Energy 186 (2019) 115873

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Multi-agent microgrid energy management based on deep learning forecaster

Mousa Afrasiabi [a], Mohammad Mohammadi [a, *], Mohammad Rastegar [a], Amin Kargarian [b]

[a] The School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
[b] The Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, 70803, USA

ARTICLE INFO

Article history:
Received 6 January 2019
Received in revised form
27 July 2019
Accepted 2 August 2019
Available online 7 August 2019

Keywords:
Microgrid energy management system
Short-term forecasting
Deep learning
Convolutional neural networks
Gated recurrent unit
Alternating direction method of multipliers

ABSTRACT

This paper presents a multi-agent day-ahead microgrid energy management framework. The objective is to minimize energy loss and operation cost of agents, including conventional distributed generators, wind turbines, photovoltaics, demands, battery storage systems, and microgrids aggregator agent. To forecast market prices, wind generation, solar generation, and load demand, a deep learning-based approach is designed based on a combination of convolutional neural networks and gated recurrent unit. Each agent utilizes the designed learning approach and its own historical data to forecast its required parameters/data for scheduling purposes. To preserve the information privacy of agents, the alternating direction method of multipliers (ADMM) is utilized to find the optimal operating point of microgrid distributedly. To enhance the convergence performance of the distributed algorithm, an accelerated ADMM is presented based on the concept of over-relaxation. In the proposed framework, the agents do not need to share with other parties either their historical data for forecasting purposes or commercially sensitive information for scheduling purposes. The proposed framework is tested on a realistic test system. The forecast values obtained by the proposed forecasting method are compared with several other methods and the accelerated distributed algorithm is compared with the standard ADMM and analytical target cascading.

1. Introduction

Microgrid energy management system (MEMS) has been proposed to manage the demand and suppliers economically [1]. MEMS strategies can be categorized into centralized and distributed approaches from the operational point of view [2]. Centralized approaches need a central agent that collects information from other agents. The central agent processes large amounts of data to determine setpoints of demand and supplier agents [3]. Distributed energy management, aka multi-agent energy management, enhances computational efficiency and preserves the information privacy of autonomous agents. Distributed MEMS allows each autonomous agent to optimize its individual cost function in a parallel or sequential manner. A distributed MEMS is designed in Ref. [4] based on the predictor-corrector proximal multiplier. In Ref. [5], a MEMS strategy is proposed based on dual decomposition to maximize DG owners' and end-users' profit. In Ref. [6], a microgrid economic dispatch problem is solved based on the alternating direction method of multipliers (ADMM). In Ref. [7], a form of asynchronous ADMM is presented to minimize the cost of energy. The standard ADMM is utilized in Ref. [8] for the cooperative management of several microgrids.

A MEMS strategy must guarantee reliable grid operation considering uncertainties of renewable generation, demand, and market prices. Several papers design deterministic MEMS strategies. For instance, in Ref. [9], a deterministic bi-level distributed economic dispatch is designed based on a diffusion algorithm. However, the majority of existing papers have considered the uncertainty of parameters. For instance, in Ref. [10], a model predictive control-based MEMS is presented to consider uncertainties in an island microgrid. A multi-objective MEMS framework based on stochastic programming is presented in Ref. [11]. A scenario-based day-ahead MEMS is presented in Ref. [12] for multi-carrier energy systems.

The recent uncertainty-based MEMS can be classified into four

* Corresponding author.
E-mail addresses: musa.afra@shirazu.ac.ir (M. Afrasiabi), m.mohammadi@shirazu.ac.ir (M. Mohammadi), mohammadrastegar@shirazu.ac.ir (M. Rastegar), kargarian@lsu.edu (A. Kargarian).

https://doi.org/10.1016/j.energy.2019.115873
0360-5442/

Nomenclature

Sets and Indices

Λ	Set of lines
H	Hidden state index
n/m	Index for regulating/curtailable load
r	Reset gate index
t	Index for time intervals
u	Update gate index
wi/he	Width/height index

Variables, Parameters, and Functions

σ	Flexibility coefficient of the curtailable load
$b(.)$	Bias matrix of pooling layer
$B(m,L)$	Bias matrix of m^{th} map at L^{th} layer
C^n/C^m	Cost of regulating/curtailable load
$E^{BESS}(t)$	Stored energy in the BESS at time interval t
E^{BESS}_{res}	The energy level that must remain in BESS at the end of the scheduling horizon
E_{sch}	Total energy schedule energy consumption
$f(.)$	Activation function
$P(.)$	Pooling function
$P_{ij}(t)/I_{ij}(t)/Q_{ij}(t)$	Active power flow/Square magnitude of the current/Reactive power flow in the line between buses i and j at time interval t
$P_j(t)/Q_j(t)$	Active/Reactive power consumption on bus j at time interval t
$P_{l,cur}(t)$	Power of curtailable load
R	Recurrent parameter
S	Step size of filter
$S_i^{CDG}(t)/S_i^{PV}(t)/S_i^{WT}(t)$	Complex power generated by CDGs/PVs/WTs on bus i at time t
$S_i^l(t)/S_i^{BESS}(t)$	Consumed complex power of load l/BESSs connected to bus i at time interval t
$u(m,L,t)/r(m,L,t)$	Update/Reset gate of GRU layers for the m^{th} output map in the L^{th} layer at time interval t
W	Weight matrices corresponding to input
X	Input dataset.
z_{ij}/r_{ij}	Impedance/Resistance/Reactance of the line ij
Δt	Time interval duration
C_n^l	Cost function of regulating load l
$P_m^{sch}(t)$	Power of nonresponsive load m at time interval t
$P_n^l(t)$	Power consumption of regulating load l at time interval t
$P_n^{sch}(t)$	Scheduled power of the regulating load l at time interval t
R^u/R^{down}	CDG's ramp up and down capabilities
$x(n,L-1)$	The n^{th} input map at L^{th} layer
α^k/β^k	Lagrange multipliers associated with the active/reactive power at bus i
η^{BESS}	BESS efficiency
ρ	Step sizes in each agent cost function
ζ^{CDG}/ζ^l	Trade-off parameters between cost and active power loss minimization
P^{BESS}	Charging/Discharging rate of BESS at time interval t
$P^{exc}(t)$	Traded power between the aggregator and the upstream grid at time interval t
$S_i(t)$	Net consumed complex power on bus i at time interval t
V_j	Square voltage magnitude of bus j
$y_{re}(t)/y_f(t)$	Real/Forecasted values

different groups including real-time, stochastic, robust, and forecasted-based schemes. Real-time energy management strategies handle uncertainties indirectly by reducing the operational time step to track the small variations of the uncertain variables [13]. Real-time MEMSs need updating short-term forecasting at every small time step and encounters many communication challenges, which is severally increased with increasing the number of microgrid agents. In stochastic MEMSs, uncertain parameters are modeled through capturing full (estimated probability density function (PDF)) or some statistical information (e.g., statistical moments). Scenario-based methods are other solutions for considering uncertainties in MEMS. In Ref. [14], a large number of scenarios is produced to model uncertainties in electrical load and electricity market. A decentralized energy management framework is presented in Ref. [15], and the uncertainty of renewable generation is modeled using scenarios generated by the Monte-Carlo simulation. Robust optimization problems can be an alternative to cover uncertainties based on the worst-case scenario [16]. In Ref. [17], robust optimization is utilized for a microgrid energy management considering uncertainties of renewable generation. Short-term forecasting engines are used in MEMS to minimize errors between the forecast and real values. Unlike other schemes, forecasting schemes and energy management procedure are independent of each other. Forecasting methods are used in the first stage, and the energy management is effectuated in the second stage based on the forecast values. This lowers the computational complexity of MEMS. However, the accuracy of forecasts has direct impacts on MEMS results. Thus, precise methods are needed to have reliable results from energy management.

The random behavior of renewable generation, demand, and market prices can play an important role in practical microgrid energy management. Short-term forecasting engines are used in MEMS to minimize errors between the forecast and real values. The farther the true realization of a random parameter is from its forecast value, the less efficient the MEMS will be. Forecasting is more crucial for multi-agent distributed energy management, where agents might not be aware of all sources of uncertainties. A central agent might gather information of all random parameters, forecast them, and send forecast values to autonomous agents. However, this procedure is in contradiction with information privacy in multi-agent systems. Considering the privacy concern of autonomous agents, accurate agent-owned forecast tools are required to support multi-agent MEMS.

The short-term forecasting of uncertain variables is a complex procedure that depends on weather and environmental conditions, customer consumption habits, social cultures, and economic conditions [18]. Five typical techniques, i.e., persistence, physical, statistical, artificial intelligence, and hybrid techniques, have been proposed in the literature to forecast uncertain parameters [19]. The K-means clustering has been utilized in Ref. [20] to forecast electrical/heat demand and photovoltaic generations. In Ref. [21], neural network ensemble (NNE) is implemented to predict renewable generation and load demand for MEMS. In Ref. [22], an artificial neural network (ANN) is utilized to forecast load consumption for microgrid management. The accuracy of these short-term forecasting techniques might become degraded dramatically because of high computational complexity, shallow architecture, hand-engineered features, and spectral analysis [23]. For instance, the residential load forecast depends on customer habits, events, and environmental factors that cause a high degree of uncertainty.

