Deep Attention GRU-GRBM with Dropout for Fault Location in Power Distribution Networks

1st Mahdi Khodayar
Department of Computer Science
University of Tulsa
Tulsa, Oklahoma, United States
mahdi-khodayar@utulsa.edu

2nd Ali Farajzadeh Bavil
Department of Computer Science
University of Tulsa
Tulsa, Oklahoma, United States
ali-farajzadeh@utulsa.edu

3rd Mohammad E. Khodayar
Department of Electrical and Computer Engineering
Southern Methodist University
Dallas, Texas, United States
mkhodayar@smu.edu

***Abstract*—Effective fault location algorithms contribute to reducing the recovery and restoration time and improve the resilience of the power distribution networks. The existing machine learning-based approaches for fault location exhibit limitations, notably the absence of unsupervised feature learning, disregarding the capture of semantic features, and overlooking task-relevant features. This paper introduces the deep-attention Gated Recurrent Unit Gaussian Restricted Boltzmann Machine (GRU-GRBM) framework for fault location and classification. It combines an attention-enhanced GRU for accurate task-relevant temporal feature extraction, a GRBM-based autoencoder for unsupervised generative feature learning, and a sparse deep Rectified Linear Unit (ReLU) network with a mutual information (MI)-based dropout technique for supervised estimation of fault location and class. The proposed structure is shown to outperform the state-of-the-art methods on the IEEE 123-bus system through generative feature extraction, attention mechanisms, and feature sparseness.**

***Index Terms*—Fault Classification, Fault Location, Power Distribution Networks**

I. Introduction

Maintaining a continuous electricity supply poses challenges, mainly attributed to common faults such as single line-to-ground, line-to-line, double-line-to-ground, and three-phase faults. Recent research has focused on locating and classifying faults through methods categorized as impedance-based, traveling wave-based, and machine learning-based approaches. Impedance-based techniques, utilized in [1], determine fault location by analyzing the network impedance derived from voltage and current measurements. Travel wave-based fault location methods, introduced in [2], analyze time delays in the arrival times of electrical waves. While effective in theory, the practical implementation of these methods is often economically challenging.

Machine learning-based fault location methods mainly leveraging deep learning algorithms, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Graph Convolutional Networks (GCN), represent a notable advancement in learning patterns for precise fault location estimation. The method in [3] employs a one-dimensional CNN to map time-domain measurements of three-phase voltage and current signals to the corresponding fault locations. In [4], the adaptive CNN (ACNN) which consisted of a two-dimensional CNN demonstrated robustness against the variation in system parameters and load currents in selecting the faulted line. Another CNN-based approach presented in [5] used capsule neural networks and spectrogram time-frequency analysis for fault location. The analysis yielded a faulty image of the voltage signal as input data and the capsule neural network extracted nuanced features within deep layers. Leveraging the capabilities of GCNs in extracting spatiotemporal features from graph-structured data, the model proposed by [6] demonstrates improved generalization performance and robustness against topology modifications, fault resistance variation, and measurement noise. The effectiveness of hybrid methods was highlighted in [7] where authors proposed a CNN-LSTM structure for fault location and classification in power distribution cables. The proposed approach incorporated the voltage and current measurements and was shown to be robust to system parameters.

The current machine learning methodologies have several limitations: 1) The absence of unsupervised feature learning and limited generalization capabilities in these methods hinders their ability to generate a robust representation for power system measurements; 2) These methods primarily capture discriminative features of the power system, neglecting the acquisition of meaningful semantic features related to the power network; 3) These approaches do not prioritize the acquisition of task-relevant deep learning features, potentially resulting in the inclusion of irrelevant features for fault classification; and 4) The reliance on dense neural networks in these methods necessitates large datasets and makes them susceptible to overfitting problems.

Motivated by these drawbacks, this paper proposes a deep-attention Gated Recurrent Unit Gaussian Restricted Boltzmann Machine (GRU-GRBM) autoencoding method for fault location. First, the distribution power system measurements are fed to a novel attention-enhanced GRU to capture attention-guided task-relevant temporal features. These features are then used by a novel GRBM-based autoencoder structure designed to learn the temporal features in an unsupervised and generative manner. The generative features captured in the latent space of the proposed autoencoder are observed by a novel sparse deep Rectified Linear Unit (ReLU) network to

979-8-3503-3120-2 /24/$31.00 ©2024 IEEE

locate and classify the faults. The presented sparse network uses an innovative MI-based dropout technique to provide a sparse representation of the measurements which enhances the generalization capacity and reduces the data requirements of the proposed framework. We test the proposed approach on the IEEE 123-bus system and show the superiority of this approach compared to the state-of-the-art method, which is due to generative feature extraction, attention mechanism, and feature sparsity.

