

# Generalization of Graph Neural Network Models for Distribution Grid Fault Detection

Burak Karabulut<sup>†,\*</sup>

Carlo Manna<sup>\*</sup>

Chris Develder<sup>†</sup>

<sup>†</sup>*Dept. of Information Technology, IDLab  
Ghent University – imec, Ghent, Belgium  
{burak.karabulut, chris.develder}@ugent.be*

<sup>\*</sup>*Water and Energy Transition Unit  
VITO, Mol, Belgium  
{burak.karabulut, carlo.manna}@vito.be*

**Abstract**—Fault detection in power distribution grids is critical for ensuring system reliability and preventing costly outages. Moreover, fault detection methodologies should remain robust to evolving grid topologies caused by factors such as reconfigurations, equipment failures, and Distributed Energy Resource (DER) integration. Current data-driven state-of-the-art methods use Recurrent Neural Networks (RNNs) for temporal modeling and Graph Neural Networks (GNNs) for spatial learning, in an RNN+GNN pipeline setting (RGNN in short). Specifically, for power system fault diagnosis, Graph Convolutional Networks (GCNs) have been adopted. Yet, various more advanced GNN architectures have been proposed and adopted in domains outside of power systems. In this paper, we set out to systematically and consistently benchmark various GNN architectures in an RNN+GNN pipeline model. Specifically, to the best of our knowledge, we are the first to (i) propose to use GraphSAGE and Graph Attention (GAT, GATv2) in an RGNN for fault diagnosis, and (ii) provide a comprehensive benchmark against earlier proposed RGNN solutions (RGCN) as well as pure RNN models (especially Gated Recurrent Unit (GRU)), particularly (iii) exploring their generalization potential for deployment in different settings than those used for training them. Our experimental results on the IEEE 123-node distribution network show that RGATv2 has superior generalization capabilities, maintaining high performance with an F1-score reduction of  $\sim 12\%$  across different topology settings. In contrast, pure RNN models largely fail, experiencing an F1-score reduction of up to  $\sim 60\%$ , while other RGNN variants also exhibit significant performance degradation, i.e., up to  $\sim 25\%$  lower F1-scores.

**Index Terms**—Power Systems, Fault Detection, Time Series, Graph Neural Networks, Recurrent Neural Networks

## I. INTRODUCTION

Fault diagnosis in power distribution grids is essential for maintaining grid reliability and preventing costly outages [1]. In particular, faults, typically involving short circuits caused by various factors such as weather, equipment failures, or insulation breakdowns [2], must be detected and localized quickly, often within milliseconds, to avoid further damage in the grid and minimize downtime for users. Such fault detection today has become way more challenging, due to increased (1) deployment of DERs, and (2) electrification (e.g., EV chargers). First of all, they change the conventional fault propagation patterns; e.g., faults may propagate from multiple directions within the grid as the power flow becomes bidirectional [3]. Second, inherent variability and intermittency [4] of renewable sources (e.g., variations in generation due to weather conditions) can lead to voltage fluctuations and load

imbalances [5], and thus amplify the need for adaptable fault diagnosis strategies.

Besides the increased complexity of (distribution) grids in terms of distributed generation and new loads, their topological configuration also changes. Indeed, distribution grids operate under diverse and evolving topologies due to switching operations, network reconfigurations, and equipment failures [2]. Also, new measurement devices such as Phasor Measurement Units (PMUs) and smart meters, may be introduced, while existing ones can malfunction or be retired, requiring continuous updates to measurement configurations [6]. Thus, effective fault diagnosis methods should not only be accurate and robust, but also able to adapt to evolving grid structures and diverse configurations. Such fault diagnosis in power systems comprises three tasks: *fault detection* determines whether a fault is present, *fault classification* identifies the fault type or affected phase, and *fault localization* pinpoints the fault's location in the network [7]. Existing methods for fault diagnosis in the power system literature [8]–[10], can be broadly classified as model-based methods and data-driven methods, which we discuss in turn in the next paragraphs.

*Model-based methods* can be categorized into analytical methods, protection-based methods, and signal-based methods. Analytical methods use mathematical models, such as impedance-based approaches [1], [11], [12], matrix-based approaches [13], and voltage-sag-based approaches [1], [14], to estimate fault location and infer fault type using voltage and current measurements. Protection-based methods, including overcurrent relays, differential protection, and directional relays, operate on predefined thresholds and relay-based schemes to detect and classify faults based on deviations from normal operating conditions. Signal-based methods focus on analyzing high-frequency transient signals to detect and locate faults, such as traveling wave analysis and wavelet-based techniques. Traveling wave methods are particularly effective for fault detection and localization, but require high-frequency sensors, which increase deployment costs and may not be feasible in distribution systems [13], [15]. Model-based methods typically use predefined thresholds, heuristic rules, and analytical models to detect, classify, and locate faults [1], [11], [16]. Although model-based methods are effective in conventional grid settings, they face challenges (e.g., false alarms, reduced performance in grids with distributed generation) in modern distribution networks due to their reliance on static assump-

tions about grid topology, operating conditions, and fault characteristics.

