Multi-Quantile Recurrent Neural Network for Feeder-Level Probabilistic Energy Disaggregation Considering Roof-Top Solar Energy

Xiao-Yu Zhang[a,b]*, Chris Watkins[b],Stefanie Kuenzel[a]

[a]*Department of Electronic Engineering, Royal Holloway, University of London, Egham Hill, Egham,TW20 0EX, UK*
[b]*Department of Computer Science, Royal Holloway, University of London, Egham Hill, Egham,TW20 0EX, UK*

Abstract

The purpose of feeder-level energy disaggregation is to decouple the net load measured at the feeder-head into various components. This technology is vital for power system utilities since increased visibility of controllable loads enables the realization of demand-side management strategies. However, energy disaggregation at the feeder level is difficult to realize since the high penetration of embedded generation masks the actual demand and different loads are highly aggregated. In this paper, the solar energy at the grid supply point is separated from the net load at first via either an unsupervised upscaling method or the supervised gradient boosting regression tree (GBRT) method. To deal with the uncertainty of the load components, the probabilistic energy disaggregation models based on multi-quantile recurrent neural network model (multi-quantile long short-term memory (MQ-LSTM) model and multi-quantile gated recurrent unit (MQ-GRU) model) are proposed to disaggregate the demand load into thermostatically controlled loads (TCLs), non-thermostatically controlled loads (non-TCLs), and non-controllable loads. A variety of relevant information, including feeder measurements, meteorological measurements, calendar information, is adopted as the input features of the model. Instead of providing point prediction, the probabilistic model estimates the conditional quantiles and provides prediction intervals. A comprehensive case study is implemented to compare the proposed model with other state-of-the-art models (multi-quantile convolutional neural network (MQ-CNN), quantile gradient boosting regression tree (Q-GBRT), Quantile Light gradient boosting machine (Q-LGB)) from training time, reliability, sharpness, and overall performance aspects. The result shows that the MQ-LSTM can estimate reliable and sharp Prediction Intervals for target load components. And it shows the best performance among all algorithms with the shortest training time. Finally, a transfer learning algorithm is proposed to overcome the difficulty to obtain enough training data, and the model is pre-trained via synthetic data generated from a public database and then tested on the local dataset. The result confirms that the proposed energy disaggregation model is transferable and can be applied to other feeders easily.

Keywords:
Probabilistic energy disaggregation, behind-the-meter PV generation,quantile regression, deep neural network, machine learning, electricity distribution networks;

1. Introduction

1.1. Motivation

Demand response (DR) plays a critical role in the future smart grid, introducing flexibility and controllability in the end-use consumers' power consumption patterns. As a result, peak demands are reduced, and the mismatch between generation and demand is minimized. In incentive-based DR schemes, the utility would control specific loads directly during a certain period for load shaping. So, understanding the portion of controllable loads is vital for the utility to design DR strategies.

Loads can be divided into critical loads and controllable loads [1]. Meanwhile, controllable loads can be further divided into thermostatically controlled loads (TCLs) and non-thermostatically controlled loads (non-TCLs). TCLs (e.g., heating, ventilation and air conditioning (HVAC), air conditioner (AC), heat pumps, furnace, and refrigerators) occupy 30-40% of the overall demand load [2], and TCLs are widely adopted in DR for their thermal inertia capability. Recent research shows that by optimizing the operation of HVAC systems, 45% of energy would be saved [3]. Moreover, Electric Vehicles (EVs) have high flexibility to schedule the charging/discharging slot, and this characteristic can benefit DSR by shaving the peak load. In addition, the high penetration of embedded generation (e.g., solar photovoltaic) masks the ground truth demand. However, most DR frameworks are planned for pure load, the poor visibility of the actual load caused by embedded generation influences the efficiency of the existing DR schemes. Hence, it is essential to increase the visibility of the load components by disaggregating the net load measured at feeder/substation level into renewable energy generation, TCLs, and non-TCLs.

1.2. Literature Review

Artificial intelligence (AI) based feeder-level energy disaggregation approaches are introduced in [4-13]; the objective of this method is to decouple the feeder-level net demand into load components. The above approaches can be further divided into model-based [9-11] and measurement-based [4-8] methods.

The model-based method is presented in [9-11], which combines the ZIP load model with artificial neural network algorithms. Synthetic data is built based on the ZIP/exponential load model, and then Monte Carlo simulation is used to generate synthetic training and validation data. By changing the weights of load components and voltage, a few active power and reactive power measurements are obtained, which are used for model training/validation. Moreover, a two-layer feedforward shallow neural network is built to estimate the portion of each load category from the total load demand measured at the substation level [10]. A multi-modal long short-term memory (LSTM) is introduced in [9] to identify the time-varying ZIP load and induction motor (IM) model parameters. The accuracy of the algorithm is increased by considering different modalities of the input data. The advantage of this method is that the dataset can be easily constructed, referring to the ZIP/ exponential load model. The limitation of this approach is that the dataset used in the case study is synthetic, and the trained model cannot be used in a real-world case.

Unlike the model-based method that uses a synthetic dataset, the

* Corresponding author.
E-mail address: Xiaoyu.Zhang.2018@live.rhul.ac.uk

Table 1.
Nomenclatures.

List of Abbreviations			
AACE	Absolute Average Coverage Error	Q-GBRT	Quantile Gradient Boosting Regression Tree
AC	Air Conditioner	Q-LGB	Quantile LightGBM
AI	Artificial Intelligence	RNN	Recurrent Neural Network
BTM	Behind-the-Meter	SCADA	Supervisory Control and Data Acquisition
DHI	Diffuse Horizontal Irradiance	TCL	Thermostatically Controlled Load
DNI	Direct Normal Irradiance	WS	Winkler Score
DNN	Deep Neural Network		
DR	Demand Response	*List of Symbols*	
DSM	Demand-Side Management	$A(k)$	Approximation Coefficients
DWT	Discrete Wavelet Transform	$C_{t,\tau}$	Cell State
EV	Electric Vehicle	$D(k)$	Detail Coefficients
GB	Gradient Boosting Machine	E_τ	Quantile Optimization Function
GBQR	Gradient Boosted Quantile Regression	$f_{t,\tau}$	Forgot Gate
GRU	Gated Recurrent Unit	$F_Y(y)$	Cumulative Distribution Function of Y
GHI	Global Horizontal Irradiance	G_t^{PV}	PV Generation
GSP	Grid Supply Point	$i_{t,\tau}$	Input Gate
HVAC	Heating, Ventilation and Air conditioning	L_t^{EV}	Electric Vehicle Demand
LSTM	Long Short-Term Memory	L_t^{feeder}	Total Demand Load
MQRNN	Multi-Quantile Recurrent Neural Network	L_t^{FURN}	Furnace Demand
MQ-GRU	Multi-Quantile Gated Recurrent Unit	$L_t^{non-TCL}$	Non-TCLs Demand
MQ-LSTM	Multi-Quantile Long Short-Term Memory	L_t^{AC}	Air Conditioner Demand
MQ-CNN	Multi-Quantile Convolutional Neural Network	Net_t^{feeder}	Net Load Demand
NCEI	National Centres for Environmental Information	$o_{t,\tau}$	Output Gate
NILM	Nonintrusive Load Monitoring	$q_Y(\tau)$	τ-quantile of a Random Variable Y
Non-TCL	Non-Thermostatically Controlled Load	Thr	Soft Thresholding
nMAE	Normalized Mean Absolute Error	U_i	Upper Boundary of PIs
OL	Other Loads	$\rho_\tau(\mu)$	Pinball Loss Function
PIs	Prediction Intervals	ρ	Pearson Correlation Coefficient
PICP	Prediction Interval Coverage Probability	ε_t	random noise
PINC	Prediction Interval Nominal Confidence		
PV	Photovoltaic		

measurement-based approach utilizes real-world smart meter measurements. Ledva *et al.* [5] proposed an online learning method to separate air conditioners demand (AC demand) from the demand load. Household-level smart meter measurements provided by the Pecan Street Dataport [14] are aggregated to build a feeder-level load. Then an online learning algorithm, Dynamic Fixed Share (DFS), is adopted to perform energy disaggregation by considering measurements from both substation and weather stations. Based on [5], an improved algorithm that combines model-based method and measurement-based method is presented in [6]. Substation, feeder, and smart meter measurements (active power, reactive power, complex voltage, and complex current) are utilized together to enhance the flexibility of the algorithm. The online learning algorithm, Dynamic Mirror Descent (DMD), keeps iterating for both measurement-based updates and model-based updates. The difficulty of the measurement-based approach exists in the difficulty of obtaining data to train the model.

Researchers further work on increasing the visibility of behind-the-meter (BTM) solar energy by decoupling the solar energy from the net load [4, 8]. Unlike traditional demand load, the generation of solar energy is highly related to solar irradiance data and meteorological data. In [8], a regional nonintrusive load monitoring (regional NILM) algorithm is proposed to disaggregate solar energy and electric vehicles (EVs) loads from the substation demand. The data used for the case study is a combination of three data sources (substation demand dataset, solar energy dataset, and EV dataset), where each component is separated individually using their proposed three-stage disaggregation framework. The substation demand is the first forecast via empirical mode decomposition (EMD), then the solar energy is estimated by matching the linear correlation between the solar irradiance and the PV outputs. Finally, the EVs load is estimated via the limited activation matching pursuit (LAMP) method. [4] views the energy disaggregation as a partially labelled dictionary learning problem. By training offline model with historical datasets with partial labels, the system can efficiently separate three categories of load that includes solar energy. However, in practical application, the situation is more complex than the case study they implemented. There are more than three categories of load are aggregated at the same time. Other solar energy disaggregation methods include linear regression, Kalman filter [12, 13], gradient boosting regression tree (GBRT) [15], multi-layer perceptron (MLP) neural network [16], Gaussian mixture modelling (GMM) [17]. Nonetheless, these approaches only focus on separating solar energy, and other load components remain unseparated from their research.

Another research area that correlates to the proposed method is probabilistic estimation. Probabilistic estimation was used in power systems and energy discipline with great success, e.g. Load forecasting [18-22], locational marginal prices forecasting [23], probabilistic real-time thermal rating (RTTR) forecasting [24, 25], and wind forecasting tasks [26]. Probabilistic estimation utilizes a variety of approaches such as quantile regression (QR), quantile GBRT (Q-GBRT), regression neural network (QRNN) methods to estimate the results in the forms of quantiles prediction intervals (PIs), etc. As a typical uncertainty quantification approach, PIs set the upper and the lower bounds to quantify the level of uncertainty, and the corresponding PI nominal confidence (PINC) is provided as well (for instance, PINC equals 95% with 0.975 as upper bound and 0.025 as lower bound).

1.3. Contribution

In the literature for feeder-level energy disaggregation, some knowledge gaps should be filled and can be concluded as follows.

(1) In the literature, only grid measurements (e.g., active/ reactive power, voltage) are utilized as inputs of the model. However, load demand is influenced by other variables such as meteorological data, calendar data. Hence, a model which considers all relevant variables should be proposed.
(2) Although both machine learning and deep learning algorithms are introduced in the literature, the uncertainty of energy disaggregation is not discussed.

