cross-entropy loss, dropout regularization, and mini-batch learning to enhance model stability and generalization.

The dataset was divided into training, validation, and test sets in a 70:15:15 ratio, ensuring class balance via stratified sampling. Prior to training, all data underwent preprocessing including normalization and encoding, while missing values were imputed using a median-based strategy and noise was reduced through outlier filtering. To evaluate the model's adaptability and robustness, additional simulated fault scenarios and an external benchmark dataset were employed, enabling assessment of performance under diverse and challenging conditions. Hyperparameters were selected through grid search on the validation set, and network fine-tuning was performed iteratively to optimize accuracy while preventing overfitting.

4.4 Handling noise and missing data

Raw input signals may contain missing or corrupted readings due to sensor faults or communication errors. These are handled using linear interpolation for short gaps or zero-filling for longer missing sequences. Additionally, denoising through moving-average filtering ensures that transient noise does not adversely affect feature extraction. This preprocessing ensures robust and reliable inputs, allowing the ST-GCN model to maintain high performance even under imperfect real-world conditions.Spatial graph convolution for electrical topology encoding $I^{(m+1)}$ is expressed using equation 1,

$$I^{(m+1)} = \Delta\left(\tilde{E}^{-1/2}\tilde{B}\tilde{E}^{-1/2}I^{(m)}X^{(m)}\right) \quad (1)$$

Equation 1 explains the spatial graph convolution for electrical topology encoding where an electrical structure is represented using normalized carries out a spatial graph convolution.In this $I^{(m)}$ is the feature matrix at layer, where O is the number of electrical nodes and G_m is the feature dimension, $\tilde{B}$ is the adjacency matrix with self-loops, where B encodes node connectivity, $\tilde{E}$ is the diagonal matrix with node degrees of $\tilde{B}$, $X^{(m)}$ is the

trainable weight matrix at layer, and $\Delta(.)$ is the nonlinear activation function.

To represent how defects change over time through system dynamics, temporal dependencies must be described once spatial dependencies have been incorporated.

Temporal modeling using LSTM over node features (i_u, d_u) is expressed using equation 2,

$$(i_u, d_u) = MTUN\big(I_u, (i_{u-1}, d_{u-1})\big) \quad (2)$$

Equation 2 explains the temporal modeling using LSTM over node features represented by this equation.

In this I_u is the spatially convolved features at time from previous GCN layer, i_u is the hidden state at time representing short-term memory, d_u is the cell state at time representing long-term memory, $MTUN(.)$ is the recurrent unit mapping inputs and previous states to new states, and i_{u-1}, d_{u-1} are the previous time step's hidden and cell states.

These spatiotemporal images are delivered to a fully linked decoder after temporal encoding for reliable and real-time fault detection in weak power lines.

Neighborhood-aware node feature aggregation $y_j^{(m+1)}$ is expressed using equation 3,

$$y_j^{(m+1)} = \partial\left(\sum_{k\in O(j)} \frac{1}{\sqrt{e_j e_k}} X^{(m)} y_k^{(m)}\right) \quad (3)$$

Equation 3 explains node feature aggregation uses degree-normalized weighting to gather information from its neighbors to update the feature image of the node at layer.

In this $y_j^{(m)}$ is the feature vector of node at layer, $O(j)$ is the set of neighboring nodes connected to node, e_j, e_k are the degrees of nodes, $X^{(m)}$ is the trainable weight matrix at layer, and $\partial(.)$ is the nonlinear activation function.

The following equation results from the need to explain the sequential fluctuation of node states brought on by temporally fault propagation by neighbor feature aggregation.

Temporal flow integration via gated recurrent units $i_j^{(u)}$ is expressed using equation 4,

$$i_j^{(u)} = HSV\big(y_j^{(u)}, i_j^{(u-1)}\big) \quad (4)$$

Equation 4 explains the temporal flow integration via gated recurrent units is the temporal evolution of node characteristics over time is captured by this equation using a GRU cell.

In this $y_j^{(u)}$ is the aggregated spatial feature of node at time, $i_j^{(u)}$ is the updated hidden state representing temporal memory for node, $i_j^{(u-1)}$ is the hidden state from the previous time step, and $HSV(.)$ is the gated recurrent unit function that encodes temporal relationships in sequential data.

High-resolution spatio-temporal fault characterization is made possible by the fault signature encoder.

Temporal convolution over spatial feature embedding $A_{j,u}^{(l)}$ is expressed using equation 5,

$$A_{j,u}^{(l)} = \sum_{\partial=0}^{L-1} \sum_{g=1}^{G} \gamma_{g,\partial}^{(l)} * I_{j,u-\partial}^{(g)} + c^{(l)} \quad (5)$$

Equation 5 explains that the temporal convolution over spatial feature embedding is the spatial properties to a 1D longitudinal convolution represented by this equation.

In this $I_{j,u-\partial}^{(g)}$ is the input feature from node, $\gamma_{g,\partial}^{(l)}$ is the temporal kernel weight for feature, L is the temporal kernel width, G is the number of input features, $A_{j,u}^{(l)}$ is the output of the temporal convolution for node, and $c^{(l)}$ is the bias term for the output channel.

4.5 Node embedding, neighborhood aggregation, and temporal flow

At each period, these nodes (e.g., sensors, breakers, transformers) are modeled using feature vectors that include voltage, current, harmonic, and phase data. These are ranked in a high-dimensional space using a node embedding algorithm that preserves both feature similarity and spatial locality.

In neighborhood aggregation, localized fault effects are considered in that features of neighboring nodes are combined with those of the original node to update its representation. It processes the temporal flow through a 1D convolution along the time dimension, allowing the model to learn how dependencies operate between multiple timesteps. Such a structure enables the ST-GCN to learn both sudden and slow shifts in system behavior.

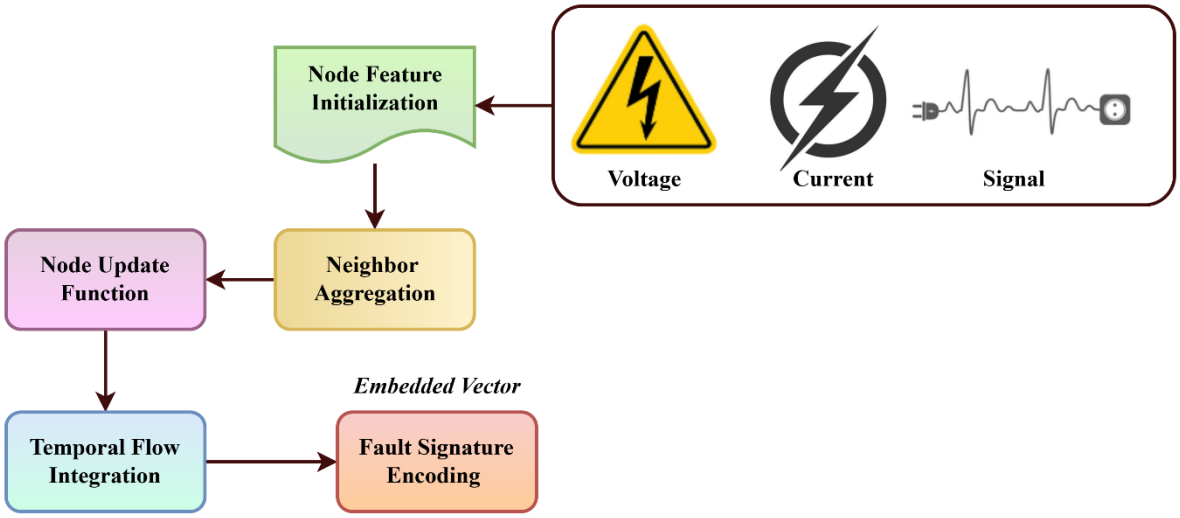

Figure 3: Node-Level ST-GCN operation flow

Figure 3 illustrates the spatio-temporal feature extraction pipeline in a graph neural network for fault diagnosis. It begins with voltage, current, and signal data used to initialize node features. Through neighbor aggregation, the
local structural context is captured. The node update function refines feature states, which are then passed through temporal flow integration to model sequential dependencies. Finally, a fault signature encoding step captures dynamic fault patterns, enabling robust anomaly detection and localization in weak electrical grid topologies [23].