II. Problem Formulation

Let us define a dataset for fault location and classification with N samples. Each sample represents a fault location and class in the distribution network. The following are the problem's essential components: The number of samples in the dataset, N, each of which represents a different power system fault scenario. The input data for each sample i is represented as X_i. It is made up of 6-dimensional time series measurements from the observable buses, which include the angles and magnitudes of the positive, negative, and zero sequence components of the observable buses' voltage. Every X_i is a matrix with shape (T, D), where D denotes the measurements' dimensionality and T denotes the number of time steps of the measurements taken after the fault occurred. The ground truth data for each sample X_i consists of the $n \times n$ fault class matrix C_i, where n is the number of buses in the power system. The fault class that exists between buses i and j is represented by each element (i, j) of C_i, with 0 denoting the "no-fault" class. The ground truth also consists of a fault location matrix L_i, which is an $n \times n$ matrix. Each element (i, j) of L_i is a normalized number between 0 and 1, denoting the location of the fault on the line connecting buses i and j; Similar to fault classes, here, 0 denotes the absence of a fault on that line. In this formulation, the distance between the fault and the bus $min(i, j)$ defines the fault location between buses i and j. Our goal is to develop and train a model that computes the occurred fault class and location matrices $\hat{C}_i$ and $\hat{L}_i$ for each sample X_i $1 \leq i \leq N$ in the dataset using the time series measurements X_i as input. The objective is to reduce the difference between the ground truth matrices C_i and L_i for each sample i and their predicted matrices $\hat{C}_i$ and $\hat{L}_i$, respectively.

III. Proposed Method

Fig. 1 shows the proposed framework for fault location in the power distribution network. First, the power system measurements are observed by a novel attention-enhanced gated recurrent unit (AGRU) to capture task-specific temporal features. Then, the AGRU's features are used by a novel GRBM-based autoencoder to capture the unsupervised generative features from the power system's data. Finally, the generative features of the autoencoder are utilized by a new sparse deep ReLU network that uses a novel MI-based dropout technique for sparse fault classification and location.

A. Attention Gated Recurrent Unit

The AGRU captures temporal dependencies in sequential data while giving varying attention to different parts of the input sequence to ensure that task-relevant temporal features h_t are captured for each time step t of each fault sample. The proposed AGRU contains an input gate that controls which information is allowed to enter its temporal memory cell at each time step. The input gate is computed by:

$$i_t = \sigma(W_i \cdot [x_t, r_t \cdot h_{t-1}] + b_i) \tag{1}$$

where x_t is the input at time step t which is a D-dimensional vector of measurements; h_{t-1} is the previous hidden state (temporal feature vector from the previous time step); r_t is the relevance score from the attention mechanism; W_i and b_i are the weight and bias parameters for the input gate, respectively. Also, σ is the sigmoid activation function. The AGRU has a reset gate that controls which information from the previous hidden state is forgotten or retained. The reset gate is computed by:

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r) \tag{2}$$

where W_r, U_r, and b_r are the weights and bias parameters for the reset gate, respectively. We define a candidate update vector n_t that represents the new information that can be added to the memory cell at each time step t. The candidate vector is computed using $n_t = tanh(W_n * x_t + r_t * U_n * h_{t-1} + b_n)$ where W_n, U_n, and b_n are weights and bias parameter for the candidate update, respectively. In this formulation, $tanh$ is the hyperbolic tangent activation function. The memory cell of the AGRU stores time-dependent information of the power systems measurements over time and is updated using the input and reset gates using $c_t = i_t * n_t + (1 - i_t) * c_{t-1}$. We define a relevance score $\bar{r}_t$ that determines how much attention to give to each input measurement at each time step t. This attention score is computed using an un-normalized attention credit for each d-th measurement of t-th time step: $e_{t,d} = v^T * tanh(W_e * h_{t-1} + U_e * x_{t,d} + b_e)$ where v, W_e, U_e, and b_e are weights and bias parameters for the attention mechanism. The relevance scores are normalized across the measurements using the attention score normalization $\alpha_{t,d} = \frac{e^{e_{t,d}}}{\sum_{k=1}^{D} e^{e_{t,k}}}$ where $\alpha_{t,d}$ is the normalized relevance score for the d-th measurement at time step t. The final relevance score $\bar{r}_t$ is a weighted sum of the input measurements computed by $\bar{r}_t = \sum_{d=1}^{D}(\alpha_{t,d} * x_{t,d})$ where $\bar{r}_t$ is computed for the d-th feature of X_t denoted by $x_{t,d}$. The hidden state or temporal feature vector at each time step t is updated based on the memory cell and the attention-weighted input using:

$$h_t = (1 - z_t) * h_{t-1} + z_t * c_t + \gamma * \bar{r}_t \tag{3}$$

where z_t is the update gate that controls the trade-off between the previous hidden state and the new information from the memory cell and the attention mechanism. In this formulation, γ is a weight parameter. The AGRU processes the input time series measurements at each time step and computes temporal feature vectors, giving varying attention to different measurements based on the relevance scores. Using this

neural architecture, one can compute the attention-enhanced temporal features of each fault sample X_i using H_i defined by $H_i =< h_1^i, h_2^i, ..., h_T^i >$ where each h_t^i is the t-th temporal feature computed for sample X_i using the proposed AGRU.

B. GRBM Encoder f_{enc}^{RBM}

In this step, we employ a GRBM encoder to extract deep generative features from H_i corresponding to each fault sample X_i, denoted by $\tilde{H}_i$. The GRBM consists of two layers: a visible layer and a hidden layer. The visible layer corresponds to the temporal features captured by the AGRU (denoted as H_i), while the hidden layer represents the learned, lower-dimensional representation or generative features of the temporal features. The visible layer units associated with the temporal features are captured by the AGRU. These features are real-valued and denoted as V_i, where i represents the i-th fault sample. Each V_i is a vector of AGRU's temporal features of X_i with $\omega = T * d_h$ elements where d_h is the dimension of the AGRU's temporal features. Thus, V_i is written as $V_i = [V_{i,1}, V_{i,2}, ..., V_{i,\omega}]$. The hidden layer units $\tilde{H}_i$ are binary units that capture features of V_i in a generative fashion. Each $\tilde{H}_i = [\tilde{H}_{i,1}, \tilde{H}_{i,2}, ..., \tilde{H}_{i,M}]$ is a binary vector with M elements, representing the activations of the hidden units. The activation of a hidden unit $\tilde{H}_{i,j}$ for the i-th fault sample X_i is computed by:

$$p(\tilde{H}_{i,j} = 1|H_i) = \sigma\left(b_j + \sum_{k=1} w_{jk} V_{i,k}\right) \tag{4}$$

where $\tilde{H}_{i,j}$ is the binary activation of the j-th hidden unit for the i-th sample; b_j is the bias term for the j-th hidden unit.; w_{jk} is the weight between the j-th hidden unit and the k-th feature in V_i,; and $V_{i,k}$ is the k-th feature V_i vector. The hidden layer's mean and standard deviation for the i-th fault sample are computed using $E[\tilde{H}_{i,j}|H_i] = p(\tilde{H}_{i,j} = 1|H_i)$ and $Std[\tilde{H}_{i,j}|H_i] = \sqrt{E[\tilde{H}_{i,j}|H_i](1 - E[\tilde{H}_{i,j}|H_i])}$, respectively. To sample the hidden units, the mean and standard deviation computed above could be used. For each hidden unit $\tilde{H}_{i,j}$, one can sample from a Bernoulli distribution $\tilde{H}_{i,j} \sim$ Bernoulli$(E[h_{ij}|H_i])$. To compute the visible units given the hidden vector we model the distribution of the real-values visible units as a Gaussian distribution. Therefore, the activation of a visible unit $V_{i,k}$ for the i-th fault sample can be computed as $V_{i,k} = \mu_k + \sigma_k \epsilon_{i,k}$ where μ_k is the mean of the Gaussian distribution for the k-th visible unit and σ_k is the standard deviation, while $\epsilon_{i,k}$ is a random sample from a standard Gaussian distribution $\mathcal{N}(0, 1)$. For each configuration of values $(V_i, \tilde{H}_i)$, the generative energy of the GRBM is defined as:

$$F_{en}^{enc}(V_i, \tilde{H}_i) = \sum_{k=1}^{T*d_h} (V_{i,k} - b_k)^2 - \sum_{k=1}^{T*d_h} \sum_{j=1}^{M} V_{i,k} W_{k,j} \tilde{H}_{i,j} - \sum_{j=1}^{M} a_j \tilde{H}_{i,j} \tag{5}$$

where b_i is the bias of the visible layer, $W_{k,j}$ is the weight connecting each unit $V_{i,k}$ to $\tilde{H}_{i,j}$, and a_j is the bias of the hidden unit $\tilde{H}_{i,j}$. In this model, the compressed generative features for the i-th fault sample are represented as a binary vector $\tilde{H}_i$ with M elements. This GRBM-based encoder captures the binary activations of the hidden units based on the real-valued temporal features of the AGRU, providing a lower-dimensional representation of the input data.

C. GRBM Decoder f_{dec}^{GRBM}

The decoding GRBM observes each generative feature $\tilde{H}_i$ of a fault sample X_i and computes a latent feature vector $\bar{H}_i$ which has the same dimension as H_i. Similar to the encoder, we train this decoder in an unsupervised fashion to initialize $\bar{H}_i$. Then, we train it using a supervised loss function to reconstruct H_i by generating a reconstructed H_i in its hidden layer $\bar{H}_i$. Therefore, after the supervised training, $\bar{H}_i \simeq H_i$. The supervised training updates the parameters of both the encoder and decoder in this autoencoding architecture.

D. Deep ReLU neural network with Dropout Regularization

To obtain the fault class C_i and fault location L_i matrices, we define a deep ReLU network with dropout regularization with Q layers z^l $1 \leq l \leq Q$. The activation of each computational layer l for an input fault X is defined by:

$$a^1 = X,\ z^l = W^l * a^{l-1} + b^l,\ a^l = ReLU(z^l) * d^l \tag{6}$$

Here, $ReLU$ is the ReLU activation function and a^l is a masked activation defined by a dropout regularizer. During each forward and backward pass, a binary dropout mask vector d^l is sampled for each hidden layer l. The dropout mask d^l has the same dimension as the output of layer l (i.e., a^l), and its elements are sampled independently from a Bernoulli distribution using $d_i^l \sim Bernoulli(1 - p)$ where i represents the index of a neuron in layer l. This means that with probability p, a neuron is dropped out (assigned a value of 0), and with probability $(1 - p)$, it is retained. During the forward pass, the masked activations a^l $1 \leq l \leq Q$ are computed considering d^l. The element-wise multiplication sets the values of neurons that were dropped out ($d_i^l = 0$) to zero, effectively deactivating them for that forward pass. For the backward pass, we also need to scale the activations to account for the dropout. This is typically done by dividing the retained activations by $(1 - p)$ using $a^l = a^l/(1 - p)$. This scaling ensures that the expected value of the activations remains the same as during inference, where no dropout is applied. It helps in maintaining the correct signal flow and gradients during training. This regularization process helps prevent overfitting and improves the generalization of the proposed deep framework.

E. Training

The proposed framework seeks to minimize the following loss function to train the AGRU, f_{enc}^{RBM}, f_{dec}^{RBM}, and Deep

ReLU network using a fault dataset $\{X_i\}_{i=1}^N$ with N faults:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{C_i} + \alpha \mathcal{L}_{L_i} + \beta E_{enc} + \gamma E_{dec} + \kappa E_{rec} \quad (7)$$

Here, α, β, γ, κ, and ζ are loss coefficients known as error hyperparameters. Furthermore, E_{enc} and E_{dec} are the average error functions (energy functions) of the encoder and decoder RBMs for the N samples, respectively. $E_{rec} = \frac{1}{N}\sum_{i=1}^{N} ||H_i - \bar{H}_i||_2^2$ is the reconstruction loss function of the encoding-decoding RBMs. We train the proposed framework using the cross-entropy loss function for fault classification. For every fault sample X_i in the dataset, the loss calculates the difference between the computed and actual classes via $\mathcal{L}_{C_i}(C_i, \hat{C}_i) = -\sum_{(j,k)} C_i(j,k)\log(\hat{C}_i(j,k))$, where (j,k) indicates the entry of the j-th row and k-th column of the matrix. Moreover, we utilize the mean squared error for the fault location of a sample X_i, which is defined as $\mathcal{L}_{L_i}(L_i, \hat{L}_i) = \sum_{j,k}(L_i(j,k) - \hat{L}_i(j,k))^2$. Algorithm 1 is presented to train the proposed fault classification framework in an end-to-end fashion. In this algorithm, we use Batch Gradient Descent (BGD) and Contrastive Divergence (CD) to train the neural networks and minimize $\mathcal{L}$.