(3) The transferability of the energy disaggregation technique is not investigated, it is vital to validate whether the proposed method can be used in different areas.

To fill the knowledge gaps as mentioned above, a feeder-level probabilistic energy disaggregation scheme is proposed in this paper. Detailed novelties of this work are listed as follows.

(1) The scheme utilizes a multi-quantile long short-term memory neural network (MQ-LSTM) to disaggregate various load components (TCLs, Non-TCLs, PV generation, and other loads). The proposed model enables both online and offline modes, and a variety of features are selected as the input variables of the proposed model, including feeder measurement, meteorological measurement, calendar and holiday information, and environmental measurement. The proposed model is also compared with other state-of-the-art quantile regression models (multi-quantile gated recurrent unit (MQ-GRU), multi-quantile convolutional neural network (MQ-CNN), quantile gradient boosting regression tree (Q-GBRT), and quantile light gradient boosting machine (Q-LGB)). In previous data-driven methods, only one or two components are separated from the netload.

(2) A Discrete Wavelet Transform (DWT)-based data denoising method is adopted to filter out the noise which exists in the power measurements.

(3) Two solar energy separation methods are proposed, utilizing the unsupervised upscaling method or the supervised gradient boosting method to separate the solar energy generated by the rooftop PVs from the net load measured at the grid supply point (GSP) on a real-time basis.

(4) A transfer learning model is introduced to transfer the energy disaggregation model trained with a public dataset to a local dataset. The transferability overcomes the problem of the data shortage.

2. The Preliminaries

2.1. Problem Statement

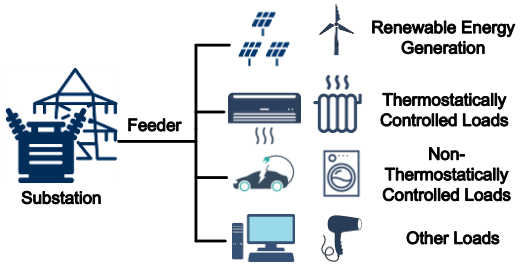
The target of this paper is to disaggregate the overall feeder-level load demand into four components, which are: TCLs, non-TCLs, renewable generation, and other loads (OL) in both real-time and offline mode (See Fig. 1(a)). From Fig. 1(b), the percentages of different loads under the feeder demand are presented, referring to [4]. From the figure, it is found that AC occupies 33% of the overall demand load. In this paper, air conditioner and furnace are selected as the TCLs, and EVs are considered as non-TCL load, rooftop PVs are considered as an embedded generation. Assuming the feeder-level net load is measured as Net_t^{feeder}, the problem can be expressed by the following formula:

$$Net_t^{feeder} = \mathrm{L_t^{feeder}} + \mathrm{G_t^{PV}} + \varepsilon_t \quad (1\text{-}1)$$

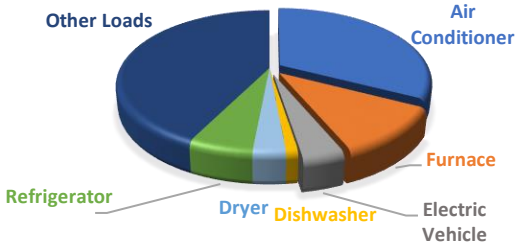
$$= L_t^{TCL} + L_t^{non-TCL} + L_t^{OL} + G_t^{PV} + \varepsilon_t \quad (1\text{-}2)$$

$$= L_t^{AC} + L_t^{FURN} + L_t^{EV} + L_t^{OL} + G_t^{PV} + \varepsilon_t \quad (1\text{-}3)$$

where L_t^{feeder} is the actual demand load, L_t^{TCL} is the TCLs demand, $L_t^{non-TCL}$ is the non-TCLs demand, L_t^{OL} is OL demand, G_t^{PV} is the PV generation, L_t^{AC} is the AC demand, L_t^{FURN} is the furnace demand, L_t^{EV} is the EVs demand, and ε_t is the random noise.



(a)



(b)

Fig. 1. (a) Load components under substation/feeder; (b) Portion of loads under the feeder demand (Data source: Pecan Street Dataport [4]).

2.2. Comparison Among Similar Problems

Three similar problems should be distinct in this paper, which are feeder-level energy disaggregation, load forecasting and house-level NILM. Household-level NILM is a technique to obtain individuals' appliance consumption from overall household-level power consumption without installing intrusive sensors, such as a smart plug or smart sensors. Since most appliances have unique characteristics in load curve or voltage curve, it is easy to separate every single appliance from the overall load with algorithms such as Hidden Markov Model (HMM), Recurrent Neural Network (RNN), and K-Nearest Neighbors (KNN). But when the situation comes to feeder level or substation level, the load curve is highly aggregated and contains hundreds even millions of household-level load curves; the characteristic of the single appliance is difficult to be detected with power measurements only. Meanwhile, load forecasting technology aims to make a prediction of demand load with both long-term and short-term horizons, given historical demand load data.

2.3. Input Variables and Data Analysis

Input variables of the energy disaggregation system are classified into four categories, including feeder measurement, meteorological measurement, time measurement, and solar irradiance, as shown in Table 2. Meanwhile, all input variables can be divided into two categories, which are numerical variables and categorical variables. Numerical variables represent the values that can be measured and placed in logical order. By contrast, categorical variables take values that are names or tags, and the number of potential values is often limited to a fixed series. These categorical variables cannot be recognized by DNN models and must be converted into a numerical form. The conversion method adopted in this work is one-hot encoding [27]. Instead of providing a single integer only, one-hot encoding provides a set of binary variables. A detailed description of the input variables is presented as follows:

Table 2
Input variables of the energy disaggregation model.

Feature type	Description	Mark
Feeder measurement		
Feeder active power flow	One week (672) lagging values and current value	$L_{t-672}, L_{t-671}, \cdots L_{t-1}, L_t$
Meteorological measurement		
Past temperature values	One week (672) lagging values and current value	$T_{t-672}, T_{t-648}, \cdots T_{t-1}, T_t$
Humidity	Humidity in current time step	HM
Wind speed	Wind speed in current time step	WS
Pressure	Pressure in current time step	P
Weather description	10 binary values for each weather condition	$WD_1, WD_2 \cdots WD_{10}$
Cloud cover	Cloud cover rate in current time step	C
Calendar information		
Day type	2 binary values for each type of day (weekday/weekend)	D_1, D_2
Holiday	4 binary values for a normal day or current/previous/after day is a holiday	L_1, L_2, L_3, L_4
Season	4 binary values for each season in one year	S_1, S_2, S_3, S_4
Month	12 binary values for each month in one year	$M_1, M_2 \cdots M_{12}$
Hour	24 binary values for each hour in one day	$H_1, H_2, \cdots H_{24}$
Solar irradiance for PV separation		
GHI	GHI in current time step	GHI
DNI	DNI in current time step	DNI

DHI	DHI in current time step	DHI
Latitude	Latitude of the PV site	Lat
Longitude	Longitude of the PV site	$Long$

2.3.1. Feeder-level Demand and Appliance Load Data

The feeder models used for this research are selected from standard feeder models provided by GridLAB-D [28]. The feeder types are classified depending on the residence description, ranging from light rural to moderate urban (with apparent power from 948 kW to 17021 kW). To construct the feeder-level demand load data, individual household-level smart meter data from Pecan Street Dataport (Dataport) [14] are added up to match the capacity of the feeder model. Dataport is the world's largest residential energy dataset, and it contains more than 700 houses, and each house measures around 20 electrical appliances in Texas, US. Moreover, the interval resolution of the smart meter data is 15 minutes. In this work, a total number of 3691 houses are aggregated to match the R2-25.00-1 feeder model (demand capacity is 17021 kW). As the demand load from the individual houses is added together, feeder-level appliance demands, including AC, furnace, EV, and PV, are also obtained. It is noticed that the furnace in the Texas area not only plays the role of a heating system to provide heat during winter but is also used to circulate cooled air during other seasons.

2.3.2. Meteorological Measurement

Both the demand load and PV generation are correlated with meteorological data strongly, so it is vital to include meteorological measurements into the input variables. In this paper, meteorological data resources from the geographical point N 30° 15' 59.9976", W 97° 43' 59.9880" is used (Austin, Texas, US), the data is provided by National Centres for Environmental Information (NCEI) [18]. Numerical variables, ambient temperature $\boldsymbol{T}$, humidity $\boldsymbol{HM}$, pressure $\boldsymbol{P}$, wind speed $\boldsymbol{WS}$, cloud cover $\boldsymbol{C}$ are chosen from the NCEI dataset. For ambient temperature, 673 variables are generated, spanning the last week and current temperature measurements:

$$\boldsymbol{T} = [\boldsymbol{T_t}, \boldsymbol{T_{t-1}}, \cdots \boldsymbol{T_{t-671}}, \boldsymbol{T_{t-672}}] \quad (2)$$

Fig. 2 shows the correlation between $\boldsymbol{T}$ and different loads at the feeder level. From the figure, it is observed that TCLs are highly influenced by temperature and relevant weather variables (e.g., humidity, wind speed). AC has a positive correlation with temperature; as $\boldsymbol{T}$ increases, the power consumption of AC rises as well. Since the furnace has a dual role (heating and circulating) in this research when $T < 13℃$, the correlation between $\boldsymbol{T}$ and furnace demand load is negative, and when $T > 18℃$, the correlation turns to positive. Meanwhile, $\boldsymbol{T}$ has little influence on the non-TCL demand (such as EV) as the curve is flat throughout different temperature periods.

Moreover, categorical meteorological data, weather description WD also has an extraordinary impact on the demand load. 10 different weather conditions are described in the NCEI dataset, which is: Mist, Clouds, Snow, Clear, Rain, Drizzle, Haze, Thunderstorm, Fog, Dust.

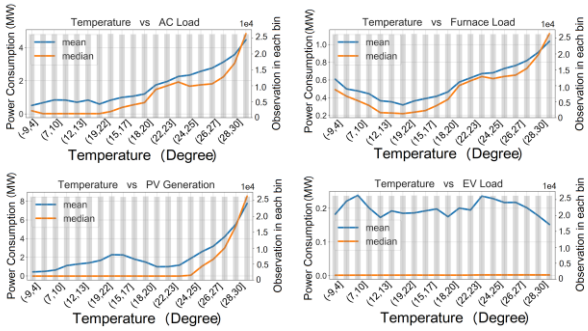


Fig. 2. Correlation temperature and various loads (Data source: Pecan Street Dataport [4])

2.3.3. Calendar Data and Holiday Information

Calendar data and holiday information is another vital factor that influences consumers' behaviours and the electricity events that happened inside their houses. As shown in Table 2, time information variables include:

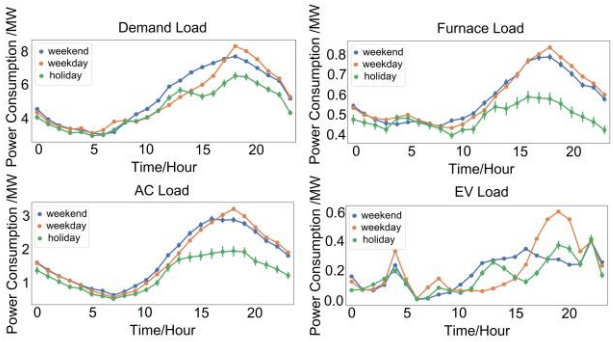


Fig. 3. Net/appliance load profiles under weekday, weekend, and holiday.

