IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 5, 2018, accepted December 8, 2018, date of publication December 19, 2018, date of current version January 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2888839

Deep Learning-Based Sustainable Data Center Energy Cost Minimization With Temporal MACRO/MICRO Scale Management

DONG-KI KANG, EUN-JU YANG, AND CHAN-HYUN YOUN, (Member, IEEE)
School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Chan-Hyun Youn (chyoun@kaist.ac.kr)

This work was supported in part by the Cross-Ministry Giga KOREA Project grant through the Korean Government(MSIT) (Development of Mobile Edge Computing Platform Technology for URLLC Services) under Grant GK18P0400, and in part by the Institute for Information and Communications Technology Promotion (IITP) grant through the Korean Government(MSIT) (Service Mobility Support Distributed Cloud Technology) under Grant 2017-0-00294.

ABSTRACT Recently, distributed sustainable data centers based on renewable power generators have been deployed in order to efficiently reduce both the energy cost and carbon emission. Dynamic right-sizing (DRS) and frequency scaling (FS) have been considered as promising solutions to tune the computing capacity corresponding to the dynamic renewable power capacity. However, in existing works, the inaccurate power prediction and the uncoordinated decision making of DRS/FS still lead to low service quality and high energy cost. In this paper, we propose a novel joint optimization method for energy efficient distributed sustainable data centers. The proposed method adopts long short-term memory approach to improve the prediction accuracy of renewable power capacity for a long period, and unsupervised deep learning (DL) solver to resolve the coordinated DRS/FS optimization. Furthermore, we present the MACRO/MICRO (MAMI) time scale-based data center management technique to achieve both high energy efficiency and low wake-up transition overhead of DRS. To evaluate the proposed DL-based MAMI optimizer, we use the real trace data of renewable power capacity from the U.S. Measurement and Instrumentation Data Center. The experimental results demonstrate that our method reduces the energy cost by 25% compared with conventional metaheuristics while guaranteeing the service response delay requirement.

INDEX TERMS Energy efficiency, renewable energy sources, mathematical programming, unsupervised learning, economic forecasting.

I. INTRODUCTION

The modern geo-distributed sustainable data centers reduces the energy cost by using two renewable power sources, solar (or photovoltaic (PV)) power and the wind power [1]–[3]. In contrast to conventional grid power from coal and nuclear plants, sustainable power is (almost) free of fuel charge and does not generate the significant carbon emission. After the adoption of Paris climate assignment under the United Nations Framework Convention on Climate Change (UNFCCC) in Dec., 2015, countries encourage IaaS suppliers to operate their data centers based on renewable energy [4]. Facebook has depolyed multiple renewable energy generators in Los Lunas, New Mexico, Papillion, etc, to supply the energy requirement from geo-distributed data centers [5]. Apple has developed its own on-site renewable energy generators of solar, water, wind, and biogas fuel cells. They have reduced CO_2 emission by 590,000 metric tons [6].

The exploitation of the on-site renewable power generation should be carefully treated due to the associated intermittency and the uncertainty. The capacity of solar power depends on the solar radiation, and the wind power depends on the regional wind speed [7], [8]. Even in the same region, fluctuations in power capacity may arise due to changes in dynamic weather conditions [9]. The geo-distributed sustainable data centers need the adaptive request dispatching (RD, i.e., mapping of data centers to requests) corresponding to dynamic renewable power capacity, to maximize their energy efficiency. For power capacity prediction, previous papers have proposed various methods such as nonlinear autoregressive models with exogenous inputs (NARX) [10], estimated weighted moving average (EWMA) [11], and Kalman filter [12]. Even though such methods show good predictive performance however, they still exhibit low prediction accuracy for irregular renewable power curves during long period.

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Meanwhile, guaranteeing the service quality is also a critical concern for sustainable data centers. There are two widely used methods for energy efficient data centers, dynamic right sizing (DRS) [13] and CPU frequency scaling (FS) [14], [15]. In the DRS, the number of powered-up servers is adjusted proportionally to the amount of incoming workloads. The DRS removes the static power consumption of idle servers, and results in both the energy saving and the high resource utilization. The FS indirectly tunes the power supply for running servers. The FS can adaptively control the CPU frequency of servers based on time-varying workloads and power capacities and results in achieving the efficient dynamic power consumption. However, the coordinated management of DRS/FS is generally not easy to be designed due to following problems. First, while FS overhead is negligible, the server wake-up transition overhead by DRS actuation is serious [13]. Too frequent DRS triggering causes non-negligible long server downtime, additional energy consumption, and cooling delay [9], [16], [17]. Second, many previous works have not considered both the DRS and FS as decision variables simultaneously since the coordination of DRS/FS makes the optimization problem non-convex [18]–[20]. Obviously, if we manage the DRS and FS separately, then we fail to achieve a better energy efficient data center in views of static and dynamic power consumption of servers.

In this paper, we propose the novel joint optimization algorithm for energy efficient sustainable data centers. Our work has following contributions.

- The objective of our optimization method is to find the optimal decision of DRS, FS, and RD for geo-distributed sustainable data centers in order to minimize energy cost while guaranteeing the service latency bound.
- We propose the MACRO/MICRO (MAMI) time scale based management for energy efficient sustainable distributed data centers. In MACRO time scale, the entire decision variables of DRS, FS, and RD are optimized simultaneously over the long time period. In MICRO time scale, given the fixed DRS decision, the decision of FS and RD are re-optimized corresponding to the workload variation for the short-term period. This approach efficiently reduces the server wake-up transition overheads caused by frequent DRS actuation.
- We adopt recurrent neural network (RNN) based long short term memory (LSTM) method [21], which is the emerging artificial intelligent approach in order to improve the prediction accuracy for a long period. We present the LSTM input/output data structures dedicated to renewable power capacity prediction.
- We design the unsupervised deep learning (DL) solver for non-convex optimization problem. Due to the benefit of unsupervised learning (i.e., not requiring labeling data), we can utilize both the actual and synthetic data to train the deep neural network (DNN) model [22]. The proposed DL based MAMI optimizer ensures the rapid output derivation once the DNN model is trained, in contrast to conventional metaheuristics which require non-negligible re-compute time for each state.

In order to demonstrate the performance of the proposed DL based MAMI optimizer, we use real traces of solar radiation, outside air temperature, and wind speed at various geo-graphical locations, provided by the U.S. Measurement and Instrumentation Data Center (MIDC) [23]. Our MAMI optimizer integrating the LSTM predictor and the DL solver, outperforms conventional metaheuristics in views of both the energy cost and the service quaility ensurance of data centers. Without violating the service latency bound, it reduces the energy cost about 25% compared to genetic algorithm (GA) [24]. Furthermore, the proposed optimizer requires the acceptable computation time ($< 1s$) for DRS/FS/RD decision making while the GA requires the undesirable time ($> 1hr$) for each decision making step.

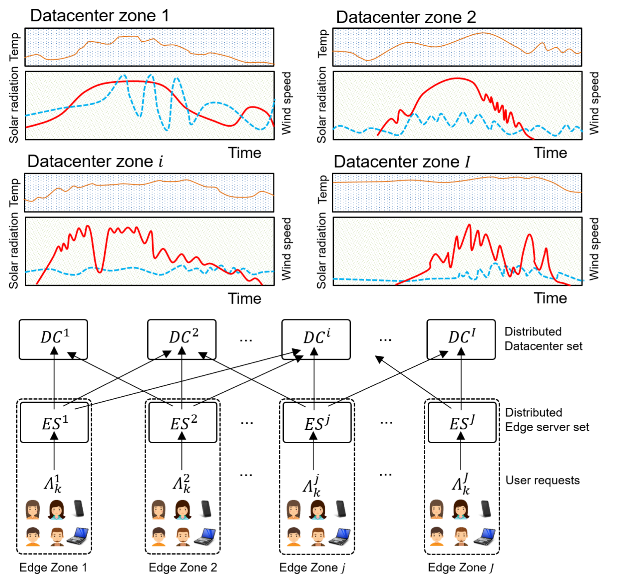


FIGURE 1. Fully connected structure of edge servers and geo-distributed sustainable data centers.

II. ENERGY COST MODEL FOR SUSTAINABLE DATA CENTERS WITH MULTIPLE EDGE SERVERS

We consider I sustainable data centers (DCs) with wind farms and solar panels, and J multiple edge servers. The edge servers (ESs) are scattered over multiple distributed edge zones which are populated areas (e.g., cities) to aggregate and transfer user requests to data centers [25], [26]. We assume that all the DCs and ESs are fully-connected each other [27]. This is shown in Fig.1. Let $[x] = \{1, \cdots, x\}$ denote the set of integer numbers which has x cardinality. Let T_c and K denote the considered continuous time length and the number of entire discrete time slots, respectively. The one time slot length is defined as $\delta = \frac{T_c}{K}$. Then the discrete time slot indices $k = 1, 2, \cdots, K$ are mapped to countinous time intervals $[0, \delta)$, $[\delta, 2\delta)$, $\cdots$, $[(K-1)\delta, K\delta)$, respectively.