Algorithm 1 Training Algorithm

while Parameters not converged
- Randomly select a batch $B = \{X_j\}_{j=1}^{\tau}$ from N samples
- Compute AGRU Features H_j for all samples in B
- Train the encoding GRBM using the CD method:
  - For each sample X_j with input temporal feature H_j generate a negative sample $\hat{H}_j$ using Gibbs sampling
  - Update encoding GRBM's parameters using gradient descent with gradient of GRBM's parameters at $\hat{X}_j$
- Train the decoding GRBM using the CD method
- Jointly train the AGRU, encoder, decoder, and the sparse ReLU network to minimize $\mathcal{L}$ using BGD

IV. Numerical Results

In this study, the sequence components of the voltage are used as the input features. The transient stability analysis is performed using Digsilent PowerFactory on the IEEE 123-bus system where the fault is applied on 0%, 20%, 40%, 60%, and 80% of the distribution lines. In this experiment the power network is partially observable and only 30% of the buses have observable measurements. The fault is applied at $t = 0.1$ sec and cleared at $t = 0.2$ sec. There are 5650 samples in the dataset. Of the total data, we use 10% for validation, 15% for testing, and 75% for training. Following [8], we employ heuristic search in conjunction with validation to determine the best values for the proposed framework's hyperparameters, with the validation fault location Root Mean Squared Error (RMSE) serving as the primary search criterion. The size of the GRU hidden feature vector is 35, the number of hidden units in the encoder RBM is 40, the number of hidden layers in the Sparse ReLU network is 3, and the size of each layer is 45

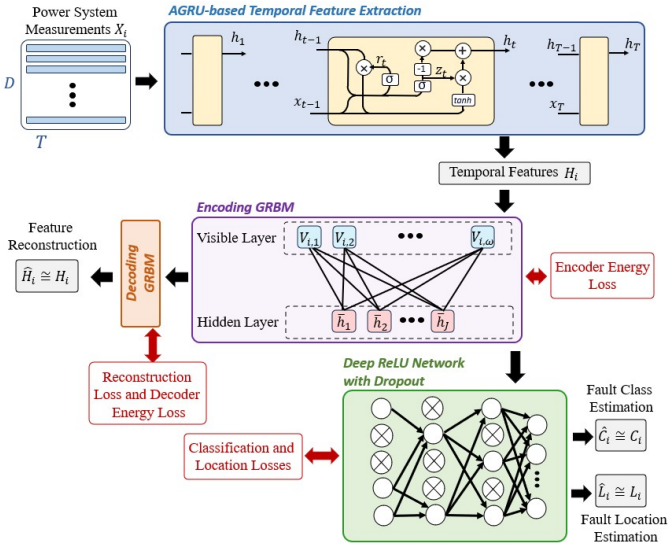

Fig. 1. Proposed deep attention GRU-GRBM with dropout for fault location.

units in the ideal hyperparameter set. Here, $\alpha = 0.8$, $\beta = 1.2$, $\gamma = 1.3$, $\kappa = 0.85$, and $\zeta = 0.95$. The BGD algorithm has a learning rate of 10^{-2} and a batch size of $\tau = 40$. To create a sparse neural network, we also set the sparse ReLU network's Bernoulli parameter to 0.35. A PC equipped with an Intel Core i7 CPU and a single GeForce RTX 4090 GPU is used to train and test the model. The fault location performance in this study is presented as the Mean Absolute Error (MAE), RMSE, and Mean Absolute Percentage Error (MAPE). In our study, we also employ classification accuracy metrics including F-score, Precision, Recall, and Accuracy for fault classification.