1) Type of the day. Day types include weekday, weekend, and holiday. Including the day types enable the disaggregation model to be sensitive to the variation of the week. In Austin, Texas, 14 days are marked as a holiday in 2018, referring to [29]. Considering the influence of holiday on residents would span before or after the holiday, the day before the holiday and the day after the holiday is also viewed as new variables. Hence, four binary variables are used to represent a holiday. Fig. 3 make a comparison among typical load profile during weekday, weekend, and holiday. The peak loads of both overall demand load, TCLs, and EV are higher than other types, while the peak of loads during weekends is clipped. A dramatic reduction of all demands is observed during holidays, especially for furnace and AC loads. This is due to a part of residents leaving their houses to travel somewhere else rather than staying at home.

2) Season S. Seasonal variation (Spring, Summer, Autumn, Winter) is also a critical factor that influences the demands, and consumers prefer different electricity appliances during different seasons. For instance, AC is typically used during summer for cooling, and the heating system is more preferred in winter for heating purposes. In this paper, four binary variables S_1, S_2, S_3, S_4 are used to represent the season, e.g., [0,0,0,1] is used to represent spring. Fig. 4 uses the Stats-Violin plot to show the distributions of the load demand of different load components in four seasons. From the figure, it is found that the power consumption of AC load in summer and autumn is much larger than it in spring or winter. As for the furnace, since it plays a dual role as a heating system and air circulation device, the demand for the furnace is high in both summer and winter. The distribution of EV does not show any difference among various seasons, which shows that the EV charging/discharging activities are not influenced by seasons significantly. Finally, solar energy generation is influenced by the season significantly as the PVs generate more power during summer and autumn.

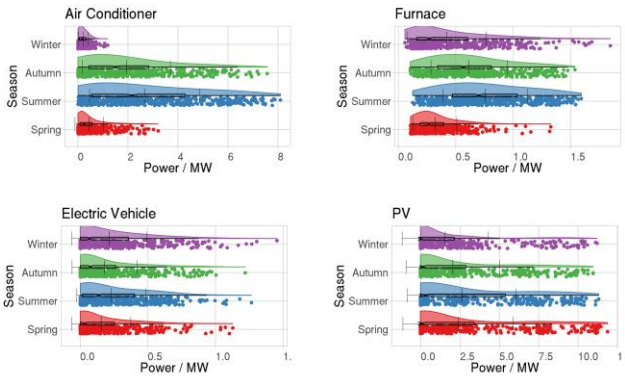


Fig. 4. Stats-Violin plot of appliance load profiles under seasons.

3) The Hour of the day H and the month of the year M. 24 binary variables and 12 binary variables are used to represent hour and month, respectively.

2.3.4. Solar Irradiance Measurement for Solar Energy Separation

Rooftop PV generation is highly correlated with solar irradiance measurement. The irradiance measurements and weather data at the exact

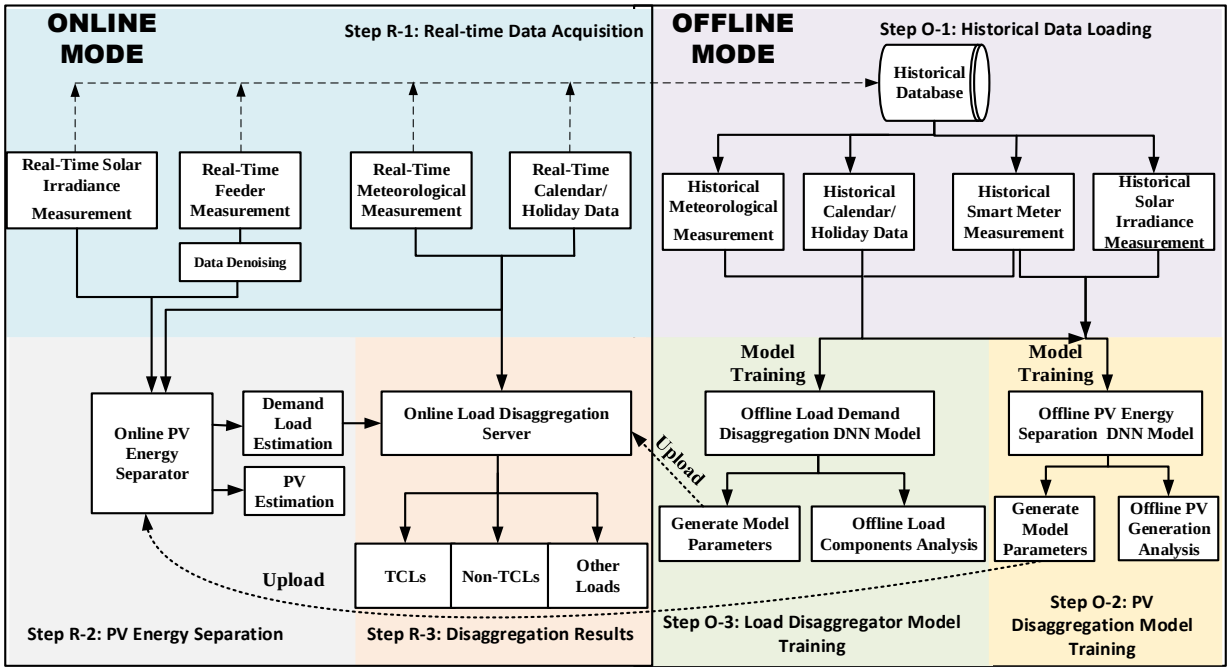


Fig. 5. Online/Offline PV energy disaggregation framework.

location are obtained from the National Climatic Data Center (NCDC) [30]. Satellite-driven data include Global Horizontal Irradiance (GHI), Direct Normal Irradiance (DNI), Diffuse Horizontal Irradiance (DHI), latitude, longitude, etc.

3. Energy Disaggregation Scheme

3.1. System Overview

The framework of the proposed multi-quantile RNN energy disaggregation system contains two modes, which as online and offline, which are shown in the block diagram of Fig. 5. In the offline mode, the energy disaggregation model is trained with a historical dataset, and offline analysis is also implemented for grid planning purposes. In online mode, the real-time net load measurement is disaggregated into individual load components.

3.1.1. Offline Mode

The offline mode has two functions: (1) training the DNN model and uploading the trained model parameters to the cloud server; (2) analysing the load components of historical feeder demand. Typically, the offline mode should be operated by the power utility or relevant third parties to build the energy disaggregation models.

Step 1: Historical data loading. Historical data is loaded from the historical database. This database contains historical data for meteorological measurements, calendar/holiday information, solar irradiance, and smart meter telemetry. The historical smart meter data contains both household-level and appliance-level power consumption data, so the smart meter data can be aggregated to generate feeder-level demand load, as stated in Section 2.3.1. It is noticed that for an area where smart meter data is not available, a transfer learning approach should be utilized, which will be introduced in Section 5.3**.**

Step 2: Training the PV separation model. Given historical data as mentioned above as input, historical PV generation as outputs, a gradient boosting (GB) machine learning model is trained to implement the solar energy disaggregation task. After the training process, the trained model, as well as model parameters, are uploaded into the cloud server, where the online PV separation is implemented. The trained offline model can also be used for data analysis and grid planning purposes, given historical data.

Step 3: Training load disaggregation model. Step 3 trains the offline model to implement demand load disaggregation. There are two points of difference compared to Step 2. Firstly, the input variables and outputs of the models are different. Compared to the PV separation model, the load disaggregation model does not require solar irradiance data. In contrast, holiday information is added to the inputs as manual activities are highly influenced by such special events. Secondly, instead of taking PV generation as input, the load disaggregation model takes the portions of each load component as outputs. Furthermore, the trained model and model parameters are also uploaded to the cloud server for online estimation purposes.

3.1.2. Online Mode

In online mode, the power utility would like to use the online server to analyse the load components on a real-time basis. The models that are trained in offline mode are uploaded to the online server, so the utility can implement online computing without training the models at the same time.

Step 1: Real-time measurements collection. The utility receives the real-time feeder demand measurement from the feeder-level smart meter or supervisory control and data acquisition (SCADA) system. Meanwhile, the utility can also access the real-time meteorological measurement provided by the local weather station. The real-time calendar data are generated by the system or online server such as Google Calendar [31]. Moreover, holiday data are provided by the local government. All real-time measurements are synchronized and preprocessed (normalization for numerical variables and one-hot encoding for categorical variables) before feeding into the online server. In addition, the noise that exists in the feeder measurement and communication would influence the performance of the disaggregation server. Hence, a data denoising method is adopted to filter out noise. A detailed description of the denoising method is introduced in the following section. Primarily, real-time satellite solar irradiance data provided by NCDC [21] is also obtained to separate solar energy components. All real-time data is also saved into a historical database to update the offline models frequently.

Step 2: Real-time PV generation separation. Before disaggregating the feeder demand into individual load components, the PV generation components are separated from the net load as the negative loads would impact the detection of other positive loads. The online PV separator receives denoised feeder demand as well as other real-time measurements as inputs. It is noticed that the holiday

information is not needed for the PV separator as human activities do not influence PV generations. The outputs of the PV separator are PV generation and demand at the current timestep.

Step 3: Real-time Energy disaggregation. The estimated load of the PV separator is then fed into an online energy disaggregation server along with real-time meteorological measurement and calendar/holiday information. The online load disaggregation server obtains the DNN model and model parameters from the offline mode, to disaggregate the estimated demand load into three components, which are: TCLs, Non-TCLs, and OLs.

3.2. Data Denoising

The theory of the DWT-based denoising technique [32, 33] is to decompose the original signal into the high frequency and low-frequency components, and then thresholding the high-frequency components for denoising purposes, before reconstructing the de-noised signal. Assume y_i represents the original noisy data, while ϵ_i is the white Gaussian noise, and σ shows the intensity of the noise. A detailed introduction of DWT is presented in Appendix A. The purpose of the signal denoising is to remove noise and find the best estimation of the underlying signal $x(i)$:

$$\mathbf{y}(\boldsymbol{i}) = \boldsymbol{x}(\boldsymbol{i}) + \boldsymbol{\sigma}\boldsymbol{\epsilon}_{\boldsymbol{i}}, \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{n} \tag{3}$$

where n is the total sample numbers of the discrete form of the signal. A two-level DWT decomposition process is shown in Fig. 6. The denoising approach includes three steps: signal decomposition, denoising, and reconstruction.

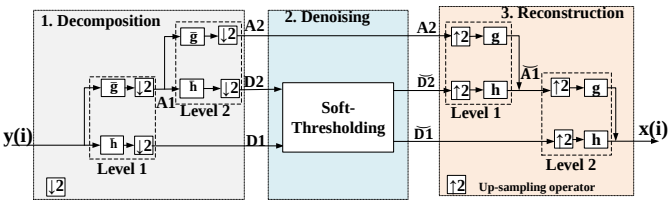


Fig. 6. DWT-based data denoising technology.