Following the extraction of spatiotemporal mappings of features, the model uses a distinguished loss function and a Bayesian output distribution to maximize its prediction ability.

Multi-class cross-entropy loss for fault classification M is expressed using equation 6,

$$M = -\frac{1}{O}\sum_{j=1}^{O}\sum_{d=1}^{D} z_j^{(d)} \log\left(\hat{z}_j^{(d)}\right) \quad (6)$$

Equation 6 explains the multi-class cross-entropy loss for fault classification calculates the categorized cross-entropy loss.

In this O is the total number of labeled fault samples, D is the number of distinct fault classes, $z_j^{(d)}$ is the ground truth label indicator for sample, $\log(.)$ is the natural logarithm function, $\hat{z}_j^{(d)}$ is the predicted probability for the sample, and M is the total loss used for training optimization.

By changing network weights to enhance the alignment between anticipated and genuine fault classes, to reliably categorize fault occurrences in weak power systems.

Softmax-based fault class prediction from ST-GCN embedding $\hat{z}_j^{(d)}$ is expressed using equation 7,

$$\hat{z}_j^{(d)} = \frac{f^{v_j^{(d)}}}{\sum_{k=1}^{D} f^{v_j^{(k)}}} \quad (7)$$

Equation 7 explains that the soft max-based fault class prediction from the ST-GCN embedding function is applied to the resulting logits of the dense classifying layer for a node in this equation.

In this $\hat{z}_j^{(d)}$ is the predicted probability that the node belongs to the fault class, $v_j^{(d)}$ is the logit value for node, D is the total number of distinct fault categories, and f is the Euler's number, used in exponentiation.

The system must choose whether to initiate a real-time reaction after obtaining the soft max probability by determining if the fault prediction's confidence level is above a predetermined threshold.

Real-time fault decision and alert dispatch rule A_i is expressed using equation 8,

$$A_i = \begin{cases} 1 & if\ \max_d \hat{z}_j^{(d)} > \partial \\ 0 & otherwise \end{cases} \quad (8)$$

Equation 8 explains the real-time fault decision and the alert dispatch rule determines if a predetermined confidence level is exceeded by the highest anticipated fault likelihood for a node.

In this $\hat{z}_j^{(d)}$ is the predicted softmax probability for node, ∂ is the decision threshold for triggering fault alerts, $\max_d \hat{z}_j^{(d)}$ is the maximum predicted probability over all fault classes for node, and A_i is the real-time fault alert status for the node.

4.3 Architecture pipeline and training paradigm

The proposed ST-GCN framework can help learn intricate node-level spatial and temporal relationships to achieve successful fault localization in weak electrical systems. The first layer will be an input that receives a time series of graph-structured information, where each graph instant contains node-level features, such as voltage, current, or frequency, and an adjacency matrix, which determines the network's connectivity. These input graphs are fed to a chain of spatial-temporal blocks, with each block adding some form of spatial-temporal convolution block, a temporal convolution block here to model temporal change, with a graph convolution block to model spatial co-occurrence.

Residual connections and batch normalization layers are incorporated within each block to enhance training stability and accelerate convergence. Their fully connected layers are then supplied with the flattened high-level features obtained from these spatio-temporal blocks. They are used to perform the ultimate classification or regression, i.e., identifying the fault type or estimating the fault location. Multi-class classification is performed using a softmax activation at the output layer, containing a cross-entropy loss-based training network. The Adam optimizer is applied in the training process, based on mini-batch gradient descent, and methods to avoid overfitting are employed, including dropout regularization. The power networks can be part of a dynamic, sensor-based system that accurately diagnoses faults in real-time via this architecture.

Figure 4 depicts the ST-GCN-based fault classification pipeline using historical sensor datasets. The workflow begins by normalizing and segmenting the temporal data, including labeled faults and timestamps. A combination of spatial GCN and 1D temporal convolution extracts spatio-temporal features. These features feed into a dense classification head with a softmax activation to predict fault types. Training utilizes cross-entropy loss and the Adam optimizer, enabling robust learning of dynamic fault patterns in electrical grid sensor data.

In order to respond quickly to an alarm, the system's mitigation module logs a date and time, node ID, projected fault, the lesson, and confidence score.

Node-level anomaly score via temporal embedding deviation $T_j(u)$ is expressed using equation 9,

$$T_j(u) = \left\| i_j(u) - \partial_j \right\|_2 \quad (9)$$

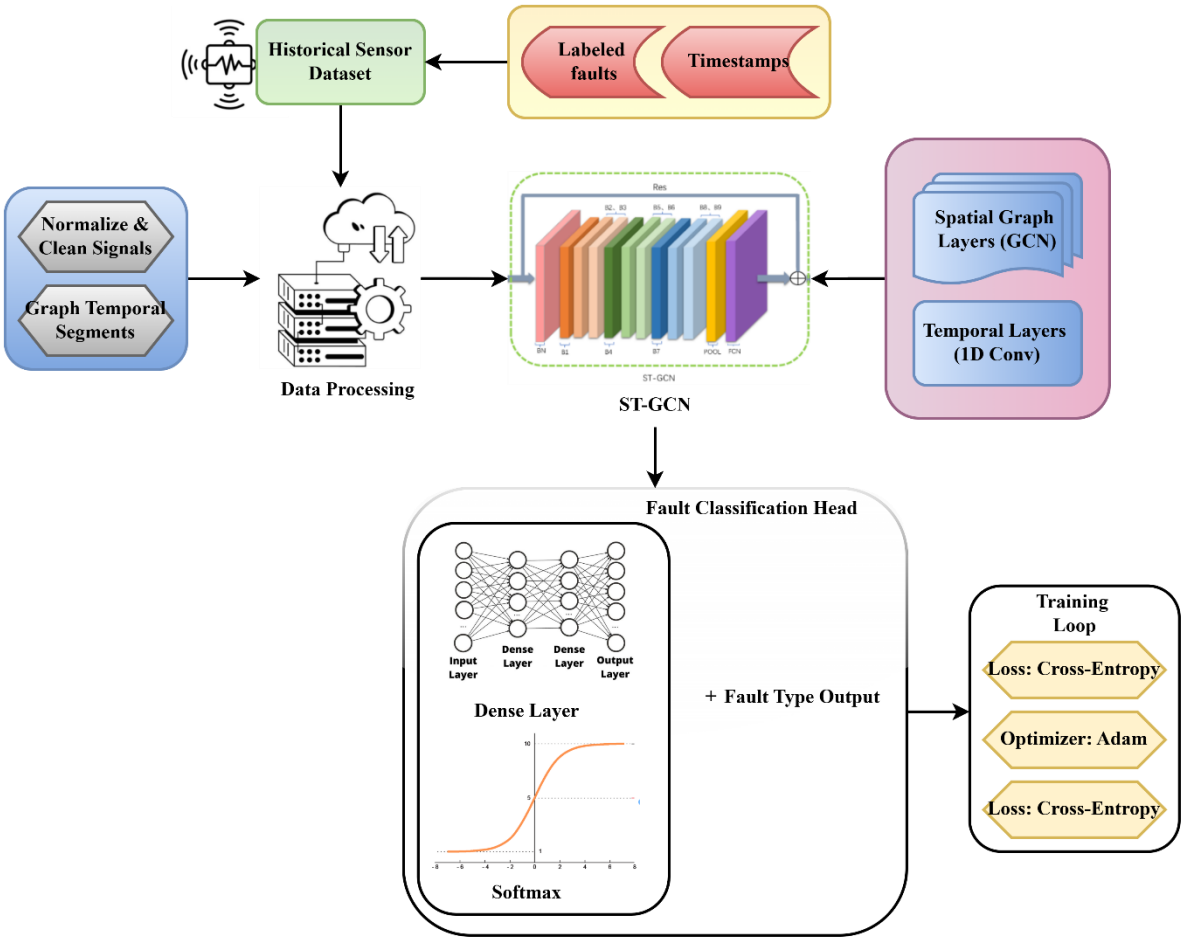

Figure 4: Full model training pipeline

Equation 9 explains the node-level anomaly score via temporal embedding deviation by calculating the standard deviation (L2) of the variance between the present-day.