As a solution, deep neural networks (DNNs) have been recently presented. Deep neural networks (DNNs) are promising architectures to provide comprehensive and detailed information from raw data. DNNs can be used to tackle the abovementioned challenges for short-term forecasting [24]. In recent years, several DNN-based time series forecasting techniques are presented in the literature. In general, they can be categorized into deep auto-encoder (DAE), deep belief neural networks (DBNN), convolutional neural networks (CNN), and recurrent neural networks (RNN). DAE and DBNN suffer from inability in understanding long dependencies across time series samples, which belongs to look-ahead times [25]. CNN is a DNN-based method with the capability of extracting the inherent features of time series with a few memory cells and parameters. However, CNN might lose some of the crucial information due to difficulty in selecting the window size of the filter. RNN-based methods, such as long short-term memory (LSTM) and gated recurrent unit (GRU), show efficient performance for accurate modeling of complex nonlinear time-dependent parameters, such as price, disaggregated load, solar irradiation, and wind speed. GRU networks often extract features that are not discovered by LSTM networks, and it is less complex than LSTM. However, the inherent parameters of the RNN-based approach extensively grow with the increase in layer numbers. It makes the model captures the noise but might lead to overfitting [26]. To overcome this problem, CNN can be combined with GRU for feature extraction. This combined architecture captures long-term relations in the input more efficiently.

Combining learning methods to form new algorithms has been considered as a potential approach to enhance the performance of learning processes. With such a combination, drawbacks of a learner might become resolved by another learner. However, combining learners is not a straightforward process as it depends on learners and the considered application of this combined learning algorithm. A random combination of learners might even worsen the over learning and forecast performance. Since CNN and GRU are, to some extent, the complement of each other, it is reasonable to combine them and create a new learning structure that is more capable of either of these deep networks.

In this paper, a day-ahead multi-agent microgrid energy management framework is proposed based on deep neural networks and alternating direction method of multipliers. The microgrid is operated in a multi-agent structure, consisting of conventional distributed generator (CDG) agents, wind turbines (WT) agents, photovoltaic (PV) agents, demand agents, battery energy storage system (BESS) agents, and microgrid aggregator agent. First, a composite CNN-GRU method is designed for each agent to forecast wind generation, solar generation, demand, and market prices. The CNN-GRU architecture leverages the advantages of both CNN (efficient modeling of complex nonlinear temporal parameters) and GRU (extracting the inherent features of time series). The proposed two-block forecasting method is a generic time series forecasting that utilizes raw data. The proposed method is capable of extracting features without handcrafted feature extraction, which leads to better computational efficiency and applicability to real-time microgrid energy management applications. The forecast data is used to formulate a local optimization problem for each agent. While CDG, demand, and BESS agents aim at maximizing their benefit, the microgrid aggregator agent formulates a convex OPF problem to minimize energy loss with respect to the network constraints and energy exchange with the upstream grid. To find the optimal operating point of the system, an accelerated version of ADMM is presented based on the concept of over-relaxation and augmented Lagrangian relaxation. This distributed algorithm outperforms the standard ADMM. Both the forecasting method and the solution algorithm are distributed, and the autonomous agents do not need to share their sensitive information with other parties either for forecasting or scheduling purposes. The simulation results on a modified CIGRE low-voltage network show the efficiency of the forecast approach as compared to several other learning methods (ARIMA, NNE, KNN, LSTM, GRU, CNN, and CNN-LSTM) and effectiveness of the accelerated ADMM algorithm for microgrid energy management as compared to the centralized MEMS, the standard ADMM, and analytical target cascading (ATC). The main contributions of the paper are summarized as follows:

- A composite CNN-GRU deep learning time series forecasting method is designed for each agent to predict market prices, wind generation, solar generation, and load demand with a high level of accuracy.
- An accelerated ADMM algorithm is presented to enhance the convergence performance of the microgrid distributed energy management problem.
- A comparison is provided using predicted and real data to demonstrate the effectiveness and advantage of the proposed forecasting engines for multi-agent microgrid energy management.

The rest of this paper is organized as follows. In Section 2, the problem and agent models are described. The designed forecaster is presented in Section 3. The accelerated ADMM algorithm is presented in Section 4. Simulation results are discussed in Section 5, and concluding remarks are provided in Section 6.

2. Problem description and agent model

The structure of the considered microgrid system is shown in Fig. 1. The considered problem is a day-ahead multi-agent microgrid energy management with the aim of minimizing agents operational cost and system losses with respect to the network and agents constraints. Six types of agents are defined as: conventional distributed generator agent, wind turbine agent, photovoltaic agent, demand agent, battery storage system agent, microgrid aggregator agent. The first five agents are resource agents that contribute to power generation and consumption. The wind and solar agents predict their generation profiles and send them to the microgrid aggregator agent. The CDG, BESS, and demand agents solve their local optimization problems, determine their active and reactive powers generation and consumption, and send these values to the microgrid aggregator agent. The aggregator solves its optimization taking into account network operational constraints, energy loss, and power exchange with the upstream grid. The aggregator determines power mismatch on each bus and sends the mismatch signal corresponding to each bus to the agents located on that bus. The microgrid aggregator plays the role of a supervisory authority to coordinate the autonomous agents without gathering agents' sensitive information.

2.1. Wind turbine and photovoltaic agents

No generation cost is considered for wind and solar generations. In this paper, the models in Ref. [27] are adopted for solar and wind generations modeling. A deep neural network-based method described in Section 3 is developed to forecast wind and solar generations. The wind and solar agents send their predictions to the microgrid aggregator agent.

2.2. Demand agent

The demand is categorized into two types, namely responsive and nonresponsive. The responsive demand is considered as the

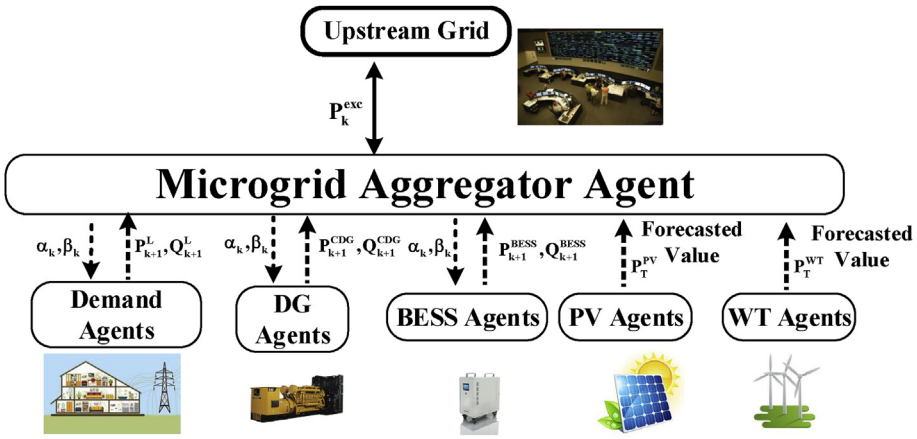

Fig. 1. Multi-agent microgrid structure.

curtailable and regulating load that can be shifted and regulated through a predetermined time interval $[t_{\min}, t_{\max}]$. The cost of participation in demand response is:

$$C_n^l = \sum_{t=1}^{T}\left(\left(P_n^{sch}\left(t\right) - P_n^l\left(t\right)\right)\right)^2 \tag{1}$$

The power consumption by the regulating load l at time interval t is bounded as:

$$P_{n,min}^l \le P_n^l(t) \le P_{n,max}^l\ ,\ \forall t \tag{2}$$

The total scheduled energy consumption needs to be maintained equal to the total forecast energy consumption.

$$\sum_{t=t_{\min}}^{t_{\max}} P_n^l(t)\Delta t = E_{sch} \tag{3}$$

Curtailable loads may reduce some percentage of the total energy consumption at certain times. The cost of the customer is:

$$C_m\left(P_m^l\right) = \sum_{t=1}^{T}\left(P_m^{sch}(t)\left(1 - e^{-\sigma P_{cur}^l(t)}\right)\right) \tag{4}$$

$$P_{cur}^l(t) = P_m^{sch}(t) - P_m^l(t)\ ,\ \forall t \tag{5}$$

The curtailable load is bounded for the satisfaction of customers.

$$P_{\min,cur}^l \le P_{cur}^l(t) \le P_{\max,cur}^l\ ,\ \forall t \tag{6}$$

The total cost function of a controllable load is:

$$C_l = C_n + C_m \tag{7}$$

2.3. Conventional distributed generator agent

Conventional DG units are controllable energy resources whose generation cost is a quadratic function of its real power generation P^{CDG}.

$$C\left(P^{CDG}\left(t\right)\right) = a^{CDG}\left(P^{CDG}(t)\Delta t\right)^2 + b^{CDG}P^{CDG}\left(t\right)\Delta t + c^{CDG} \tag{8}$$

The operational constraints of a CDG are:

$$P_{\min}^{CDG} \le P^{CDG}(t) \le P_{\max}^{CDG}\ ,\ \forall t \tag{9}$$

$$\begin{cases} P^{CDG}(t) - P^{CDG}(t-1) \le R^{up}P_{\max}^{CDG} \\ P^{CDG}(t-1) - P^{CDG}(t) \le R^{down}P_{\max}^{CDG} \end{cases} \tag{10}$$

$$\left(P^{CDG}(t)\right)^2 + \left(Q^{CDG}(t)\right)^2 \le \left(S^{CDG}\right)^2\ ,\ \forall t \tag{11}$$

2.4. Battery energy storage system agent

The cost function of BESS is modeled by (12).

$$C\left(P^{BESS}\left(t\right)\right) = a^{BESS}\sum_{t=1}^{T}P^{BESS}\left(t\right)^2 - b^{BESS}\sum_{t=1}^{T-1}\left(P^{BESS}\left(t+1\right) P^{BESS}\left(t\right)\right) + c^{BESS}\sum_{t=1}^{T}\left(\min\left(E^{BESS}\left(t\right) - d^{BESS}E_{\max}^{BESS}, 0\right)\right)^2 \tag{12}$$

Parameter d^{BESS} penalizes the cost function when the BESS energy level is less than $d^{BESS} \times E_{\max}^{BESS}$. The operational constraints of BESS are:

$$P_{\min}^{BESS} \le P^{BESS}(t) \le P_{\max}^{BESS}\ ,\ \forall t \tag{13}$$

$$\left(P^{BESS}(t)\right)^2 + \left(Q^{BESS}(t)^2\right) \le \left(S^{BESS}\right)^2\ ,\ \forall t \tag{14}$$

$$E^{BESS}(t+1) = E^{BESS}(t)\eta^{BESS} + P^{BESS}(t)\Delta t\ ,\ \forall t \tag{15}$$

$$E_{\min}^{BESS} \le E^{BESS}(t) \le E_{\max}^{BESS}\ ,\ \forall t \tag{16}$$

$$E^{BESS}(T) \ge E_{res}^{BESS} \tag{17}$$

Table 1
Proposed deep learning method: CNN-GRU architecture for forecasting engines.

