Sparse Attention Graph Gated Recurrent Unit for Spatiotemporal Behind-The-Meter Load and PV Disaggregation

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

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

Mohsen Saffari
Department of Computer Science
Department of Computer Science
Tulsa, Oklahoma, United States
mohsen-saffari@utulsa.edu

***Abstract*—The increasing adoption of rooftop photovoltaic (PV) power generation systems in residential areas necessitates accurate monitoring and disaggregation of behind-the-meter (BTM) load and PV power. Despite recent advancements, existing BTM disaggregation approaches suffer from three major drawbacks: neglecting task-relevant spatiotemporal features, overfitting, and lack of a sparse neural architecture which leads to high sample complexity. This paper addresses them by introducing a deep sparse attention graph recurrent framework. This framework conceptualizes a set of neighboring residential units as a graph where the nodes are the net load values of the units and the edges show the mutual information (MI) of these measurements. We develop an Attention Gated Recurrent Unit (AGRU) to capture enhanced temporal characteristics of the net load. We employ a novel low-rank Dictionary Learning (DL) method to discern spatiotemporal features of these measurements and further utilize a Rectified Linear Unit (ReLU) neural network that incorporates an MI-based dropout to provide a sparse model for the estimation of the BTM load and PV. Experimental results validate the effectiveness of our proposed model, exhibiting superior performance on the Ausgrid dataset in BTM load and PV power estimation compared to state-of-the-art methods.**

***Index Terms*—BTM load and PV disaggregation, Graph GRU, Attention, Deep Learning**

I. Introduction

Rooftop Photovoltaic (PV) power installations, commonly positioned behind the meter, limit visibility for the distribution system operator resulting in inaccurate load forecasting and nodal voltage fluctuations [1], [2]. To enhance the reliability of the distribution system, two approaches—model-based and data-driven—are employed for the disaggregation of PV generation from net demand. Model-based techniques use the physical characteristics of the resources to separate PV generation. The Linear Regression Strategy (LRS) structure in [1] employed linear regression for this task based on substation and solar irradiance measurements. However, these approaches face challenges due to uncertainties in model parameters and the risk of overestimation during PV power generation failure [1].

Artificial Intelligence (AI) and data-driven methods are divided into supervised and unsupervised approaches. Unsupervised methods employ net demand and environmental metrics, as demonstrated by the probabilistic Bayesian Structural Time Series (BSTS) [4] introduced at the feeder level. In contrast, when providing information on BTM generation, supervised techniques involve the analysis of labeled data. For instance, studies presented in [5], [6] focus on the mapping of the original feature space to a representative sparse latent space through constrained optimization problems. Specifically, the authors of [6] capture the device contributions through learned temporal features and dictionary learning. Expanding on supervised methodologies, the authors of [1] introduced the Repeated Game Theory with Vector Payoff (RGVP) structure, integrating data clustering and game-theoretic learning. This structure serves as a semi-supervised source separator, utilizing a repository of candidate load and solar exemplars. Moreover, the authors in [2] apply a dense graph learning method that considers space-dependent and time-dependent features of net load measurements of residential units to estimate their behind-the-meter PV power and load. However, the captured features are not necessarily task-relevant and the model requires large amounts of data due to its large parameter space and dense structure.

The existing works in BTM disaggregation [1], [2], [4]–[6] have three major drawbacks: 1) They do not study capturing task-relevant spatiotemporal features from the input data. That is, the contribution of each feature to the task is not taken into account in the feature extraction process. 2) They merely capture dense deep learning features, hence, they are prone to overfitting issues; and 3) They do not provide a sparse neural architecture, hence, they require large amounts of data samples to train their models.