To address such challenges, early *data-driven methods* have been increasingly adopted to overcome the limitations of traditional model-based approaches, e.g., [17]–[20]. Early artificial intelligence (AI) approaches, such as decision trees, k-nearest neighbors (kNN), and support vector machines (SVMs), have been used to classify faults based on manually extracted features [8]. More recent models based on deep learning, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have shown promise by directly learning patterns from raw or minimally processed data. CNNs have been particularly effective in analyzing spatial features, while RNNs (e.g., Long Short-Term Memory (LSTM) or GRU models) leverage temporal dependencies in sequential data [7]. However, traditional AI models are designed primarily for fairly rigorously structured data (e.g., matrices of image pixels, time series at a fixed measurement frequency). Power systems, in contrast, are networks of interconnected nodes (buses) and edges (power lines), where relationships between data points are defined by non-uniform electrical connectivity rather than fixed spatial arrangements such as physical distance. Thus, methods such as CNNs struggle to fully exploit the underlying structure of the grid, particularly as new measurement configurations emerge due to the integration of DERs [16], [21].

Graph learning approaches, as GNNs pioneered by Gori et al. [22], [23], have been increasingly adopted to overcome aforementioned limitations of traditional AI models. GNNs are specifically designed to process data structured as graphs: power grid components such as buses and their interconnecting distribution power lines can naturally be represented as graph nodes and edges. Using message-passing algorithms, GNNs are able to capture dependencies between nodes and edges [21]. MansourLakouraj et al. [13] used spectral Graph Convolutional Networks (GCNs) for fault localization by leveraging the graph structure of the power system to model spatial dependencies. Spectral GCNs operate on the graph's Laplacian matrix, thus allowing for the incorporation of the global graph structure. Spectral methods require recalculating the entire graph at each step, making them computationally expensive and less efficient for the tasks that focus on local graph structures rather than global ones [21].

On the other hand, unlike spectral methods, spatial GCNs are more efficient as they operate directly on the graph's node and edge features, aggregating information from neighboring nodes to capture local dependencies and avoiding the redundant recalculation of the entire graph. Hence, for tasks such as fault localization, spatial GCNs are often preferred. For example, [16] and [24] used spatial GCNs, specifically from the framework proposed by Kipf and Welling [25], for the fault localization task. In [2], the authors propose a Gated Graph Neural Network (GGNN) that leverages GNNs, while also using gated mechanisms to update the hidden states of the nodes over time. However, both of these approaches do not account for the time-series nature of fault events, as the proposed

methodology models only spatial relationships. In contrast, Bang et al. [26] experimented with a 1D-Convolutional GNN, where they use 1D-CNN to capture temporal dependencies, and then a GCN [25] to extract spatial correlations for fault diagnosis tasks. Later, in [7], they enhanced this hybrid approach by incorporating Long Short-Term Memory (LSTM) networks. Their approach effectively captures both spatial and temporal dependencies.

However, the adaptability of GCNs [26] to evolving grid topologies is limited due to the reliance of the model on a fixed adjacency matrix, preventing it from generalizing to unseen nodes when the graph structure changes. In contrast, the work proposed in [11] incorporates Graph Attention Networks (GATs), which use an attention mechanism to learn node importance based on the local graph structure. This ability to focus on relevant nodes in the graph, rather than relying on a fixed adjacency matrix to calculate node representations, makes GATs more adaptable to the changes in graph topology. Ngo et al. [27] extend GAT models by incorporating 1D-CNN to extract temporal information to improve fault diagnosis accuracy by using both voltage and branch current inputs. However, the robustness of their model under evolving grid conditions remains untested.

In conclusion, GNN-based models show great promise for distribution grid fault diagnosis, yet (i) there has not been a rigorous performance comparison of the various recent GNN architectures for fault diagnosis, and (ii) their conceptual potential to deal with variations/extensions of the grid in terms of measurement devices and/or topological changes has not been studied. The current paper forms a pilot study to try and address these issues, in particular by

- (1) Proposing RNN+GNN architectures previously unexplored for power system fault detection i.e., the RGSAGE and RGAT models;
- (2) Conducting a systematic quantitative benchmarking of the proposed models against the state-of-the-art RGCN model as well as non-GNN models (based on GRUs); and
- (3) Evaluating the generalization capabilities of the various GNN-based solutions for deployments with varying number of PMUs, on a realistic, sizeable distribution grid configuration (the IEEE 123-node network).

We also discuss next steps by which we will extend the current pilot to a broader scope of topological variations and fault diagnosis applications.

## II. METHODOLOGY

The limitations identified from the literature review above highlight the need for a structured evaluation of GNN architectures and their ability to handle evolving grid topologies. Next, we discuss the RNN+GNN pipeline components and particularly the GNN models in more detail.

### A. Temporal Feature Extraction – Recurrent Neural Networks

Fault data in distribution grids inherently constitute time series data. To capture these temporal dependencies, recurrent

neural networks (RNNs) are widely used [28]. Common RNN variants include Long Short-Term Memory (LSTM) cells and Gated Recurrent Units (GRUs). Faults in power systems can typically be inferred from a relatively limited time window of measurements. Given the simpler architecture of GRUs, which are also computationally more efficient than LSTMs [29], we will use GRU cells as part of our RGNN model to extract temporal dependencies. To assess the contribution of the GNN part in our RGNN pipeline, we will benchmark them against pure RNN solutions, namely (i) a shared *GRU* model applied in parallel to all the PMUs individually, to locally infer a failure (and then use majority voting to assess whether a fault occurs), and (ii) an *aggregated GRU* model, where we apply a shared GRU model to each PMU node and then join their outputs with a max pooling layer to output a single fault classification.