In this $i_j(u)$ is the Node embedding vector at time, ∂_j is the temporal mean embedding of a node over a reference window of length, $\|.\|_2$ is the L2 (Euclidean) norm, and $T_j(u)$ is the anomaly score indicating deviation magnitude for node.

5 Fault intelligence layer

5.1 Real-time fault detection mechanism

The real-time identification model runs on the incoming data. The module utilizes the ST-GCN model, which has been previously trained. The system flags the presence of a fault once deviations from expected patterns are detected. The system, equipped with a low-latency temporal convolution, can operate with minimal delay, detecting faults as soon as they appear within a few milliseconds. It is beneficial in situations where a quick response time is crucial for safety and asset protection.

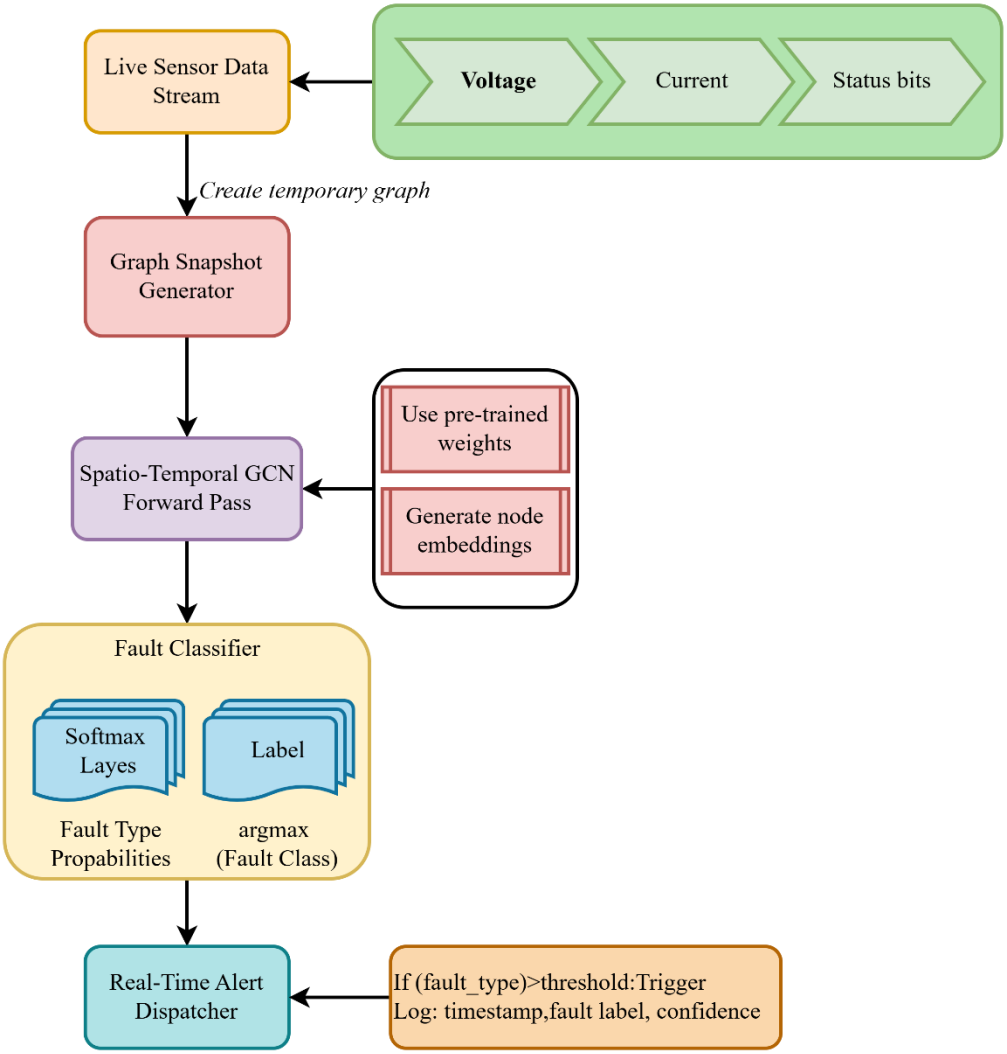

Figure 5: Real-Time Fault Detection Flow (ST-GCN Inference Pipeline)

Figure 5 illustrates the real-time ST-GCN-based fault detection pipeline for smart electrical grids. Live voltage, current, and status bit streams are converted into graph snapshots. A pre-trained ST-GCN performs forward propagation to generate node embeddings. The output is passed to a softmax-based fault classifier, which determines the most probable fault class. If fault probability exceeds a threshold, a real-time alert dispatcher logs the timestamp, fault label, and confidence, enabling rapid fault mitigation and network resilience.

A confidence score that takes into consideration both signal variability and topological agreement among nearby nodes is estimated using anomaly scores.

Confidence score from signal stability and neighborhood consistency C_j is expressed using equation 10,

$$C_j = \frac{\exp(\beta * U_j + \gamma * B_j)}{\sum_{k \in L} \exp(\beta * U_j + \gamma * B_j)} \quad (10)$$

Equation 10 explains that the confidence score from signal stability and neighborhood consistency is calculated.

In this U_j is the signal deviation sharpness for node normalized, B_j is the structural consistency score based on anomalous neighbor alignment, β, γ are the weighting factors for signal and structural terms, L is the set of top candidate nodes based on anomaly scores, and C_j is the final normalized confidence score for node fault prediction.

The node designated as the probable fault origin is the one with the greatest confidence score.

Accuracy evaluation Ac is expressed using equation 11,

$$Ac = \frac{UQ + UO}{UQ + UO + GQ + GO} \quad (11)$$

Equation 11 explains the accuracy evaluation is the ratio of accurate fault forecasts, both positive and negative, to all predictions is known as accuracy.

In this UQ is the true positives correctly identified as fault cases, UO is the true negatives correctly identified as non-fault cases, GQ is the false positives incorrect fault predictions, and GO is the false negatives are missed fault cases.

5.2 Localization engine for fault origin tracing

Once a fault has been observed, the model applies the learned spatial embeddings to backtrack the fault to its root cause. The method that can enable the system to identify the most likely source of failure is to analyze which nodes contributed the most significant amount of anomaly through an attention mechanism or gradient-based attribution. The model identifies the node(s), sector, or component that operators can isolate and fix within a very short time, eliminating the need for time-consuming inspections.