Motivated by these drawbacks, this paper develops a novel deep sparse attention graph recurrent framework for spatiotemporal BTM load and PV disaggregation. First, the net load of residential units is modeled using a dynamic graph where the nodes show the net load measurements of the units and edges show the correlations between these measurements. Then, a novel attention graph GRU is designed to capture the attention-enhanced space-time features of the input dynamic graph. The proposed attention mechanism helps the GRU to find

task-relevant features in the feature extraction process. The spatiotemporal features are then used in a novel low-rank DL structure to capture a sparse feature vector. The sparse features help the model overcome overfitting challenges and enhance the generalization capacity of the proposed framework. Finally, a novel deep ReLU neural network with a mutual information (MI)-based dropout mechanism is developed and trained to estimate the BTM load and PV measurements of the residential units. The proposed dropout technique uses the MI between the activations of the hidden units of the deep ReLU neural network to find a feature mask that removes units that have high MI (i.e., less informative units). This method prevents the neural network from relying too much on specific neurons and encourages the learning of more robust and generalized features. It also helps the proposed method to break up the co-adaptation of neurons. That is, the neurons will not rely on the presence of specific neuron activations to produce meaningful features, which encourages more independent feature learning in the neural architecture.

II. Problem Formulation

Consider a set of N local residential units with rooftop PV panels. Each unit, s_t^i, consumes load L_t^i and generates PV power PV_t^i. A smart meter measures the BTM net electricity demand which is a combination of these values computed by $R_t^i = L_t^i - PV_t^i$. Given the net loads $\{R_t^i\}_{i=1}^N$ in a time window $[t, t+m]$ of length $m+1$, the goal of this study is to find the values of L_t^i as well as PV_t^i for all time instances in the range $[t, t+m]$. In this formulation, the net loads of the units are assumed to have spatiotemporal correlations since the units have correlated weather factors (e.g., temperature, cloud cover, etc.). Thus, we seek to find the spatiotemporal correlations between the nodes and use such relationships to enhance the disaggregation performance. For this problem, one can consider a dataset $\{S_t\}_{t\in[1,M]}$ with M samples. For each sample S_t, we have a set of net load values $\{R_{\bar{t}}^i\}_{1\le i\le N,\ t\le\bar{t}\le t+m}$ as well as a set of ground truth BTM load and PV values denoted by $\{L_{\bar{t}}^i\}_{1\le i\le N,\ t\le\bar{t}\le t+m}$ and $\{PV_{\bar{t}}^i\}_{1\le i\le N,\ t\le\bar{t}\le t+m}$, respectively. In this formulation, we denote the model's estimation of $L_{\bar{t}}^i$ and $PV_{\bar{t}}^i$ by $\hat{L}_{\bar{t}}^i$ and $\hat{PV}_{\bar{t}}^i$, respectively, and the objective is to train a data-driven model that accurately estimates the BTM load and PV for each sample in the dataset.

III. Proposed Method

Fig. 1 shows the overview of the proposed deep sparse attention graph GRU framework for spatiotemporal BTM load disaggregation. First, we model each sample S_t using a spatiotemporal graph $G_{\bar{t}} = (V_{\bar{t}}, E_{\bar{t}})\ t \le \bar{t} \le t+m$ where $V_{\bar{t}}$ is the set of N nodes that represent the net loads of residential units, and $E_{\bar{t}}$ is the set of edges $e_{\bar{t}}^{i,j}$ connecting two units i and j that represent the mutual information (MI) of $\tilde{R}_t^i = [R_t^i, R_{t+1}^i, ..., R_{t+m}^i]$ with $\tilde{R}_t^j$. An attention graph GRU is developed to find the spatiotemporal features of the input graph. Then, a low-rank DL method is devised to capture the sparse features of the spatiotemporal data representation.

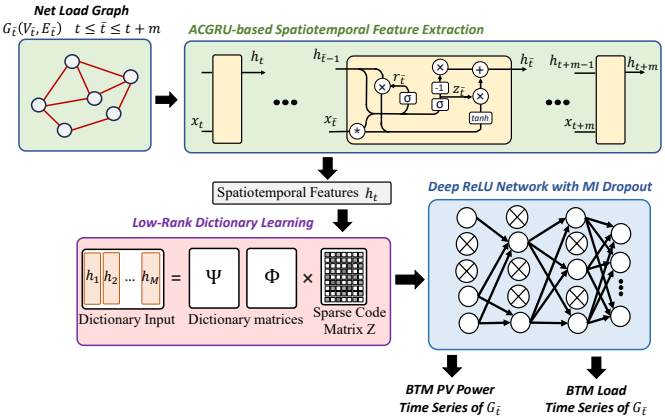


Fig. 1. Proposed deep sparse attention recurrent model for BTM load and PV disaggregation

Finally, a novel deep ReLU network with MI-based dropout is trained to estimate the BTM load and PV values of the residential units.