### B. Spatial Feature Extraction – Graph Neural Networks

A distribution network is naturally represented as a graph  $G = (V, E, A, H)$ , where  $V$  is the set of  $N$  nodes (buses), i.e.,  $|V| = N$ , and  $E$  is the set of edges (branches) representing physical connections between nodes. The network structure is captured by the adjacency matrix  $A \in \mathbb{R}^{N \times N}$  (with  $A_{uv} = 1$  if nodes  $u$  and  $v$  are connected by an edge). In GNNs, each node  $v \in V$  is associated with a feature vector  $H_v \in \mathbb{R}^F$ , forming the node feature matrix  $H \in \mathbb{R}^{N \times F}$ .

Graph learning aims to map node and edge attributes to target outputs to target outputs, modeled as:

$$\hat{y} = f(G, \theta), \quad (1)$$

where  $\theta$  represents the model parameters, and  $\hat{y}$  is the predicted output. GNNs iteratively aggregate information across nodes, in multiple layers:

$$H^{(k+1)} = f(H^{(k)}, A; \theta), \quad (2)$$

where  $H^{(k)} \in \mathbb{R}^{N \times d_k}$  is the node representation matrix. This is composed of row vectors  $h_v^{(k)}$ , representing each node  $v \in V$  with a  $d_k$ -dimensional feature vector at layer  $k$  (typically the same  $\forall k$ ). This iterative aggregation process can be viewed as a message passing framework, where each node  $u$  updates its representation by aggregating the features from its neighbors  $v \in \mathcal{N}(u)$ . GNN variants vary in how exactly this aggregation happens and what non-linear activation functions (which we will note as  $\phi(\cdot)$ ) are used to obtain  $f$  in eq. (2). In our work, we consider four GNN variants:

1) *Graph Convolutional Network (GCN)* aggregates neighbor features using a normalized adjacency matrix  $\hat{A}$  [25], updating the feature matrix  $H^{(k+1)}$  as:

$$H^{(k+1)} = \phi(\hat{A}H^{(k)}W), \quad W \in \mathbb{R}^{d_k \times d_k} \quad (3)$$

$$\text{with } \hat{A} = D^{-1/2}A'D^{-1/2}, \quad \hat{A} \in \mathbb{R}^{N \times N}. \quad (4)$$

Here  $W$  is a learnable weight matrix,  $A' = A + I$ ,  $D$  is the degree matrix, and  $d_k$  is the dimensionality of the node feature vector at layer  $k$ . Note that this assumes a fixed topology, defined through  $A$ .

2) *GraphSAGE (Graph Sample and Aggregation)* [30] updates a node's representation using a learned aggregation function that does not explicitly rely on the graph structure (as opposed to GCNs, which need a fixed  $A$ ). This implies that a trained GraphSAGE can easily be used for a different graph topology at inference. The updated feature vector for node  $v$  at layer  $k + 1$  is given by:<sup>1</sup>

$$h_v^{(k+1)} = \phi\left(W \cdot \text{concat}\left(h_v^{(k)}, \text{AGG}\left(\{h_u^{(k)} | u \in \mathcal{N}(v)\}\right)\right)\right), \quad (5)$$

where  $W$  is a learnable weight matrix,  $\text{concat}$  is the concatenation operation, and  $\text{AGG}$  represents an aggregation function. Common options for  $\text{AGG}$  include max pooling, average pooling, and more complex aggregators such as LSTMs.

3) *Improved Graph Attention Network (GATv2)* introduces an attention-based message passing mechanism that adaptively weighs the contributions of neighboring nodes [31]. The representation of a node  $v$  at layer  $k + 1$  is computed as:

$$h_v^{(k+1)} = \phi\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} W h_u^{(k)}\right) \quad \text{with} \quad (6)$$

$$\alpha_{vu} = \frac{\exp\left(a^T \cdot \text{LeakyReLU}(W_1 h_v^{(k)} + W_2 h_u^{(k)})\right)}{\sum_{j \in \mathcal{N}(v)} \exp\left(a^T \cdot \text{LeakyReLU}(W_1 h_v^{(k)} + W_2 h_j^{(k)})\right)}, \quad (7)$$

where  $a$  is a learnable attention vector and  $\text{LeakyReLU}$  denotes a nonlinear activation. This attention mechanism allows the model to focus on the most relevant neighbors based on the learned coefficients. The earlier Graph Attention Network (GAT) proposed by Velickovic et al. [32] computed attention as a linear combination of features  $(W h_v^{(k)} || W h_u^{(k)})$  which applies the same linear transformation  $W$  to both nodes before concatenation, and the nonlinearity is applied after attention is computed. In contrast, improved GAT (GATv2) provides (i) more flexibility by using separate learnable weights  $(W_1 h_v + W_2 h_u)$  for each node, which accounts for asymmetric neighbor contributions, and (ii) greater expressiveness by applying nonlinearity before attention calculation.