The geo-distributed sustainable DCs get their power supply from two main electricity facilities, grid power and renewable power generators. Let sr_k^i, oat_k^i, and v_k^i denote the weather condition variables; solar radiation, outside air temperature (OAT), and the wind speed in the i-th DC at time slot k, respectively. Let spc_k^i and wpc_k^i denote the solar power and the wind power capacity based on the weather condition in the i-th DC at time slot k, respectively. Let m_k^i and f_k^i denote the number of powered-up servers and the CPU frequency value (the same value is set for all servers in DC) in the i-th DC at time slot k. Let Λ_k^j and λ_k^{ij} denote the number of user requests entered to the j-th ES and the number of the dispatched requests from the j-th ES to the i-th DC at time slot k, respectively. Let $\lambda_k^i = \sum_{j=1}^{J} \lambda_k^{ij}$ denote the sum of user requests from entire edge servers. This structure is shown in Fig.2.

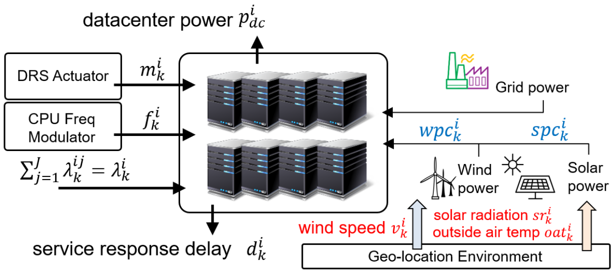


FIGURE 2. Dynamic Right Sizing (DRS) actuation, CPU Frequency Scaling (FS), and Request Dispatching (RD) for energy efficient sustainable data centers, with solar panels and wind farms.

A. RENEWABLE POWER CAPACITY MODEL

1) POWER CAPACITY OF SOLAR PANEL

The solar power is generated by sunlight through the solar panel converting sunlight into electrical power. The available solar power capacity spc_k^i is defined as follows [28]:

$$spc_k^i = \eta^i S^i sr_k^i (1 - 0.005(oat_k^i - 25)), \tag{1}$$

where S^i and η^i represent the size of panel area (m^2) and the solar panel conversion efficiency (%), respectively. sr_k^i and oat_k^i represent the amount of solar radiance (w/m^2) and the outside air temperature (C), respectively; and they might be varied according to the dynamic weather condition.

2) POWER CAPACITY OF WIND FARM

The amount of electricity generated by wind farms depends on the regional wind speed and the blade characteristics. The approximated wind power capacity wpc_k^i is defined as follows [29]:

$$wpc_k^i = \begin{cases} 0, & \text{if } v_k^i < v_{ci} \\ p_r * \dfrac{v_k^i - v_{ci}}{v_r - v_{ci}}, & \text{if } v_{ci} \leqslant v_k^i \leqslant v_r \\ p_r, & \text{if } v_r \leqslant v_k^i \leqslant v_{co} \\ 0, & \text{if } v_k^i > v_{co} \end{cases}, \tag{2}$$

where p_r, v_{ci}, v_r, v_{co} are rated electrical power (w/m), cut-in wind speed (m/s), rated wind speed (m/s), and cut-off wind speed (m/s), respectively. In particular, v_{ci} represents the wind speed at which the fan blades first start to rotate and produce the power (usually $3-5m/s$) [7], [30]. v_{co} represents the upper limit of the permissible wind speed to protect the turbine from the risk of damage due to strong wind speed (typically $25-30m/s$).

B. DATA CENTER POWER CONSUMPTION MODEL

Let u and f denote the average CPU utilization and the CPU frequency of the arbitrary server in the data center, respectively. According to [31], the server power consumption p based on u and f can be approximated as follows:

$$p(f, u) = \alpha_3 fu + \alpha_2 f + \alpha_1 u + \alpha_0. \tag{3}$$

These power model coefficients α_0, α_1, α_2, α_3 are derived from curve-fitting methods. Note that the relationship $u = \frac{\lambda}{f}$ holds where λ is the service arrival rate [31]. We assume that the user requests dispatched to the data center are evenly distributed to all the powered-up servers. That is, the average CPU utilization for single server in the i-th DC at time slot k is defined as $u_k^i = \frac{\lambda_k^i}{m_k^i f_k^i}$. Then, the total power consumption of the i-th DC at time slot k is defined as follows:

$$\begin{aligned} p_{dc}^i(m_k^i, f_k^i, \lambda_k^i) &= m_k^i(\alpha_3^i f_k^i \frac{\lambda_k^i}{m_k^i f_k^i} + \alpha_2^i f_k^i + \alpha_1^i \frac{\lambda_k^i}{m_k^i f_k^i} + \alpha_0^i) \\ &= \alpha_3^i \lambda_k^i + \alpha_2^i m_k^i f_k^i + \alpha_1^i \frac{\lambda_k^i}{f_k^i} + \alpha_0^i m_k^i. \end{aligned} \tag{4}$$

Note that the static power consumption $\alpha_0^i\, m_k^i$ can be reduced only by powering down the idle servers via DRS actuation, since the adjustment of f_k^i and λ_k^i affects only the dynamic power consumption.

C. DATA CENTER SERVICE RESPONSE LATENCY MODEL

For simplicity, we consider only the transactional workload as the type of user request in this paper. The performance of transactional workloads can be assesed by using the service response latency. We use $M/M/m$ queueing model to model the average service response latency for each DC. The average service response latency d^i for the i-th DC is defined as follows:

$$d^i(m_k^i, f_k^i, \lambda_k^i) = \frac{P_Q}{m_k^i f_k^i - \lambda_k^i} + \frac{1}{f_k^i}, \tag{5}$$

where the first term of Eq.(5) denotes the average queue waiting time, and the second term denotes the average service time. We simply assume that $P_Q = 1$ which represents the probability of requests waiting in the queue refer to [3] and [32].

D. DATA CENTER ENERGY COST MINIMIZATION PROBLEM FORMULATION

Based on the Eq.(1)-(5), the energy cost model for the i-th DC at time slot k is defined as follows:

$$C_k^i = d^i(m_k^i, f_k^i, \lambda_k^i) * (p_{dc}^i(m_k^i, f_k^i, \lambda_k^i) - spc_k^i - wpc_k^i)^+, \tag{6}$$

where $(x)^+ = \max(x, 0)$. For simplicity, the additional operation cost for solar panels and wind farms is assumed to be free of charge in this paper. If $spc_k^i + wpc_k^i \geqslant p_{dc}^i(m_k^i, f_k^i, \lambda_k^i)$, then the i-th DC is able to process the assigned user requests $\lambda_k^i = \sum_{\forall j \in [J]} \lambda_k^{ij}$ with zero energy cost. Otherwise, the excess energy consumption is supported by the grid power generator, and the energy cost increases as the the amount of excess energy consumption increases.

The objective of this paper is to find optimal decision for DC energy cost minmiziation. The decision vector contains the number of powered-up servers $\mathbf{m}$, the CPU frequency $\mathbf{f}$, and the request dispatching map $\boldsymbol{\lambda}$. The predicted renewable power capacity vectors $\mathbf{spc}$, $\mathbf{wpc}$ and user requests $\boldsymbol{\Lambda}$ are given over multiple time slots $\forall k \in [K]$. We unfold these vectors as follows:

- $\mathbf{m} = (m_1^1, \cdots, m_1^I, m_2^1, \cdots, m_K^I) \in \mathbb{R}^{IK}$
- $\mathbf{f} = (f_1^1, \cdots, f_1^I, f_2^1, \cdots, f_K^I) \in \mathbb{R}^{IK}$
- $\boldsymbol{\lambda} = (\lambda_1^{11}, \lambda_1^{12}, \cdots, \lambda_1^{1J}, \lambda_1^{21}, \lambda_1^{IJ}, \lambda_2^{11}, \cdots, \lambda_K^{IJ}) \in \mathbb{R}^{IJK}$
- $\mathbf{spc} = (spc_1^1, \cdots, spc_1^I, spc_2^1, \cdots, spc_K^I) \in \mathbb{R}^{IK}$
- $\mathbf{wpc} = (wpc_1^1, \cdots, wpc_1^I, wpc_2^1, \cdots, wpc_K^I) \in \mathbb{R}^{IK}$
- $\boldsymbol{\Lambda} = (\Lambda_1^1, \cdots, \Lambda_1^J, \Lambda_2^1, \cdots, \Lambda_K^J) \in \mathbb{R}^{JK}$

Now, we formulate the problem for the energy cost minimization of distributed data centers as follows:

Problem 1 (Data Center Energy Cost Minimization):

$$\min_{\mathbf{m},\mathbf{f},\boldsymbol{\lambda}} C := \sum_{k=1}^{K}\sum_{i=1}^{I} C_k^i + \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{j=1}^{J} \lambda_k^{ij} t^{ij} \rho^{ij}, \tag{7}$$

$$\text{subject to } \frac{1}{m_k^i f_k^i - \lambda_k^i} + \frac{1}{f_k^i} \leqslant \bar{d}, \quad \forall i \in [I],\ \forall k \in [K], \tag{8}$$

$$m_k^i f_k^i - \lambda_k^i > 0, \quad \forall i \in [I],\ \forall k \in [K], \tag{9}$$

$$\underline{m}^i \leqslant m_k^i \leqslant \overline{m}^i, \quad \forall i \in [I],\ \forall k \in [K], \tag{10}$$

$$\underline{f}^i \leqslant f_k^i \leqslant \overline{f}^i, \quad \forall i \in [I],\ \forall k \in [K], \tag{11}$$

$$\sum_{i=1}^{I} \lambda_k^{ij} = \Lambda_k^j, \quad \forall j \in [J],\ \forall k \in [K]. \tag{12}$$

Here, t^{ij} and ρ^{ij} are the trasfer time and the link power consumption per data unit of user requests from j-th ES to i-th DC, respectively. The constraint (8) indicates that the average service response latency can not exceed the predefined latency upper bound $\bar{d}$. The constraint (9) indicates that the number of user requests to dispatch can not exceed the available computing capacity of the destination DC. If it is violated, then the involved service response latency may be infinitely increased. The constraint (10) indicates that the number of powered-up servers should be adjusted within the number of total servers in the DC. The constraint (11) represents the available CPU frequency scaling range. The constraint (12) indicates that the user requests arrived at the certain ES should be entirely dispatched to distributed DCs.