Step 1. Decomposition. The decomposition process utilizes two pairs of high-pass and low-pass filters. The noisy signal y(i) can be treated as the initial approximation signal. By passing y(i) through a series of high-pass and low-pass filters, the detail coefficients are given via a high-pass filter, and the approximation coefficients are given via a low-pass filter. At each decomposition level, the decomposed signals are downsampled by two to satisfy Nyquist's criterion. The kth level decomposition functions are:

$$A(k) = \sum_n x(n)h(2k-n) \tag{4}$$
$$D(k) = \sum_n x(n)g(2k-n) \tag{5}$$

where $x(n)$ represents the input signal, h and g are low-pass and high-pass filters, respectively. And k is the number of total decomposition levels. After passing each pair of filters, downsampling by two is implemented to compute the lower-level detail coefficients $D(k)$ and approximation coefficients $A(k)$, respectively.

Step 2. Thresholding. The detail coefficients $D(k)$ are shrunk by adopting a thresholding function Thr. Thresholding can be divided into hard and soft thresholding. In this paper, the soft thresholding is adopted, the magnitude of coefficients is more significant than the thresholding is softened. The soft threshold values can be computed using (6):

$$Thr = \delta_{mad}\sqrt{2ln\,(N)} \tag{6}$$

δ_{mad} in (6) is the standard deviation of the noise and can be calculated via (7):

$$\delta_{mad} = \frac{MAD}{0.6745} \tag{7}$$

where MAD is the median value of wavelet coefficients, and N is the total amount of coefficients. After the threshold is determined, the soft thresholding function is applied to reduce the magnitude of the coefficient as (8):

$$\breve{D} = \begin{cases} D - Thr & if\, x \geq Thr \\ D + Thr & if\, x \leq -Thr \\ 0 & if\, |x| \leq Thr \end{cases} \tag{8}$$

where $\breve{\text{D}}$ is the updated detail coefficients.

Step 3. Reconstruction. The shrunk coefficients detail coefficients $\breve{\text{D}}(k)$ and approximation coefficients $A(k)$ are reconstructed again via Inverse Discrete Wavelet Transform (IDWT) to the denoised signal. The down-sampling function in the decomposition is replaced by the up-sampling function in reconstruction.

3.3. Behind-the-Meter Solar Energy Detection – Two Methods

Referring to Step O-1 in Fig. 5, a PV separator should be trained to detect the BTM solar energy and separate the solar energy G_t^{PV} and ground truth demand load L_t^{feeder} from the net load Net_t^{feeder}, see Equation (1-1). In this work, two solar energy detection algorithms are proposed, which are an unsupervised upscaling method (UM), and a supervised gradient boosting regression tree (GBRT)- based algorithm.

3.3.1. Method I: Unsupervised Upscaling Method

As shown in Fig. 7 (a), the PV generation highly correlates with the ambient temperature T and solar irradiance. More solar energy is generated given larger GHI and higher T. Moreover, Fig. 7 (b) plots the PV generation and GHI in one week together. From the figure, it is observed that the shape of the PV generation curve is highly overlapping with the curve of GHI at the same area. The unsupervised learning approach utilizes real-time GHI measurement and historical feeder measurements only to estimate the PV outputs.

3.3.1.1. Estimate PV Capacity by Edge Detection

The PV capacity C under the feeder is first estimated via an edge detection method. Assuming the load demand under the feeder keeps stable, given historical feeder demand before PVs installed $P_{without\ PV,t}$ and feeder demand after PVs installed $P_{with\ PV,t}$, the mismatching between $P_{without\ PV,t}$ and $P_{with\ PV,t}$ can be calculated by (9):

$$Error_t = P_{without\ PV,t} - P_{with\ PV,t} \tag{9}$$

As shown in Figure 7(c), the PV capacity C is equal to the maximum of $Error$ throughout the whole year approximately:

$$C \approx max\ (Error_t) \tag{10}$$

3.3.1.2. Estimate PV Output

The PV output is estimated via normalized GHI and PV capacity C:

$$P_{PV,t} = C \cdot GHI_t \tag{11}$$

The unsupervised method is easy to implement and does not require training the model, and only a few measurements are needed. This method is highly suitable for areas that lack smart meters. The unsupervised method does not consider other relevant variables such as temperature, cloud cover rate, so the model cannot provide an exact estimation.

3.3.2. Method II: Supervised Gradient Boosting -based Method

The supervised GBRT-based solar energy detection algorithm requires training the machine learning model in offline mode before uploading the model to the cloud server in the online mode. Apart from GHI, and feeder demand measurements, historical PV outputs, as well as other variables introduced in Table 2 and Figure 5, are also adopted as input features. For areas where historical PV outputs are not available, a synthetic data generation approach introduced in [15] can be used to generate training data. The approach can generate PV outputs via System Advisor Model (SAM) simulation software [34] and combines the synthetic PV outputs with historical demand load data to simulate the feeder with solar energy penetrated.

The core component, the GBRT algorithm, is a kind of machine learning algorithm that produces a prediction model from a series of weak prediction models [35]. Usually, the GBRT algorithm contains three elements: a differentiable loss function for optimization, a squared error is adopted as the

loss function for regression task; a weak prediction model to make a prediction, a decision tree is used as the weak model in GBRT; and an additive model that can add all weak models together and minimize the losses, see Algorithm 1.

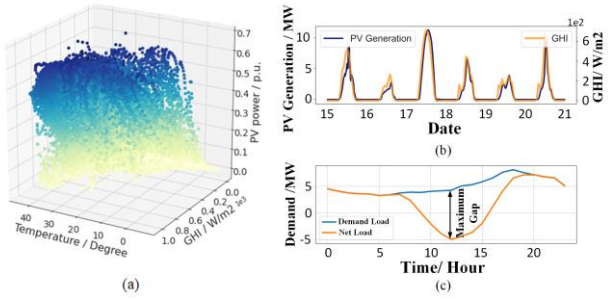


Fig. 7. (a) 3D plot of the combined effect of temperature and GHI to PV output (b) Comparison of PV output and GHI; (c) Comparison of net load and demand load.

Algorithm 1: Gradient Boosting algorithm

1. **Input:** Dataset $(x,y)_{i=1}^{n}$, where x the input features and y the target, $F(x)$ the prediction model, loss function $L(y,F(x)) = \frac{1}{2}(y-F(x))^2$, learning rate ν $(0<\nu<1)$, Iteration number M.
2. Initialization. Set the $F_0(x) = \underset{\gamma}{argmin}\sum_{i=1}^{n} L(y_i,\gamma)$;
3. For m =1 to M:
   a. Compute pseudo-residuals
   $r_{im} = -\left[\frac{\partial L(y_i,F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} = y_i - F(x_i)$
   for $i = 1,2\cdots n$;
   b. Fit a weak learner (regression tree in this case) to the r_{im} values (training model with data $\{(x_i,r_{im})\}_{i=1}^{n}$And create terminal regions R_{jm} for $j = 1,2\cdots J_m$;
   c. Compute $\gamma_{jm} = \underset{\gamma}{argmin}\sum_{x_i\in R_{ij}} L(y_i,F_{m-1}(x)+\gamma)$
   d. Update $F_m(x) = F_{m-1}(x) + \nu\sum_{j=1}^{J_m}\gamma_{jm}$.
4. **Output**: $F_M(x)$.

3.4. Domestic Loads Disaggregation at Feeder-Level

The demand load L_t^{feeder} estimated by the solar energy separator introduced in Section 3.3 is then used as input variables of the energy disaggregation model. The purpose of the model is to separate demand load L_t^{feeder} into TCLs (AC and furnace), Non-TCLs (EV) and OL, as illustrated in (1).

In this work, one-week historical load demand $\boldsymbol{L}_F$ with interval 24 h are generated as the input variables of the energy disaggregation model:

$$\boldsymbol{L}_F = \left[L_t^{feeder}, L_{t-24h}^{feeder}, \cdots, L_{t-648h}^{feeder}, L_{t-672h}^{feeder}\right] \tag{12}$$

The core component of the energy disaggregation model is the multi-quantile long short-term memory (MQ-LSTM). The detailed description of the MQ-LSTM algorithm is introduced as follows.

3.4.1.1. Multi-Quantile Long Short-Term Memory (MQ-LSTM)

MQ-LSTM is a technology that is built based on traditional LSTM, and it enables the LSTM neural network to make probabilistic predictions by combining quantile regression (QR) with LSTM units. A more comprehensive analysis of dependent variables can be obtained by the measures of central tendency and statistical dispersion provided by QR [36]. To implement probabilistic estimation, a set of quantiles should be set in ahead $\boldsymbol{\tau} = \tau_1, \tau_2, \cdots, \tau_M$, and M is the total quantiles number. The τth quantile (τ-quantile) of a random variable Y can be defined as:

$$q_Y(\tau) = F_Y^{-1}(\tau) = inf\{y: F_Y(y) \geq \tau\} \quad 0<\tau<1 \tag{13}$$

where $F_Y(y)$ is the cumulative distribution function of Y and can be expressed as:

$$F_Y(y) = P(Y \leq y) \tag{14}$$

The pinball loss function of QR is presented in (15):

$$\rho_\tau(\mu) = \begin{cases}\tau\mu & if\ \mu \geq 0\\ (\tau-1)\mu & if\ \mu < 0\end{cases} \tag{15}$$

MQ-LSTM requires training M models individually, and each model is an LSTM model. LSTM is a variation of a recurrent neural network; it can process entire sequences of data and learn long-term dependencies. The LSTM unit regulates information by relying on a structure known as a gate. The gate is constituted by a sigmoid activation function σ and a pointwise multiplication operation. The sigmoid activation function only has "0" and "1" values; a value of "0" means the gate is closed, and "1" means the gate is open, and all information can go through the gate. There are three gates in the LSTM unit, which are forgot gate $f_{t,\tau}$, input gate $i_{t,\tau}$ and output gate $o_{t,\tau}$. With the regulations of the gates, the information of the cell state $C_{t,\tau}$ is updated to retain critical information from the previous sequence.