### C. Proposed RGNN framework

In the literature, both RNNs and GNNs have been adopted individually for fault diagnosis in power distribution systems, e.g., respectively in [33] and [1]. However, this means the model either focuses on extracting temporal (RNN) or spatial (GNN) information, but not both. Hence, combining RNNs and GNNs into an RGNN have been proposed to jointly model temporal and spatial dependencies. In our work, we implement an RGNN pipeline where the input is first processed by a GRU cell to capture temporal dependencies, followed by a GNN model to extract spatial relationships. We will particularly

<sup>1</sup>The original GraphSAGE model alternatively proposes a sampled subset  $\mathcal{N}'(v) \subset \mathcal{N}(v)$  rather than the full set of neighbors. Given the relatively small size of power distribution networks (as opposed to e.g., social media graphs), we apply GraphSAGE in its full-batch setting without any subsampling.

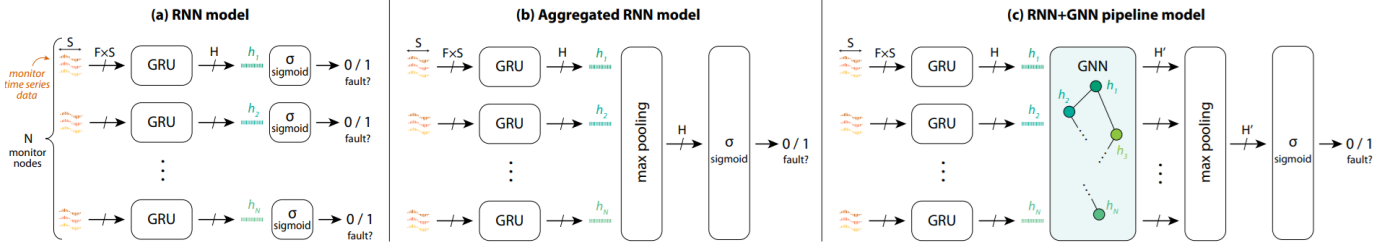


Fig. 1. Illustration of the (a) shared RNN for local inference per PMU, (b) aggregated RNN model, and (c) RNN+GNN pipeline architecture. The GNN layers consist of message-passing, dropout, and batch normalization, with each variant using its respective architecture (see §II-B). Further,  $N$  represents the number of PMU nodes,  $F$  is the number of features (e.g., phase voltages, currents),  $S$  is the time series sequence length,  $H$  denotes the dimension of the GRU hidden state representation (and hence output), and  $H'$  is the dimension of the GNN layer's output node representations.

explore various recent GNN models. For that GNN part, we are the first to consider GraphSAGE and graph attention (GAT, GATv2) models — which incorporate attention mechanisms and inductive learning capabilities, respectively — for the considered power distribution fault diagnosis application. Figure 1 summarizes our proposed pipeline architecture. For each of the  $N$  PMU nodes, the input time series of  $S$  timesteps (see details in §III-B), is processed by a GRU. The node representations given by the GRU outputs are then processed by a GNN model (with the PMUs being the nodes and their connected edges defined based on the grid topology; details in §III-B). The final GNN layer's node representations are then aggregated with a max pooling layer, and finally a sigmoid produces the single binary classification output.

### III. EXPERIMENTAL SETUP

#### A. Fault Types

Faults in power systems are typically classified into two types: open circuit and short circuit faults [34]. Open circuit faults occur when conductors break, disrupting circuit continuity, while short circuit faults result from unintended contact between conductors or with the ground, causing abnormal current surges. Short circuit faults, especially due to their frequency and impact on system stability, are the focus of most detection studies. These faults can be further categorized into asymmetrical faults (line-to-ground (LG), line-to-line (LL), and double line-to-ground (LLG)), and symmetrical faults such as three-phase faults. LG faults are the most common, making up approximately 75%–80% of power system faults [34]–[36], and are the focus of our study.

#### B. Simulation Setup and Data Collection

The proposed models are data-driven, and their performance depends on high-quality data. Given the rarity of faults and the limited measurement infrastructure to capture them, we simulate data using OpenDSS [37] and the PyDSS interface [38]. We conduct our experiments on the IEEE 123-bus feeder, a widely-used benchmark system in fault diagnosis studies [1], [7], [26], operating at a nominal voltage of 13.2 kV and a frequency of 60 Hz. Fault durations typically range from 20 ms to 50 ms in distribution systems; for consistency, we consider a fault duration of 20 ms. As indicated in Fig. 2, faults are

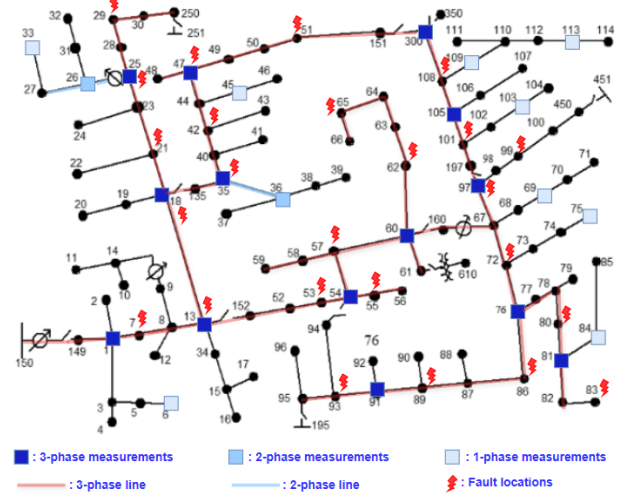


Fig. 2. IEEE 123-node feeder with fault locations and voltage measurements.

dynamically injected at 25 locations, and measurements are gathered by  $N = 25$  PMUs. We collect measurement data at 1 ms resolution, thus providing 20 data points per fault for detailed fault profile analysis.