Note that Problem 1 has two main challenges. First, too frequent variation of $\mathbf{m}$ may aggravate the DRS wake-up transition overheads [13], [16]. This indicates that the DC energy cost should be optimized with the minimum number of DRS triggering as possible. Second, the multiplication of $\mathbf{m}$ and $\mathbf{f}$ in the constraints (8), (9) makes the Problem 1 non-convex optimization. This prevents that the general convex optimizers solve the Problem 1. To overcome these issues, we propose the novel optimization method in next section.

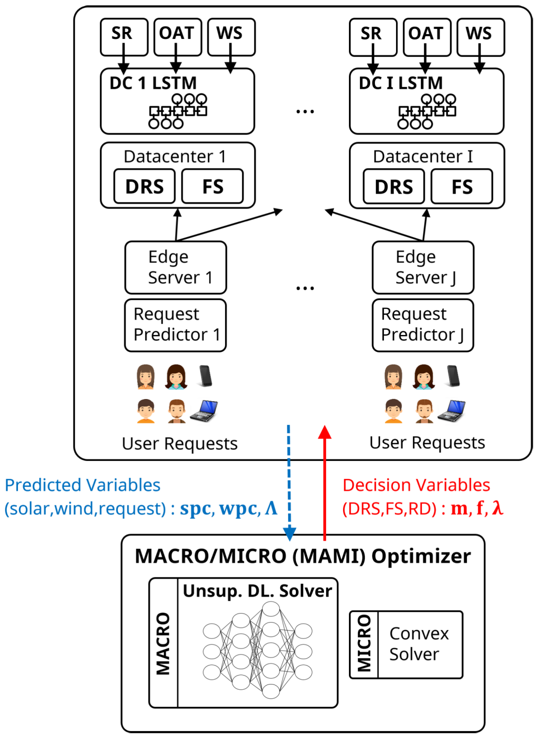


FIGURE 3. The structure of the proposed MACRO/MICRO (MAMI) optimizer, which consists of unsupervised deep learning (DL) solver (for MACRO time scale decision making) and convex solver (for MICRO scale decision making).

III. TEMPORAL MACRO/MICRO (MAMI) SCALE BASED DATA CENTER ENERGY COST OPTIMIZATION

In this section, we introduce our proposed deep learning (DL) based MACRO/MICRO (MAMI) time scale optimizer. Fig.3 shows the structure of the proposed MAMI optimzer for geo-distributed sustainable data centers. The MAMI optimizer consists of unsupervised DL solver (for MACRO time scale decision making) and the convex solver (for MICRO time scale decision making). The long short term memory (LSTM) based predictor in each DC reports predicted renewable power capacity sequence to the MAMI optimizer. The request predictor in each ES reports predicted user request sequence to the MAMI optimizer. We show a detailed explanation of the DL solver and the LSTM predictor in section 4.

The MAMI optimizer differentiates the time scale for the variable $\mathbf{m}$ that is sensitive to switching overhead and the variables $\mathbf{f}$ and $\boldsymbol{\lambda}$ that are not sensitive to switching overhead. On the MACRO time scale, the MAMI optimizer fulfills the decision making for optimal $\mathbf{m}'$, $\mathbf{f}$, and $\boldsymbol{\lambda}$. Here, $\mathbf{m}'$ is the new decision vector defined for the shortened time slots

$k' = 1, \cdots, K'$, and it replaces the original decision vector $\mathbf{m}$. On the MICRO time scale, given fixed $\mathbf{m}'$, the MAMI optimizer elaborately recalculates $\mathbf{f}$ and $\boldsymbol{\lambda}$ according to changes in solar power capacity, wind power capacity, and the user demands.

A. MACRO TIME SCALE DECISION MAKING

Let K' denote the number of MACRO time slots such that $K' < K$. Let $\delta' = q \cdot \delta$ denote the MACRO time scale sampling interval where q is the positive integer number. Then, the MACRO time slot indices $k' = 0, 1, \cdots, K'$ are mapped to $0, 1q, 2q, \cdots, K$. This relationship is shown in Fig.4. For the MACRO time scale decision making, we newly define $\mathbf{m}' = (m_1'^1, m_1'^2, \cdots, m_{K'}'^I) \in \mathbb{R}^{IK'}$ instead of using $\mathbf{m} \in \mathbb{R}^{IK}$. Then we formulate the MACRO time scale DC energy cost minimization problem as follows:

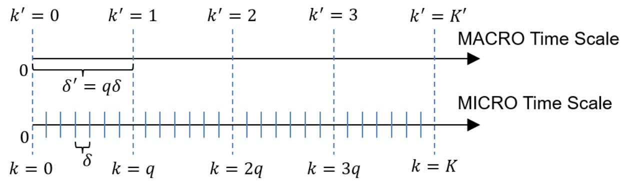


FIGURE 4. MACRO/MICRO (MAMI) time scale control period.

Problem 2 (MACRO Time Scale Data Center Energy Cost Minimization):

$$\min_{\mathbf{m}',\mathbf{f},\boldsymbol{\lambda}} C' := \sum_{k'=1}^{K'}\sum_{k=1}^{q}\sum_{i=1}^{I} C^i_{k',k} + \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{j=1}^{J} \lambda_k^{ij} t^{ij} \rho^{ij}, \quad (13)$$

$$\text{subject to } \frac{1}{m'^i_{k'} f^i_{qk'+k} - \lambda^i_{qk'+k}} + \frac{1}{f^i_{qk'+k}} \leqslant \bar{d},$$
$$\forall i \in [I], \quad \forall k \in [q], \ \forall k' \in [K'], \quad (14)$$
$$m'^i_{k'} f^i_{qk'+k} - \lambda^i_{qk'+k} > 0, \quad \forall i \in [I], \ \forall k \in [q], \ \forall k' \in [K'], \quad (15)$$
$$\underline{m}^i \leqslant m'^i_{k'} \leqslant \overline{m}^i, \quad \forall i \in [I], \ \forall k' \in [K'], \quad (16)$$
$$\underline{f}^i \leqslant f^i_{qk'+k} \leqslant \overline{f}^i, \quad \forall i \in [I], \ \forall k \in [q], \ \forall k' \in [K'], \quad (17)$$
$$\sum_{i=1}^{I} \lambda^{ij}_{qk'+k} = \Lambda^j_{qk'+k}, \quad \forall j \in [J], \ \forall k \in [q], \ \forall k' \in [K'], \quad (18)$$

where

$$C^i_{k',k} = d^i(m'^i_{k'}, f^i_{qk'+k}, \lambda^i_{qk'+k}) * (p^i_{dc}(m'^i_{k'}, f^i_{qk'+k}, \lambda^i_{qk'+k}) - spc^i_{qk'+k} - wpc^i_{qk'+k})^+.$$

As the size of q increases, the dimensionality of $\mathbf{m}'$ decreases. We can properly tune the DRS wake-up transition overhead by adjusting q. Note that replacing $\mathbf{m}$ using $\mathbf{m}'$ has two advantageous points compared to naively adding the equality constraints to the Problem 1. First, we decrease the computation complexity by reducing the dimensionality of the decision vector. Second, we avoid the additional computation burden for equality constraints.

Note that Problem 2 is the non-convex problem due to the constraints (14), (15) which include the multiplication of decision variables m and f. In section 4, we show that our unsupervised DL solver efficiently resolves this non-convex problem.

B. MICRO TIME SCALE DECISION MAKING

The prediction of renewable power capacity does not always match with the actual data due to the intermittent and fluctuated properties of the weather condition (e.g., abrupt clouds and winds, drastic drop of temperature). Although the MACRO time scale based decision making efficiently reduces the DRS wake-up overheads however, it does not cope well with the renewable power capacity fluctuation immediately due to the long sampling period of δ'.

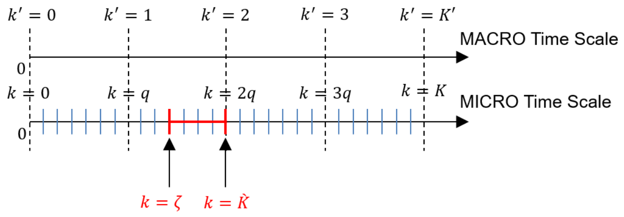


FIGURE 5. MICRO time scale partial optimization for $\grave{\mathbf{f}}$ and $\grave{\boldsymbol{\lambda}}$ during interval $[k = \zeta, k = \grave{K}]$ at $k = \zeta$.