The responsibility of the forget gate is to delete the information from the cell state C_t. As shown in (16), the forget gate $f_{t,\tau}$ takes two inputs, x_t and $h_{t-1,\tau}$, where h_{t-1} is the hidden state from the previous cell, and x_t is the input to the present stage. If the output of $f_{t,\tau}$ is closer to "1", that is, to keep, or else the information is forgotten.

$$f_{t,\tau} = \sigma\left(W_{f,\tau}\left[x_t, h_{t-1,\tau}\right] + b_{f,\tau}\right) \tag{16}$$

As for the input gate $i_{t,\tau}$, x_t and $h_{t-1,\tau}$ is passed through a sigmoid function to determine the values to be updated, see (17). Also, x_t and $h_{t-1,\tau}$ is passed into a tanh function to squish values between [-1,1] to create a new candidate cell state value $\widetilde{C_{t,\tau}}$, see (18). Finally, the new cell state $C_{t,\tau}$ is determined given $i_{t,\tau}$ and $\widetilde{C_{t,\tau}}$, shown in (17-19):

$$i_{t,\tau} = \sigma\left(W_{i,\tau}\left[x_t, h_{t-1,\tau}\right] + b_{i,\tau}\right) \tag{17}$$
$$\widetilde{C_{t,\tau}} = tanh\left(W_{c,\tau}\left[x_t, h_{t-1,\tau}\right] + b_{c,\tau}\right) \tag{18}$$
$$C_{t,\tau} = f_{t,\tau} \odot C_{t-1,\tau} + i_{t,\tau} \odot \widetilde{C_{t,\tau}} \tag{19}$$

Finally, the output of the cell and the hidden state is determined by the output gate $o_{t,\tau}$:

$$o_{t,\tau} = \sigma\left(W_{o,\tau}\left[x_t, h_{t-1,\tau}\right] + b_{o,\tau}\right) \tag{20}$$
$$h_{t,\tau} = o_{t,\tau} \odot tanh\left(C_{t,\tau}\right) \tag{21}$$

where $W_{f,\tau}$, $W_{i,\tau}$, $W_{c,\tau}$, $W_{o,\tau}$ are weight matrices; and $b_{f,\tau}$, $b_{i,\tau}$, $b_{c,\tau}$, $b_{o,\tau}$ are the bias. Typically, a fully connected layer $z_{t,q}$ is connected between the LSTM layer and output layer $f_\tau(x_t)$. The function of the fully connected layer is to convert $h_{t,q}$ into proper output size. Hence, the output of the MQ-LSTM neural network is:

$$z_{t,\tau} = \sigma\left(W_{h,\tau} \cdot h_{t,\tau} + b_{h,\tau}\right) \tag{22}$$

$$f_\tau(x_t) = W_{z,\tau} \cdot z_{t,\tau} + b_{z,\tau} \tag{23}$$

where $W_{z,\tau}$ is The MQ-LSTM model is optimized by minimising the quantile optimization function E_τ:

$$E_\tau = \frac{1}{N}\sum_{t=1}^{N}\rho_\tau\left(y_t - f_\tau(x_t)\right) \tag{24}$$

where N is the total number of data, y_t is the ith ground truth value.

3.4.3. Model Description

The techniques introduced in previous sections are combined to construct the energy disaggregation model. The model takes the MQ-LSTM as the core component, the variables introduced in Table 2 are adopted as the input variables of the model. Since multiple types of independent variables are considered, three input layers are designed. As shown in Fig. 8, The input layers take the demand load sequence $\boldsymbol{L}_F$, temperature sequence $\boldsymbol{T}$, and other variables as input, respectively. LSTM layers are then applied to the first two input layers to process the sequence data, and a one-hot encoder is adopted to transfer categorical variables (e.g., the hour of the day, the season of the year) into numerical data. Then a concatenate is used to merge three input layers together. Two fully connected layers are designed to enable the network to better extract features and learn the input data. A quantile layer evaluates multiple outputs for different quantiles.

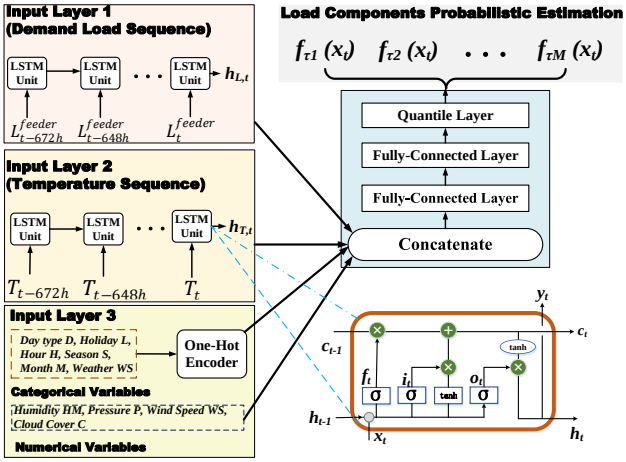


Fig. 8. The main structure of the MQ-LSTM-based energy disaggregation algorithm.

4. Evaluation Criteria

4.1. Software and Hardware Platform

To implement the proposed simulation case study, a variety of open access packages and libraries based on Python 3.7 are adopted. PyWavelets package [37] is adopted to implement the DWT-based data denoising method. Scikit-Learn package [38], LightGBM package [39] are used for implementing Q-GBRT and Q-LGB algorithms, respectively. Moreover, TensorFlow 2 [40] is utilized as the platform to construct a quantile deep neural network. As for hardware, the simulation and computation are implemented on a high computation ability computer equipped with a Core i7-7700HQ CPU, NVIDIA GTX 1060 GPU 2.80 GHz (8 cores), and 8 GB RAM.

4.2. Performance Metrics

In this paper, four categories of point prediction metrics are used to evaluate the performance of solar energy separators, and three probability density prediction metrics are adopted to assess the efficiency of the energy disaggregation model.

4.2.1. Metrics for Point Prediction

To assess the performance of the proposed energy separator, four performance metrics are adopted, which are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Rooted Mean Squared Error (RMSE), and R^2. Detailed formulas are shown as follows:

(1) Normalized Mean Absolute Error (nMAE):

$$nMAE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|}{C} \tag{25}$$

(2) Normalized Rooted Mean Squared Error (nRMSE):

$$nRMSE = \frac{1}{C}\sqrt{\frac{\left(\sum_{i=1}^{N}[y_i-\hat{y}_i|^2\right)}{N}} \tag{26}$$

(3) R-Square (R^2):

$$R^2 = 1-\frac{\sum_i (y_i-\hat{y}_i)^2}{\sum_i (y_i-\bar{y})^2} \tag{27}$$

(4) Pearson Correlation Coefficient (ρ):

$$\rho = \frac{\sum_{i=1}^{N}(y_i-\bar{y})(\hat{y}_i-\bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2\sum_{i=1}^{n}(\hat{y}_i-\bar{\hat{y}})^2}} \tag{28}$$

The value of R^2 and ρ are between -1 and 1, the closer the value to 1, the stronger correlation between the two variables.

4.2.2. Metrics for Probability Density Prediction

Traditional performance metrics, as introduced above, are usually used to evaluate points prediction results, and they cannot assess the probability density prediction results precisely. Hence, to better evaluate the performance of algorithms, both the reliability and sharpness of PIs are thoroughly investigated.

4.2.2.1. Reliability Aspect

Reliability indicates whether the quantile regression model can capture the targets into their predicted PIs efficiently. Prediction interval coverage probability (PICP) and absolute average coverage error (AACE) are introduced to assess the reliability of the model.

(1) PICP: As an essential metric adopted to assess probability density prediction, PICP indicates the probability that ground truth values are within the prediction interval (between lower and upper boundary). The values of PICP ranges from 0% to 100%, and the more significant PICP value represents more ground truth values fall into the predicted interval. The formula to calculate PICP is:

$$PICP = \frac{1}{N}\sum_{i=1}^{N}\varepsilon_i \qquad \varepsilon_i = \begin{cases}1, & y_i \in [L_i, U_i]\\ 0, & y_i \notin [L_i, U_i]\end{cases} \tag{29}$$

where N is the number of testing data, ε_i is the Boolean value, L_i is the lower boundary, and U_i is the upper boundary.

(2) Absolute Average Coverage Error (AACE): AACE indicates the deviation of PICP to PINC, the expected PICP value. The equation of AACE is:

$$AACE = |PICP - PINC| \tag{30}$$

where PINC =1 – α, and α is nominal proportions. Smaller AACE represents a more precise coverage probability provided by PIs.

4.2.2.2. Sharpness Aspect

The performance of the model cannot be thoroughly investigated with reliability metrics only since a more comprehensive PI can include more target points in it and achieve a higher PICP value. However, a wide PI has poor performance in tracking the variation and fluctuation of the target curve. Hence, the sharpness of the PIs is also extremely important for probabilistic estimation. Proposed by Winkler in 1972 [41], Winkler Score (WS) is used to assess the width of the interval with a penalty once the observation is outside the interval. WS is defined as:

$$WS = \begin{cases}\Delta_i & L_i \le y_i \le U_i\\ \Delta_i + 2(L_i - y_i)/\alpha & y_i < L_i\\ \Delta_i + 2(y_i - U_i)/\alpha & y_i > U_i\end{cases} \tag{31}$$

where Δ_t is the width of PIs at time point i and $\Delta_t = U_t - L_t$. As for WS, lower scores are associated with narrower intervals and better estimation of the PIs provided. Finally, the reliability and sharpness are combined to present a comprehensive metric named Score:

$$Score = \frac{WS}{PICP} \tag{32}$$

A smaller value of $Score$ indicates the model has a better performance in both reliability and sharpness.

5. Case Study

To evaluate the proposed energy disaggregation model, two case studies are introduced in this section. The first case study compares the proposed MQ-LSTM algorithm with other advanced quantile regression models. The second case study investigates the transferability of the proposed energy disaggregation model.

5.1. Benchmark Models

Following state-of-the-art algorithms are adopted in the case studies:

(1) Multi-Quantile Gated Recurrent Unit (MQ-GRU) [19];
(2) Multi-Quantile Convolutional Neural Network (MQ-CNN) [42];
(3) Quantile Light Gradient Boosting Machine (Q-LGB) [39];
(4) Quantile Gradient Boosting Regression Tree (Q-GBRT) [19].

5.2. Case Study I: Comparison of the Proposed Algorithms with Other Methods

The interval resolution of the data adopted in this case study is 5 min, and a denoising level 2 is applied to the original data to filter out noises that existed in the measurements. Moreover, four weeks historical feeder load demand, as well as temperature record, are adopted as input variables to enable the LSTM neural network helpful extract information from the past. As for the dataset for training, the dataset is constructed referred to Section 2.3.1, and the data is split into a training dataset (1st January 2018 to 1st August 2018), validation data set (2nd August 2018 to 15th September), and testing dataset (16th September to 31st December 2018), respectively.

5.2.1. Solar Energy Separation

In the case study, both the performance of supervised and unsupervised solar energy separation methods is evaluated. Given the net load measured at the feeder side, the solar energy separator aims to estimate the PV generation on a real-time basis. Figure 9 presents a comprehensive analysis of the two algorithms. The estimating curves evaluated by two models, as well as the ground truth PV generation, are shown in Figure 9 (c). The actual value is shown in light blue shading, while the red solid curve and orange solid curve represents the estimating results from the GBRT and UM models, respectively. From the figure, the values estimated by the GBRT method tracks the ground truth values with high accuracy, while the UM method cannot estimate the peak values generated by the PV. The evaluation metrics of the two methods are shown in the radar chart (Fig. 9 (a)) and Table 4, and the best metrics are highlighted with grey shading. It is observed that the GBRT method is superior to the UM method in all metrics. The nRMSE values of UM and GBRT are 12.41% and 4.68%, while the RMSE values of UM and GBRT methods are 1.54 MW and 0.58 MW, respectively. Moreover, the nMAE values of UM and GBRT methods are 6.44% and 2.55%. Meanwhile, the correlation metrics, R2 and ρ, provide clearer evidence that the GBRT method is far superior compared to the UM method. Figure 9 (b) utilizes a scatter plot to visualize the correlation of estimated values with the ground truth values. R2 and ρ of the GBRT method reach 96% and 95%, which means the estimating values are highly correlated with the actual values, while R2 and ρ of UM are 73% and 54% only. Although the GBRT method has superior performance, it requires pretraining the model before adopting it to real-time applications. Meanwhile, UM has lower accuracy but high flexibility.