From the measurement data, we recorded 60 ms fault windows, starting 40 ms before the fault. As input to our classification models (GRU or RNN), we feed a time series of  $S = 20$  timesteps, by considering sliding windows of 20 ms from the collected samples. From a total of 3 load scenarios, 25 fault locations, and 25 PMUs, 1,875 unique PMU data streams are collected. Each of these PMU's data contributes 41 windows, 21 non-faulty and 20 faulty, resulting in a nearly balanced dataset mixture of faults and fault-free cases. Table I summarizes our data generation configurations: in total, 76,875 sliding windows were collected, including 37,500 fault cases and 39,375 non-fault cases. The full dataset was then randomly split into a ~68% training, ~17% validation, and ~15% test set.

For evaluating model robustness to changing PMU configurations, different PMU configurations are subsampled from the entire 25-PMU setup (see Table II). We note that the configurations with a lower number of PMUs are subsets of

TABLE I  
DATASET INFORMATION

Elements	Value	Number
Fault type	LG	1
Fault resistance ( $\Omega$ )	0.1, 1.0, 10	3
Fault position (Buses)	7, 13, 18, 21, 25, 29, 35, 42, 47, 51, 53, 55, 57, 62, 65, 72, 80, 83, 86, 89, 93, 97, 99, 101, 108	25
PMU location	1, 6, 13, 18, 25, 26, 33, 35, 36, 45, 47, 54, 60, 69, 75, 76, 81, 84, 91, 97, 103, 105, 109, 113, 300	25
Load Scenarios (p.u.)	0.7, 1.0, 1.3	3

Dataset Distribution			
Category	Train Set	Validation Set	Test Set
Fault Cases	25,610	6,402	5,488
No Fault Cases	26,890	6,723	5,762
Total Cases	52,500	13,125	11,250

TABLE II  
LIST OF PMUS USED IN EACH CONFIGURATION. NODES DROPPED FROM ONE SUBSET TO THE NEXT ARE TYPESET IN RED.

# PMUs	Bus locations (see Fig. 2)
25	1, 6, 13, 18, 25, 26, 33, 35, 36, 45, 47, 54, 60, 69, 75, 76, 81, 84, 91, 97, 103, 105, 109, 113, 300
19	1, <del>6</del> , 13, 18, 25, 26, 33, 35, <del>36</del> , 45, 47, 54, 60, 69, 75, <del>76</del> , 81, <del>84</del> , 91, 97, <del>103</del> , 105, 109, <del>113</del> , 300
15	1, 6, 13, 18, <del>25</del> , 26, <del>33</del> , 35, 36, <del>45</del> , 47, 54, 60, 69, 75, 76, 81, 84, <del>91</del> , 97, <del>103</del> , 105, 109, <del>113</del> , 300
11	1, 6, 13, <del>18</del> , 25, 26, 33, 35, 36, 45, 47, <del>54</del> , <del>60</del> , <del>69</del> , 75, 76, 81, 84, 91, 97, <del>103</del> , 105, 109, <del>113</del> , 300
7	<del>1</del> , 6, 13, 18, 25, <del>26</del> , 33, 35, 36, 45, 47, 54, 60, 69, <del>75</del> , 76, 81, 84, 91, <del>97</del> , <del>103</del> , 105, 109, <del>113</del> , 300

those with a higher number. Those with fewer PMUs include mostly higher-degree nodes, which should be more informative than peripheral ones, thus reflecting real-world PMU settings. Modern distribution grids evolve due to DER integration (e.g., solar panels, wind turbines, residential battery storage) and electrification (e.g., EV chargers). Since DERs are typically connected at leaf nodes, new measurement devices are often introduced at these points, while existing peripheral measurement devices may be retired or repurposed. Our approach reflects these dynamic updates in measurement infrastructure.

### C. Model Training and Evaluation

We used Pytorch with the PyG library [39] for GNNs. We used three-phase voltages, currents, and their respective phase angles as features and applied Z-score feature normalization (resulting in a mean of 0 and a standard deviation of 1). Our GRU and GNN layers have a hidden state size of 128, and we adopt binary cross-entropy with logits as loss function for our fault classification objective. To alleviate overfitting, we apply dropout and batch normalization, and attention dropout for the GAT models. All models are trained for 35 epochs using the AdamW optimizer [40].

For the GNN part, we only consider the  $N$  measured buses as graph nodes, ensuring that the graph is tailored for

fault detection tasks and aligns with real-world measurement constraints. Other works [1], [41] include unmeasured buses with zero feature values, which may be beneficial for fault localization but could lead to noise propagation and over-smoothing issues [42]. Excluding unmeasured buses also limits the graph size and computational complexity. In the GraphSAGE-based models (RGSAGE), given the small graph size ( $N \leq 25$ ), we use the full-batch setting and adopt either max or mean pooling as the aggregation function (as a lightweight yet effective alternative to more complex methods such as LSTMs [30]).