The goal of the MICRO time scale based decision making is to re-calculate the sophisticated frequency scaling/request dispatching, so as to complement the MACRO time scale management. Suppose that the the nontrivial prediction error for renewable power capacity is identified at $k = \zeta$. In addition, let spc'^i_k and wpc'^i_k denote the newly predicted solar and wind power capacity for time slot k, respectively. This is shown in Fig.5. The MAMI optimizer partially recalculates $\mathbf{f}$ and $\boldsymbol{\lambda}$ corresponding to the renewaled prediction. To do this, we define the MICRO time scale based DC energy cost minimization problem as follows:

Problem 3 (MICRO Time Scale Data Center Energy Cost Minimization):

$$\min_{\grave{\mathbf{f}},\grave{\boldsymbol{\lambda}};\mathbf{m}'} C'' := \sum_{k=\zeta}^{\grave{K}}\sum_{i=1}^{I} C''^i_k + \sum_{k=\zeta}^{\grave{K}}\sum_{i=1}^{I}\sum_{j=1}^{J} \lambda_k^{ij} t^{ij} \rho^{ij}, \quad (19)$$

$$\text{subject to } \frac{1}{m''^i f^i_k - \lambda^i_k} + \frac{1}{f^i_k} \leqslant \bar{d}, \quad \forall i \in [I],$$
$$k = \zeta, \zeta+1, \cdots, \grave{K}, \quad (20)$$
$$m''^i f^i_k - \lambda^i_k > 0, \quad \forall i \in [I], \ k = \zeta, \zeta+1, \cdots, \grave{K}, \quad (21)$$
$$\underline{f}^i \leqslant f^i_k \leqslant \overline{f}^i, \quad \forall i \in [I], k = \zeta, \zeta+1, \cdots, \grave{K}, \quad (22)$$

$$\sum_{i=1}^{I} \lambda_k^{ij} = \Lambda_k^j, \quad \forall j \in [J], \qquad k = \zeta, \zeta+1, \cdots, \grave{K}, \quad (23)$$

where

$$\grave{K} = \zeta + q - mod(\zeta, q),$$
$$m''^i = m_k'^i, \quad k = \lceil \frac{\zeta}{q} \rceil,$$
$$\grave{\mathbf{f}} = (f_\zeta^1, \cdots, f_\zeta^I, f_{\zeta+1}^1, \cdots, f_{\grave{K}}^I) \in \mathbb{R}^{I \cdot (\grave{K}-\zeta)},$$
$$\grave{\boldsymbol{\lambda}} = (\lambda_\zeta^{11}, \cdots, \lambda_\zeta^{IJ}, \lambda_{\zeta+1}^{11}, \cdots, \lambda_{\grave{K}}^{IJ}) \in \mathbb{R}^{IJ \cdot (\grave{K}-\zeta)},$$
$$C_k''^i = d^i(m''^i, f_k^i, \lambda_k^i) * (p_{dc}^i(m''^i, f_k^i, \lambda_k^i) - spc_k'^i - wpc_k'^i)^+.$$

Here, mod(a, b) represents the remainder after a is divided by b. The number of powered-up servers is fixed as constant m''^i derived from the optimal decision variable for the MACRO time scale problem. The MICRO time scale Problem 3 is resolved over the partial time slots from the current time slot index $k = \zeta$ to the next MACRO time slot index $k = \grave{K}$. At every MACRO time slot, the MICRO time scale optimization can be iteratively conducted corresponding to the prediction error of renewable power capacity.

Contrary to the Problem 2, the MICRO time scale Problem 3 can be resolved by using the standard convex optimizer, since m''^i is not the decision variable. Therefore, our DL solver only focuses on the MACRO time scale Problem 2.

IV. UNSUPERVISED DEEP LEARNING SOLVER WITH THE LONG SHORT TERM MEMORY PREDICTOR

In this section, we explain the unsupervised deep learning (DL) solver for non-convex MACRO time scale Problem 2. Furthermore, we introduce the long short term memory (LSTM) predictor to foresee the future renewable power capacity sequence.

A. LONG SHORT TERM MEMORY PREDICTOR FOR RENEWABLE POWER CAPACITY

The structure of the LSTM network model is shown in the left part of Fig.6. The notations $\mathbf{u}_t$, $\mathbf{h}_t$, $\mathbf{c}_t$, $\mathbf{y}_t$ denote input data block, hidden state block, cell state block, and output response block at training time step t, respectively. The output from hidden state $\mathbf{h}_t$ is derived from the non-linear hyperbolic tangent function $tanh(\mathbf{x}) = \frac{sinh(\mathbf{x})}{cosh(\mathbf{x})} = \frac{e^{2\mathbf{x}}-1}{e^{2\mathbf{x}}+1}$. The backpropagation of LSTM updates the weight parameters between each block based on the estimation error of output response sequence compared to the actual one. For details, see [33].

Obviously, the weather condition shows date and time-related periodical patterns (e.g., as shown in Fig.8). We define the date variable as the integer number $dt \in [1, 365]$ (days) and the time variable as one within $hr \in [1, 24]$ (hours). The input data sequence with length L is defined as follows:

$$(\mathbf{u}_t^i, \mathbf{u}_{t+1}^i, \cdots, \mathbf{u}_{t+L}^i), \quad (24)$$

where

$$\mathbf{u}_t^i = (dt_t^i,\ hr_t^i,\ spc_t^i,\ wpc_t^i).$$

We also use L for the length of the predicted output responsce sequence. Then, the output response sequence is defined as follows:

$$(\mathbf{y}_{t+L+1}^i, \mathbf{y}_{t+L+2}^i, \cdots, \mathbf{y}_{t+2L}^i), \quad (25)$$

where

$$\mathbf{y}_t^i = (spc_t^i,\ wpc_t^i).$$

After $\mathbf{h}^i$ is updated, the predicted output response sequence $(\mathbf{y}_{t+L+1}^i, \cdots, \mathbf{y}_{t+2L}^i)$ is compared to the measured one $(\check{\mathbf{y}}_{t+L+1}^i, \cdots, \check{\mathbf{y}}_{t+2L}^i)$ where $\check{\mathbf{y}}_{t+L+l}^i$ is the measured output data at training time step $t + L + l$. The backpropagation is conducted based on the estimation error over the training time slots $t, \cdots, t + L$, and entire network weight parameter matrices are updated. When closing the enough accuracy level, the training of LSTM network model is done.

Each DC conducts the inferencing for LSTM network model. At time slot k, all the DCs send the predicted renewable power capacity sequence with $L = K$ to the MAMI optimizer. If the non-negligible prediction error is identified in i-th DC at certain time slot $k = \zeta$, then the LSTM based predictor re-trains the network model and the i-th DC sends the updated output response to the MAMI optimizer. The MICRO scale solver (i.e., convex optimizer) resolves the Problem 3 with updated prediction data of renewable power capacity.

B. UNSUPERVISED DEEP LEARNING BASED SOLVER FOR MACRO TIME SCALE OPTIMIZATION

The goal of our unsupervised DL solver is to find the (approximated) optimal decision vector for the non-convex MACRO time scale Problem 2. To do this, we design the customized loss function for the DNN model training. The conventional loss function structures for supervised and unsupervised DL method are defined as follows [22], [34]:

$$E_{N_{mb}}^s(\mathbf{w}) = \frac{1}{N_{mb}} \sum_{l=1}^{N_{mb}} \theta(\check{\mathbf{z}}_{(l)}, \mathbf{z}_{(l)} = DNN(\mathbf{v}_{(l)}; \mathbf{w})). \quad (26)$$

$$E_{N_{mb}}^u(\mathbf{w}) = \frac{1}{N_{mb}} \sum_{l=1}^{N_{mb}} \theta'(\mathbf{z}_{(l)} = DNN(\mathbf{v}_{(l)}; \mathbf{w})). \quad (27)$$

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \alpha_{t-1} \nabla_{\mathbf{w}} E_{N_{mb}}(\mathbf{w}_{t-1}). \quad (28)$$

Here, $\mathbf{w}, \theta, \theta', E_{N_{mb}}^s$, and $E_{N_{mb}}^u$ represent the network weight parameters, the supervised DNN loss function, unsupervised DNN loss function, average value of supervised DNN loss function, and the average value of unsupervised DNN loss function, respectively. N_{mb} and DNN represent the minibatch size and the DNN model, respectively. $\mathbf{v}_{(l)}$, $\check{\mathbf{z}}_{(l)}$, $\mathbf{z}_{(l)}$ represent the input data, the label data corresponding to $\mathbf{v}_{(l)}$, and the estimated output data corresponding to $\mathbf{v}_{(l)}$, respectively. t and α represent the DNN training time slot index and the learning rate, respectively. Note that Eq.(28) represents the gradient based formula for updating network weight parameters.