A. Attention Graph GRU

For each sample S_t, we define an attention graph GRU model that observes each graph snapshot $G_{\bar{t}}\ \bar{t} \in [t, t+m]$ and generates a spatiotemporal feature vector $h_{\bar{t}}$. As shown in Fig. 1, in this space-time recurrent model, first, each $G_{\bar{t}}$ is fed to an attention graph convolution model defined by:

$$\begin{aligned} A_{\bar{t}} &= softmax(U_t * \phi * U_{\bar{t}}^\top) \\ h'_{\bar{t}} &= A_{\bar{t}} * (U_{\bar{t}} * \theta * U_{\bar{t}}^\top * X_{\bar{t}}) \end{aligned} \tag{1}$$

where $h'_{\bar{t}}$ is the updated feature representation after the graph convolution, $U_{\bar{t}}$ is the matrix of eigenvectors of the Laplacian matrix of $G_{\bar{t}}$, θ is the filter weight matrix, $X_{\bar{t}}$ is the input node feature, and ϕ is the learnable weight matrix for the attention mechanism. Here, $A_{\bar{t}}$ is the attention matrix obtained by applying the softmax function to the element-wise product of $U_{\bar{t}} * \phi * U_{\bar{t}}^\top$. The attention mechanism allows the convolution to focus on different parts of the graph and adaptively weigh the contributions of neighboring nodes during the convolution process. The computed $h'_{\bar{t}}$ is then fed to the GRU to compute the spatiotemporal features $h_{\bar{t}}$ corresponding to each graph $G_{\bar{t}}$ at each time step $\bar{t}$ using:

$$\begin{aligned} x_{\bar{t}} &= h'_{\bar{t}} \\ r_{\bar{t}} &= \sigma(W_r * [h_{\bar{t}-1}, x_{\bar{t}}]) \\ z_{\bar{t}} &= \sigma(W_z * [h_{\bar{t}-1}, x_{\bar{t}}]) \\ \tilde{h}_{\bar{t}} &= tanh(W_h * [r_{\bar{t}} * h_{\bar{t}-1}, x_t]) \\ h_{\bar{t}} &= (1 - z_{\bar{t}}) * h_{\bar{t}-1} + z_{\bar{t}} * \tilde{h}_{\bar{t}} \end{aligned} \tag{2}$$

where $x_{\bar{t}}$ is the input at time step $\bar{t}$. Here, $r_{\bar{t}}$, $z_{\bar{t}}$, and $h_{\bar{t}}$ are the reset gate, update gate, and hidden state at time step $\bar{t}$. The $\tilde{h}_{\bar{t}}$ is the candidate hidden state at time step $\bar{t}$. W_r, W_z, and W_h are weight matrices for the reset gate, update gate, and candidate hidden state, respectively. σ denotes the sigmoid activation function while $tanh$ is the hyperbolic

tangent activation function. The reset gate decides what to forget from the previous hidden state, and the update gate determines how much of the candidate hidden state should influence the current state. The candidate hidden state is a temporary value that combines information from the reset gate and the current input to calculate a potential new hidden state. We define h_t as the concatenation of $h_{\bar{t}}$ for all time steps $\bar{t} \in [t, t+m]$ for a sample G_t.

B. Low-Rank Dictionary Learning (DL)

We define a low-rank sparse coding scheme to learn a sparse feature vector α_t for each spatiotemporal feature h_t by:

$$\begin{aligned} &\min_{D,Z} ||H - DZ||_F^2 + \lambda ||Z||_{2,1} \\ &s.t. \;\; rank(D) = r \end{aligned} \tag{3}$$