Forecaster	Layer	Filter #cell	Activation Function	Loss Function/Optimizer
Load	Convolution 2D	(2,1) × 512	ReLU	MAPE/ADAM
	GRU	#128 cell	ReLU	
	Dense	#64	Linear	
PV	Convolution 2D	(2,1) × 512	ReLU	MSE/ADAM
	GRU	#128 + 0.2 dropout	ReLU	
	GRU	#128 + 0.2 dropout	ReLU	
	Dense	#64	ReLU	
Wind	Convolution 2D	(2,1) × 512 +0.2 dropout	ReLU	MAPE/ADAM
	Convolution 2D	(2,1) × 512	ReLU	
	GRU	#256 + 0.25 dropout	ReLU	
	GRU	#256 + 0.25 dropout	ReLU	
	Dense	#64	ReLU	
Price	Convolution 2D	(2,1) × 512	ReLU	MAE/ADAM
	GRU	#512	ReLU	
	GRU	#256	ReLU	
	GRU	#128	ReLU	
	Dense	#64	Linear	

2.5. Microgrid aggregator agent

The microgrid aggregator agent receives resource agents' information (active and reactive powers generation/consumption by generator/demand agents) and formulates an OPF problem to manage the mismatch between supply and demand, branch flows, bus injections, energy loss, and power exchange with the upstream grid. The aggregator agent solves its OPF problem, determines the mismatches at each bus and sends mismatch signals corresponding to a bus to agents located on that bus. The net complex power consumption on bus i at time t is calculated as:

$$\sum_{i}^{N^{bus}} S_i(t) = \sum_{i}^{N^{bus}} S_i^{l}(t) + \sum_{i}^{N^{bus}} S_i^{BESS}(t) - \sum_{i}^{N^{bus}} S_i^{CDG}(t) - \sum_{i}^{N^{bus}} S_i^{PV}(t) - \sum_{i}^{N^{bus}} S_i^{WT}(t)\ ,\ \forall i \in \left\{1,\ldots,N^{bus}\right\}\ ,\ \forall t \tag{18}$$

Considering the microgrid as a radial network [28], the power flow equations are described as follows:

$$P_{ij}(t) - r_{ij}I_{ij}(t) - P_j(t) = \sum_{\substack{(j,k)\in\Lambda \\ k\neq i}} P_{jk}(t) \tag{19}$$

$$Q_{ij}(t) - x_{ij}I_{ij}(t) - Q_j = \sum_{\substack{(j,k)\in\Lambda \\ k\neq i}} Q_{jk}(t) \tag{20}$$

$$V_j(t) = V_i(t) - 2\left(r_{ij}P_{ij}(t) + x_{ij}Q_{ij}\left(t\right)\right) + \left(|z_{ij}|\right)^2 I_{ij}(t) \tag{21}$$

$$I_{ij}(t)V_i(t) = \left(P_{ij}(t)\right)^2 + \left(Q_{ij}(t)\right)^2 \tag{22}$$

Bus voltages are bounded as:

$$v_i^{\min} \le \left|v_i(t)\right| \le v_i^{\max}\ ,\ \forall t \tag{23}$$

where $v_i(t)$ ($V_i(t) = |v_i(t)|^2$) is the voltage magnitude at bus i at time interval t. The cost of energy purchased from the main grid at time interval t is:

$$C\left(P^{exc}(t)\right) = P^{exc}(t)\pi^{f}(t)\Delta t \tag{24}$$

$P^{exc}(t)$ will be negative if the microgrid aggregator sells power to the main grid, and it will be positive if this agent purchases power from the main grid. Equation (22) makes the optimization problem nonconvex. This equation is relaxed by the following inequality constraint:

$$I_{ij}(t)V_i(t) \ge \left(P_{ij}(t)\right)^2 + \left(Q_{ij}(t)\right)^2\ ,\ \forall t\ ,\ \forall (i,j)\in\varepsilon \tag{25}$$

This relaxation might be exact if the bus voltages are kept around the nominal value, and the power flow at each line is not too large [6]. Simulation results will show the precision of the relaxation in the case of low DG penetration levels.

2.6. Proposed forecast engine

To solve the multi-agent energy management problem, demand, solar generation, wind generation, and electricity prices from the upstream grid need to be forecasted at each time interval. In this section, a forecast engine is designed for each agent to predict its uncertain parameters. Since the agents operate independently, the forecasting engines perform independently to protect the privacy of the agents. The general framework with CNN, GRU, and Dense layers order is held for all other agents. However, the layers of each block are different for each agent. As an instance, the parameters of the proposed CNN-GRU for the residential load at bus 3, PV, wind speed, and price are shown in Table 1. The architecture of the proposed CNN-GRU engine includes three main components: i) CNN block, ii) GRU block and iii) dense block. First, the historical data is prepared with a representation learning technique based on converting data from 1D to 2D and shaping data into 4D tensor. Then, the CNN block, which is composed of convolution layers and max-pooling layers, extracts features. The output of the CNN block acts as the input of the GRU block. Finally, the dense block controls the dimension of the GRU output and displays the forecasting engines' outputs. The detailed descriptions are explained in the following subsections. The mean absolute percentage error (MAPE) for residential loads and wind speed, mean square error (MSE) for PV power, and mean average error (MAE) for the price of a sample are considered as loss functions. The Adam algorithm [28] is utilized to minimize the MAPE/MSE/MAE. Furthermore, the dropout technique, presented in Ref. [29], is used to tackle the overfitting

problem. The utilized activation function is rectified linear unit (ReLU) to mitigate the numerical instabilities caused by the vanishing gradients [30].

2.7. Input data preparation

One of the advantages of the designed structure is that it does not require any pre-processing techniques. Data preparation is carried out based on raw data, including historical prices, loads, solar irradiation, and wind speed. The 1D time series is not sufficient for complete feature extraction. Therefore, the representation learning technique is applied in which the initial dataset, which is extensively assessed using real input, is converted to 2D-vectors through "look-back (LB)" that depends on the sample size. Assuming LB = 2 [31], the input data are arranged in the form of:

$$X_{LB}=\underbrace{\begin{bmatrix} NaN & NaN & \cdots & X_{n-3} & X_{n-2} \\ NaN & X_1 & \cdots & X_{n-2} & X_{n-1} \\ X_1 & X_2 & \cdots & X_{n-1} & X_n \end{bmatrix}}_{\text{Time Series Samples}}\begin{matrix}\} & X_{input}\\ \} & Y_{output}\end{matrix} \tag{26}$$

Time series (TS) data, as the input, is converted from a $TS\times 1$ vector to a $TS \times LB$ vector. Then, the $TS \times LB$ vector is converted to 4D tensors as $TS \times (1 \times LB \times 2 \times 1)$ to enhance features. The structure of each block of the CNN-GRU is described in the next subsections.

2.8. Convolutional neural networks block

This block contains two layers, including convolution layer and max-pooling layer. The output shape of CNN does not match the required input shape of RNN-based methods, and it is necessary to match the output of the CNN block and the input of the GRU block. To this end, the output of CNN is converted to multiple sequences to arrange GRU block inputs based on *Time distributed* wrapper, which ensures the CNN and GRU connections.

Convolution layers process two-dimensional inputs by convolving the input network, multiple linear filters, and learned weights to detect spatial features in the local structure of the input network in CNN. This is called 2D-CNN [32]. Convolution layers become deeper with taking inputs from a feature map. The feature map is the output of the neurons drawn across the entire previous layer. The output of the feature map is the result of a predefined activation of the neurons. The output of m^{th} feature map at L^{th} layer, $y(m,L)$, is given as:

$$y(m,L)=f\{\ [W\otimes x(n,L-1)\ ]+B(m,L)\ \}\quad \forall m=1,\ldots,y_{wi}\ \forall n=1,\ldots,y_{he} \tag{27}$$

$$y_{wi}=\frac{x_{wi}-W_{wi}}{S}+1 \tag{28}$$

$$y_{he}=\frac{x_{he}-W_{he}}{S}+1 \tag{29}$$

The symbol $\otimes$ shows a convolution operation. In the designed forecasting engines, the output of each convolution layer is followed by a max-pooling layer that maps features from the previous layer and makes the convolution layers highly overlapped with each other. Thus, max-pooling mitigates the overfitting problem, improves computational cost, and reduces information redundancy and size [33]. The m^{th} output map at the L^{th} layer of the max-pooling layer P, $y(m,L)$, with ReLU activation function is represented as:

$$y(m,L)=f\{[\omega(m,L)P(x(m,L-1)]+b(m,L)\}\quad \forall m=1,\ldots,y_{wi} \tag{30}$$

As depicted in Table 1, the convolution layer output is a 4D tensor, i.e., (sample, channels, rows, cols, and filters). It is considered as the input of the max-pooling layer. The input vector of max-pooling is a vector with the size of $TS \times 1 \times 2 \times 2 \times 512$. The max-pooling layer subsamples enable the designed structure to capture spatial information at multiple scales. The output of max pooling-layer is a vector with the size of $TS \times 1 \times \frac{LB}{2} \times 2 \times 2 \times 512$. In the last layer of the CNN block, the output of the max-pooling layer is flattened with a flattening technique. This technique converts the data to a 1D-vector with $TS \times 1024$ size and feeds it into the GRU block.