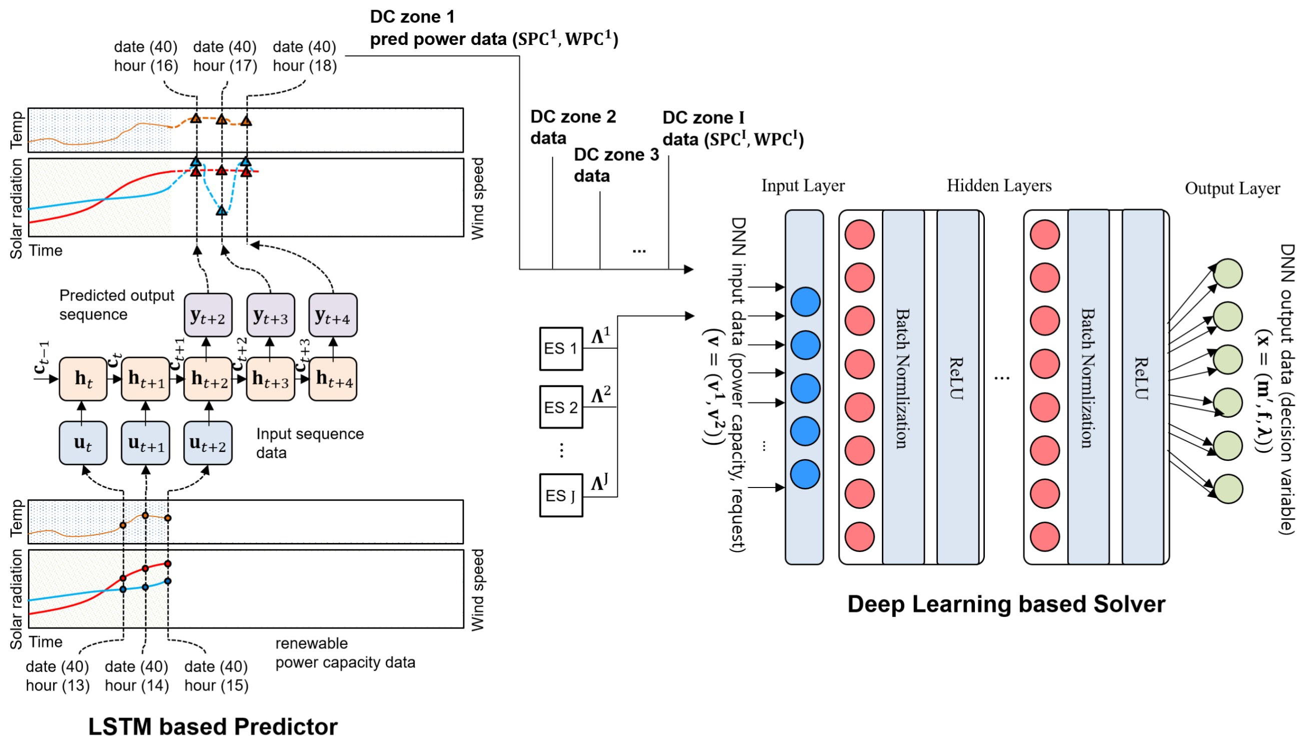


FIGURE 6. Proposed structure of unsupervised DL solver for MACRO Time Scale Problem and LSTM based renewable power capacity predictor.

In particular, we adopt the Eq.(27) to our proposed DL solver. In order to find the feasible solution for the MACRO time scale Problem 2, we add the simple penalty term Φ, which penalizes the violation of constraints (14) - (18). Consequently, we define our unsupervised DNN loss function as follows:

$$\theta'(\mathbf{v};\boldsymbol{\phi}) = C'(\mathbf{x} = DNN(\mathbf{v};\mathbf{w})) + \Phi(\mathbf{x}), \tag{29}$$

where

$$\begin{aligned}\Phi(\mathbf{x}) = &\sum_{k'=1}^{K'}\sum_{k=1}^{q}\sum_{i=1}^{I}\phi_d\Big(\frac{1}{m'^i_{k'}f^i_{qk'+k}-\lambda^i_{qk'+k}}+\frac{1}{f^i_{qk'+k}}-\bar{d}\Big)^+\\ &-\sum_{k'=1}^{K'}\sum_{k=1}^{q}\sum_{i=1}^{I}\phi_{posi}(m'^i_{k'}f^i_{qk'+k}-\lambda^i_{qk'+k})^-\\ &+\sum_{k'=1}^{K'}\sum_{k=1}^{q}\sum_{i=1}^{I}\{\phi_{\underline{f}}(\underline{f}^i-f^i_{qk'+k})^+ + \phi_{\overline{f}}(f^i_{qk'+k}-\overline{f}^i)^+\}\\ &+\sum_{k'=1}^{K'}\sum_{i=1}^{I}\{\phi_{\underline{m}}(\underline{m}^i-m^i_{k'})^+ + \phi_{\overline{m}}(m^i_{k'}-\overline{m}^i)^+\}\\ &+\sum_{k=1}^{K}\sum_{j=1}^{J}\phi_\lambda\Big|\sum_{i=1}^{I}\lambda^{ij}_k-\Lambda^j_k\Big|,\end{aligned}$$

and $\boldsymbol{\phi} = (\phi_d, \phi_{posi}, \phi_{\underline{f}}, \phi_{\overline{f}}, \phi_{\underline{m}}, \phi_{\overline{m}}, \phi_\lambda)$ is the set of penalty weight values for contraint violation. Here, $(x)^- = \min(x, 0)$.

The inferencing input data $\mathbf{v}$ and the inferencing output data $\mathbf{x}$ in our DNN loss function are defined as follows:

- $\mathbf{v}^1 = (\mathbf{y}^1_{t+L+1}, \cdots, \mathbf{y}^I_{t+L+1}, \mathbf{y}^1_{t+L+2}, \cdots, \mathbf{y}^I_{t+2L}) \in \mathbb{R}^{2IL}$
- $\mathbf{v}^2 = (\Lambda^1_{t+L+1}, \cdots, \Lambda^J_{t+L+1}, \Lambda^1_{t+L+2}, \cdots, \Lambda^J_{t+2L}) \in \mathbb{R}^{JL}$
- $\mathbf{v} = (\mathbf{v}^1, \mathbf{v}^2) \in \mathbb{R}^{2IL+JL}$
- $\mathbf{x} = (\mathbf{m}', \mathbf{f}, \boldsymbol{\lambda}) \in \mathbb{R}^{IK'+IK+IJK}$

For DNN training, the MAMI optimizer collects the historical data of renewable power capacity and user requests. If we do not have enough training dataset, then we can use the synthetic dataset which is randomly generated according to the predetermined distribution within the available ranges. The structure of the unsupervised DL solver is shown in the right part of Fig.6. The batch normalization technique [35] is applied to each hidden layer to avoid the gradient vanishing and exploding problems. We use the well-known rectified linear unit (ReLU) [36], [37] as the activation function for each neuron in our DNN model.

Algorithm 1 describes the MACRO time scale DC energy cost minimization procedures. The MAMI optimizer gathers the predicted renewable power capacity $\mathbf{v}^1 = (\mathbf{y}^1_{t+L+1}, \cdots, \mathbf{y}^I_{t+2L})$ from LSTM predictor in each DC $i = 1, \cdots, I$ (line 01). The MAMI optimizer gathers the amount of predicted user requests $\mathbf{v}^2 = (\Lambda^1_{t+L+1}, \cdots, \Lambda^J_{t+2L})$ from each ES $j = 1, \cdots, J$ (line 02). The input data vector $\mathbf{v}$ for DNN inferencing is generated resulting from integration of $\mathbf{v}^1$ and $\mathbf{v}^2$ (line 03). The decision variable $\mathbf{x} = (\mathbf{m}', \mathbf{f}, \boldsymbol{\lambda})$ is derived from inferencing the DNN model DNN (line 04).

Algorithm 1 MACRO Time Scale DC Energy Cost Minimization

```
INPUT
Trained DNN model $DNN$ with weight parameters $\mathbf{w}$
OUTPUT
MACRO time scale optimal decision $\mathbf{x} = (\mathbf{m}', \mathbf{f}, \boldsymbol{\lambda})$
01 : gather predicted renewable power capacity from LSTM predictor in each DC $1, \cdots, I$,
        $\mathbf{v}^1 = (\mathbf{y}^1_{t+L+1}, \cdots, \mathbf{y}^I_{t+2L})$
02 : gather predicted user requests from edge servers (ESs),
        $\mathbf{v}^2 = (\Lambda^1_{t+L+1}, \cdots, \Lambda^J_{t+2L})$
03 : generate DNN input data vector $\mathbf{v} = (\mathbf{v}^1, \mathbf{v}^2)$
04 : $\mathbf{x} = (\mathbf{m}', \mathbf{f}, \boldsymbol{\lambda}) \longleftarrow DNN(\mathbf{v}; \mathbf{w})$
05 : return $\mathbf{x}$
```

Algorithm 2 MICRO Time Scale DC Energy Cost Minimization

```
INPUT
Starting MICRO time slot index, $k = \zeta$
Number of powered-up servers $m''^1, \cdots, m''^I$ by Eq.(24)
Newly predicted solar power capacity
    $spc'^i_k, k = \zeta, \cdots, \grave{K}, \forall i \in [I]$
Newly predicted wind power capacity
    $wpc'^i_k, k = \zeta, \cdots, \grave{K}, \forall i \in [I]$
OUTPUT
MICRO time scale partial optimal decision $\grave{\mathbf{f}}, \grave{\boldsymbol{\lambda}}$
01 : build $C''$ by using Eq.(19)
02 : $\grave{\mathbf{f}}, \grave{\boldsymbol{\lambda}} \longleftarrow$ solving MICRO time scale Problem 3
03 : return $\grave{\mathbf{f}}, \grave{\boldsymbol{\lambda}}$
```

TABLE 1. Parameters for data center energy cost minimization problem.