where $H = [h_1, h_2, ..., h_M] \in \mathbb{R}^{d_h \times M}$ is the matrix of spatiotemporal features of the dataset, D is a dictionary matrix that stores patterns of samples in H while $Z = [z_1, z_2, ..., z_M] \in \mathbb{R}^{d_z \times M}$ is the matrix of sparse codes (sparse features). We set rank r for D. In this formulation, the sparsity term $\lambda ||Z||_{2,1}$ with coefficient λ ensures that the captured features in Z are sparse. This term uses the $L_{2,1}$ norm of Z defined by $||Z||_{2,1} = \sum_{i=1}^{d_z} \sqrt{\sum_{j=1}^{M} {Z_{i,j}}^2}$ where $Z_{i,j}$ is the (i,j)-th element in Z. The constraint in (3) may be expressed by multiplying two dictionary matrices $\Psi \in \mathbb{R}^{d_h \times r}$ and $\Phi \in \mathbb{R}^{r \times d_h}$ with rank r. By applying spectral analysis [7] in (3), we take into account the local sample structure and rewrite the optimization for extracting sparse features as:

$$\min_{\Psi,\Phi,Z} \quad J = ||H - \Psi\Phi Z||_F^2 + \lambda_1 tr(Z\Lambda Z^\intercal) + \lambda_2 ||Z||_{2,1} \tag{4}$$

where λ_1 and λ_2 are the coefficients of the objective. Here, Λ is the Laplacian matrix corresponding to a radial basis function kernel $S_{i,j}^{\Lambda} = exp(\frac{-||h_i - h_j||_2^2}{\sigma^2})$ defined in the space of H. By defining $D = \Psi\Phi$, the feature correlation is considered in a low-rank space. To obtain the optimal Z denoted by Z^*, we propose the a two-stage update procedure:

1) Optimizing Dictionaries: we assume Z is fixed, and set $\frac{\partial J}{\partial \Psi} = 0$ which results in the optimal primary dictionary $\Psi^* = HZ^\intercal\Phi^\intercal(\Phi Q_w \Phi^\intercal)^{-1}$ with $Q_w = ZZ^\intercal$. Using Ψ^*, the optimization is expressed as $\max_\Phi tr((\Phi Q_w \Phi^\intercal)^{-1}\Phi Q_b \Phi^\intercal)$ with $Q_b = ZH^\intercal HZ^\intercal$. Thus, the optimal secondary dictionary Φ^* is computed by the eigenvector matrix of ${Q_w}^{-1}Q_b$ corresponding to the top r eigenvalues.

2) Optimizing Sparse Codes: We compute the optimal sparse codes Z^* using:

$$\min_Z J_3 = ||H - \Psi||_F^2 + \lambda_1 tr(Z\Lambda Z^\intercal) + \lambda_2 ||Z||_{2,1} \tag{5}$$

Here, we set the derivative of the objective w.r.t Z to zero, which leads to $((\Psi\Phi)^\intercal\Psi\Phi + \lambda\Omega)Z + Z(\alpha\Lambda) = (\Psi\Phi)^\intercal H$ with a $d_z \times d_z$ dimensional diagonal square matrix Ω with diagonal entries $\Omega_{i,i} = (2||Z_{:,i}||_2 + \epsilon)^{-1}$ where ϵ is considered a small positive constant. This equality is a Sylvester equation, hence, can be solved using the Bartels–Stewart method [8] to obtain Z^*.

C. Deep ReLU Discriminator with Dropout Mechanism

Our objective is to compute $\hat{L}_{\bar{t}}^i \simeq L_{\bar{t}}^i$ as well as $\hat{PV}_{\bar{t}}^i \simeq {PV_{\bar{t}}}^i$ for $i = 1, 2, ..., N$ and $\bar{t} \in [t, t+m]$ in a graph G_t. Therefore, in this section, we define a deep neural network with ReLU activation functions accompanied by a total of L latent computational layers, each indexed by l within the set $1, 2, ..., L$ to map each h_t $1 \leq t \leq M$ corresponding to G_t to its BTM load and PV values. The layers of the proposed neural network are intrinsically defined by tunable parameters in the form of weights and biases, designated as W_l and b_l. Each layer l receives an input vector h^l and yields an output represented as O^l. The feed-forward propagation process for i-th hidden unit in this deep ReLU neural architecture is written as:

$$h_i^{l+1} = W_i^{l+1} O_i^l + b_i^{l+1} \tag{6}$$

$$O_i^{l+1} = ReLU(h_i^{l+1}) \tag{7}$$

While each dimension of the latent space h^l provides some information to the output layer, not all of this information is necessary for accurately predicting the labels Y. Some of the dimensions in the latent space may not be relevant to the target variable and could lead to inaccurate predictions. To avoid wasting computational resources and ensure that the relevant features are captured in the latent space, we propose a new dropout technique that enhances the sparsity of the latent representation in our deep ReLU neural network. Using mutual information, we developed a dropout layer to evaluate the relationship between two random variables U and V from an entropy perspective. Entropy serves as a metric for quantifying the level of uncertainty within a random variable. A high value of entropy indicates that each event within the variable has approximately equal chances of occurring, whereas a low value implies varying probabilities of occurrence for different events. The MI metric between two random variables U and V is calculated as:

$$I(U,V) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mathcal{P}(U,V) \log\left(\frac{\mathcal{P}(U,V)}{\mathcal{P}(U)\mathcal{P}(V)}\right) du \; dv \tag{8}$$

Based on (8), two random variable U and V are independent when $I(U,V) = 0$. Within our deep ReLU neural network, we identify the most critical set of neurons based on their capacity to convey valuable and task-relevant information using the defined mutual information measure. To this end, we introduce a two-part mask function f_M. The initial component of f_M calculates the MI between the activations of a hidden unit, considering a batch of input data, and the corresponding target vector for that same batch. The second part of the mask evaluates the hidden unit's significance by quantifying the MI between the activation function of the current hidden neuron and the set of units that are currently selected. Mathematically, this function can be formally expressed as:

$$f_M(O_i^l, \mathcal{Y}, S) = I(O_i^l, \mathcal{Y}) - \frac{1}{k} \sum_{O_j^l \in S} I(O_i^l, O_j^l) \tag{9}$$

where O_i^l denotes the output activation vector of i-th neuron of l-th layer for a batch of input samples $\{h_i\}_{i=1}^B$ and $\mathcal{Y}$ is the actual label corresponding to the same batch of training data. Here, B is the batch size. Also, S is the set of already selected neurons, and k is the cardinality of the set S. When a neuron has a higher MI value with its activation and target vectors and a lower value of MI with the selected neurons, it is more likely to be chosen through the MI dropout. In simpler terms, this results in a decreased probability of removal from the network. In our approach, we incorporate the mask layer defined in (9) after each hidden layer within our proposed deep sparse ReLU neural network to retain the most crucial hidden neurons and drop the remaining neurons in each layer throughout the model's training process.

D. Training Process

Algorithm 1 shows the training procedure of the proposed attention graph GRU, dictionary matrices Ψ and Φ, and the dropout-enhanced deep ReLU neural network. The presented algorithm first computes the sparse spatiotemporal features of the net load values of all nodes for a batch of samples randomly selected from the dataset. Then, the dictionary matrices as well as the sparse codes are trained to represent the spatiotemporal features using sparse feature extraction. Next, the sparse codes are used to compute the BTM loads and PV values of the residential units. Finally, the loss functions of the BTM load and PV values are propagated to train the framework in an end-to-end fashion. Here, the loss for BTM load estimation of a sample $S_{\bar{t}}$ is computed by $\frac{1}{N}\frac{1}{m+1}\sum_{i=1}^{N}\sum_{t=\bar{t}}^{\bar{t}+m}||\hat{L}_t^i - L_t^i||_2^2$. Similarly, the loss corresponding to BTM PV estimation for the sample is written by $\frac{1}{N}\frac{1}{m+1}\sum_{i=1}^{N}\sum_{t=\bar{t}}^{\bar{t}+m}||\hat{PV}_t^i - PV_t^i||_2^2$.

Algorithm 1 Training Algorithm

while Parameters not converged
- Randomly select a batch $B = \{\bar{S}_k\}_{k=1}^K$ from N samples
- For each sample $\bar{S}_k$, compute the attention Graph GRU feature denoted by h_k
- Create a dictionary dataset $H_B = \{h_k\}_{k=1}^K$ with K feature vectors (columns).
- Compute $\Psi^* = HZ^\intercal\Phi^\intercal(\Phi Q_w \Phi^\intercal)^{-1}$
- Compute Z^* in (5) using Bartels–Stewart method [8]
- Compute the BTM loads $\{\hat{L}_{\bar{t}}^i \simeq L_{\bar{t}}^i\}_{1\le i\le N}$
- Compute the BTM PV values $\{\hat{PV}_{\bar{t}}^i \simeq PV_{\bar{t}}^i\}_{1\le i\le N}$
- Backpropagate the BTM load and PV loss to train the framework in an end-to-end fashion