Table 3
Performance of two solar energy separation methods.

Algorithms	nRMSE (%)	RMSE (MW)	nMAE (%)	R^2	ρ
Unsupervised Algorithm	12.41	1.54	6.44	0.73	0.54
GBRT Algorithm	4.68	0.58	2.55	0.96	0.95

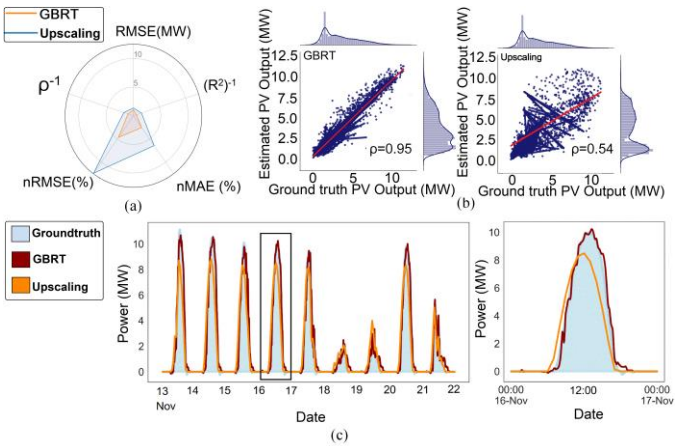


Fig. 9. (a) Radar chart of performance metrics to two PV separation algorithms; (b) Scatter plot of estimated PV power versus ground truth PV energy for unsupervised upscaling and gradient boosting methods, with the Pearson correlation. (b) Comparison of solar energy estimated by the PV separator and ground truth value.

5.2.2. Load Components Disaggregation

The estimated demand load L_t^{feeder} is then fed into the energy disaggregation model to obtain the power consumption of individual appliances: AC, furnace, and EV. To thoroughly investigate the performance of the proposed model it is compared with MQ-GRU, MQ-CNN, Q-LGB, and Q-GBRT algorithms in four metrics: PICP, WS, Score, and training time.

The training time of each algorithm is shown in Table 4; both the proposed MQ-LSTM, MQ-GRU, and MQ-CNN are trained on a GPU-based Tensorflow platform, while Q-LGB, Q-GBQT models are trained on a computer without GPU used. GPUs are suitable for training deep neural network models because they can process multiple computations at the same time. The computer installs many cores, which allows it to better and faster compute multiple parallel processes. Although a high computation ability CPU (Core i7-7700HQ CPU) is adopted, it still takes nearly 275 min (4.6 h) to finish the evaluation of the Q-GBRT model. The training time of the Q-GBRT model is much longer than other algorithms, which demonstrates that the Q-GBRT model is less practical and flexible in application. In contrast, the training time of the Q-LGB model is less than the Q-GBRT model, while it only takes 37.22 min to finish the training process on average. When it turns to GPU-based training models, the training time of the proposed MQ-LSTM is 93% less than the time of the Q-GBRT model. And the training time of MQ-GRU and MQ-CNN are all shorter than 25 min. From the result, apart from the Q-GBRT model, the training time of all other models are considerable in industrial application.

Table 4
Comparison of training time (min).

Appliance\Algorithm	MQ-LSTM	MQ-GRU	MQ-CNN	Q-LGB	Q-GBRT
AC	17.67	16.65	20.12	37.69	277.12
Furnace	16.78	17.23	22.32	36.26	273.45
EV	17.23	18.67	21.87	37.72	274.90
Average	17.22	17.52	21.43	37.22	275.16

The reliability of the probabilistic models is assessed by metrics PINC and AACE, respectively. The comparison results are shown in Fig. 10 as well as Table 5. In Fig. 10, the PICPs of five algorithms to three appliance loads are the plot. In an ideal situation, the value of PICP is equal to PINC, as shown in the solid red curve (1:1 line). From the figure, among all curves, the curve of the proposed MQ-LSTM is closest to the ideal curve, meaning that the MQ-LSTM model is better than other benchmarks for energy disaggregation tasks, and it can produce reliable PIs. The maximum AACEs of the PIs evaluated by the MQ-LSTM model, which is the metric to show the deviation between PICP and PINC, are 2.79%, 8.10%, and 10.12% for AC, Furnace, EV load, respectively. As for AC load, the other four algorithms also show merit reliability as the maximum AACEs of MQ-GRU, MQ-CNN, Q-LGB, Q-GBRT are 5.42%, 8.30%, 18.63%, and 12.63%, respectively. However, the reliability of PIs computed by MQ-CNN and Q-LGB for Furnace load is considerably low as these algorithms overestimate PICP, the disaggregation result is unreliable for the power system industry. Furthermore, as for EV load, PIs evaluated by both five algorithms are well-calibrated as all five curves overlap with the ideal curve precisely. To summarize, the reliability of PIs provided by MQ-LSTM is higher than any other algorithm for the different load tasks.

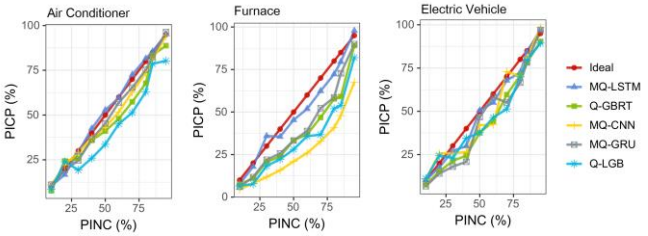


Figure 10. PI reliability diagrams: PICP of five algorithms as a function of PI nominal coverage.

The sharpness is assessed by metric WS, as indicated in the boxplots in Fig. 11 and Table 6. From the figure, MQ-CNN is the bluntest model. The average WS of its PIs, is 2.70, 3.70, and 5.10 for AC, Furnace, and EV loads. Meanwhile, the Q-LGB and Q-GBRT are the sharpest models among all algorithms, and the sharpness of MQ-LSTM and MQ-GRU is among the models mentioned above. However, it should be noticed that the sharpness should be analyzed along with reliable performance, which can be visualized via metric Score.

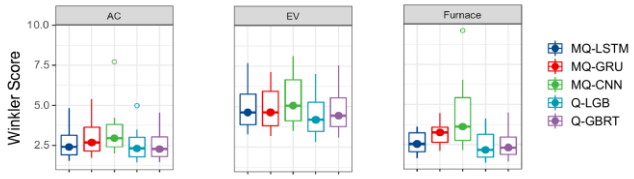


Fig. 11. Boxplot of the Winkler Score.

From Table 5, by comparing the Score of each model for different PINC, the more miniature value Score is, the better probabilistic estimation the model can provide. It is noticed that the Score value of the proposed MQ-LSTM is the smallest among most cases and PICPs, except for EV cases. The MQ-LSTM model achieves the best performance in disaggregating AC load components. The Score of MQ-LSTM reduced 37.82% and 18.37% when PINC is 95%, compared to the Scores of MQ-CNN and Q-LGB models, respectively. The superior of MQ-LSTM is even more apparent when disaggregating Furnace load components. From the table, we can find that the deviation between other models and the proposed MQ-LSTM increases with the decrease of PINC. When PINC reaches 20%, the values of Score for other algorithms are almost double the value of MQ-LSTM. This result demonstrates that the proposed method has distinct estimation performance in different quantiles when disaggregating Furnace load from the overall demand load. However, when it comes to the EV case, the performance of Q-LGB turns better than MQ-LSTM with the reduction of PINC.

Fig. 12 presents the PIs evaluated by the proposed MQ-LSTM model with different confidence levels and the actual appliance load curve (AC, Furnace, and EV loads). As for AC load, it is observed almost all actual curves are between the upper and lower bounds. The solid yellow line, which represents the median estimation, tracks the variation of AC demand load precisely. Meanwhile, the width of PIs, especially 95% PI, are the smallest among all appliance loads. The estimation of the Furnace load is quite similar to the AC components, as the Furnace in Austin, Texas area plays dual roles: heating and circulating air during the usage of AC. Hence, the demand load curve of the furnace is correlated to the curve of AC demand. Most of the ground truth data are within estimated PIs; the median curve of PIs overlaps with the actual furnace load curves. However, the width of PIs is wider than the PIs of estimated AC loads. This demonstrates the reliability of the MQ-LSTM model for Furnace load has equal performance as it has for AC loads, but the sharpness of the model on Furnace is not as good as it is for AC loads. As for EV loads, the component is harder to be separated from the overall demand load for two reasons: Firstly, the portion of EV load is relatively small compared to the portion of AC load or Furnace load; secondly, the average operation duration of EVs is also shorter than other cases. From Fig.12, the PIs estimated by MQ-LSTM are compared with the actual EV load curve; from the figure, it is found that although the model cannot estimate the exact load curve, most all operation duration durations are estimated precisely.

The probability density curves obtained by the proposed MQ-LSTM are presented in Fig. 13. The actual values of different load components during the time are investigated, the grey shading curves are the probability density function (PDF) with 95% confidence level, and the red vertical line is the actual value of a specific hour, while the black dash vertical line shows the maximum probability point of the probability density curve. First of all, all selected actual values are in the middle of the PDFs. As for AC and Furnace load components, the maximum probability points almost overlap with the actual values, except for 1 pm and 2 pm, and the model has the highest accuracy when estimating the maximum and minimum values of the target loads. However, the PDF graph for EV load is not in good shape, as the maximum probability point of the PDF is not as high as expected, and the shape of the PDF is not strictly following the Gaussian distribution.

Fig. 14 shows the estimation results of all quantile regression models for three load components from 23rd September 2018 to 28th September 2018. The figure provides a more detailed view of the performance of the individual energy disaggregation models. The colour shadings are the estimated PIs between 90-quantile and 10-quantile, and the solid red curve is the ground truth curve of the load component, while other colour solid curves are the median values of estimated PIs. From the figure, it is found that the ground truth curve is within the PIs of all models, and the PIs of MQ-LSTM and MQ-GRU can track the fluctuations of the actual values with high accuracy. Among all PIs, the PIs provided by the MQ-CNN has the most significant interval, while the widths of Q-LGB and Q-GBRT are relatively small, but the errors between PIs of these two models and the actual values are also considerably higher. To summarise, the proposed MQ-LSTM energy disaggregation algorithm has superior performance in both reliability and sharpness for all load components investigated in this paper.

5.3. Case Study IV: Transferability of Proposed Scheme

One major issue of the proposed energy disaggregation method is that the model requires historical feeder-level demand load data to train the machine learning/ neural network models before adopting them for industrial application. However, such kind of data is not always available for most areas, and the issue limits a broader application. Hence, it is essential to investigate the transferability of the proposed energy disaggregation model. The transfer learning process is shown in Fig. 15, as a transductive transfer learning problem (source data labels are available, but target data labels are unavailable [43]), the deep neural network model is pre-trained, with data in the source domain, the difference between the distributions of the source domain and target domain can be minimized by modifying the source domain (adjusting the portions of different load components, adjusting seasonal and trend, etc.). The source data is also normalized and resampled to fit the target data. Then the model is fine-tuned with the training set of the target domain, and the model is tested with the testing set of the target domain.