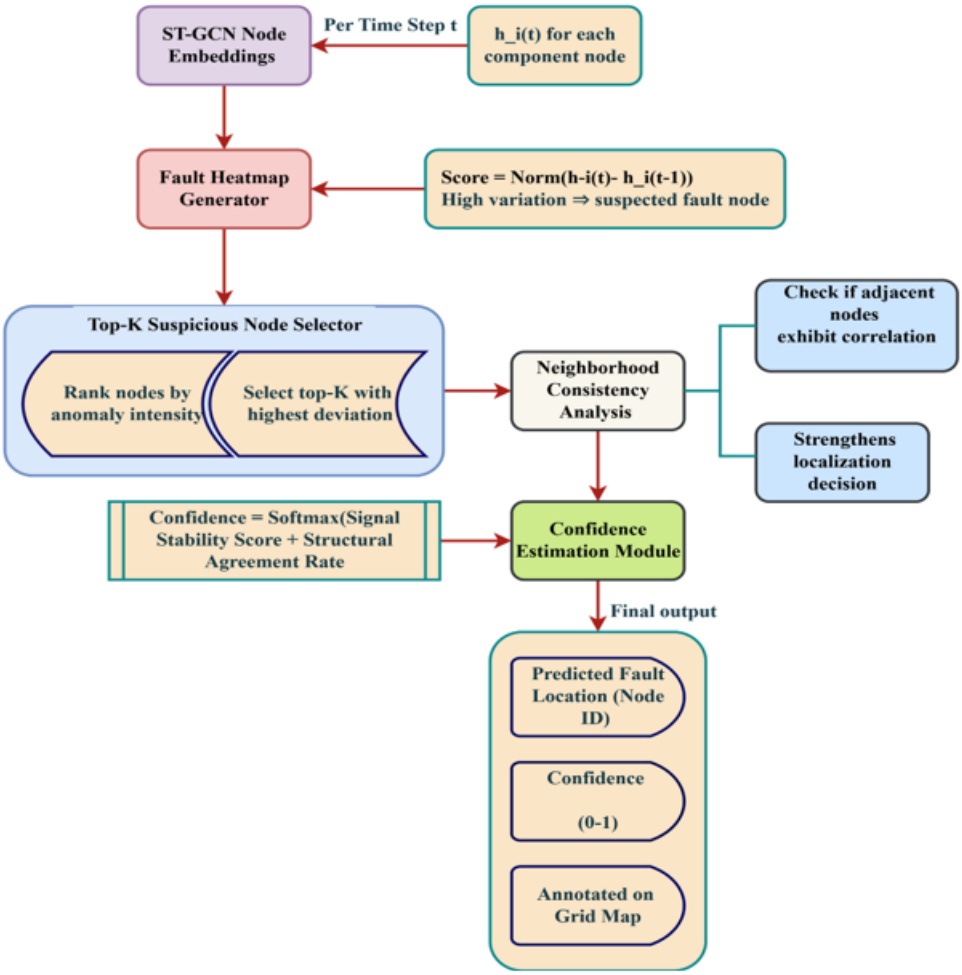

Figure 6: Fault origin tracing + confidence score pipeline

This figure 6 illustrates the fault localization pipeline using ST-GCN node embeddings. At each timestep, node representations $hi(t), h\_i(t), hi(t)$ are evaluated for deviation, generating a fault heatmap. The Top-K suspicious nodes with the highest anomaly scores are selected. Neighborhood consistency analysis validates these anomalies using structural correlations. A confidence estimation module computes a softmax-based score by combining signal stability and structural agreement. The final output includes predicted fault location (node ID), confidence score (0–1), and visual annotation on the smart grid map.

Algorithm: Fault Origin Tracing and Confidence Scoring using ST-GCN

Inputs:

- $Node\ embeddings\ over\ time: H = \{h\_i(t)\}$
- $Graph\ structure: G(V, E)$
- $Parameters: \tau\_deviation, K, \alpha\_stability, \beta\_structure$

Outputs:

- $Fault\_Node\_ID: Predicted\ faulty\ node$
- $Confidence\_Score \in [0, 1]$
- $Annotated\_Grid\_Map$

```
Step 1: Compute deviation scores
for each node i in G:
  for each timestep t:
    deviation_score[i][t] = compute_deviation(h_i(t))
Step 2: Generate heatmap
for each node i:
  if deviation_score[i][t] > τ_deviation:
    heatmap[i] = deviation_score[i][t]
  else:
    heatmap[i] = 0
Step 3: Select Top − K suspicious nodes
top_k_nodes = select_top_k(heatmap, K)
Step 4: Neighborhood consistency check
for each node i in top_k_nodes:
  neighbors = get_neighbors(i, G)
  valid_neighbors = 0
  for each neighbor j in neighbors:
    if deviation_score[j][t] > τ_deviation:
      valid_neighbors += 1
  if valid_neighbors < len(neighbors) * 0.5: # threshold_ratio = 0.5
    remove i from top_k_nodes
Step 5: Confidence estimation
for each node i in top_k_nodes:
  signal_stability = compute_signal_stability(h_i(t))
  structure_agreement = compute_structure_agreement(i, neighbors)
  score[i] = α_stability * signal_stability + β_structure * structure_agreement
```

$confidence\_score[i] = softmax(score[i])$

Step 6: *Final fault prediction*

$Fault\_Node\_ID = argmax(confidence\_score[i])$

$Confidence\_Score = confidence\_score[Fault\_Node\_ID]$

Step 7: *Annotate and return*

$annotate\_grid\_map(Fault\_Node\_ID, Confidence\_Score)$

$return\ Fault\_Node\_ID, Confidence\_Score, Annotated\_Grid\_Map$

This algorithm 1 identifies faults in a smart grid by analyzing ST-GCN node embeddings over time. It computes deviation scores, selects the Top-K anomalous nodes, verifies them using neighborhood consistency, and calculates a confidence score combining signal and structural cues. The output is a fault location with visual and numerical confidence.

5.3 Anomaly scoring and confidence estimation

The model also has an anomaly scoring function that measures the degree of deviation of the current signal from normal operational behavior. Each of the detections is accompanied by a confidence score, in turn constituted of the softmax output probabilities or as a Bayesian model of uncertainty. The scores help prioritize tasks and minimize false alarms. To illustrate, a fault with an oddity score of 0.91 and a confidence of 98 per cent is handled as high-priority over low-confidence ones.

5.4 Metrics for evaluation

Evaluating fault detection systems in smart grids demands robust and diverse metrics beyond simple accuracy. This section introduces key performance indicators, including F1-score, fault localization accuracy, latency, confusion matrix analysis, precision-recall curves, loss convergence, and scalability. Together, they offer a comprehensive evaluation of model reliability, responsiveness, and adaptability to network growth.

The F1-score is calculated next for a balanced evaluation since, although accuracy indicates overall correctness, it may mask class imbalance.

F1 score analysis G1 is expressed using equation 12,

$$G1 = \frac{2 * Pn * Rl}{Pn + Rl} \quad (12)$$

Equation 12 explains the F1 score analysis emphasizes the balance across errors in judgment and false negatives by combining accuracy and recall using a harmonic mean.

In this Pn is the proportion of predicted faults that are true, Rl is the proportion of actual faults detected correctly, and G1 is the composite performance indicator for classification under imbalance.

Real-time systems require spatial assessment in addition to classification metrics, which brings us to the following metric fault localization accuracy.

Fault localization accuracy LA is expressed using equation 13,

$$LcAc = \frac{\sum_{j=1}^{L} 1[o_j = \hat{o}_j]}{L} \quad (13)$$

Equation 13 explains that the fault localization accuracy is the correctness of fault nodes prediction among the top predicted nodes in comparison to actual fault sites is calculated.

In this L is the number of test fault cases or evaluated fault events, o_j is the ground truth fault node for case, $\hat{o}_j$ is the predicted fault node for case, $1[.]$ is the indicator function, and $LcAc$ is the proportion of correctly localized faults.

Temporal responsiveness is crucial, even though localization gauges accuracy. Latency is the next indicator that measures system responsiveness.

Detection time Lcy is expressed using equation 14,

$$Lcy = \frac{1}{L}\sum_{j=1}^{L}\left(u_j^d - u_j^f\right) \quad (14)$$

Equation 14 explains that the detection time is the mean time interval between a fault's actual occurrence and the model's detection, is known as latency.

In this u_j^f is the time stamp when the fault occurred, u_j^d is the time stamp when the fault was detected, L is the number of evaluated fault instances, and Lcy is the average response delay in milliseconds.

The confusion matrix offers a statistical and visual representation of prediction results to evaluate the misclassification distribution.

Confusion matrix entry computation D_{jk} is expressed using equation 15,

$$D_{jk} = \sum_{l=1}^{O} 1[z_l = j \cap \hat{z}_l = j] \quad (15)$$

Equation 15 explains that the confusion matrix entry computation is the frequency with which a class was anticipated to be a class across all samples, is shown by the confusion matrix column.

In this D_{jk} is the count of instances with the true class, z_l is the actual class label of sample, $\hat{z}_l$ is the predicted class label of sample, O is the total number of test samples, and $1[.]$ is the logical indicator function.

A precision-recall curve is used to display the performance trade-off when precision and recall measurements are retrieved from the confusion matrix.

Precision vs. recall curve construction QS_u is expressed using equation 16,

$$QS_u = (Ps_u, Rl_u), \ \forall_u \in [0,1] \quad (16)$$

Equation 16 explains to shows classification resilience at different confidence levels.