IV. Numerical Results

A. Dataset

In this research, we use the Ausgrid dataset [9] to train, validate, and test our proposed model and compare it with recent benchmarks. The Ausgrid dataset contains rooftop solar power measurements of 300 residential units near Sydney from July 1 in 2010 to June 30 in 2013. The dataset contains 30-minute time intervals between its measurements. We use the measured loads and PV power of the residential units from the first day of July 2010 up to the end of June in 2012. We employ 70% of the collected data for training the model, 15% for validation, and 15% for test. In this study, we consider the Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) as our performance metrics. In this dataset, we consider the time window length $m+1$ to be equal to 48.

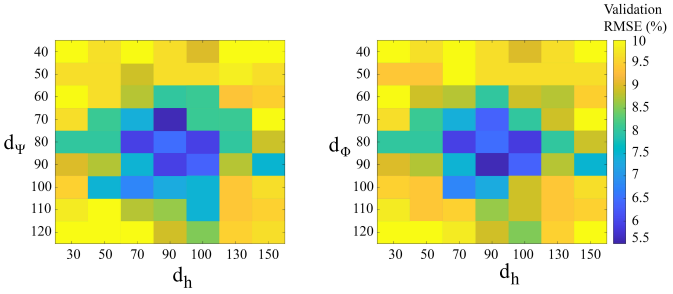


Fig. 2. Hyperparameter selection using the average validation RMSE of load and PV measurements for the residential units in the Ausgrid dataset.

B. Hyperparameter Selection

The proposed model has various hyperparameters including the objective coefficients (i.e., λ_1 and λ_2), the number of dictionary atoms (i.e., columns) of Ψ and Φ, denoted by d_Ψ and d_Φ, respectively, as well as the dimension of the GRU's hidden feature d_h, number of deep ReLU network's hidden layers L and number of neurons in hidden layers Q. We use grid search with the following domains defined for these hyperparameters: $d_\Psi, d_\Phi \in [40, 120]$, $d_h \in [30, 150]$, $L \in [2, 5]$, $Q \in [40, 120]$, $\lambda_1 \in [0, 10]$ and $\lambda_2 \in [0, 10]$. We compute the average validation RMSE of load and PV estimations for all residential units in the validation dataset to find the optimal configuration with the minimum validation error. Fig. 2 shows the relations of three hyperparameters d_h, d_Ψ, and d_Φ using the average validation RMSE of load and PV of the Ausgrid residential units. In this grid search, the following configuration is the optimal set of hyperparameters with the lower validation error: $\lambda_1 = 3, \lambda_2 = 5, d_h = 90, d_\Psi = 70, d_\Phi = 90, L = 3$, and $Q = 60$. In the optimal neural architecture, the dimension of the GRU's latent feature layer is 90, and the deep ReLU neural network has three layers where each layer contains 60 neurons.

As shown in Fig. 2, the moderate numbers for the temporal feature dimension d_h as well as number of dictionary atoms d_Ψ and d_Φ give the optimal validation RMSE. Decreasing these values would lead to a reduction of the generalization capacity of the model due to having a limited number of tunable parameters. Also, increasing the values of these dimensions would lead to overfitting issues and, hence, would increase the validation RMSE.

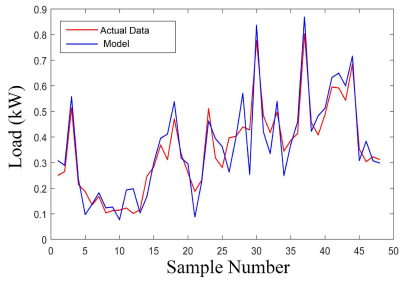
C. Experimental Settings

In this study, we train and test the proposed method and the benchmarks on a computer with one Intel Core i7-11700

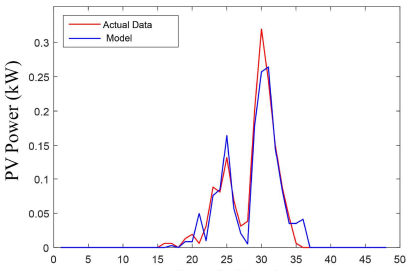
processor and one NVIDIA GeForce RTX 3090 GPU. The presented model is developed, trained, and evaluated using Python 3 and Tensorflow with GPU support [10]. The offline training process of the proposed neural architecture takes 25.41 minutes and the running time for each test sample is 15.03 milliseconds in the test process. The feed-forward algorithm in the neural network has a low time complexity, which accounts for the short test running time of the proposed method.