2.9. Gated recurrent unit block

Two GRU layers are implemented in the block that each of them is characterized by an update gate (u), a reset gate (r), and a hidden state (h). The update gate controls the spatial and temporal information that should be accepted as new input features. This gate stores the current memory across multiple time intervals when an important feature is detected in memory content from a previous feature. The update gate is mathematically described as follows:

$$u(m,L,t)=f\ [\ W_u y(m,L,t)+R_u h(L,t-1)\ B_u(m,L)] \tag{31}$$

The reset gate determines whether the memory cell removes unnecessary detected features. The information of the previous layer will be discarded if the reset gate is close to zero. Then, the reset gate would make the unit acts if the next processed unit is the first in the sequence. The reset gate $R(m,L,t)$ is governed by:

$$r(m,L,t)=f\ [\ W_r y(m,L,t)+R_r h(L,t-1)+ B_r(m,L)] \tag{32}$$

Both gates are characterized by hidden state $h(L,t)$ and candidate state $h'(L,t)$, which are the memory of GRU layers. These states are described as:

$$h(L,t)=[\ 1-u(m,L,t)h(L,t-1)]+ u(m,L,t)h'(L,t) \tag{33}$$

$$h'(L,t)=f\ [\ W_h y(m,L,t)+R_h h(L,t-1)+B_h(m,L)] \tag{34}$$

Two GRU layers are used to construct a $TS \times 128$ vector and transform it into a $TS \times 62$ vector with 10% drop out.

2.10. Dense block

The third block of the proposed structure is the dense block that is added to connect all hidden states in GRU and control the

dimensions of the GRU output. The m^{th} output map at the L^{th} layer passes a dense layer to construct the final output map as:

$$y(m,L) = W_f f\left[W.y(m,L) + B_f(m,L)\right] \tag{35}$$

In the dense block dense layers are used to transform the output of GRU block into a $TS \times LB$ vector. The second dense layer utilizes a linear activation function to create final forecast results.

4. Multi-agent distributed microgrid scheduling

After predicting the uncertain parameters, a day-ahead scheduling subproblem is formulated for each agent using its local forecast parameters. These subproblems can be solved by mathematical optimization methods or metaheuristic algorithms. Since all autonomous agents' subproblems formulated in Section 2 are convex (we also convexify the aggregator agent's optimization), mathematical methods are more efficient than metaheuristic algorithms. ADMM can be used to coordinate the optimization subproblems. However, the conventional ADMM may perform poorly, if various problem components are poorly conditioned or a set of highly accurate results is needed [34]. The over-relaxation technique presented in Ref. [34] is a potential solution to overcome the conventional ADMM problems and increase its convergence speed.

If a user wants to formulate a centralized MEMS, the objective function would include three terms: minimizing the cost of purchasing energy from the main grid by the aggregator agent; minimizing the power losses (handled by the aggregator agent), and the constraints would be (2), (3), (6), (9), (10), (11), (13)–(17), (19)–(21), (23), and (25).

$$\begin{aligned} &\min \zeta^{MA} C\left(P^{exc}(t)\right) + \zeta^{loss} \sum_{(i,j)\in\varepsilon} r_{ij} I_{ij}(t) + \\ &\zeta^{l} C\left(P^{l}(t)\right) + \zeta^{CDG} C\left(P^{CDG}(t)\right) + \\ &\zeta^{BESS} C\left(P^{BESS}(t)\right) \\ &s.t\ (2),(3),(6),(9-11),(13-17),(19-21),(23),(25) \end{aligned} \tag{36}$$

Based on this centralized problem, the agents' optimization subproblems in the context of distributed optimization are formulated in the subsequent sections.

4.1. Demand agent

The forecasted load by CNN-GRU should be supplied by CDGs and renewable DGs. The demand agents are responsible for adjusting the controllable load consumption and solving the following problem:

$$\begin{aligned} &\min_{S^l}\ \zeta^{l} C\left(P^{l}(t)\right) + \frac{\rho_1^l}{2}\left|\left|P^{l} - P_{k+1} + \alpha_k\right|\right|_2^2 + \\ &\frac{\rho_2^l}{2}\left|\left|Q^{l} - Q_{k+1} + \beta_k\right|\right|_2^2,\ s.t\ (2),(3),(6) \end{aligned} \tag{37}$$

4.2. Conventional distributed generation agent

Unlike WT and PV agents, CDG agents control their output power with respect to operational constraints. Each CDG agent solves the following optimization problem:

$$\begin{aligned} &\min_{S^l}\ \zeta^{CDG} C\left(P^{CDG}(t)\right) + \frac{\rho_1^{CDG}}{2}\left|\left|P^{CDG} - P_{k+1} + \right.\right. \\ &\left.\left.\alpha_k\right|\right|_2^2 + \frac{\rho_2^{CDG}}{2}\left|\left|Q^{CDG} - Q_{k+1} + \beta_k\right|\right|_2^2,\ s.t\ (9)-(11) \end{aligned} \tag{38}$$

4.3. Battery energy storage agent

Each BESS agent solves the following optimization to minimize its cost:

$$\begin{aligned} &\min_{S^{BESS}}\ \zeta^{BESS} C\left(P^{BESS}(t)\right) + \frac{\rho_1^{BESS}}{2}\left|\left|P^{BESS} - P_{k+1} + \right.\right. \\ &\left.\left.\alpha_k\right|\right|_2^2 + \frac{\rho_2^{BESS}}{2}\left|\left|Q^{BESS} - Q_{k+1} + \beta_k\right|\right|_2^2 \\ &s.t\ (13)-(17) \end{aligned} \tag{39}$$

4.4. Microgrid aggregator agent

The microgrid aggregator agent is responsible for monitoring the power balance and bus voltages. This agent solves an optimization problem according to the forecast market prices and the information received from other agents. The information exchanged between the aggregator agent and other agents include the active and reactive power of buses and lagrangian multiplier. The aggregator problem is:

$$\begin{aligned} &\min_{\substack{P(t),Q(t),V(t)\\ I(t),S(t)}}\ \zeta^{MA} C\left(P^{exc}(t)\right) + \zeta^{loss} \sum_{(i,j)\in\varepsilon} r_{ij} I_{ij}(t) + \\ &\frac{\rho_1^{MA}}{2}\left|\left|P_{k+1}^{l} + P_{k+1}^{BESS} - P_{k+1}^{CDG} - P_{k+1}^{RDG} - P_{k+1} + \alpha_k\right|\right|_2^2 \\ &+\frac{\rho_2^{MA}}{2}\left|\left|Q_{k+1}^{l} + Q_{k+1}^{BESS} - Q_{k+1}^{CDG} - Q_{k+1}^{RDG} - Q_{k+1} + \beta_k\right|\right|_2^2 \\ &s.t\ (19-21),\ (23)-(25) \end{aligned} \tag{40}$$

where $P(t)$, $Q(t)$, and $S(t)$ include $\{P_i, P_{ij}\}$, $\{Q_i, Q_{ij}\}$, and $\{S_i, S_{ij}\}$.

4.5. Accelerated alternating direction method of multipliers with over-relaxation

To enhance the convergence speed of the distributed multi-agent microgrid energy management approach, a relaxation parameter μ is used for each consistency constraint and an over-relaxation technique is implemented to accelerate ADMM [35]. Two sets of auxiliary variables are defined, one set for active power ($\widehat{P}$) and another set for reactive power ($\widehat{Q}$). The values of the auxiliary variables at iteration $k+1$ are calculated as:

$$\widehat{P}_{k+1} = \mu_1\left(P_{k+1}^{l} + P_{k+1}^{BESS} - P_{k+1}^{CDG} - P_{k+1}^{RDG}\right) + (1-\mu_1)P_k \tag{41}$$

$$\widehat{Q}_{k+1} = \mu_2\left(Q_{k+1}^{l} + Q_{k+1}^{BESS} - Q_{k+1}^{CDG} - Q_{k+1}^{RDG}\right) + (1-\mu_2)Q_k \tag{42}$$

If $\mu_k > 1$, over-relaxation will be implemented, while $\mu_k < 1$ leads to under-relaxation. The relaxation variable is often selected

between 1.5 and 1.8 [36]. Therefore, $\mu_k = 1.75$ is selected based on trial and error. Replacing the auxiliary variables in (40), the following optimization subproblem is formulated for a microgrid aggregator agent:

$$\min_{\substack{P(t),Q(t),V(t)\\ I(t),S(t)}} \zeta^{MA} C\left(P^{exc}(t)\right) + \zeta^{loss} \sum_{(i,j)\in\varepsilon} r_{ij} I_{ij}(t) + \frac{\rho_1}{2}\left\|\widehat{P}_{k+1} - P_{k+1} + \alpha_k\right\|_2^2 + \frac{\rho_2}{2}\left\|\widehat{Q}_{k+1} - Q_{k+1} + \beta_k\right\|_2^2 \quad s.t\ (19-21),\ (23)-(25) \tag{43}$$

After solving the optimization subproblems at each time interval, the system operator updates the Lagrange multipliers using the latest values of the auxiliary variables and active and reactive powers.

$$\alpha_{k+1} = \alpha_k + \left(\widehat{P}_{k+1} - P_k(t)\right) \tag{44}$$

$$\beta_{k+1} = \beta_k + \left(\widehat{Q}_{k+1} - Q_k(t)\right) \tag{45}$$

The stopping criteria, which are based on the active and reactive powers, are the primal residual $r_{k+1}^{P/Q}$ and the dual residual $s_{k+1}^{P/Q}$.