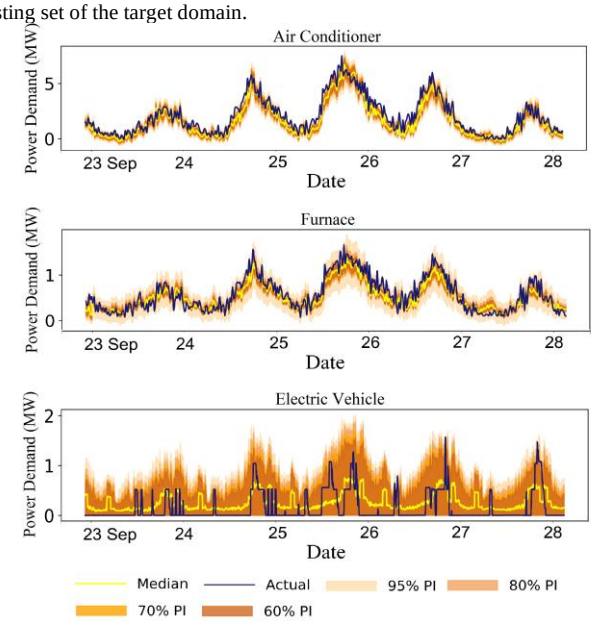


Fig. 12. PIs of the MQ-LSTM energy disaggregation model with various confidence levels for AC, Furnace, EV loads.

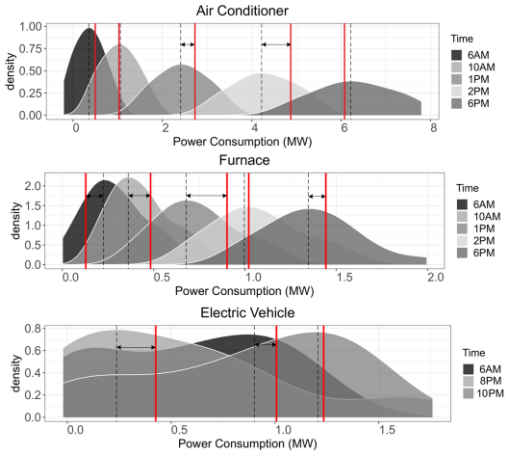


Fig. 13. Probability density curves obtained by MQ-LSTM energy disaggregation model for AC, Furnace, EV loads.

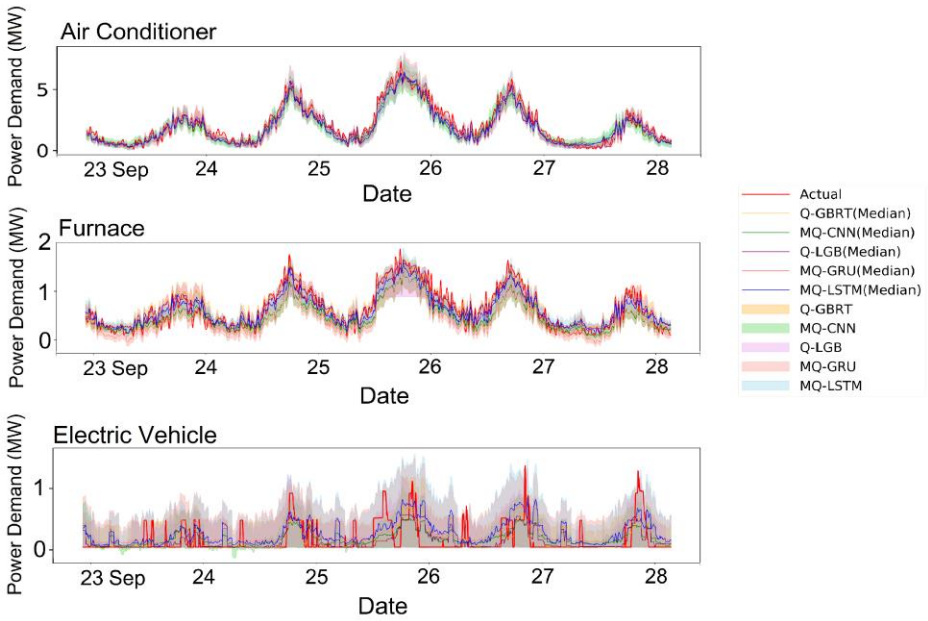


Fig. 14. Comparison of the results of energy disintegration implemented by various algorithms. The solid curve is the median estimate, the color shading is the range between the estimate curve of quantiles 10 and 90, while the red solid curve is the ground truth load.

Table 5
Probabilistic energy disaggregation evaluation metrics.

PINC (%)	Appliance	Metrics	MQ-LSTM	MQ-GRU	MQ-CNN	Q-LGB	Q-GBRT
95%	AC	PICP (%)	95.23	96.35	94.44	80.282	88.77
		WS	5.81	5.39	7.71	4.997	4.56
		AACE (%)	0.20	1.35	0.56	0.147	6.23
		Score	5.08	5.60	8.17	6.223	5.14
	Furnace	PICP (%)	97.80	89.93	67.60	82.00	89.20
		WS	3.61	3.51	9.61	4.12	3.57
		AACE (%)	2.80	5.07	27.40	13.00	5.73
		Score	3.69	3.91	14.22	5.03	4.00
	EV	PICP (%)	97.00	97.33	98.40	89.40	90.40
		WS	7.15	7.69	8.12	7.02	7.56
		AACE (%)	2.00	2.33	3.40	5.60	4.60
		Score	7.37	7.90	8.26	7.85	8.37
85%	AC	PICP (%)	85.46	82.47	80.71	78.66	83.33
		WS	3.57	4.05	4.22	3.48	3.54
		AACE (%)	0.46	2.53	4.29	6.34	1.67
		Score	4.17	4.91	5.23	4.43	4.25
	Furnace	PICP (%)	79.56	72.76	47.92	53.89	59.11
		WS	3.48	4.46	6.53	3.96	4.49
		AACE (%)	5.44	12.24	37.08	31.11	25.89
		Score	4.37	6.13	13.63	7.35	7.6
	EV	PICP (%)	84.29	78.24	80.94	79.44	78.21
		WS	6.29	6.55	7.41	6.32	6.91
		AACE (%)	0.71	6.76	4.06	5.56	6.79
		Score	7.46	8.37	9.15	7.96	8.83
70%	AC	PICP (%)	72.56	65.38	62.88	51.37	57.37
		WS	2.81	3.24	3.48	2.70	2.67
		AACE (%)	2.56	4.62	7.12	18.63	12.63
		Score	3.88	4.96	5.33	5.25	4.65
	Furnace	PICP (%)	62.38	52.29	32.80	36.74	46.82
		WS	3.04	3.59	4.57	2.77	2.80
		AACE (%)	7.62	17.71	37.20	33.26	23.18
		Score	4.87	6.86	13.93	7.54	5.97
	EV	PICP (%)	67.82	55.06	72.87	51.36	59.60
		WS	5.36	5.45	5.86	4.99	5.12
		AACE (%)	2.18	14.94	2.87	18.64	10.40
		Score	7.90	9.91	8.04	9.71	8.60
40%	AC	PICP (%)	42.43	36.58	36.97	25.94	35.86
		WS	2.05	2.29	2.56	1.94	1.90
		AACE (%)	2.49	3.42	3.03	14.06	4.14
		Score	4.83	6.27	6.92	7.46	5.30
	Furnace	PICP (%)	35.53	25.66	15.76	22.27	23.85
		WS	2.21	2.83	2.98	1.85	1.08
		AACE (%)	4.47	14.34	24.24	17.73	16.15
		Score	6.21	11.03	18.90	8.31	7.88
	EV	PICP (%)	29.88	20.83	26.40	34.46	24.46
		WS	4.09	4.05	4.33	3.77	4.02
		AACE (%)	10.12	19.17	13.60	5.54	15.54
		Score	13.69	19.42	16.40	10.93	16.45

Table 6
Performance of transfer learning.

PINC (%)	95%			85%			70%			40%		
Appliance/ Metrics	AC	Furnace	EV	AC	Furnace	EV	AC	Furnace	EV	AC	Furnace	EV
PICP (%)	91.55	89.35	97.92	82.18	80.61	88.54	60.65	62.85	59.99	43.17	26.39	32.70
WS	10.97	16.00	8.61	6.97	12.83	6.85	5.45	8.17	5.65	3.63	5.82	3.88
AACE (%)	3.45	5.65	12.94	2.82	4.39	3.54	9.35	7.15	10.01	3.17	13.61	7.30
Score	11.99	17.90	8.79	8.48	15.91	7.73	8.99	13.00	9.41	8.40	22.04	11.87

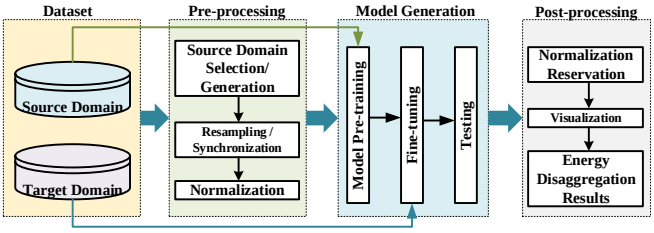


Fig. 15. Block diagram of the transfer learning process.

In this case study, the semi-synthetic data set constructed from Austin, Texas data of Dataport between 2018 and 2019 (sampling frequency is 1 Min) is used as the public training dataset, and the New York data from the Dataport during 2019 (sampling frequency is 15 Min) is used as the local data to be disaggregated (testing dataset). Since the geographical location, sampling frequency, climate condition, portion of load components are all different, we can view the two datasets are entirely different, and the result would be reliable. An investigation about the installation rate of AC, Furnace, and EV in the local area is implemented to obtain the portion of different load components roughly. Referring to the investigation result, the training dataset is adjusted to make the training dataset more like the testing data. Like Case Study I, the solar energy component is separated at the first stage and followed by the demand load components to be disaggregated from the overall demand load.

Table 6 presents the performance metrics of the transfer learning model. Compared to the results concluded in Table 5, the results of the transfer learning model are not as good as the typical energy disaggregation model, as the WS, AACE, and Score are more significant than the results in Case Study I. However, referring to Fig. 16, the probabilistic models can still provide precise estimation results as most of the actual values are within the estimated PIs with narrow widths. The results show that the proposed MQ-LSTM-based energy disaggregation model has good transferability; by selecting a suitable source domain, the model can be well-trained and adopted to other feeder models.