We compare the CNN [3], ACNN [4], Wavelet CNN [9], LSTM [9], GRU [10], Capsule Network (CapsNet) [11], CNN-LSTM [7], and CNN-GRU [7] with the proposed GRU-GRBM framework. The fault location and classification outcomes of the proposed method and the benchmarks are shown in Tables I and II, respectively. The WCNN shows better location and classification accuracies in comparison with the CNN model due to its wavelet decomposition-based preprocessing stage that makes it a more robust model to data noise and uncertainty. The time-dependent models, GRU and LSTM, outperform the CNN and WCNN models, which are merely spatial feature extraction techniques, as the tables demonstrate. As an example, the LSTM increases CNN's classification accuracy by 2.34%. Due to its smaller parameter count, which helps it prevent overfitting better than the LSTM model, the GRU performs better than the LSTM. The ACNN shows a better performance in location and classification tasks compared to the GRU as it leverages an attention mechanism that ensures the task-relevant features of the input are present in the latent space of the neural network. The ACNN provides 7.51% lower location RMSE and 0.94% higher classification accuracy compared to the GRU. CapsNet performs better than the GRU and ACNN because it can handle hierarchical features more effectively and enhances the deep learning features' resilience and generalization ability. For example, the

TABLE I
FAULT LOCATION RESULTS OF THE PROPOSED METHOD AND COMPARED BENCHMARKS

Model	RMSE	MAE	MAPE (%)
CNN	28.452	17.329	18.530
WCNN	27.052	16.074	17.191
LSTM	26.324	15.204	16.812
GRU	23.740	12.849	12.039
ACNN	21.956	11.302	11.105
CapsNet	21.307	10.804	10.912
CNN-LSTM	17.821	8.563	8.027
CNN-GRU	16.032	7.401	7.042
Proposed	14.751	5.603	5.913

TABLE II
FAULT CLASSIFICATION RESULTS OF THE PROPOSED METHOD AND COMPARED BENCHMARKS

Model	Precision	Recall	Accuracy (%)	F-scroe
CNN	0.7762	0.7609	78.34	0.7684
WCNN	0.7903	0.7882	79.11	0.7892
LSTM	0.8217	0.8143	80.68	0.8179
GRU	0.8664	0.8503	85.21	0.8582
ACNN	0.8733	0.8608	86.01	0.8670
CapsNet	0.8812	0.8732	86.93	0.8771
CNN-LSTM	0.8953	0.9041	90.02	0.8996
CNN-GRU	0.9154	0.9133	91.47	0.9143
Proposed	0.9607	0.9570	95.81	0.9588

F-score and classification accuracy of GRU are improved by 2.2% and 1.72%, respectively, by the CapsNet. In comparison to deep learning models that only take into account time- or space-dependent characteristics, the spatiotemporal feature-extracting deep learning models yield higher accuracies. For example, as Tables I and II demonstrate, the CNN-LSTM model enhances the CapsNet's location RMSE and MAPE by 16.36% and 2.88%, respectively. The CNN-GRU has a smaller tunable parameter space because it employs a GRU model as opposed to an LSTM as in the CNN-LSTM. As a result, even with fewer training data, it has more generalization power and is more resistant to overfitting. In comparison to the CNN-LSTM model, the CNN-GRU produces a location MAPE that is 0.98% lower and a classification accuracy that is 1.45% higher, as indicated by the tables. When compared to the most recent benchmarks, the proposed approach demonstrates much higher classification and location accuracies. The proposed attention GRU-GRBM outperforms the best-compared benchmark, CNN-GRU, with a 4.34% greater classification accuracy and a 16.03% lower location MAPE, as the tables demonstrate. Our method is more accurate compared to other approaches since it uses RBM-based autoencoding for generative unsupervised feature learning and MI-based sparsity loss for sparse feature extraction as well as task-relevant features found by our attention mechanism. Some of the estimated test fault locations of the CNN-GRU and the proposed technique are displayed in Figures 2 and 3, respectively. As the figure illustrates, the proposed method's generative feature extraction, attention-guided pattern recognition, and sparse feature learning capabilities enable it to track the real locations of the AG and ABC faults than the CNN-GRU.