$$r_{k+1}^{p} = \left(P_{k+1}^{l} + P_{k+1}^{BESS} - P_{k+1}^{CDG} - P_{k+1}^{RDG}\right) - P_{k+1}^{i} \tag{46}$$

$$r_{k+1}^{Q} = \left(Q_{k+1}^{l} + Q_{k+1}^{BESS} - Q_{k+1}^{CDG} - Q_{k+1}^{RDG}\right) - Q_{k+1}^{i} \tag{47}$$

$$s_{k+1}^{P} = -\rho_1\left(P_{k+1}^{i} - P_{k}^{i}\right) \tag{48}$$

$$s_{k+1}^{Q} = -\rho_2\left(Q_{k+1}^{i} - Q_{k}^{i}\right) \tag{49}$$

The solution is obtained when the norms of primal and dual residuals are less than a threshold.

$$\left\|r_{k+1}^{Q}\right\| \le \varepsilon_{pri}^{P} \;\&\; \left\|r_{k+1}^{Q}\right\| \le \varepsilon_{pri}^{Q} \;\&\; \left\|r_{k+1}^{Q}\right\| \le \varepsilon_{dual}^{P} \;\&\; \left\|r_{k+1}^{Q}\right\| \le \varepsilon_{dual}^{Q} \tag{50}$$

where ε_{pri}^{P}, ε_{pri}^{Q}, ε_{dual}^{P}, and ε_{dual}^{Q} are determined as:

$$\varepsilon_{pri}^{P} = \sqrt{T}\,\varepsilon^{abs} + \varepsilon^{rel}\, max\left\{\left\|P_k^{l} + P_k^{BESS} - P_k^{CDG} - P_k^{RDG}\right\|_2^2, \left\|-P_k^{i}\right\|_2^2\right\} \tag{51}$$

$$\varepsilon_{pri}^{Q} = \sqrt{T}\,\varepsilon^{abs} + \varepsilon^{rel}\, max\left\{\left\|Q_k^{l} + Q_k^{BESS} - Q_k^{CDG} - Q_k^{RDG}\right\|_2^2, \left\|-Q_k^{i}\right\|_2^2\right\} \tag{52}$$

Parameters ε^{abs} and ε^{rel} represent absolute and relative tolerances, respectively. A reasonable value for the relative stopping criterion is $\varepsilon^{abs} = 10^{-4}$ and $\varepsilon^{rel} = 10^{-3}$. The procedure of the accelerated ADMM algorithm is summarized in Algorithm 1.

4.6. Discussion on convergence

As shown in Fig. 2, ADMM is a special case of Douglas-Rachford splitting that is itself a special case of the proximal point algorithm. The proposed accelerated ADMM is based on the general proximal point algorithm. Since the general proximal point algorithm is proven to converge faster than the proximal point algorithm [36], it implies that the general Douglas-Rachford splitting is faster than the classical Douglas-Rachford splitting, and consequently, the general (i.e., accelerated) ADMM is faster than the classical ADMM. A detailed convergence proof for the accelerated ADMM is given in Ref. [37].

5. Simulation results

A modified CIGRE low-voltage network [38], shown in Fig. 3, is used. Simulation results are presented in two parts: forecasting results, and ADMM results. The proposed CNN-GRU forecasting accuracy is compared with several popular methods. The performance of the developed accelerated ADMM is compared with the centralized MEMS, the standard ADMM, and ATC. All forecasting methods are implemented in the Keras package [39], in the graphical processing unit (GPU) with a GeForce GTX 1080 TI. The optimization algorithms are implemented using the CVX package [40] in MATLAB.

5.1. Forecasting engines results

The real price data is extracted from day-ahead information of the CAISO market [41]. Wind speed, solar irradiance, and household demand datasets are collected from February 1, 2012, to January 31, 2013, of the city of London [42]. 70% of the dataset is dedicated to training, and 30% of them is used for the testing process.

The training procedure is carried out based on the truncated backpropagation through time [43], and the testing process is carried out using the walk forward validation [44]. Training the designed CNN-GRU model with data of one year for 480 epochs takes approximately 40 min. The results show that the application of CNN-GRU method for one-step forecasting of wind speed and solar irradiation last 5.7963 ms and 5.6965 ms, respectively.

The root mean square error (RMSE), normalized root mean square error (NRMSE), and mean absolute percentage error (MAPE) metrics are reported to compare the results [23].

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}\left(y_{re}(t) - y_f(t)\right)^2}{N}} \tag{53}$$

$$NRMSE = \frac{\sqrt{\frac{\sum_{t=1}^{N}\left(y_{re}(t) - y_f(t)\right)^2}{N}}}{y_{\max} - y_{\min}} \tag{54}$$

Algorithm 1: Accelerated ADMM-based multi-agent microgrid energy management

Initialization: set $k = 0$ and randomly initialize P_k, Q_k, V_k, I_k, P_{ij}, Q_{ij}, α_k, β_k, P_k^{l}, P_k^{BESS}, Q_k^{l}, Q_k^{BESS}, P_k^{CDG}, and Q_k^{CDG} Repeat k = 1, 2, …

Solve optimization subproblems of the resource agents in parallel and send the optimal values of P_k and Q_k to the microgrid aggregator.

Calculate $\widehat{P_{k+1}}$ and $\widehat{Q_{k+1}}$ based on (41) and (42).

Solve the optimization subproblem of the microgrid aggregator agent using $\widehat{P_{k+1}}$ and $\widehat{Q_{k+1}}$.

Update Lagrange multipliers based on (44) and (45) and send the multipliers to all agents.

Terminate the loop if the convergence criteria are satisfied.

End

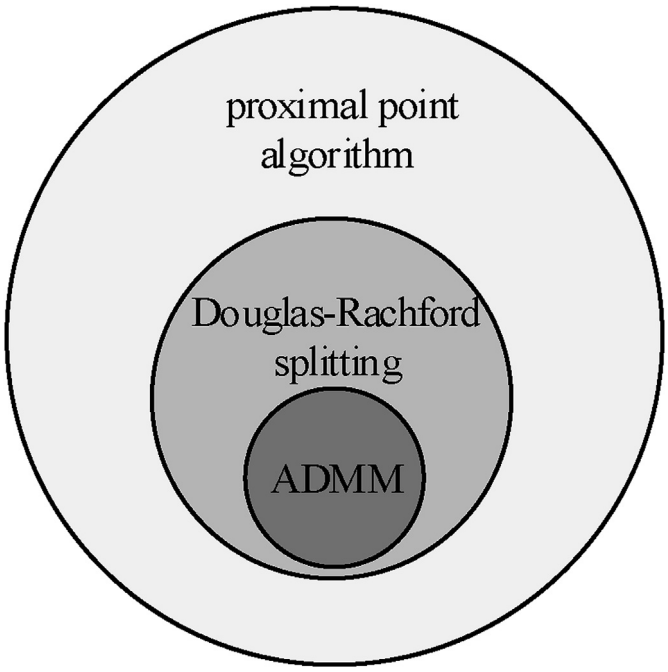

Fig. 2. ADMM is a special case of the proximal point algorithm.

$$MAPE = \frac{\sum_{t=1}^{N}\left|\frac{y_{re}(t)-y_f(t)}{y_{re}(t)}\right|}{N} \quad (55)$$

CNN-GRU results are compared with real data and forecast values obtained by several deep learning structures (i.e., CNN-LSTM, 2D CNN, LSTM, and GRU). In addition, results obtained by the k-nearest neighbor (KNN) and neural network ensemble (NNE), which are shallow structure neural networks, and autoregressive integer moving average (ARIMA), which is a statistical short-term forecasting engine, are compared with CNN-GRU results.

Fig. 4 illustrates forecast price, wind speed, solar radiation, and residential load on bus 3 obtained by the proposed CNN-GRU. Although training and testing are implemented for one year, forecast values have been shown for a sample day (January 26, 2013). The forecast resolution is 30 min. Comparing the forecast values with the real data shows the high accuracy of CNN-GRU. In addition, Tables 2–5 verify the higher accuracy of CNN-GRU as compared to other methods. For example, the resulted MAPE from CNN-GRU for solar irradiance forecasting shows 79.35% and 88.91% improvement as compared to MAPEs obtained by, respectively, KNN and ARIMA.

All consumers are assumed to be residential. There are instantaneous changes in the residential load patterns due to customers' habits, cultural relations, etc. Therefore, the residential load forecasts usually have lower accuracy than other uncertain parameters. The CNN-GRU short-term forecast for household loads connected to bus 3 is presented in Fig. 4(d). DNN-based forecasters provide a higher level of accuracy than other methods. The proposed CNN-GRU provides the most accurate results. For instance, MAPE obtained by CNN-GRU is less than half of that obtained by GRU. As shown in Tables 2–5, CNN-GRU results are more accurate than KNN, NNE, and ARIMA results. For example, the resulted MAPE from CNN-GRU for solar irradiance forecasting shows 79.35%, 73.38% and 88.91% improvements in MAPEs obtained by KNN, NNE, and ARIMA, respectively.