## IV. RESULTS AND DISCUSSION

We now discuss the quantitative performance analysis of the various RNN and RNN+GNN models, as performed on the IEEE 123-bus topology. Figure 3 shows the various architectures' fault detection performance (i.e., binary classification fault vs. no fault), for models trained on the 11-PMU setup.

Let's first focus on comparing the various models' fault detection performance, when testing them in similar conditions as they were trained for. Looking at Fig. 3, the test results for 11-PMU setup (i.e., what the model was trained for) show that all models, including GRUs, perform well on this relatively simple fault detection task ( $F1 \simeq 1$ ). This suggests that GNN-based models' conceptual strengths do not shine for this relatively simple classification problem. Yet, we hypothesize that GNNs may offer benefits over when we consider more advanced diagnosis tasks (e.g., fault localization, see also §V).

Now, more interestingly, let's focus on the models' *generalization* capabilities and consider test results for a different number of PMUs than the training set ( $N \neq 11$ ). First, for the case of *fewer* PMUs ( $N = 7$ ), we note that all methods mostly maintain high performance ( $F1$  still close to 1) — which is intuitive, as the  $N = 7$  PMUs used at inference, are included in those seen during training. Second, switching to cases with *more* PMUs, i.e., new PMU nodes are included at test time ( $N > 11$ ), we observe that the RRN-only models (GRU, aggregated GRUs) largely fail with a significant performance drop ( $F1 \leq 0.4$  on average). In contrast, the RGNN pipeline models all perform quite stronger. This better generalization capacity is what we intuitively expect from their ability to process topological relationships (i.e., structure and connectivity of the graph) to aggregate the distributed inputs into a global view.

Still, not all GNN architectures perform equally well. (i) The *RGCN* variant performs the most poorly even for a small number of added PMUs (i.e.,  $N = 15$ ). This is possibly due to GCN relying on a fixed topology (through the fixed adjacency matrix  $A$ ). (ii) Our proposed *RSAGE* approach performs slightly better (than RGCN), particularly when using max pooling.<sup>2</sup> Still, there is a significant performance drop (down to  $F1 \simeq .75$ ). (iii) Our second proposal, of using *GATv2*, is the only one that maintains performance, even for test results

<sup>2</sup>That max pooling works better for detecting faults — of which the effects may differ quite a bit from one monitor to the next — intuitively was to be expected.



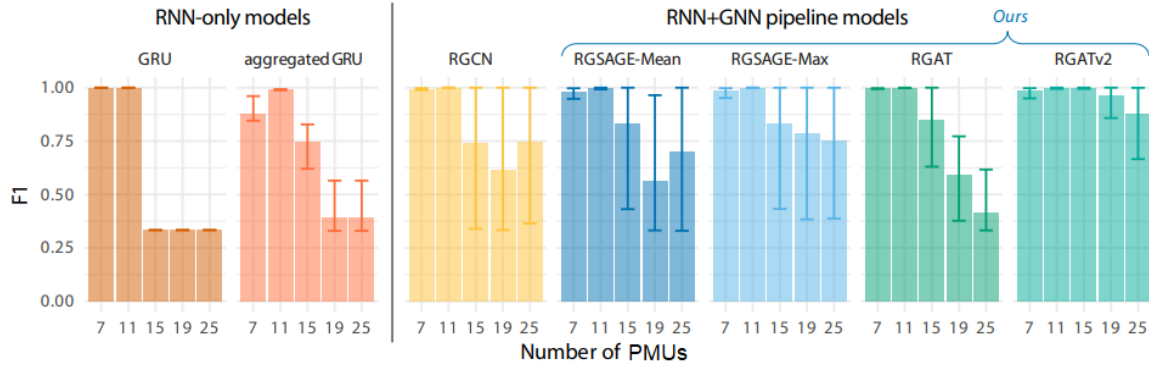


Fig. 3. F1 Scores for Fault Event Detection using RNN-only models (GRU-based; left) and GNN-based models (right). All models are trained on the 11-PMU setup and evaluated across all configurations (7–25 PMUs; see Table II for details). Error bars show the 90% confidence intervals across 5 models trained with different random seeds.

at  $N = 25$ , where the majority of the PMUs were unseen during training. The asymmetric processing across GNN edges (cf. eq. (7)) — which distinguishes it from the original GAT — seems crucial to effectively cater for different node behavior. Indeed, the added PMUs for  $N = 25$  are mostly leaf nodes (see Fig. 2), less centrally located than the initial case of  $N = 11$  PMUs seen during training.

In summary, our results show that GNN-based models are more robust compared to pure RNN-based models for distribution grid fault detection. In particular, in terms of generalization to unseen PMU configurations, our proposed RGATv2 outperforms both traditional RNN-based models and the state-of-the-art RGCN variant.

## V. CONCLUSION AND FUTURE WORK

The aim of this paper’s pilot study was to comprehensively and systematically (i) present and compare the various GNN-based models for distribution grid fault detection, and particularly (ii) analyze the generalization capacity of those models when deploying them in configurations different from those seen at model training time. For (i), we consider the IEEE 123-bus system, and train models for the simple binary classification case (fault vs. no fault). In terms of models, we compare advanced RNN+GNN models (RGNN in short) as well as pure RNN-based baselines. In particular, we are the first to propose GraphSAGE and Graph Attention (GAT/GATv2) models for this application. For (ii), we consider models trained on a  $N = 11$  PMU setup and then test for  $N \neq 11$ .