To show the robustness of the proposed model to the data

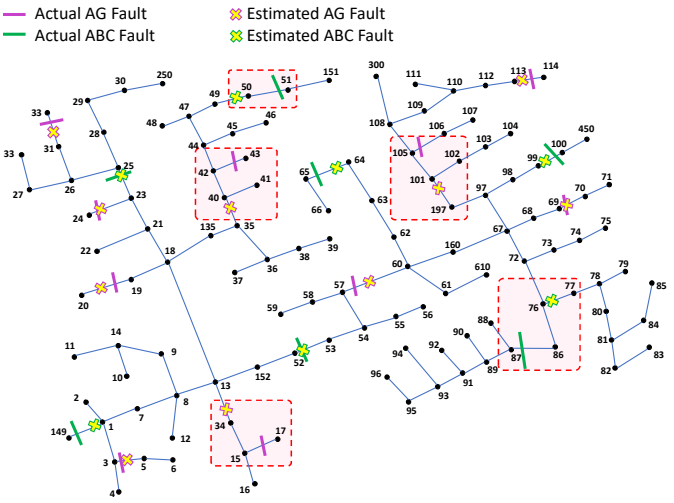

Fig. 2. Fault location results of the CNN-GRU model

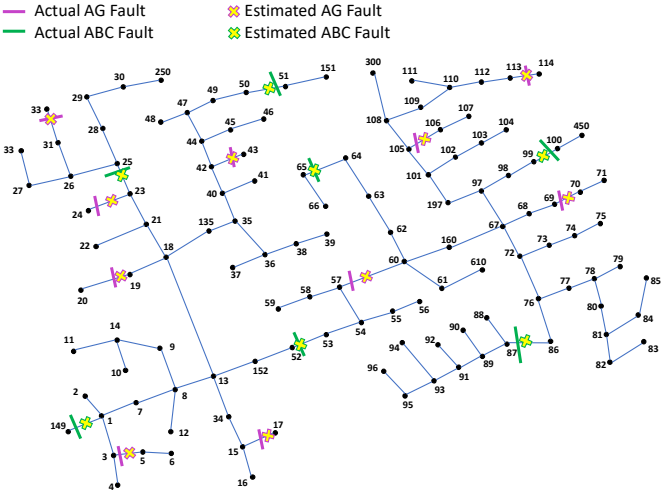

Fig. 3. Fault location results of the proposed framework

noise and uncertainty, we add Gaussian noises with zero means and different standard deviations (STDVs) to the input measurements and show the performance of the proposed method compared to the recent benchmarks. Fig. 4 shows the MAPE results of the fault location tasks for the proposed model and different benchmarks. Here, the noise STDV is changing from 0 to 0.1. As depicted in this figure, the proposed model shows a better robustness with a lower increase in the MAPE values as the noise STDV grows compared to the other benchmarks. This observation is mainly due to the generative feature extraction that helps the model capture robust latent representations of the input data.

In our standard experimental setting, only 30% of the buses have observable measurements. To compare the performance of the proposed model and the benchmarks for different input sizes (i.e., number of observed buses), we increase this number from 30% to 70% of the total number of buses, and show the F-score results of the fault classification task. Fig. 5 shows the F-score results for the compared benchmarks and the proposed

method. As shown in this figure, the proposed GRU-GRBM keeps having a higher classification accuracy compared to the recent benchmarks. As the size of the available data grows, the proposed method can better use the data to classify the faults compared to the other deep learning-based models. This is due to learning the unsupervised features of the input data and the feature sparsity of the proposed decoders.

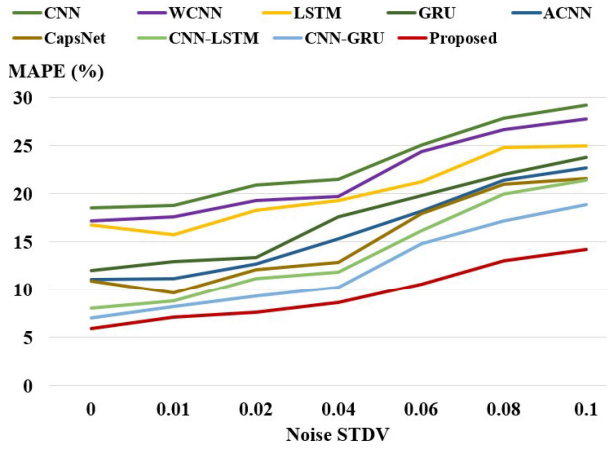

Fig. 4. Fault location MAPE of the proposed model and recent benchmarks with different Gaussian noise STDVs.