5.2. Multi-agent microgrid energy management results

January 26, 2013, is selected to study the impacts of forecasting techniques on the multi-agent energy management algorithm. The parameters of CDG on bus 5 are illustrated in Table 6. The maximum charging rate of BESSs connected to buses 6 and 13 are 2.5 kW and 1 kW, respectively. The maximum/minimum allowable charge level of BESSs on buses 6 and 13 are, respectively, 30/0.5 kWh and 10/0.1. The battery charging efficiency is 0.95. a_{BESS}, b_{BESS}, c_{BESS}, and d_{BESS} are 1, 0.75, 0.5, 0.2, respectively. Loads on buses 3, 7, 10, and 14 can be curtailed with the maximum load curtailment of 40%, 35%, 25%, and 30% of the total load, respectively. The load on bus 12 can be regulated between hours 10 and 22. The upstream grid voltage is

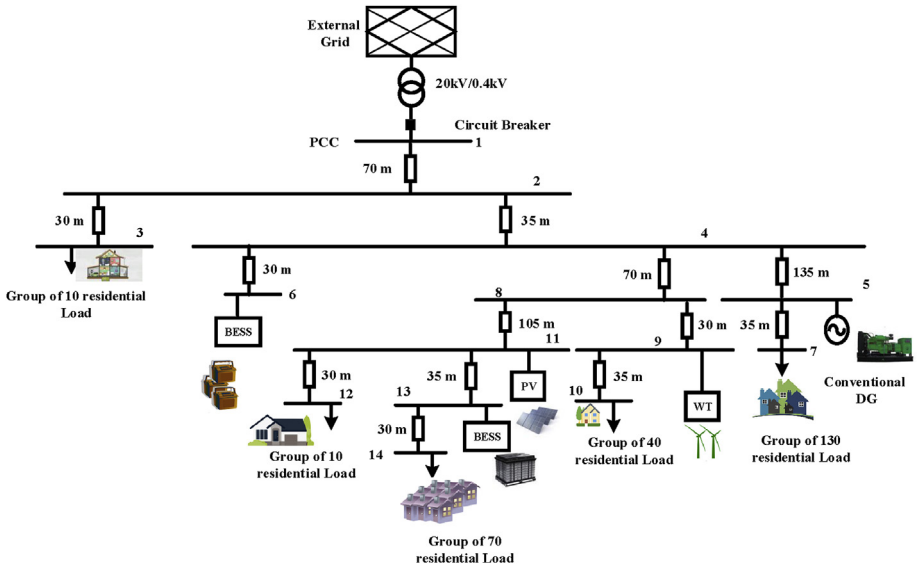

Fig. 3. Microgrid test system.

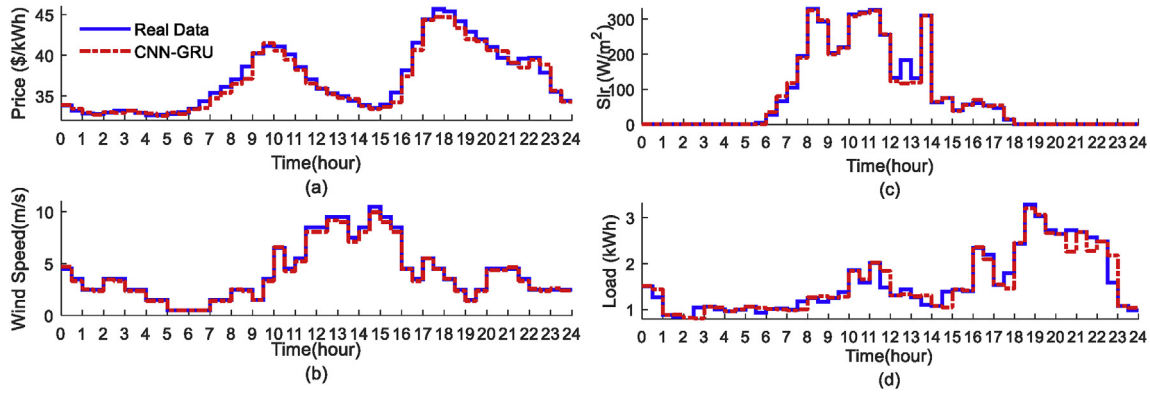

Fig. 4. Short-term forecasting results by CNN-GRU technique in a sample day; (a) Price (b) Wind speed (c) Solar irradiance (Slr) (d) Residential load.

Table 2
Performance of forecasting methods for price.

Forecasting methods	MAPE (%)	RMSE	NRMSE
CNN-GRU	0.91365	0.49417	0.037935
CNN-LSTM	0.97773	0.50422	0.038706
2D-CNN	1.0601	0.55414	0.042538
GRU	1.08	0.57172	0.043887
LSTM	1.8239	0.96092	0.073764
ARIMA	1.953	0.99353	0.099353
KNN	2.362	1.3300	0.08250
NNE	1.8642	0.98095	0.07936

Table 3
Performance of forecasting methods for wind speed.

Forecasting methods	MAPE (%)	RMSE	NRMSE
CNN-GRU	3.5875	0.21578	0.021758
CNN-LSTM	5.4813	0.34515	0.034515
2D-CNN	8.1056	0.33372	0.044556
GRU	10.218	0.45985	0.062843
LSTM	10.37	0.48	0.065322
ARIMA	21.546	0.983	0.0783
KNN	28.910	1.037	0.0860
NNE	19.773	0.8412	0.08254

Table 4
Performance of forecasting methods for solar irradiation.

Forecasting methods	MAPE (%)	RMSE	NRMSE
CNN-GRU	5.47	10.627	0.032399
CNN-LSTM	6.2253	9.7109	0.026060
2D-CNN	8.0339	7.7818	0.048386
GRU	14.505	20.997	0.064015
LSTM	16.612	20.532	0.062597
ARIMA	49.322	58.561	0.08561
KNN	26.499	54.354	0.0670
NNE	20.548	45.357	0.07856

1.02 p.u. The minimum and maximum voltage magnitudes of each bus are 0.95 and 1.05. The accelerated ADMM parameters ζ_{BESS}, ζ_{CDG}, ζ_{MA}, ζ_{loss}, and ζ_I are assumed to be 1, and $\rho_p = \rho_q = 1.33$, $\mu_p = \mu_q = 1.75$, $\varepsilon_{abc} = 10^{-3}$, and $\varepsilon_{rel} = 10^{-3}$.

Two scheduling problems are solved with the accelerated ADMM, one with the forecast data and another one with the real data. Fig. 5 shows the active power exchange between the microgrid aggregator and the upstream grid. The power is purchased (sold) from the upstream grid when the market price is low (high, i.e., hours 16 to 18). The total loss of the understudied MG by applying the proposed accelerated ADMM method is 0.634 kWh in the case of using CNN-GRU forecasted data. The total loss by using real data is 0.605 kWh. This shows less than 5% error in the result. The load on bus 12 is regulatable, and the other loads are curtailable. For the sake of comparison, Fig. 6 shows the unscheduled (load without any capability for curtailment and regulation) and scheduled load profiles on buses 7 and 12. Demand agents benefit from the load curtailment, especially during high price periods. For example, during the period [17:30, 18], the forecast price in Fig. 4(a) is the highest price, and accordingly, the highest possible curtailment, 35%, is occurred at this period. In addition, as shown in Fig. 6(b), the demand is shifted from the high price period, i.e., [16:30, 22] to the low price periods, i.e., [11, 16:30]. The CDG and BESS agents adjust their operation based on the market price and technical characteristics (e.g., voltage limits and loss minimization) considered by the microgrids aggregator agent. Fig. 7 depicts power generated by the CDG agent. The pattern of CDG active power output is similar to the forecast price in Fig. 4(a) As shown in Fig. 8, bus voltages magnitude are in the predefined range. The voltage magnitude is high whenever the total power output of PV, WT, CDG, and BESS is high, and the load is low.

Table 5
Performance of forecasting methods for residential load at bus 3.

Forecasting methods	MAPE (%)	RMSE	NRMSE
CNN-GRU	6.0342	0.16176	0.06473
CNN-LSTM	6.566	0.22657	0.090664
2D-CNN	6.6175	0.22485	0.089977
GRU	14.391	0.24194	0.14207
LSTM	14.785	0.25425	0.1424
ARIMA	20.253	0.467	0.1880
KNN	21.259	0.462	0.1930
NNE	18.936	0.4027	0.1798

Analysis of Forecasting on Microgrid Energy Management System: The distributed multi-agent energy management is solved using the forecast data provided by CNN-GRU and four other forecast methods. The MAPE index reported in Table 7 verify the superiority of CNN-GRU for energy management.

Distributed Solution Algorithm Analysis: The performance of the accelerated ADMM is compared with the standard ADMM and analytical target cascading (ATC), which are two popular distributed optimization algorithms [45]. A convergence index, *rel*, is introduced to calculate the relative error between the objective functions obtained by the centralized and distributed algorithms.

Table 6
Parameters of CDG.

Parameters	a^{CDG}	b^{CDG}	$R^{up}/Rdown$	$P^{CDG}_{max}/P^{CDG}_{min}$	S^{CDG}
value	10	30	0.3	30/0	30

$$rel = \frac{|f_{cen} - f_{dis}|}{f_{cen}} \tag{56}$$

where f_{dis} is the total cost function determined by the distributed algorithm, and f_{cen} is the optimal cost obtained by the centralized

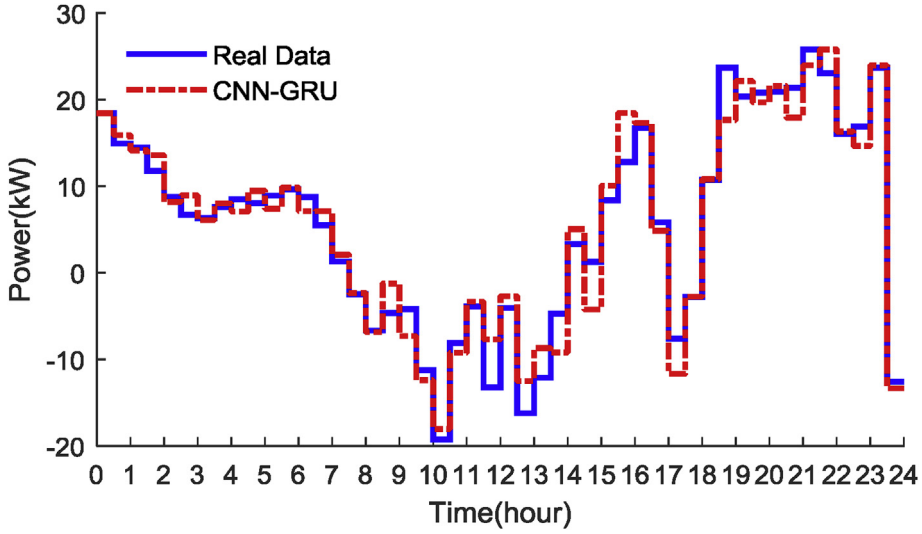

Fig. 5. The power exchange between the main grid and microgrid.