Notation	Values
number of DCs I	3
number of ESs J	5
MACRO / MICRO sampling period	$K' = 1, K = 12$
bound for powered-up servers $\underline{m}$, $\overline{m}$	$1, 20\ (\times 10^3)$
bound for CPU frequency $\underline{f}$, $\overline{f}$	0.5, 4 (Ghz)
transfer time per request t	0.1 (sec)
transfer power per request ρ	1 (W)
required frequency per request	1 (Ghz)
power coefficients $\alpha_3, \alpha_2, \alpha_1, \alpha_0$	$30, 100, 10, 1.2$
penalty factors $\phi_{\underline{m}}, \phi_{\overline{m}}, \phi_{\underline{f}}, \phi_{\overline{f}}, \phi_d, \phi_{posi}$	10^3
penalty factor ϕ_λ	10^5

Algorithm 2 describes the MICRO time scale DC energy cost minimization procedures. This algorithm is triggered whenever the significant prediction error for renewable power capacity is identified. The MICRO time scale cost function C'' is built based on the error identification point $k = \zeta$, the number of powered-up servers resulting from MACRO time scale optimization $m''^1, \cdots, m''^I$, and the newly predicted solar/wind power capacity $spc'^i_k, wpc'^i_k, k = \zeta, \cdots, \grave{K}$, $\forall i \in [I]$ (line 01). The partial optimal solutions $\grave{\mathbf{f}}$ and $\grave{\boldsymbol{\lambda}}$ over $k = \zeta, \cdots, \grave{K}$ are re-calculated via the standard convex optimizer (line 02).

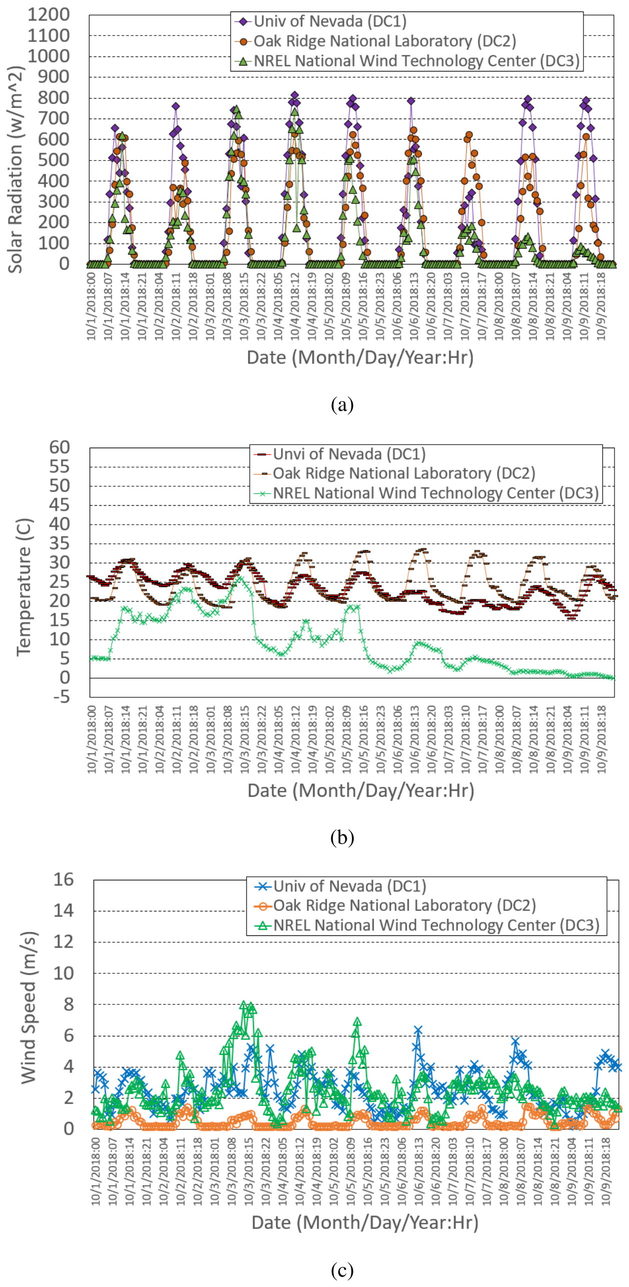
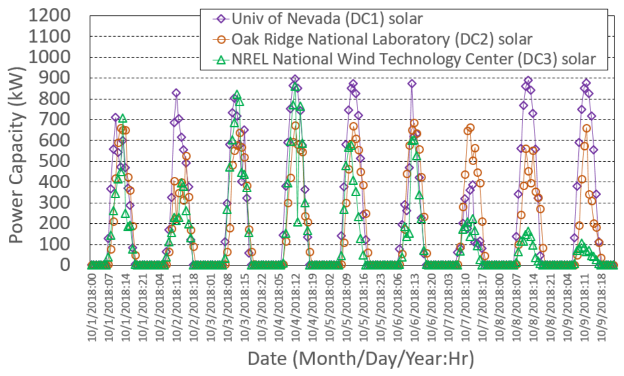


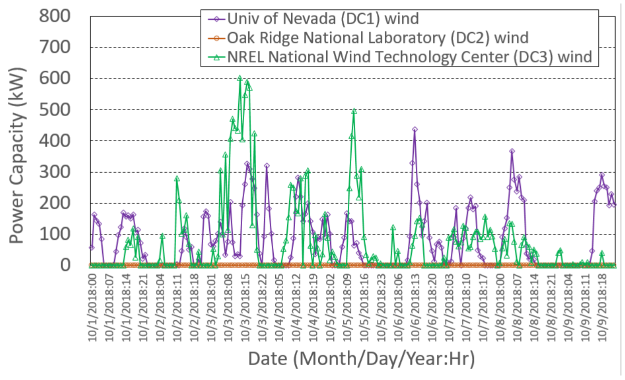
FIGURE 7. Curves of solar radiation, outside air temperatue, and wind speed during 2018.10.01-2018.10.08 (8days) at DC1 (Univ of Nevada), DC2 (Oak Ridge National Lab), and DC3 (NREL National Wind Technology Center, M2) [23]. (a) Actual solar radiation sr^1, sr^2, sr^3 (w/m^2). (b) Actual outside air temperature (OAT) oat^1, oat^2, oat^3 (C). (c) Actual wind speed v^1, v^2, v^3 (m/s).

V. EXPERIMENTAL RESULTS

In our experiment, we deploy the physical machine with CPU i7-3770, gtx1080 (8GB), memory 16GB in order to train the LSTM model and the DNN model. We use CUDA 8.0 and cuDNN 5.0 [38] to accelerate the DNN training speed based on GPU. We use Keras 2.0.8 [39] APIs with Tensorflow framework 1.4 [40] in order to design the customized model structure of LSTM and DNN. To verify the performance of the proposed MAMI optimizer, we use real trace data of solar



(a)

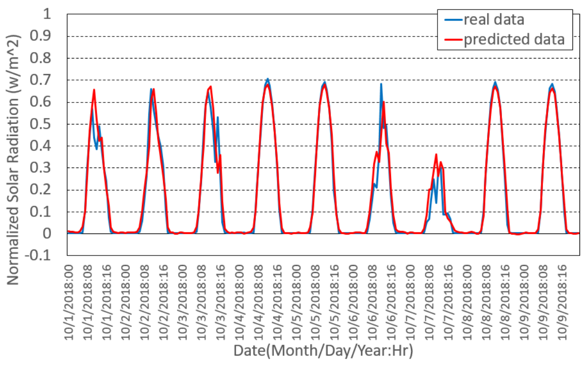


(b)

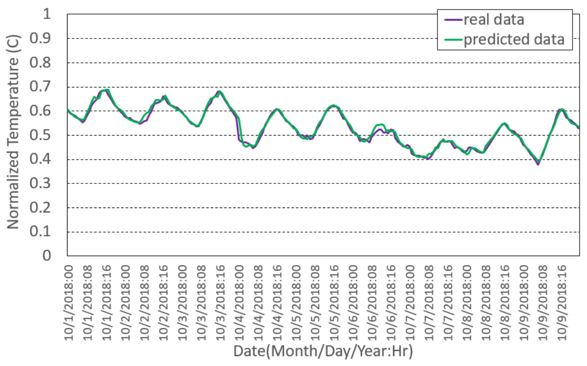
FIGURE 8. Estimated curves of solar power capacity and wind power capacity during 2018.10.01-2018.10.08 (8days) at DC1 (Univ of Nevada), DC2 (Oak Ridge National Lab), and DC3 (NREL National Wind Technology Center, M2) [23]. (a) Available solar power capacity spc^1, spc^2, spc^3 (kW). (b) Available wind power capacity wpc^1, wpc^2, wpc^3 (kW).

TABLE 2. LSTM / DNN training parameters.