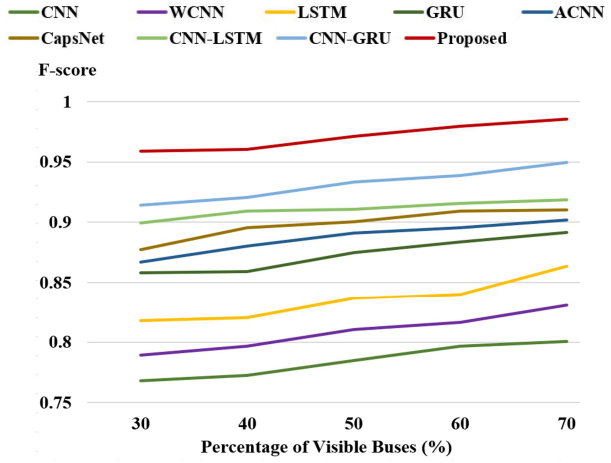

Fig. 5. Fault classification F-scores with different percentages of visible buses for the proposed GRU-GRBM and the recent benchmarks.

V. Conclusions

In this paper, a novel GRU-GRBM autoencoding method with a deep attention mechanism is proposed to locate faults. The distribution power system measurements are input to a newly built attention-enhanced GRU to capture task-relevant temporal characteristics led by attention. A unique GRBM-based autoencoder structure uses these temporal characteristics and generates unsupervised temporal information. The proposed autoencoder's generative features are fed to a new sparse deep ReLU network to discover and characterize faults. The sparse network uses an MI-based dropout mechanism to represent measurements sparsely. This method improves framework generalization and reduces data needs. The IEEE 123-bus system is applied to test the proposed method and compare it to the present state-of-the-art. Generative feature extraction, attention mechanism, and feature sparsity explain the superiority of the proposed method compared to recent benchmarks.

VI. Acknowledgement

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

References

[1] E. Personal et al., "A comparison of impedance-based fault location methods for Power Underground Distribution Systems," Energies, vol. 9, no. 12, p. 1022, 2016.

[2] D. W. P. Thomas, R. J. O. Carvalho, and E. T. Pereira, "Fault location in distribution systems based on traveling waves," 2003 IEEE Bologna Power Tech Conference Proceedings, Bologna, Italy, 2003, pp. 5 pp. Vol.2.

[3] Y. Yu, M. Li, T. Ji, and Q. H. Wu, "Fault location in distribution system using convolutional neural network based on domain transformation," in CSEE Journal of Power and Energy Systems, vol. 7, no. 3, pp. 472-484, May 2021.

[4] J. Liang, T. Jing, H. Niu, and J. Wang, "Two-terminal fault location method of distribution network based on Adaptive Convolution Neural Network," IEEE Access, vol. 8, pp. 54035–54043, 2020.

[5] H. Mirshekali, A. Keshavarz, R. Dashti, S. Hafezi, and H. R. Shaker, "Deep learning-based fault location framework in power distribution grids employing convolutional neural network based on Capsule Network," Electric Power Systems Research, vol. 223, p. 109529, 2023.

[6] J. Hu et al., "Fault location and classification for distribution systems based on Deep Graph Learning Methods," Journal of Modern Power Systems and Clean Energy, vol. 11, no. 1, pp. 35–51, 2023.

[7] Y. Yu, M. Li, T. Ji, and Q. H. Wu, "Fault location in distribution system using convolutional neural network based on domain transformation," in CSEE Journal of Power and Energy Systems, vol. 7, no. 3, pp. 472-484, May 2021.

[8] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang and G. Liu, "Convolutional Graph Autoencoder: A Generative Deep Neural Network for Probabilistic Spatio-Temporal Solar Irradiance Forecasting," in IEEE Transactions on Sustainable Energy, vol. 11, no. 2, pp. 571-583, April 2020.

[9] D. Paul and S. K. Mohanty, "Fault classification in transmission lines using wavelet and CNN," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Mar. 2019.

[10] M. Cui, M. Khodayar, C. Chen, X. Wang, Y. Zhang and M. E. Khodayar, "Deep Learning-Based Time-Varying Parameter Identification for System-Wide Load Modeling," in IEEE Transactions on Smart Grid, vol. 10, no. 6, pp. 6102-6114, Nov. 2019.

[11] H. Mirshekali, A. Keshavarz, R. Dashti, S. Hafezi, and H. R. Shaker, "Deep learning-based fault location framework in power distribution grids employing convolutional neural network based on Capsule Network," Electric Power Systems Research, vol. 223, p. 109529, Oct. 2023.