D. Results and Comparison

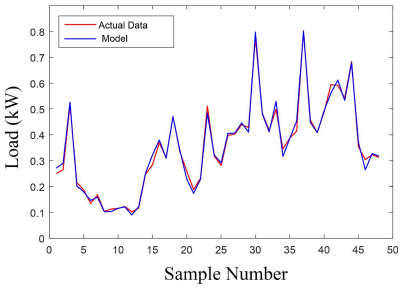
We compare the proposed framework with recent BTM load and PV disaggregation benchmarks including the linear regression strategy (LRS) [3], Bayesian structural time series (BSTS) model [4], dictionary-based energy disaggregation [5], repeated game theory with vector payoff (RGVP) [1], Graph Dictionary Learning (GDL) [11], and graph capsule network (GraphCaps) [2]. As shown in this table, LRS obtains the greatest values in RMSE and MAPE for both load and PV estimates. LRS uses the least square (LS) approach to separate the BTM load and PV from the total demand. The reason for this model's low accuracy is that complex nonlinear correlations found in the load time series cannot be well captured by linear models. The BSTS uses Bayesian rules to provide probabilistic features for the disaggregation task, hence, it captures more complex data patterns compared to the LSR as it learns probabilistic features from the data. As a result, as shown in the table, BSTS improves the results of the LSR. Moreover, the DED method improved BSTS since it leverages DL and can capture sparse features from the load data which better models the variations in load. The RVGP improves the results of DED due to applying a closed-loop game-theoric approach that can more efficiently learn the temporal dependencies between load and PV time series considering limited and noisy data samples. The GDL and GCaps define a graph neural network and capsule network for BTM load and PV disaggregation, respectively. Although these models learn the space-time relations between load and PV measurements, they do not provide data sample efficiency since they need to see many examples to tune their large number of parameters. Moreover, these methods do not explicitly capture task-relevant features while the proposed technique solves both of these issues by providing feature sparsity and attention-enhanced features. As shown in the table, the proposed framework leads to lower error functions in both PV and load estimations compared to all other benchmarks with 3.02% and 3.40% load RMSE and PV RMSE, respectively. As shown in Fig. 3, the proposed model can better capture the patterns of load and PV compared to the best benchmark, i.e., the GCaps method that employs capsule networks to capture robust features. The main reasons for this superiority are: 1) finding better sparse features due to the low-rank sparse coding and MI-based dropout, and 2) attention-enhanced spatiotemporal feature extraction that captures task-relevant recurrent features of the net load.



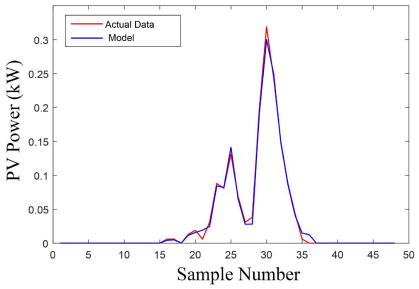
(a) Load estimation of benchmark



(b) PV estimation of benchmark



(c) Load estimation of proposed method



(d) PV estimation of proposed method

Fig. 3. BTM Load and PV estimation results of the proposed framework and the state-of-the-art benchmark, GCaps, for 48 samples showing one test day.

TABLE I
RESULTS OF BTM LOAD AND PV DISAGGREGATION USING RECENT BENCHMARKS AND THE PROPOSED METHOD.