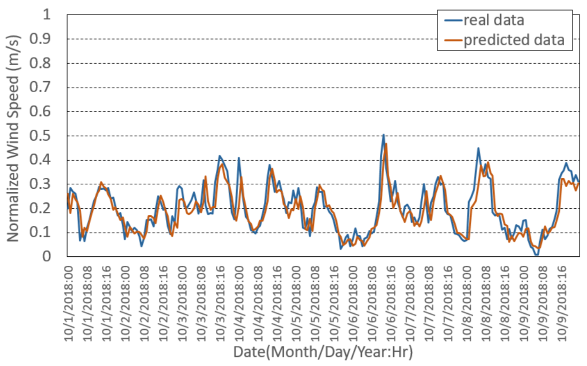
Notation	Values
number of training samples	6×10^4
number of test samples	9×10^3
LSTM batch-size	30
LSTM maximum epochs	100
LSTM hidden size	20
LSTM activation, optimizer, loss	$tanh$, $Adam$, MSE
LSTM timestep L	12
DNN batch-size	32
DNN maximum epochs	400
DNN models	1 FN - 1 BN - 1 ReLU
DNN activation, optimizer, loss	$ReLU$, $Adadelta$, θ'
DNN input dim, output dim	132, 219



(a)



(b)

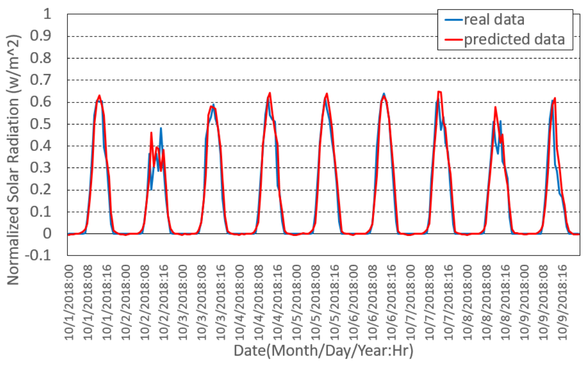


(c)

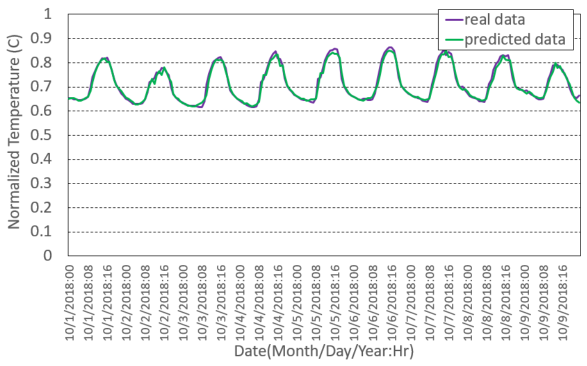
FIGURE 9. Prediction results for normalized solar radiation, outside air temperature (OAT), and wind speed during 2018.10.01-2018.10.08 (8days) at DC1 (Univ of Nevada) by the LSTM predictor. (a) Normalized solar radiation at DC1, sr^1. (b) Normalized OAT at DC1, oat^1. (c) Normalized wind speed at DC1, v^1.

radiation, outside air temperature (OAT), and wind speed from geo-distributed DC regions in U.S [23]. We consider three regions: DC1-University of Nevada, Las Vegas (Paradise, Nevada); DC2-Oak Ridge National Labotary (Eastern Tennessee); DC3-NREL National Wind Technology Center, M2 (Boulder, Colorado).

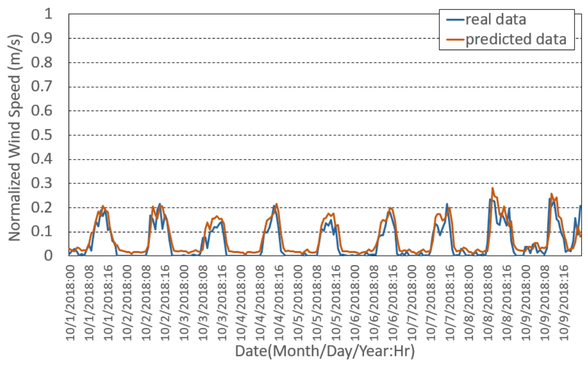
Fig.7 and Fig.8 show the curves of solar radiation, OAT, wind speed, solar power capacity, and wind power capacity during 2018.10.01-2018.10.08 (8days) at DC1 (Univ of Nevada), DC2 (Oak Ridge National Lab), and DC3 (NREL National Wind Technology Center, M2). As shown in Fig.7-(a) and Fig.8-(a), the available solar power capacity strongly depends on the amount of solar radiation (the OAT influences are low). Similarly, the available wind power capacity is proportional to the wind speed as shown in Fig.7-(c) and Fig.8-(b). Note that the solar power capacity curve shows the regular patterns over time of the day. Contrary to DC1 and DC3, the available wind power capacity of DC2 is quite low due to the slow wind speed. This indicates that the accurate prediction for the solar power



(a)

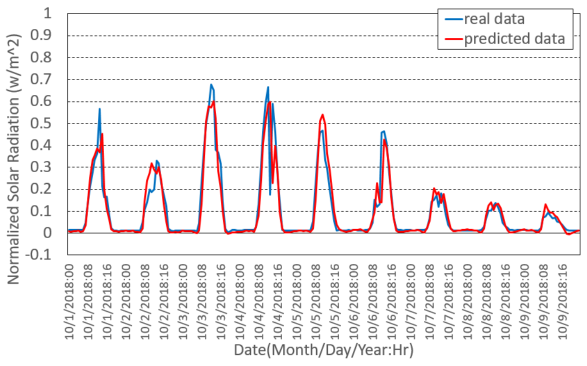


(b)

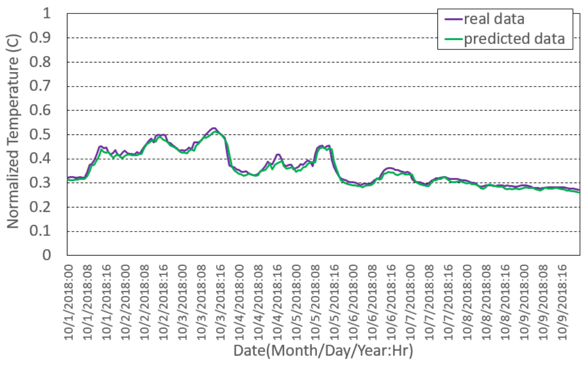


(c)

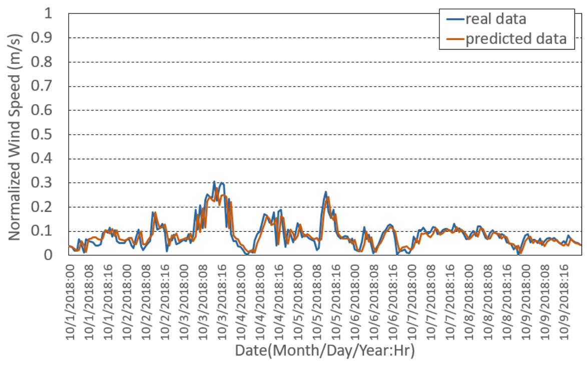
FIGURE 10. Prediction results for normalized solar radiation, outside air temperature (OAT), and wind speed during 2018.10.01-2018.10.08 (8days) at DC2 (Oak Ridge National Laboratory) by the LSTM predictor. (a) Normalized solar radiation at DC2, sr^2. (b) Normalized OAT at DC2, oat^2. (c) Normalized wind speed at DC2, v^2.



(a)



(b)



(c)

FIGURE 11. Prediction results for normalized solar radiation, outside air temperature (OAT), and wind speed during 2018.10.01-2018.10.08 (8days) at DC3 (NREL National Wind Technology Center, M2) by the LSTM predictor. (a) Normalized solar radiation at DC3, sr^3. (b) Normalized OAT at DC3, oat^3. (c) Normalized wind speed at DC3, v^3.

capacity is relatively easy. Meanwhile, the curve of the wind power capacity depicts the irregular and intermittent patterns (Fig.8-(b)). This indicates that the prediction for the wind power capacity may show the lower accuracy than the solar power capacity.

The Table 1 presents the associated parameters for the energy cost minimization problem. We consider 3 geo-distributed DCs (DC1, DC2, and DC3) and 5 ESs. For simplicity, we assign the same bound of powered-up servers, bound of CPU frequency, and power coefficients for all the DCs. Every communication links between DCs and ESs have equal transfer time and power consumption per an user request. Note that we should define the proper penalty weight values for each constraint term in Eq.(30) to accurately find the feasible solution. We assign the same penalty weight values ($=10^3$) to all the constraint terms except for request dispatching constraint ($=10^5$). This differentiated assignment attempts to avoid the rejection of entered user requests, as much as possible.