6. Conclusions

Understanding the load components under the grid supply point can aid the power system utility in implementing demand-side management and shaving the peak load. In this paper, a probabilistic feeder-level energy disaggregation model based on MQ-LSTM deep neural network is proposed. The purpose of the proposed model is to disaggregate the net load into four components: renewable energy generation (rooftop PVs in this case), TCLs (AC and Furnace loads), Non-TCLs (EV load chosen as a case study), and other loads. The solar energy generated by the rooftop PVs is separated at first by adopting either unsupervised upscaling or the supervised GBRT method. Although the supervised GBRT method can provide better estimation results than the unsupervised method, the unsupervised method shows more flexibility and does not require a training dataset to train the model.

Then other load components are disaggregated via the proposed MQ-LSTM algorithm; the proposed model considers a variety of relevant variables as input features, including current and historical demand load and temperature, meteorological measurement, calendar information. Four state-of-the-art probabilistic machine learning/ deep learning models: MQ-GRU, MQ-CNN, Q-LGB, Q-GBRT algorithms are adopted as the benchmarks. Two case studies thoroughly investigate the performance of the proposed model and benchmark models in four aspects: reliability (PICP, AACE), sharpness (WS), training time and overall performance (Score). The case studies confirm that the proposed model has superior performance in disaggregating different load components. As for AC and Furnace loads, the model not only can provide reliable and sharp PIs but also can estimate the precise load curves. When it turns to EV load, although the detailed load curve is not available, the probabilistic model can estimate the intervals with high accuracy. Moreover, the transferability of the proposed model is studied as well since it is unrealistic to obtain a large amount of labelled data for training. The model is pre-trained with a public dataset (source domain) then tested with local data (target domain). The results show that the proposed model is transferable to a different area with a different interval resolution as well as different portions of load components.

Future work includes two aspects: (1) Develop an unsupervised learning model, such that a training dataset is not required; (2) Increase the accuracy of the transfer learning model by adopting a larger source domain.

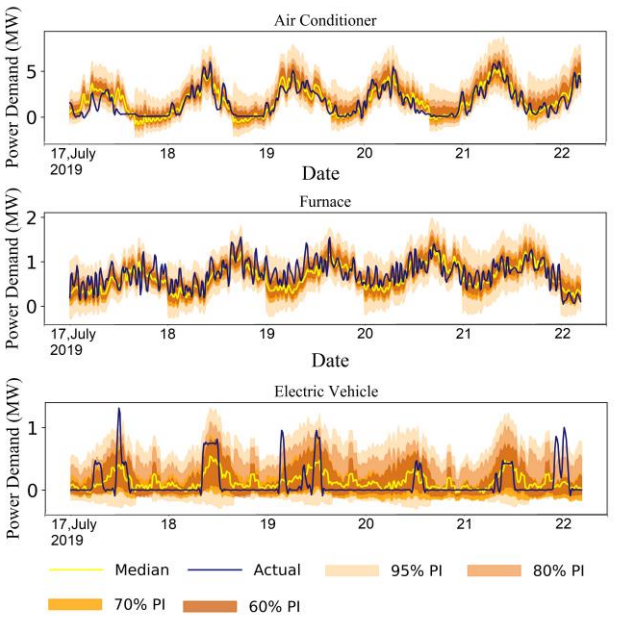


Fig. 16. PIs of the transfer learning model with various confidence levels for AC, Furnace, EV loads.

References

[1] F. Saghezchi, F. Saghezchi, A. Nascimento, and J. Rodriguez, "Game theory and pricing strategies for demand-side management in the smart grid," presented at the 2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP), 23-25 July 2014, 2014.

[2] P. Kohlhepp, H. Harb, H. Wolisz, S. Waczowicz, D. Müller, and V. Hagenmeyer, "Large-scale grid integration of residential thermal energy storages as demand-side flexibility resource: A review of international field studies," *Renewable and Sustainable Energy Reviews*, vol. 101, pp. 527-547, 2019/03/01/ 2019, doi: https://doi.org/10.1016/j.rser.2018.09.045.

[3] Y. M. Lee, R. Horesh, and L. Liberti, "Optimal HVAC Control as Demand Response with On-site Energy Storage and Generation System," *Energy Procedia*, vol. 78, pp. 2106-2111, 2015/11/01/ 2015, doi: https://doi.org/10.1016/j.egypro.2015.11.253.

[4] W. Li, M. Yi, M. Wang, Y. Wang, D. Shi, and Z. Wang, "Real-Time Energy Disaggregation at Substations With Behind-the-Meter Solar Generation," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2023-2034, 2021, doi: 10.1109/TPWRS.2020.3035639.

[5] G. S. Ledva, L. Balzano, and J. L. Mathieu, "Real-time energy disaggregation of a distribution feeder's demand using online learning," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 4730-4740, 2018.

[6] G. S. Ledva and J. L. Mathieu, "Separating feeder demand into components using substation, feeder, and smart meter measurements," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3280-3290, 2020.

[7] J. Wang *et al.*, "A Data-Driven Pivot-Point-Based Time-Series Feeder Load Disaggregation Method," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5396-5406, 2020.

[8] S. Wang, R. Li, A. Evans, and F. Li, "Regional nonintrusive load monitoring for low voltage substations and distributed energy resources," *Applied Energy*, vol. 260, p. 114225, 2020.

[9] 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," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6102-6114, 2019, doi: 10.1109/TSG.2019.2896493.

[10] Y. Xu and J. Milanovic, "Artificial-Intelligence-Based Methodology for Load Disaggregation at Bulk Supply Point," *IEEE Transactions on Power Systems*, vol. 30, no. 2, pp. 795-803, 2014, doi: 10.1109/TPWRS.2014.2337872.

[11] M. W. Asres, A. A. Girmay, C. Camarda, and G. T. Tesfamariam, "Non-intrusive load composition estimation from aggregate ZIP load models using machine learning," *International Journal of Electrical Power & Energy Systems*, vol. 105, pp. 191-200, 2019.

[12] H. Shaker, H. Zareipour, and D. Wood, "Estimating power generation of invisible solar sites using publicly available data," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2456-2465, 2016.

[13] H. Shaker, H. Zareipour, and D. Wood, "A data-driven approach for estimating the power generation of invisible solar sites," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2466-2476, 2015.

[14] P. Street, "Dataport: the world's largest energy data resource," *Pecan Street Inc*, 2015.

[15] Y. Wang, N. Zhang, Q. Chen, D. S. Kirschen, P. Li, and Q. Xia, "Data-Driven Probabilistic Net Load Forecasting With High Penetration of Behind-the-Meter PV," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3255-3264, 2018, doi: 10.1109/TPWRS.2017.2762599.

[16] X. Y. Zhang, S. Kuenzel, and C. Watkins, "Feeder-Level Deep Learning-based Photovoltaic Penetration Estimation Scheme," presented at the 2020 12th IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), 20-23 Sept. 2020, 2020.

[17] F. Bu, K. Dehghanpour, Y. Yuan, Z. Wang, and Y. Guo, "Disaggregating Customer-level Behind-the-Meter PV Generation Using Smart Meter Data and Solar Exemplars," *IEEE Transactions on Power Systems*, pp. 1-1, 2021, doi: 10.1109/TPWRS.2021.3074614.

[18] F. He, J. Zhou, L. Mo, K. Feng, G. Liu, and Z. He, "Day-ahead short-term load probability density forecasting method with a decomposition-based quantile regression forest," *Applied Energy*, vol. 262, p. 114396, 2020.

[19] S. Zhang, Y. Wang, Y. Zhang, D. Wang, and N. Zhang, "Load probability density forecasting by transforming and combining quantile forecasts," *Applied Energy*, vol. 277, p. 115600, 2020/11/01/ 2020, doi: https://doi.org/10.1016/j.apenergy.2020.115600.

[20] W. Zhang, H. Quan, and D. Srinivasan, "Parallel and reliable probabilistic load forecasting via quantile regression forest and quantile determination," *Energy*, vol. 160, pp. 810-819, 2018.

[21] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided LSTM," *Applied Energy*, vol. 235, pp. 10-20, 2019.

[22] J. Xie and T. Hong, "Variable selection methods for probabilistic load forecasting: Empirical evidence from seven states of the united states," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6039-6046, 2017.

[23] J. F. Toubeau *et al.*, "Capturing Spatio-Temporal Dependencies in the Probabilistic Forecasting of Distribution Locational Marginal Prices," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2663-2674, 2021, doi: 10.1109/TSG.2020.3047863.

[24] F. Fan, K. Bell, and D. Infield, "Probabilistic Real-Time Thermal Rating Forecasting for Overhead Lines by Conditionally Heteroscedastic Auto-Regressive Models," *IEEE Transactions on Power Delivery*, vol. 32, no. 4, pp. 1881-1890, 2017, doi: 10.1109/TPWRD.2016.2577140.

[25] F. Fan, K. Bell, and D. Infield, "Transient-state real-time thermal rating forecasting for overhead lines by an enhanced analytical method," *Electric Power Systems Research*, vol. 167, pp. 213-221, 2019/02/01/ 2019, doi: https://doi.org/10.1016/j.epsr.2018.11.003.

[26] Y. He and H. Li, "Probability density forecasting of wind power using quantile regression neural network

and kernel density estimation," *Energy conversion and management*, vol. 164, pp. 374-384, 2018.
[27] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.
[28] K. P. Schneider, Y. Chen, D. P. Chassin, R. G. Pratt, D. W. Engel, and S. E. Thompson, "Modern grid initiative distribution taxonomy final report," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2008.
[29] t. O. W. o. T. C. o. Austin. "Official City Holidays." https://www.austintexas.gov/department/official-city-holidays (accessed.
[30] I. S. Data, "National Climatic Data Center (NCDC)," *Asheville, NC*, 2001.
[31] T. Feddern-Bekcan, "Google calendar," *Journal of the Medical Library Association: JMLA*, vol. 96, no. 4, p. 394, 2008.
[32] C. Cai and P. d. B. Harrington, "Different discrete wavelet transforms applied to denoising analytical data," *Journal of chemical information and computer sciences*, vol. 38, no. 6, pp. 1161-1170, 1998.
[33] Q. Pan, L. Zhang, G. Dai, and H. Zhang, "Two denoising methods by wavelet transform," *IEEE transactions on signal processing*, vol. 47, no. 12, pp. 3401-3406, 1999.
[34] N. Blair *et al.*, "System advisor model, sam 2014.1. 14: General description," National Renewable Energy Lab.(NREL), Golden, CO (United States), 2014.
[35] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," (in English), *Frontiers in neurorobotics*, Methods vol. 7, no. 21, 2013-December-04 2013, doi: 10.3389/fnbot.2013.00021.
[36] R. Koenker and G. Bassett Jr, "Regression quantiles," *Econometrica: journal of the Econometric Society*, pp. 33-50, 1978.
[37] G. R. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary, "PyWavelets: A Python package for wavelet analysis," *Journal of Open Source Software*, vol. 4, no. 36, p. 1237, 2019.
[38] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825-2830, 2011.
[39] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146-3154, 2017.
[40] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265-283.
[41] R. L. Winkler, "A decision-theoretic approach to interval estimation," *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 187-191, 1972.
[42] G. Petneházi, "QCNN: Quantile Convolutional Neural Network," *arXiv preprint arXiv:1908.07978*, 2019.
[43] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43-76, 2020.