Model	Load		PV	
	RMSE	MAPE(%)	RMSE	MAPE(%)
LSR [3]	0.2801	10.6322	0.3249	10.874
BSTS [4]	0.2561	9.8045	0.2812	9.7052
DED [5]	0.2012	8.7651	0.2168	8.6403
RGVP [1]	0.1804	8.0451	0.1872	7.8309
GDL [11]	0.1621	5.6403	0.1751	6.0491
GCaps [2]	0.1525	5.1977	0.1563	5.5320
Proposed	0.1104	3.0208	0.1183	3.4087

V. Conclusion

This paper develops a novel sparse attention graph GRU method for BTM disaggregation of load and PV measurements in residential units in a wide area. First, the power grid is modeled as a graph where the nodes show the net load measurements of the units, and the edges represent the MI of these measurements. Then, to capture the attention-enhanced space-time properties of the input dynamic graph, a new attention GRU is devised. During feature extraction, the proposed attention mechanism aids the GRU in identifying features that are significant to the BTM disaggregation task. Then, a novel low-rank DL structure makes use of the spatiotemporal characteristics to extract a sparse feature vector. The sparse features improve the proposed framework's ability to generalize and assist the model in overcoming overfitting issues. To estimate the BTM load and PV measurements of the residential units, a new deep ReLU neural network with an MI-based dropout mechanism is created and trained. By using the MI between the activations of the deep ReLU neural network's hidden units, the suggested dropout strategy finds a feature mask that eliminates units with high MI (i.e., units that are less informative). By doing this, the neural network is kept from becoming overly dependent on certain neurons. Thus, learning more resilient and universal properties is promoted.

VI. Acknowledgement

This research is supported by the National Science Foundation under grant ECCS-2223628.

References

[1] F. Bu, K. Dehghanpour, Y. Yuan, Z. Wang, and Y. Zhang, "A datadriven game-theoretic approach for behind-the-meter PV generation disaggregation," IEEE Trans. Power Syst., vol. 35, no. 4, pp. 3133–3144, Jul. 2020.

[2] M. Saffari, M. Khodayar, M. E. Khodayar, and M. Shahidehpour, "Behind-the-meter load and PV disaggregation via deep spatiotemporal graph generative sparse coding with Capsule Network," IEEE Transactions on Neural Networks and Learning Systems, pp. 1–15, 2023.

[3] E. C. Kara, C. M. Roberts, M. Tabone, L. Alvarez, D. S. Callaway, and E. M. Stewart, "Disaggregating solar generation from feeder-level measurements," Sustain. Energy, Grids Netw., vol. 13, pp. 112–121, Mar. 2018.

[4] P. Shaffery, R. Yang, and Y. Zhang, "Bayesian structural time series for behind-the-meter photovoltaic disaggregation," in Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT), Feb. 2020, pp. 1–5.

[5] W. Li, M. Yi, M. Wang, Y. Wang, D. Shi, and Z. Wang, "Realtime energy disaggregation at substations with behind-the-meter solar generation," IEEE Trans. Power Syst., vol. 36, no. 3, pp. 2023–2034, May 2021.

[6] Khodayar, Mahdi, Jianhui Wang, and Zhaoyu Wang. "Energy disaggregation via deep temporal dictionary learning." IEEE transactions on neural networks and learning systems 31, no. 5 (2019): 1696-1709.

[7] L. Wang, Z. Xiong, G. Shi, F. Wu and W. Zeng, "Adaptive Nonlocal Sparse Representation for Dual-Camera Compressive Hyperspectral Imaging," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 10, pp. 2104-2111, 1 Oct. 2017, doi: 10.1109/TPAMI.2016.2621050.

[8] Zhang, Z. and Chen, X. (2023) "Generalized conjugate direction algorithm for solving general coupled Sylvester matrix equations", Journal of the Franklin Institute, 360(14), pp. 10409–10432.

[9] E. L. Ratnam, S. R. Weller, C. M. Kellett, and A. T. Murray, "Residential load and rooftop PV Generation: An australian distribution network dataset," International Journal of Sustainable Energy, vol. 36, no. 8, pp. 787–806, 2015.

[10] Y. Kim et al., "Efficient large-scale deep learning framework for heterogeneous Multi-GPU cluster," 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W), Jun. 2019.

[11] M. Khodayar, G. Liu, J. Wang, O. Kaynak, and M. E. Khodayar, "Spatiotemporal behind-the-meter load and PV power forecasting via Deep Graph Dictionary learning," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 10, pp. 4713–4727, 2021.