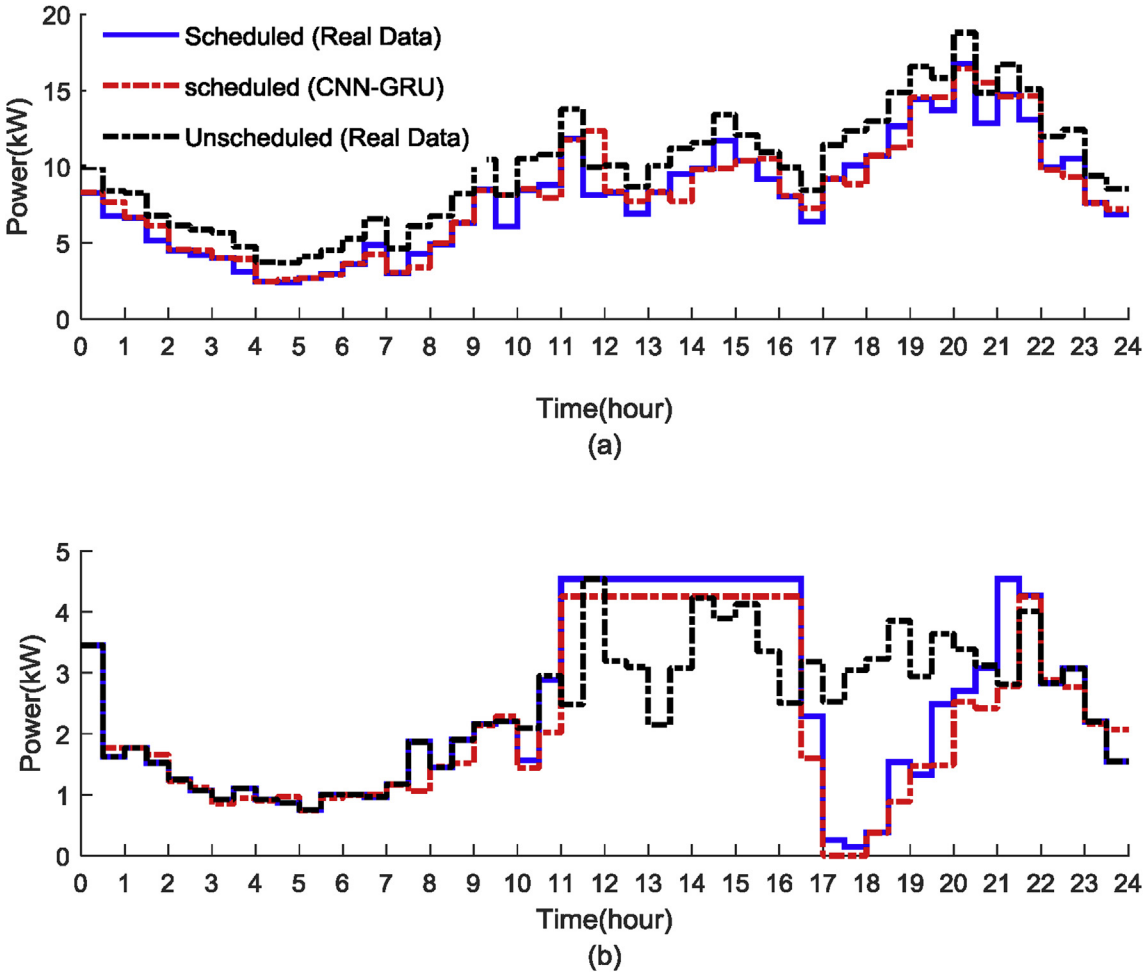

Fig. 6. Daily load profile for unscheduled and scheduled power consumption (a) Curtailable load connected to bus 7 (b) Regulated load connected to bus 12.

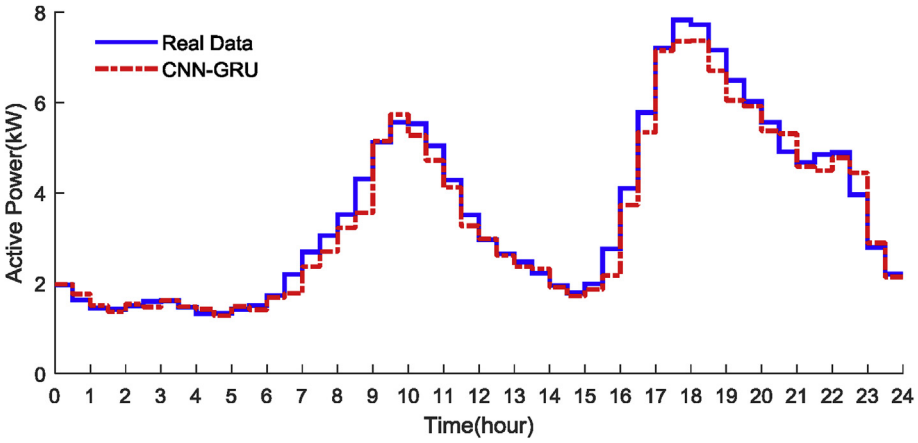

Fig. 7. CDG scheduled active power generation.

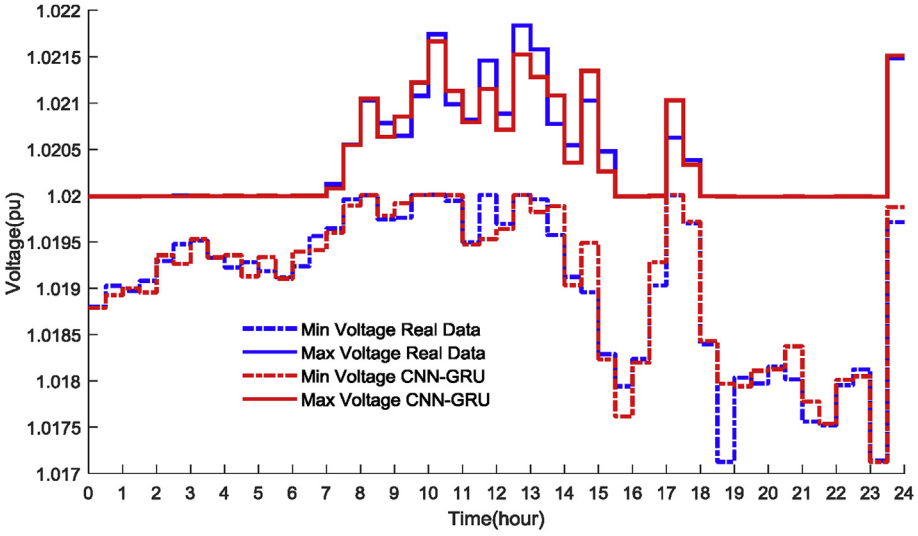

Fig. 8. Lowest and highest bus voltage in the microgrid.

Table 7
Multi-agent MEMS results MAPE% using different DNN-based forecasting techniques.

Forecasting method	Total cost	CDG	BESS at bus 6	Load at bus 7	Load at bus 12
CNN-GRU	2.017	0.605	5.5915	0.78413	1.4255
CNN-LSTM	7.039	1.048	9.7157	0.4258	1.3479
2D-CNN	2.646	1.143	6.8827	0.84993	1.3306
GRU	3.201	1.188	5.9591	1.6667	2.4528
LSTM	3.028	0.837	5.8685	1.6984	2.4558

approach. In addition, the residual norm of the power balance, i.e., $||r||$, is calculated as:

$$||r|| = ||(P_l + P_{BESS} - P_{CDG} - P_{WT} - P_{PV}) - P_i||_2 \quad (57)$$

Figs. 9 and 10 show the convergence measures over the course of iterations. The indices obtained by the proposed accelerated ADMM are below those obtained by the standard ADMM and ATC. Table 8 presents total cost, CDG cost, BESS cost, and the number of iterations for the centralized MEMS and the distributed algorithms. For instance, the CDG agent cost errors are 42.39%, 30.50%, and

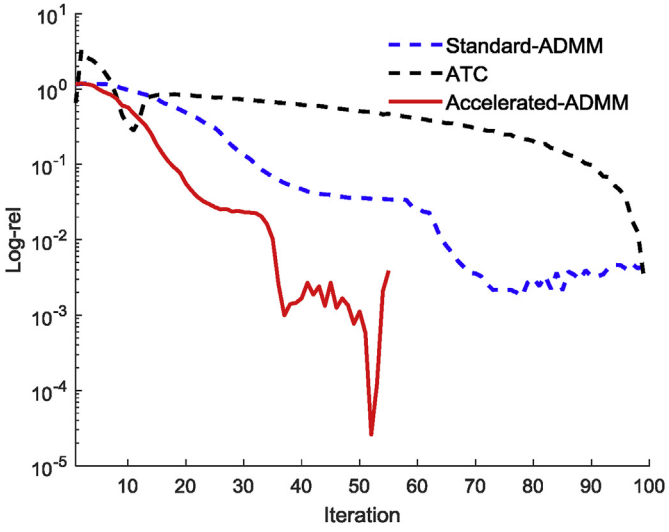

Fig. 9. Comparison between the *rel* index obtained by standard ADMM, accelerated ADMM, and ATC.

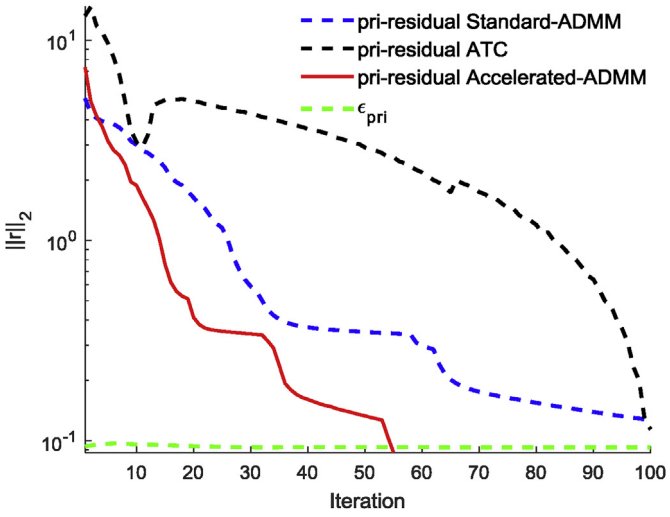

Fig. 10. Comparison between the primal residual obtained by standard ADMM, accelerated ADMM, and ATC.