Our pilot study indicates that, first of all, for fixed deployments, where models are trained and tested under the same conditions (with  $N = 11$  in our experiments), all considered models (including pure RNN models) are adequate. Still, we hypothesize this may no longer be the case when moving to more challenging diagnosis applications. Therefore, our future work aims to study fault type classification and particularly fault localization. Nevertheless, the current pilot experiments studying generalization, considering a varying number of PMU nodes, already highlighted the potential benefits of GNN-based

approaches. Specifically, our newly proposed adoption of the RGATv2 model for fault detection demonstrated superior performance. While pure RNN-based solutions largely fail (F1 below 50%) to perform when adding new PMU nodes, even the previously proposed RGCN model as well as our other proposal, RGSAGE, suffers drops in F1-score of  $\sim 25\%$ .

As indicated above, we plan to extend this pilot study to more detailed fault diagnosis tasks, especially fault location. Moreover, we will extend the generalization scenarios to a broader scope of topological variations (e.g., switching operations, network reconfigurations), in light of recent evolutions of today’s power grid. For instance, as opposed to the more conventional fault scenarios studied above, distributed energy resources (DERs) — typically renewable sources such as photovoltaics — can introduce larger variability in fault characteristics, leading to more complex fault patterns.

## ACKNOWLEDGMENT

Research reported in this publication was supported by VITO grant number VITO\_UGENT\_PhD\_2301. This research was also partially funded by the Flemish Government (AI Research Program).

## REFERENCES

- [1] K. Chen, J. Hu, Y. Zhang, Z. Yu, and J. He, “Fault location in power distribution systems via deep graph convolutional networks,” *IEEE J. Sel. Areas Commun.*, vol. 38, p. 119, Jan 2020.
- [2] J. T. de Freitas and F. G. F. Coelho, “Fault localization method for power distribution systems based on gated graph neural networks,” *Electr. Eng.*, vol. 103, pp. 2259–2266, 2021.
- [3] A. Karimi, Y. Wu, M. Koyuturk, and R. H. French, “Spatiotemporal graph neural network for performance prediction of photovoltaic power systems,” in *Proc. Proc. 3rd Annu. Conf. on Innov. Appl. of Artif. Intell. (IAAI 2021)*, (Vancouver, Canada), pp. 1–8, AAAI, Feb. 2021.
- [4] G. Ren, J. Wan, W. Wang, J. Liu, F. Hong, and D. Yu, “Quantitative insights into the differences of variability and intermittency between wind and solar resources on spatial and temporal scales in China,” *J. Renew. Sustain. Energy*, vol. 13, no. 4, p. 043307, 2021.
- [5] B. B. Adetokun, C. M. Muriithi, J. O. Ojo, and O. Oghoroda, “Impact assessment of increasing renewable energy penetration on voltage instability tendencies of power system buses using a QV-based index,” *Sci. Rep.*, vol. 13, p. 9782, 2023.
- [6] A. Ojha, V. Panwar, and P. Kant, “Optimization techniques for network reconfiguration in distribution systems: An overview,” *Natl. J. Multidiscip. Res.*, vol. 3, no. 3, pp. 68–71, 2018.