In this u is the soft max threshold for binary classification, Ps_u is the precision computed at threshold, QS_u is the tuple for plotting the precision-recall trade-off, and Rl_u is the recall computed at threshold.

As a result, the following equation uses loss convergence to monitor training progress.

Loss convergence curve M_f is expressed using equation 17,

$$M_f = \frac{1}{O_f} \sum_{j=1}^{O_f} M_j \quad (17)$$

Equation 17 explains the loss convergence curve representing the sample-wise loss, often cross-entropy.

In this M_f is the mean training loss at epoch, M_j is the loss for sample, O_f is the number of samples processed in epoch, and f is the current epoch index.

Scalability becomes essential in deployment. As a result, performance is evaluated as network size grows in the final calculation.

Scalability vs network size $\nabla P(O)$ is expressed using equation 18,

$$\nabla P(O) = \frac{P_{O_0} - P_O}{P_{O_0}} * 100 \quad (18)$$

Equation 18 explains topological growth by calculating the percentage decrease in performance of the model as the network grows from a base size.

In this P_O is the model performance metric at network size, O_0 is the baseline network size, and $\nabla P(O)$ is the percent performance degradation due to scaling.

This evaluation framework ensures balanced performance analysis using classification metrics, spatial accuracy, response latency, and visual diagnostics. It further monitors training progress and assesses scalability under topological expansion. These metrics collectively validate the model's practical effectiveness in real-time fault localization and its resilience across varying operational conditions in smart grid environments.

6 Experimental validation

6.1 Dataset overview

Smart Grid Monitoring Dataset, a Kaggle dataset [24], simulates real-life parameters of smart grid functions with sensor data that consists of fine-grained sensor data in a time-series format. It contains voltages, currents, frequencies, active and reactive power, and FFT converted features, along with labeled fault events. The dataset is most suitable for spatial-temporal modeling, which is intended for applications such as load forecasting, fault detection, and anomaly analysis. It is a highly multimodal dataset whose characteristics make it an optimal source for training and testing ST-GCN models in weak electrical systems. The parameters about the dataset is shown in Table 3.

Table 3: Parameterized table

Parameter	Description
Dataset Source	Kaggle (ziya07)
Data Type	Time-series sensor data
Key Measurements	Voltage (V), Current (I), Frequency (Hz), Active/Reactive Power (kW/kVAR)
Derived Features	FFT coefficients, Total Harmonic Distortion (THD), RMS
Sampling Interval	1-second intervals
Fault Labels	Includes labeled fault and regular operation events
Number of Sensors	Multiple smart meters and grid points (node-level data)
Applicable Tasks	Fault detection, anomaly classification, load forecasting, and pattern mining

The experiments use the Smart Grid Monitoring Dataset from Kaggle, which simulates real-life smart grid operations with fine-grained time-series sensor data. This dataset contains measurements of voltage (V), current (I), frequency (Hz), active/reactive power (kW/kVAR), and derived features such as FFT coefficients, RMS, and Total Harmonic Distortion (THD), along with labeled fault events. The dataset is highly multimodal and suitable for spatial-temporal modeling tasks such as fault detection, anomaly analysis, and load forecasting. To ensure reproducibility, the dataset is partitioned into training (70%), validation (15%), and testing (15%) subsets, and preprocessing includes denoising via moving-average filtering, normalization, and segmentation into fixed-length time windows.

Evaluation Metricst to comprehensively assess model performance, several metrics are considered. Accuracy measures the proportion of correctly predicted fault or normal instances, while the F1-score evaluates the harmonic mean of precision and recall for each fault type, which is particularly useful in imbalanced datasets. Fault localization accuracy quantifies the ratio of correctly identified faulty nodes to actual fault locations, and detection latency captures the time between fault occurrence and model prediction. Precision-Recall (PR) curves are analyzed to evaluate trade-offs at different decision thresholds, which is critical for mission-critical systems. Loss convergence behavior is monitored during training to assess learning stability and detect overfitting, and scalability tests examine performance across varying network sizes and sensor counts to validate the model's applicability in real-time and large-scale scenarios.

7 Results and discussion

The results and discussion section evaluates the ST-GCN model's effectiveness across diverse performance metrics, including classification accuracy, F1-score, fault localization precision, detection latency, and scalability. Visual tools like confusion matrices and PR curves support these findings, providing comprehensive insights into the model's robustness, responsiveness, and adaptability across real-time fault scenarios.

7.1 Accuracy evaluation

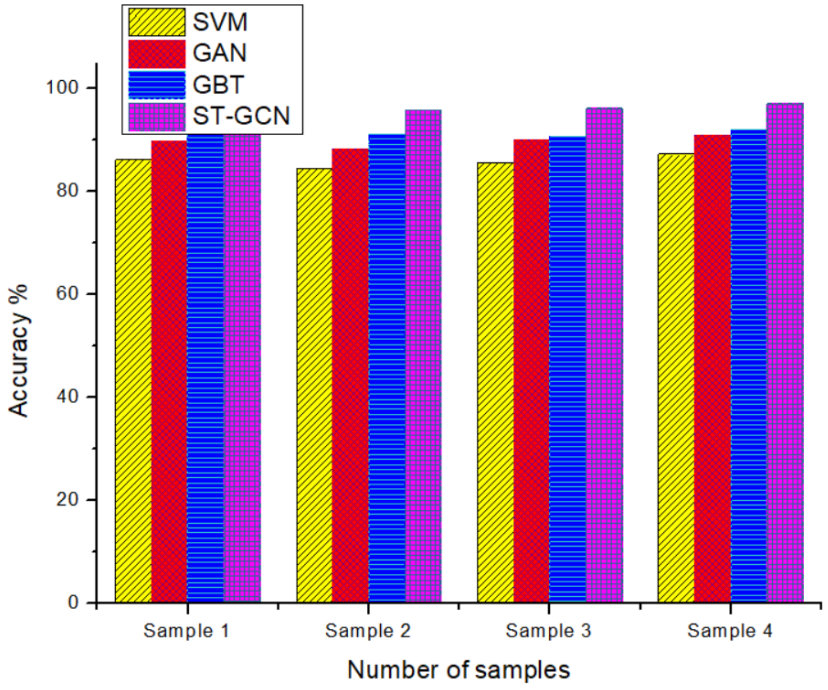

Figure 7: The analysis of accuracy evaluation

Figure 7 explains the suggested ST-GCN model was tested on a variety of simulated weak electrical system cases to assess its classification performance against a broad scope of fault types and conditions. Accuracy was taken to mean the number of correctly predicted fault or normal test levels divided by the total tests. The model demonstrated high performance that remains constant when tested on varying datasets, which in turn include differences in fault resistance, fault location, and the time at which they occur. The topology and temporal nature of the space enabled a successful network generalization, which outperformed other classical machine learning

models, including Decision Trees and SVMs. It was also most helpful in capturing the patterns of faults that were more subtle, which are those patterns that occur in low-connectivity nodes or sparse sensor input, typical of weak electrical systems made denoted using equation 11.