Table 8
Comparison of accelerated ADMM, standard ADMM, centralized and ATC.

	Centralized (benchmark)	Standard ADMM	ATC	Accelerated ADMM
Total cost ($)	3566.6	3553.4	3585.6	3557.2
CDG agent ($)	588.33	579.12	599.54	581.04
BESS agent ($)	5.354	7.624	6.987	6.523
Iteration	–	100	99	55

21.83% for standard ADMM, ATCS, and the accelerated ADMM, respectively, in comparison with the benchmark. The accelerated ADMM converges after 55 iterations (within 20 min), which is 45 and 44 iterations less than that for the standard ADMM and ATC, respectively.

6. Conclusion and future works

This paper presents a multi-agent microgrid energy management framework incorporating CNN-GRU for forecasting day-ahead market prices, PV generation, WT generation, and load demand. An accelerated ADMM algorithm is developed to coordinate decisions made by autonomous agents taking into account their information privacy.

The application of the proposed framework on the modified CIGRE low-voltage network shows the accuracy of the presented DNN-based forecaster. In some cases, the CNN-GRU method is at least 50% more accurate than several other methods such as 2D-CNN, GRU, and LSTM. In addition, comparing the results of the proposed energy management framework using forecast and real data verified that using the CNN-GRU method leads a set of results that are close to reality. Moreover, the developed accelerated ADMM-based algorithm converges faster than the standard ADMM and ATC.

The above investigations on the proposed MEMS strategy and DNN-based forecasting engines reveal that further explorations in the following directions would be worthwhile:

- Modeling the microgrid reconfiguration in the microgrid energy management.
- Developing forecasting engines with a high level of accuracy to minimize errors caused by the discrepancy between predictions and real data. To tackle this problem, one can implement the MEMS framework based on probability density forecasting and stochastic optimization.

Declaration of interests

None.

References

[1] Prinsloo G, Dobson R, Mammoli A. Synthesis of an intelligent rural village microgrid control strategy based on smartgrid multi-agent modelling and transactive energy management principles. Energy 2018;147:263–78.
[2] Sun H, Zhang B, Wang J, Sun D, Tong J, Moslehi K, Li F, Strunz K, Li F. Guest editorial distributed energy management systems. IEEE Transactions on Smart Grid 2016;7(2):971–3.
[3] Malekpour AR, Pahwa A. Stochastic networked microgrid energy management with correlated wind generators. IEEE Trans Power Syst 2017;32(5):3681–93.
[4] Shi W, Xie X, Chu CCP, Gadh R. Distributed optimal energy management in microgrids. IEEE Transactions on Smart Grid 2015;6(3):1137–46.
[5] Wang Z, Yang K, Wang X. Privacy-preserving energy scheduling in microgrid systems. IEEE Transactions on Smart Grid 2013;4(4):1810–20.
[6] Chen G, Yang Q. An admm-based distributed algorithm for economic dispatch in islanded microgrids. IEEE Transactions on Industrial Informatics 2018;14(9):3892–903.
[7] Zheng Y, Song Y, Hill DJ, Zhang Y. Multiagent system based microgrid energy management via asynchronous consensus admm. IEEE Trans Energy Convers 2018;33(2):886–8.
[8] Liu T, Tan X, Sun B, Wu Y, Tsang DH. Energy management of cooperative microgrids: a distributed optimization approach. Int J Electr Power Energy Syst 2018;96:335–46.
[9] de Azevedo R, Cintuglu MH, Ma T, Mohammed OA. Multiagent-based optimal microgrid control using fully distributed diffusion strategy. IEEE Transactions on Smart Grid 2017;8(4):1997–2008.
[10] Zhang Y, Fu L, Zhu W, Bao X, Liu C. Robust model predictive control for optimal energy management of island microgrids with uncertainties. Energy 2018;164:1229–41.
[11] Tabar VS, Jirdehi MA, Hemmati R. Energy management in microgrid based on the multi objective stochastic programming incorporating portable renewable energy resource as demand response option. Energy 2017;118:827–39.
[12] Shams MH, Shahabi M, Khodayar ME. Stochastic day-ahead scheduling of multiple energy carrier microgrids with demand response. Energy 2018;155:326–38.
[13] Elsied M, Oukaour A, Youssef T, Gualous H, Mohammed O. An advanced real time energy management system for microgrids. Energy 2016;114:742–52.
[14] Shen J, Jiang C, Liu Y, Wang X. A microgrid energy management system and risk management under an electricity market environment. IEEE Access 2016;4:2349–56.
[15] Wang Z, Chen B, Wang J, et al. Decentralized energy management system for networked microgrids in grid-connected and islanded modes. IEEE Transactions on Smart Grid 2016;7(2):1097–105.
[16] Xiang Y, Liu J, Liu Y. Robust energy management of microgrid with uncertain renewable generation and load. IEEE Transactions on Smart Grid 2016;7(2):1034–43.
[17] Guo Y, Zhao C. Islanding-aware robust energy management for microgrids. IEEE Transactions on Smart Grid 2018;9(2):1301–9.
[18] Zhang Y, Le J, Liao X, Zheng F, Li Y. A novel combination forecasting model for wind power integrating least square support vector machine, deep belief network, singular spectrum analysis and locality-sensitive hashing. Energy 2019;168:558–72.
[19] Liang Y, Niu D, Hong WC. Short term load forecasting based on feature extraction and improved general regression neural network model. Energy 2019;166:653–63.
[20] Sohn JM. Generation applications package for combined heat power in on-grid and off-grid microgrid energy management system. IEEE Access 2016;4:3444–53.
[21] Chaouachi A, Kamel RM, Andoulsi R, Nagasaka K. Multiobjective intelligent energy management for a microgrid. IEEE Trans Ind Electron 2013;60(4):1688–99.
[22] Palma-Behnke R, Benavides C, Lanas F, Severino B, Reyes L, Llanos J, Sáez D. A microgrid energy management system based on the rolling horizon strategy. IEEE Transactions on Smart Grid 2013;4(2):996–1006.
[23] Shi H, Xu M, Li R. Deep learning for household load forecasting—a novel pooling deep rnn. IEEE Transactions on Smart Grid 2018;9(5):5271–80.
[24] Teng W, Cheng H, Ding X, Liu Y, Ma Z, Mu H. Dnn-based approach for fault detection in a direct drive wind turbine. IET Renew Power Gener 2018;12(10):1164–71.
[25] Zhang S, Zhang S, Wang B, Habetler TG. Machine learning and deep learning algorithms for bearing fault diagnostics-a comprehensive review. 2019. arXiv preprint arXiv:1901.08247.

[26] Deep learning in neural networks: an overview. Neural Netw 2015;61:85–117.
[27] Alavi SA, Ahmadian A, Aliakbar-Golkar M. Optimal probabilistic energy management in a typical micro-grid based-on robust optimization and point estimate method. Energy Convers Manag 2015;95:314–25.
[28] Kingma DP, Ba J. Adam: a method for stochastic optimization. 2014. arXiv preprint arXiv:1412.6980.
[29] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.
[30] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th international conference on machine learning. ICML-10); 2010. p. 807–14.
[31] Kong W, Dong ZY, Hill DJ, Luo F, Xu Y. Short-term residential load forecasting based on resident behaviour learning. IEEE Trans Power Syst 2018;33(1):1087–8.
[32] Abdel-Hamid O, Mohamed Ar, Jiang H, Deng L, Penn G, Yu D. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing 2014;22(10):1533–45.
[33] Wang L, Scott KA, Xu L, Clausi DA. Sea ice concentration estimation during melt from dual-pol sar scenes using deep convolutional neural networks: a case study. IEEE Trans Geosci Remote Sens 2016;54(8):4524–33.
[34] Eckstein J. Parallel alternating direction multiplier decomposition of convex programs. J Optim Theory Appl 1994;80(1):39–62.
[35] Boyd S, Parikh N, Chu E, Peleato B, Eckstein J, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning 2011;3(1):1–122.
[36] Eckstein J, Bertsekas DP. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. Math Program 1992;55(1–3):293–318.
[37] Nishihara R, Lessard L, Recht B, Packard A, Jordan MI. A general analysis of the convergence of admm. 2015. arXiv preprint arXiv:1502.02009.
[38] Papathanassiou S, Hatziargyriou N, Strunz K, et al. A benchmark low voltage microgrid network. In: Proceedings of the CIGRE symposium: power systems with dispersed generation; 2005. p. 1–8.
[39] Chollet F, et al. Keras: deep learning library for theano and tensorflow. https://keras.io/k7(8); 2015.
[40] Grant M, Boyd S, Ye Y. Cvx: matlab software for disciplined convex programming. 2008.
[41] California independent system operator open acces same-time information system (oasis). [online]. available: http://oasis.caiso.com
[42] London datastore [online]. available: https://data.london.gov.uk/.
[43] Williams RJ, Peng J. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. Neural Comput 1990;2(4):490–501.
[44] Cao LJ, Tay FEH. Support vector machine with adaptive parameters in financial time series forecasting. IEEE Trans Neural Netw 2003;14(6):1506–18.
[45] Kargarian A, Mohammadi J, Guo J, Chakrabarti S, Barati M, Hug G, Kar S, Baldick R. Toward distributed/decentralized dc optimal power flow implementation in future electric power systems. IEEE Transactions on Smart Grid 2018;9(4):2574–94.