- [7] B. L. H. Nguyen, T. V. Vu, T.-T. Nguyen, M. Panwar, and R. Hovsapian, "Spatial-temporal recurrent graph neural networks for fault diagnostics in power distribution systems," *IEEE Access*, vol. 11, p. 3273292, May 2023.
- [8] K. Chen, C. Huang, and J. He, "Fault detection, classification and location for transmission lines and distribution systems: a review on the methods," *High Volt.*, vol. 1, no. 1, pp. 25–33, 2016.
- [9] P. K. Ganivada and P. Jena, "A fault location identification technique for active distribution system," *IEEE Trans. Ind. Inform.*, vol. 18, no. 5, pp. 3000–3010, 2022.
- [10] Y. Ju, X. Tian, H. Liu, and L. Ma, "Fault detection of networked dynamical systems: A survey of trends and techniques," *Int. J. Syst. Sci.*, vol. 52, pp. 3390–3409, Dec 2021.
- [11] Z. Wang, B. Huang, B. Zhou, J. Chen, and Y. Wang, "An enhanced fault localization technique for distribution networks utilizing cost-sensitive graph neural networks," *Processes*, vol. 12, no. 2312, 2024.
- [12] J. Tavoosi, M. Shirkhani, A. Azizi, S. U. Din, A. Mohammadzadeh, and S. Mobayen, "A hybrid approach for fault location in power distributed networks: Impedance-based and machine learning technique," *Electr. Power Syst. Res.*, vol. 210, p. 108073, 2022.
- [13] M. MansourLakouraj, R. Hossain, H. Livani, and M. Ben-Idris, "Application of graph neural network for fault location in pv penetrated distribution grids," in *Proc. N. Am. Power Symp. (NAPS 2021)*, IEEE, 2021.
- [14] S. Turizo, G. Ramos, and D. Celeita, "Voltage sags characterization using fault analysis and deep convolutional neural networks," *IEEE Trans. Ind. Appl.*, vol. 58, no. 3, 2022.
- [15] F. Liu, L. Xie, K. Yu, Y. Wang, X. Zeng, L. Bi, and X. Tang, "A novel fault location method based on traveling wave for multi-branch distribution network," *Electr. Power Syst. Res.*, vol. 224, p. 109753, 2023.
- [16] M. Mansour Lakouraj, R. Hossain, H. Livani, and M. Ben-Idris, "Application of graph neural network for fault location in PV penetrated distribution grids," in *Proc. N. Am. Power Symp. (NAPS 2021)*, IEEE, 2021.
- [17] H. Livani, C. Y. Evrenosoğlu, and V. A. Centeno, "A machine learning based faulty line identification for smart distribution network," in *Proc. N. Am. Power Symp. (NAPS 2013)*, pp. 1–5, IEEE, 2013.
- [18] H. Livani and C. Y. Evrenosoğlu, "A fault classification and localization method for three-terminal circuits using machine learning," *IEEE Trans. Power Deliv.*, vol. 28, pp. 2282–2290, Oct 2013.
- [19] Z. S. Hosseini, M. Mahoor, and A. Khodaei, "AMI-enabled distribution network line outage identification via multi-label SVM," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5470–5472, 2018.
- [20] 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," *Int. J. Electr. Power Energy Syst.*, vol. 133, p. 107102, 2021.
- [21] W. Liao, B. Bak-Jensen, J. R. Pillai, and Y. Wang, "A review of graph neural networks and their applications in power systems," *J. Mod. Power Syst. Clean Energy*, vol. 10, no. 2, pp. 345–360, 2022.
- [22] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, 2005.
- [23] F. Scarselli, M. Gori, A. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, 2009.
- [24] X. Ma, W. Zhen, H. Ren, G. Zhang, K. Zhang, and H. Dong, "A method for fault localization in distribution networks with high proportions of distributed generation based on graph convolutional networks," *Energies*, vol. 17, no. 22, p. 5758, 2024.
- [25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv:1609.02907*, 2016.
- [26] B. L. H. 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 *Proc. IEEE 1st Ind. Electron. Soc. Annu. On-Line Conf. (ONCON 2022)*, pp. 1–6, IEEE, 2022.
- [27] Q.-H. Ngo, B. L. H. Nguyen, J. Zhang, K. Schoder, H. Ginn, and T. Vu, "Deep graph neural network for fault detection and identification in distribution systems," 2024.
- [28] Y. Cao, J. Tang, S. Shi, D. Cai, L. Zhang, and P. Xiong, "Fault diagnosis techniques for electrical distribution network based on artificial intelligence and signal processing: A review," *Processes*, vol. 13, p. 48, 2025.
- [29] R. Cahuantzi, X. Chen, and S. Guttel, "A comparison of lstm and gru networks for learning symbolic sequences," in *Proc. Int. Conf. SAI Intell. Comput. (SAI 2023)*, vol. 739 of *Lecture Notes in Networks and Systems*, pp. 722–740, Springer, Cham, 2023.
- [30] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Conf. on Neural Inf. Process. Systems (NeurIPS 2017)*, (Long Beach, CA, USA), Neural Information Processing Systems Foundation, 2017.
- [31] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?," in *Proc. Int. Conf. on Learning Representations (ICLR 2022)*, OpenReview.net, 2022.
- [32] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. on Learning Representations (ICLR 2018)*, OpenReview.net, 2018.
- [33] J. Q. Yu, Y. Hou, A. Y. S. Lam, and V. O. K. Li, "Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks," *IEEE Trans. Smart Grid.*, vol. 10, no. 2, pp. 1694–1703, 2019.
- [34] S. S. Gururajapathy, H. Mokhlis, and H. A. Illias, "Fault location and detection techniques in power distribution systems with distributed generation: A review," *Renew. Sustain. Energy Rev.*, vol. 74, pp. 949–958, 2017.
- [35] J. D. Glover, M. S. Sarma, and T. Overbye, *Power System Analysis & Design*. Boston, MA, USA: Cengage Learning, SI version ed., 2012.
- [36] M. Mousa, S. Abdelwahed, and J. Kluss, "Review of fault types, impacts, and management solutions in smart grid systems," *Smart Grid Renew. Energy*, vol. 10, pp. 98–117, April 30 2019.
- [37] The Electric Power Research Institute (EPRI), "OpenDSS." <https://www.epri.com/pages/sa/opendss>.
- [38] National Renewable Energy Laboratory (NREL), "PyDSS interface." <https://www.nrel.gov/grid/pydss.html>.
- [39] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv:1903.02428*, 2019.
- [40] R. Llugsi, S. E. Yacoubi, A. Fontaine, and P. Lupera, "Comparison between Adam, AdaMax and Adam W optimizers to implement a weather forecast based on neural networks for the Andean city of Quito," in *Proc. IEEE 5th Ecuador Tech. Chapters Meet. (ETCM 2021)*, pp. 1–6, IEEE, 2021.
- [41] H. Sun, S. Kawano, D. N. Nikovski, T. Takano, and K. Mori, "Distribution fault location using graph neural network with both node and link attributes," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. - Europe (ISGT Europe 2021)*, (Espoo, Finland), IEEE, October 2021.
- [42] T. K. Rusch, M. M. Bronstein, and S. S. Mishra, "A survey on oversmoothing in graph neural networks," *arXiv:2303.10993*, 2023.