7.2 F1-score analysis

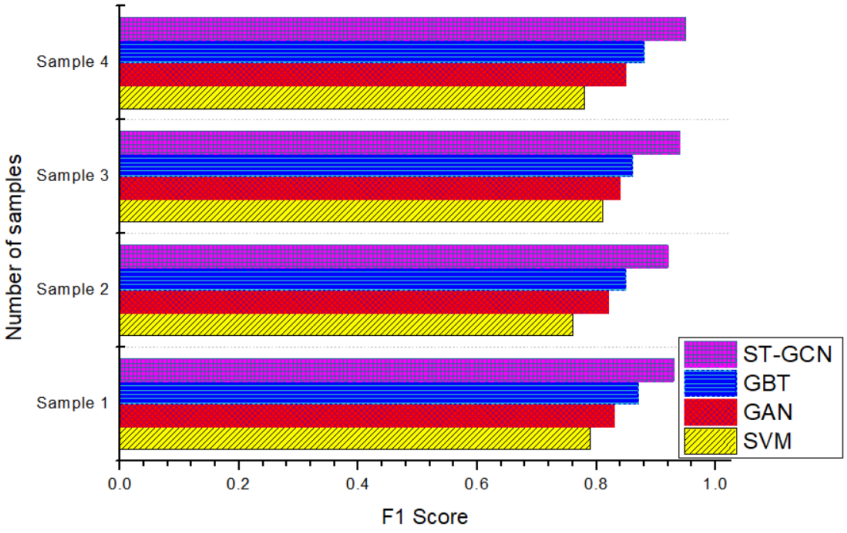

Figure 8: The Analysis of F1 Score

To evaluate the degree of precision versus recall on the model, F1-scores of each fault type were taken and explain in figure 8. This measure is essential in imbalanced data, where some faults are more common than others. ST-GCN demonstrate a high accuracy but mis-detection in the less studied types of faults. The ST-GCN achieved a substantial F1-score across all classes, indicating that it effectively reduced both false positives and false negatives. Message-passing in spatial graph and temporal sequence learning allowed the model to acquire more insights of complex, non-linear fault behaviors and hence provided an effective performance of classification regardless of edge-case scenarios were evaluated using equation 12

7.3 Fault localization accuracy

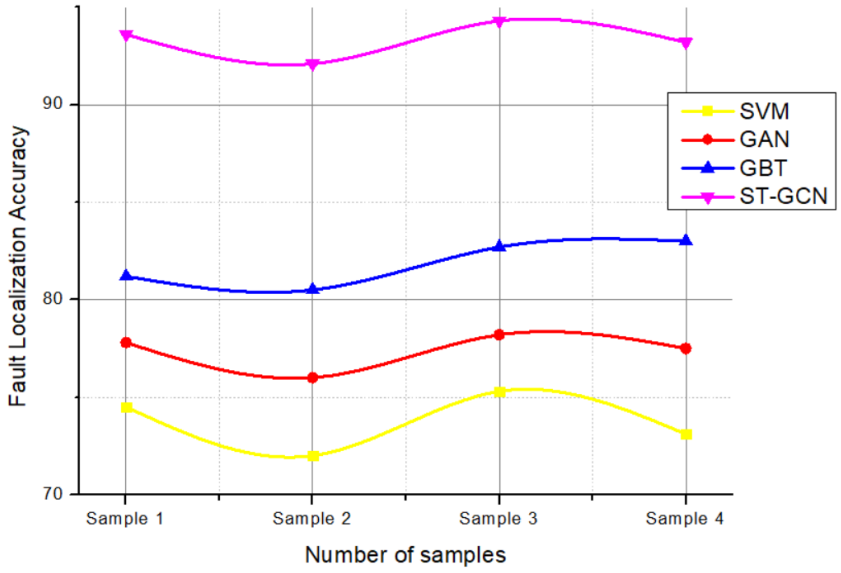

Figure 9: The Analysis of Fault Localization Accuracy

The system's primary feature is its ability to locate the source of a fault in the network accurately is shown in figure 9. The model incorporated node embeddings and the graphical topology of the electric graph to trace the disturbance signal to its root. The ratio of forecasted fault nodes to the real fault location determined the performance of the localization. The model demonstrated high fault localization accuracy across various topological arrangements, including radial layouts and meshed layouts. It was particularly effective at identifying faults deep in the network, where more basic models could be negatively impacted by signal distortion or latency. That success was dependent on the attention-like behavior of the ST-GCN architecture, which is driven by its neighborhood aggregation was calculated using equation 13.

7.4 Detection time (latency)

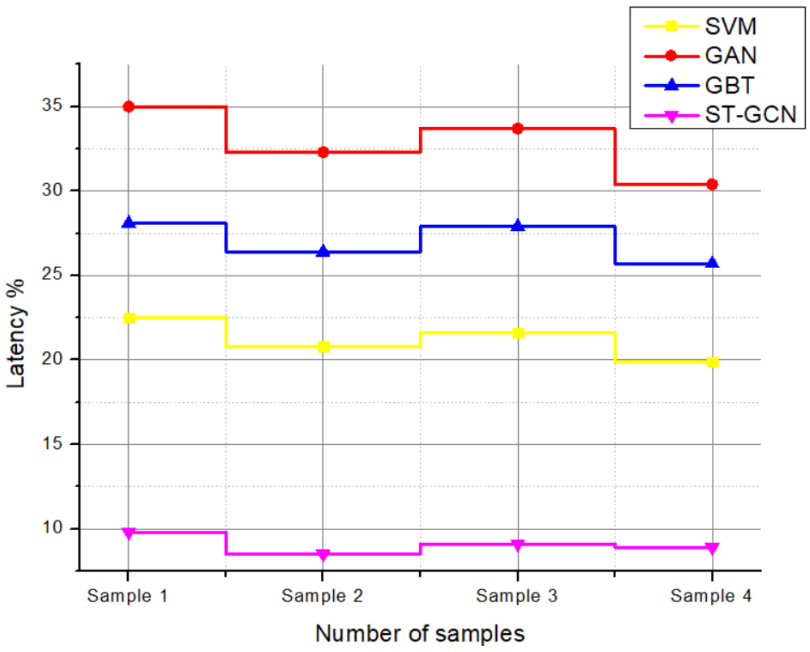

Figure 10: The analysis of detection time

In practice, the accuracy of fault detection in the smart grid is not the only parameter to consider; timeliness is also essential is shown in figure 10. There was ST-GCN model latency, which was recorded as the time taken between the introduction of a fault signal to the system and the time when prediction occurred. Since the 1D temporal convolution operations are lightweight and the model can be processed using efficient Graph processing techniques, the model can achieve a low inference time and hence be applied to real-time operations. The architecture does not require extensive historical information or data, and the network operates efficiently, ensuring that it does not compromise the reliability of the diagnosis. This renders it feasible to be combined with supervisory control and data acquisition (SCADA) systems and edge computing environments made evaluated by equation 14.

7.5Confusion Matrix Visualization

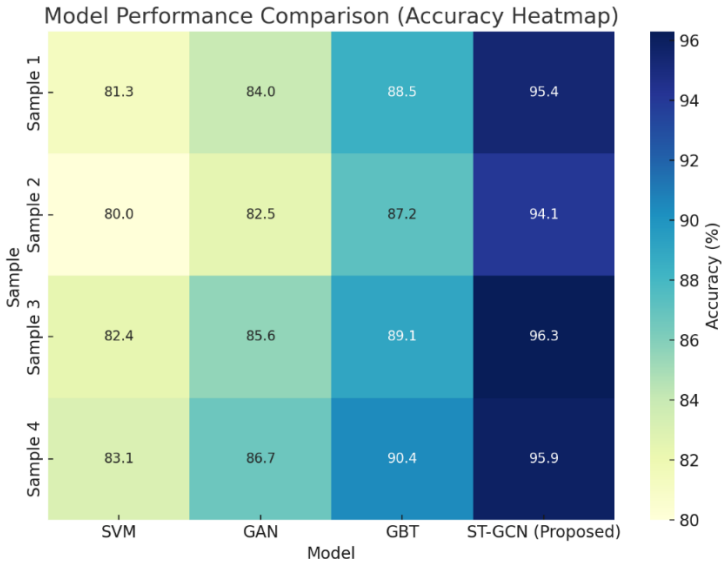

Figure 11: The confusion matrix

The effectiveness of the classification model could be visualized across all output classes, as shown in the confusion matrix is shown in figure 11. It will provide an idea of the types of faults that are most frequently misclassified and the accurate favorable rates of the classes. The higher the diagonal dominance in the matrix (of the ST-GCN model), the more precise the predictions were. A small number of off-diagonal entries indicated minor confusion among some fault types with similar electrical structure, e.g., between single-line-to-ground and double-line-to-ground faults. The visualization demonstrates that the model generates distinct and robust representations of each fault category made valuated by equation 15.

7.6 Precision vs. recall curve

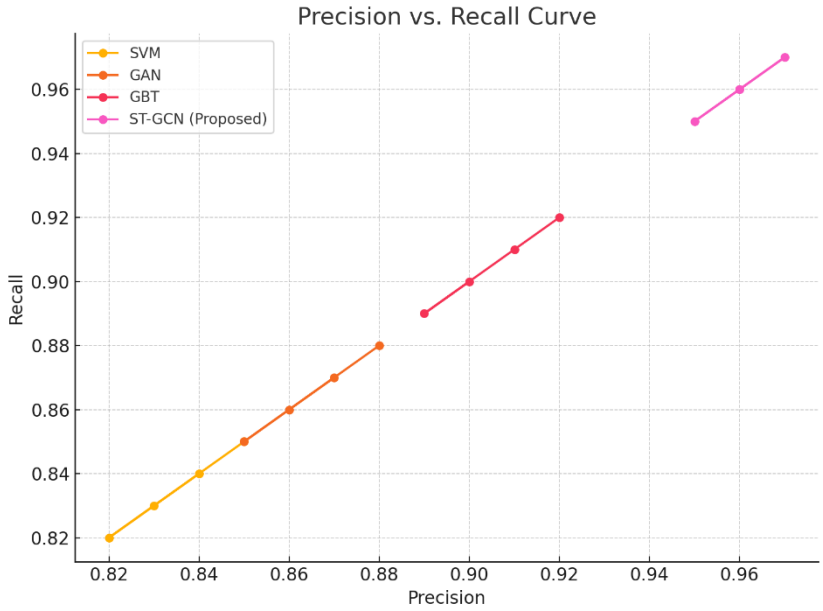

Figure 12: The analysis of precision vs recall curve

Precision-recall curves were also drawn to compare the model response against varied decision thresholds is shown in figure 12 and is calculated using equation 16. Such curves play a critical role in applications where the cost of false alarms or missed detections is high. There was a high area under the precision-recall curve (PR AUC)

over all fault conditions, indicating that under any given condition of creating the model, high precision or rejection may be attained. This is especially relevant in the field of mission-critical infrastructures, where false negative occurrences (i.e., no detection of a fault) should be avoided.

7.7 Loss convergence behavior

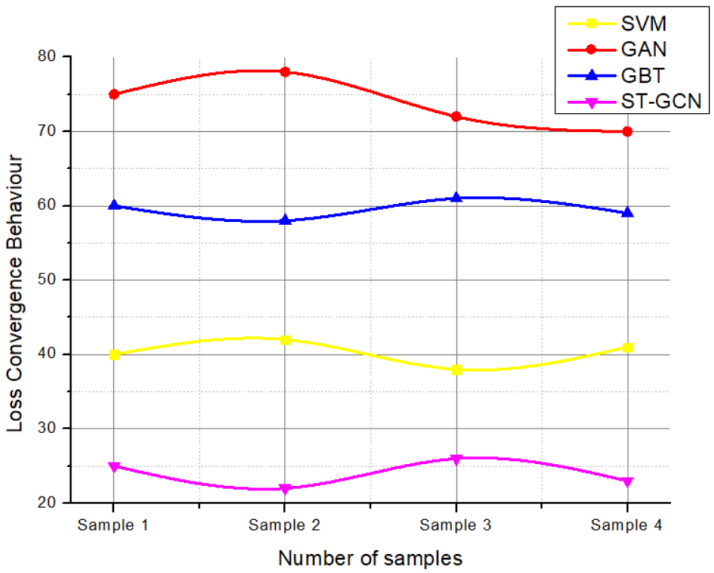

Figure 13: The analysis of loss convergence behavior

To analyze training convergence, the loss function was monitored as a function of epoch processs using the equation 17. The cross-entropy loss reflected good learning and stopped decreasing, indicating that it had stabilized and was not overfitting. Such methods of regularization, such as dropout and batch normalization, were used to achieve this stability, and early stopping mechanisms ensured that training was stopped at favorable convergence points. When the model was tested against unbalanced and noisy datasets, it exhibited untroubled and monotonic training behavior, indicating that the model is very robust and can generalize effectively is explained in figure 13.

7.8 Scalability vs network size

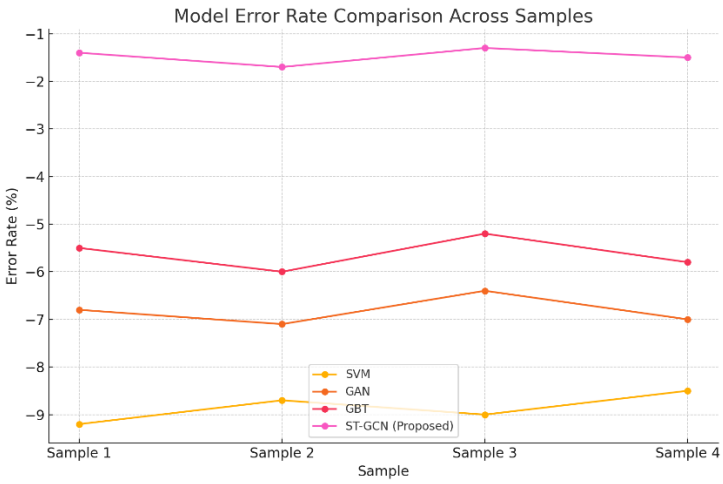

Figure 14: The analysis of scalability vs network size

Scalability tests were conducted by applying the ST-GCN model to networks of varying sizes, small networks similar to microgrids and larger networks is shown in figure 14. The model did not fail to perform as accurately and as quickly as before, though it had added a couple of nodes and edges were calculated using the equation 18. This massively parallel architecture is modular in form because it uses neighborhood aggregation and temporal segmentation of computation and distributed processes. This ensures that the model is scalable horizontally in terms of increased sensor deployments and data volumes, without significantly increasing the computational cost or deteriorating diagnostic performance.

The ST-GCN model demonstrated superior accuracy, balanced F1-scores, rapid detection, and reliable fault localization, outperforming traditional models even in sparse or large-scale networks. Its low latency, robust learning behavior, and scalable architecture validate its deployment in real-world smart grids, ensuring dependable fault detection and diagnosis in dynamic electrical environments.

7.9 Potential limitations and considerations

While the ST-GCN model demonstrates high accuracy, low latency, and robust fault localization, certain practical considerations remain. Computational overhead may increase as network size scales or when processing high-frequency sensor data, requiring efficient graph processing and optimized hardware for real-time deployment. Additionally, careful parameter tuning including the number of ST-GCN layers, temporal window size, learning rate, and batch size is necessary to maintain optimal performance and avoid overfitting. These factors should be considered when deploying the model in large-scale or resource-constrained smart grid environments.

Scalability considerations of ST-GCN

While ST-GCN demonstrates strong performance in capturing spatiotemporal patterns, its application to large-scale or high-frequency datasets poses computational and memory challenges due to the graph convolution operations and temporal modeling. To mitigate these issues, strategies such as model pruning, mini-batching, and efficient graph sampling can reduce computation, while distributed training across multiple GPUs or parallelized processing can handle larger datasets. Future work may also explore incremental or streaming-based ST-GCN variants to improve real-time processing capability without compromising accuracy.

8 Conclusion and strategic outlook

8.1Summary of findings

The presented paper introduces a new model for failure analysis and diagnosis of a weak electrical system, utilizing Spatial-Temporal Graph Convolutional Networks. The model succinctly represents the topology of the sensor data as well as its temporal texture, thereby functioning effectively in fault detection, classification, and localization. Extensive evaluation yielded excellent results in terms of accuracy, low latency, and generalizability across various network connectivities and fault scenarios.

8.2 Engineering and research implications

As an engineer, the model under consideration provided a scalable and deployable engineered solution for smart grid automation, particularly in low-voltage and loosely coupled networks. The way it was designed to work with the current infrastructure and leverage edge computing generates new avenues for decentralized diagnostics. In research, this publication fills the gap between sophisticated machine learning methods and their practical application in electrical engineering, demonstrating the effectiveness of applying graph neural networks in mission-critical systems.

8.3 Future scope: multimodal graphs, self-supervised GNNs

In the future, several directions will enhance the resiliency and agility of this framework. Multimodal graph input, e.g., integrating electrical, environmental, and cyber-physical signals through contextual awareness, could be provided. Additionally, the implementation of self-supervised learning methods on GNNs can reduce the need for large labeled datasets, making applications possible in unsupervised or semi-supervised grid setups. The better use of dynamic graph modeling and online learning capabilities will also improve the system's resilience to changing grid topologies and load profiles. While the proposed ST-GCN model demonstrates effective fault detection in electrical systems, its generalization to non-electrical networks (e.g., water distribution or traffic systems) may be limited. Future work will explore modality-agnostic approaches, adaptive temporal windows, and robustness to noisy or incomplete data to extend applicability and improve system resilience

References

[1] Y. Li, C. Xue, F. Zargari, and Yunwei Ryan Li, “From Graph Theory to Graph Neural Networks (GNNs): The Opportunities of GNNs in Power Electronics,” IEEE access, vol. 11, pp. 145067–145084, Jan. 2023, doi: https://doi.org/10.1109/access.2023.3345795.

[2] D. Chanda and N. Y. Soltani, “A Heterogeneous Graph-Based Multi-Task Learning for Fault Event Diagnosis in Smart Grid,” 2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-6, 2023.

[3] Melih BİLGEN and Necmi ALTIN, “An Overview on Reliability Analysis and Evaluation Methods Applied to Smart Grids,” Gazi Üniversitesi Fen Bilimleri Dergisi Part C Tasarım ve Teknoloji, vol. 9, no. 4, pp. 645–660, Dec. 2021, doi: https://doi.org/10.29109/gujsc.981235.

[4] J. Liu, Y. Huang, K. Chen, G. Liu, J. Yan, S. Chen, Y. Xie, Y. Yu and T. Huang, "Graph Neural Networks for Fault Diagnosis in Photovoltaic-Integrated Distribution Networks with Weak Features," Sensors, vol. 25, no. 18, 5691, Sep. 2025. https://doi.org/10.3390/s25185691

[5] H. Sun, X. Li, D. Su, J. Han, R.-H. Li, and G. Wang, "Towards Data-centric Machine Learning on Directed Graphs: a Survey," arXiv (Cornell University), Nov. 2024, doi: https://doi.org/10.48550/arxiv.2412.01849.

[6] A. Sirico and D. R. Herber, "On the Use of Geometric Deep Learning for the Iterative Classification and Down-Selection of Analog Electric Circuits," Journal of Mechanical Design, vol. 146, no. 5, Nov. 2023, doi: https://doi.org/10.1115/1.4063659.

[7] X. Zheng et al., "Towards Data-centric Graph Machine Learning: Review and Outlook," arXiv (Cornell University), Jan. 2023, doi: https://doi.org/10.48550/arxiv.2309.10979.

[8] A. Aligholian and H. Mohsenian-Rad, "GraphPMU: Event Clustering via Graph Representation Learning Using Locationally-Scarce Distribution-Level Fundamental and Harmonic PMU Measurements," IEEE Transactions on Smart Grid, vol. 14, no. 4, pp. 2960–2972, Jul. 2023, doi: https://doi.org/10.1109/tsg.2022.3225373.

[9] L. Wang, Z. Qin, T. Slangen, P. Bauer, and T. van Wijk, "Grid Impact of Electric Vehicle Fast Charging Stations: Trends, Standards, Issues and Mitigation Measures - An Overview," IEEE Open Journal of Power Electronics, vol. 2, pp. 56–74, 2021, doi: https://doi.org/10.1109/ojpel.2021.3054601.

[10] B. Fan, T. Liu, F. Zhao, H. Wu, and X. Wang, "A Review of Current-Limiting Control of Grid-Forming Inverters Under Symmetrical Disturbances," IEEE Open Journal of Power Electronics, vol. 3, pp. 955–969, 2022, doi: https://doi.org/10.1109/ojpel.2022.3227507.

[11] N. Sapountzoglou, J. Lago, and B. Raison, "Fault diagnosis in low voltage smart distribution grids using gradient boosting trees," Electric Power Systems Research, vol. 182, p. 106254, May 2020, doi: https://doi.org/10.1016/j.epsr.2020.106254.

[12] S. Yan, G. Zhang, B. Chen, C. Li, Y. Liu, and J. Yang, "Fault Diagnosis Technology of Low Voltage Circuit Breaker Based on Multilayer Perceptron," Lecture notes in electrical engineering, pp. 332–344, Jan. 2022, doi: https://doi.org/10.1007/978-981-19-1532-1_35.

[13] X. Liu, H. Wang, Z. Gao, and X. Han, "Research on fault localization of distribution transformers based on frequency response analysis and support vector machine (SVM)," Frontiers in Energy Research, vol. 12, Nov. 2024, doi: https://doi.org/10.3389/fenrg.2024.1477556.

[14] Y. M. Nsaif, Molla, A. Hussain, Afida Ayob, Yushaizad Yusof, and Muhammad, "A New Voltage Based Fault Detection Technique for Distribution Network Connected to Photovoltaic Sources Using Variational Mode Decomposition Integrated Ensemble Bagged Trees Approach," Energies, vol. 15, no. 20, pp. 7762–7762, Oct. 2022, doi: https://doi.org/10.3390/en15207762.

[15] M. Markovic, Amirhossein Sajadi, A. Florita, R. C. III, and B.-M. Hodge, "Voltage Estimation in Low-Voltage Distribution Grids with Distributed Energy Resources," IEEE Transactions on Sustainable Energy, vol. 12, no. 3, pp. 1640–1650, Feb. 2021, doi: https://doi.org/10.1109/tste.2021.3060546.

[16] H. Mirshekali, R. Dashti, A. Keshavarz, and H. R. Shaker, "Machine Learning-Based Fault Location for Smart Distribution Networks Equipped with Micro-PMU," Sensors, vol. 22, no. 3, p. 945, Jan. 2022, doi: https://doi.org/10.3390/s22030945.

[17] K. Liu, T. Kang, X. Ye, M. Bai, and Y. Fan, "A fault location method of distribution network based on XGBoost and SVM algorithm," IET Cyber-Physical Systems: Theory & Applications, vol. 6, no. 4, pp. 254–264, Oct. 2021, doi: https://doi.org/10.1049/cps2.12022.

[18] Y. Venkatachalam and T. Subbaiyan, "Intelligent fault diagnosis in power systems: A comparative analysis of machine learning-based algorithms," Expert Systems with Applications, vol. 265, p. 125945, Mar. 2025, doi: https://doi.org/10.1016/j.eswa.2024.125945.

[19] Masoumeh Zareapoor, Pourya Shamsolmoali, and J. Yang, "Oversampling Adversarial Network for Class-Imbalanced Fault Diagnosis," arXiv (Cornell University), Jan. 2020, doi: https://doi.org/10.48550/arxiv.2008.03071.

[20] X. Li, Y. Hu, J. Zheng, and M. Li, "Neural Architecture Search For Fault Diagnosis," arXiv (Cornell University), Jan. 2020, doi: https://doi.org/10.48550/arxiv.2002.07997.

[21] J. Liu, Y. Huang, K. Chen, G. Liu, J. Yan, S. Chen, Y. Xie, Y. Yu, and T. Huang, "Graph Neural Networks for Fault Diagnosis in Photovoltaic-Integrated Distribution Networks with Weak Features," Sensors, vol. 25, no. 18, p. 5691, Sep. 2025, doi: 10.3390/s25185691.

[22] D. Chanda and N. Yahya Soltani, "A Heterogeneous Graph-Based Multi-Task Learning for Fault Event Diagnosis in Smart Grid," IEEE Transactions on Power Systems, vol. 40, no. 2, pp. 1427–1438, Mar. 2025, doi: 10.1109/TPWRS.2024.3447533.

[23] Peters, I., & Kamrul, G. (2025). Applications AI-driven solar energy management system for smart grids using predictive analytics and adaptive control. Journal of Quantum Nano-Green Environmental Systems, 1(1), 14–24. https://doi.org/10.70023/qnges.251102

[24] https://www.kaggle.com/datasets/ziya07/smart-grid-monitoring-dataset