Table 2 presents the associated parameters for LSTM/DNN training. We retreive the training input dataset including hourly solar radiation, OAT and wind speed data from

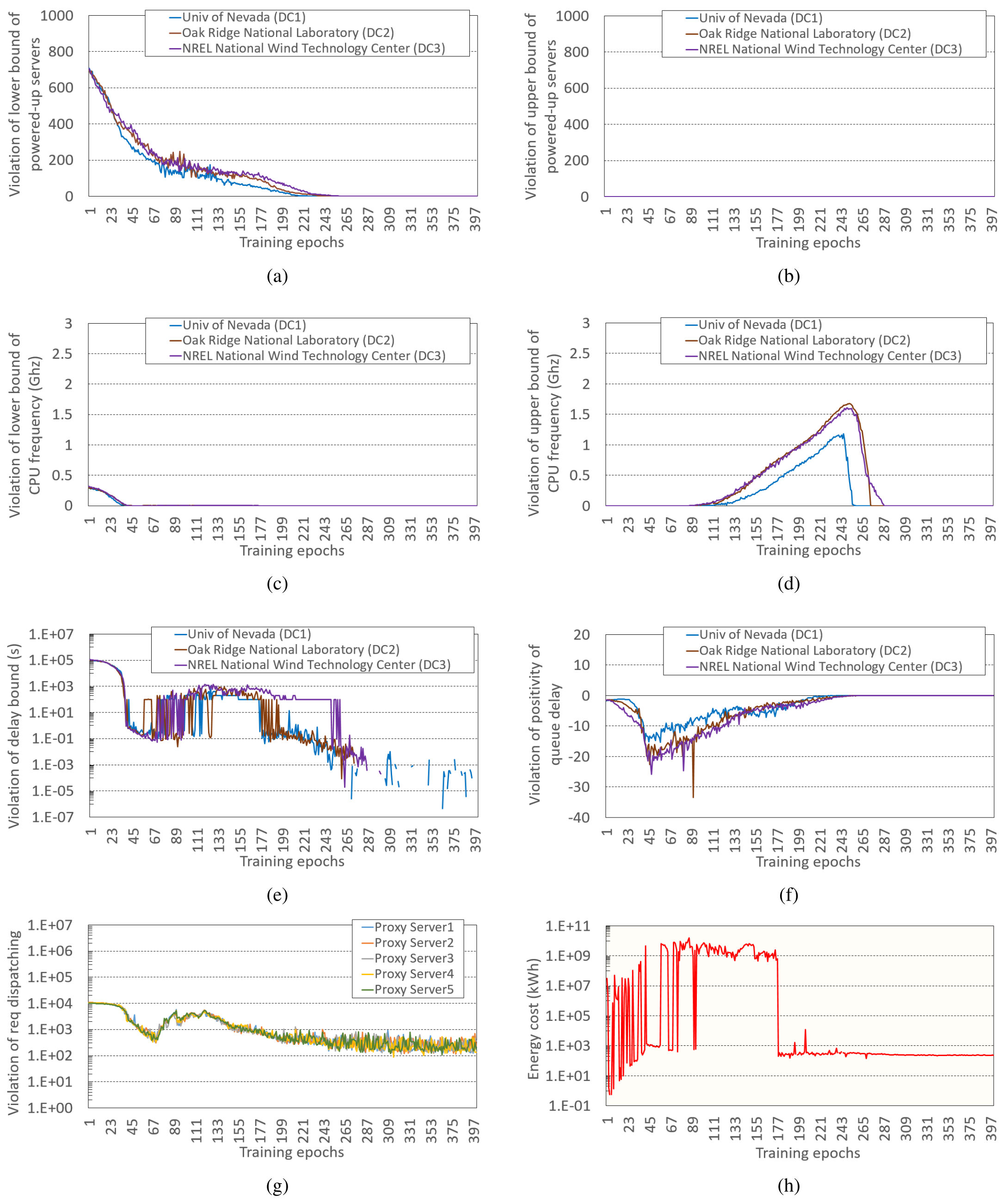


FIGURE 12. Experimental Results of DNN training progress with $\bar{d} = 0.7sec$ for MACRO time scale problem optimization. The average degree of violation for all constraints involved in the MACRO time scale problem asymptotically decreases as the DNN training progresses. The average energy cost approaches the (approximated) optimal value 240kWh as the DNN training progresses. (a) Average violation of $\underline{m}^i$: $(\underline{m}^i - m^i)^+$. (b) Average violation of $\overline{m}^i$: $(m^i - \overline{m}^i)^+$. (c) Average violation of $\underline{f}^i$: $(\underline{f}^i - f^i)^+$. (d) Average violation of $\overline{f}^i$: $(f^i - \overline{f}^i)^+$. (e) Average violation of $\bar{d}$: $(d^i - \bar{d})^+$. (f) Average violation of queue delay positivity: $(m^i * f^i - \lambda^i)^-$. (g) Average violation of request dispatching: $|\sum_{i=1}^{I} \lambda^{ij} - \Lambda^j|$. (h) Average energy cost: C'.

DC1-DC3 during 2010.10.08 - 2018.10.09 (8years). The total number of data is about 7×10^5. We split the entire dataset into two groups, training dataset (6×10^5) and test dataset (9×10^4) for training validation. We set the maximum epochs for LSTM training as 100, however the early-stopping is possible when the LSTM loss function value reaches the predefined accuracy level. The associated SGD optimizer for LSTM is *Adam*, which exploits the adaptive estimates of lower-order moments for stochastic gradient based optimization [41]. We adopt *mean squared error* (MSE) as the LSTM loss function [21], [42]. To foresee the hourly renewable power capacity for a half day, we set the LSTM timestep length as $L = 12$.

For DNN training, we use the training datset which is composed of historical renewable power capacity, and the historical amount of user requests. For renewable power capacity

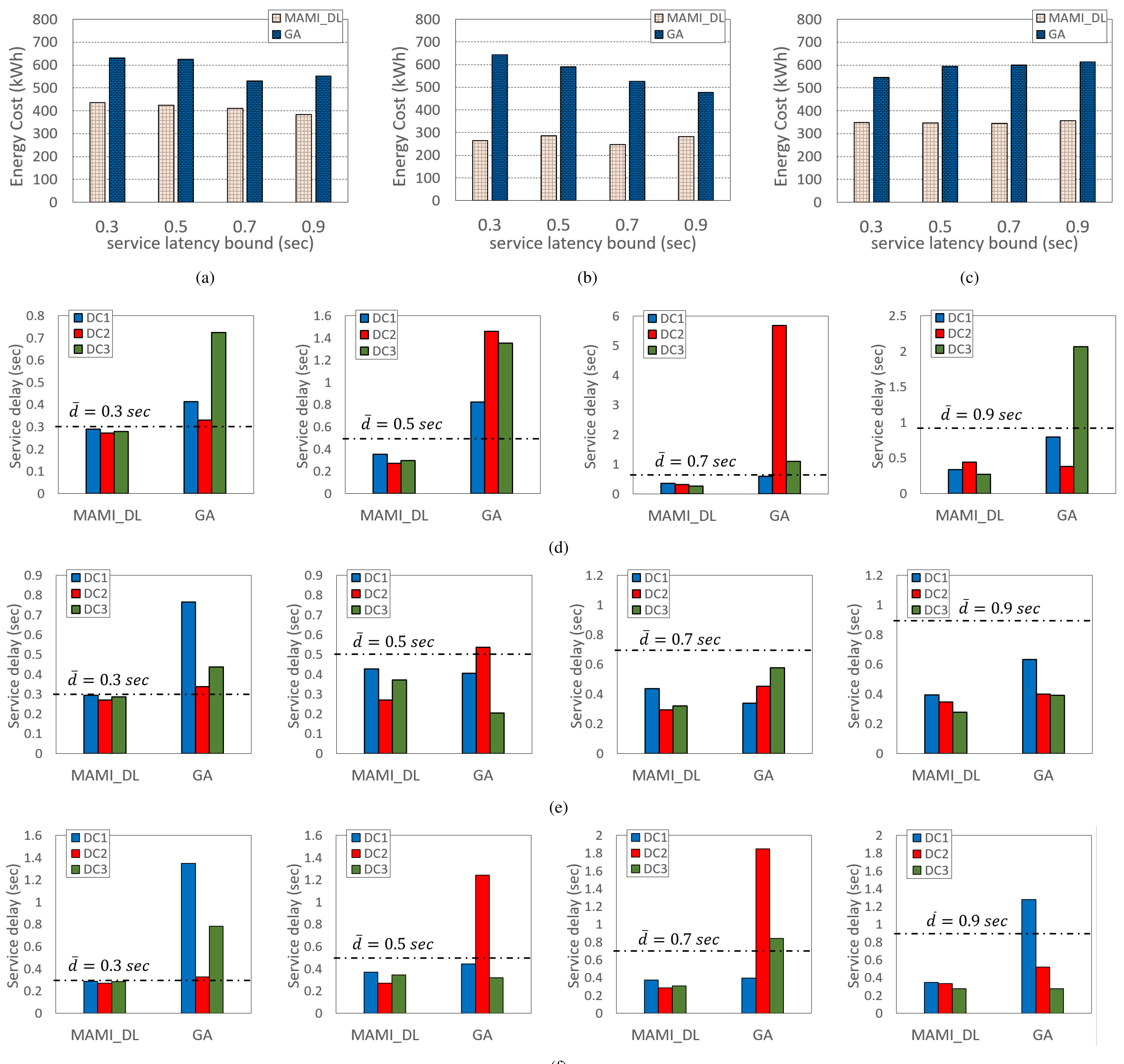


FIGURE 13. Comparision results of proposed DL based MAMI optimizer and the rank based genetic algorithm [24] in temrs of energy cost (kWh) and the service response latency (sec). (a) Energy cost at 2018.10.10 (case1). (b) Energy cost at 2018.10.15 (case2). (c) Energy cost at 2018.10.20 (case3). (d) Service response latency with $\bar{d} = 0.3s,\ 0.5s,\ 0.7s,\ 0.9s$ at 2018.10.10 (case1). (e) Service response latency with $\bar{d} = 0.3s,\ 0.5s,\ 0.7s,\ 0.9s$ at 2018.10.15 (case2). (f) Service response latency with $\bar{d} = 0.3s,\ 0.5s,\ 0.7s,\ 0.9s$ at 2018.10.20 (case3).

data, we adopt the same input data used for LSTM training. For user request data, we generate the synthetic raw dataset following the normal distribution with mean value 10^4 and standard deviation 9×10^3. The dimension of the DNN input data is $(2I + J) * K$. The dimension of the DNN output data is $I * K' + I * K + J * I * K$. Therefore, for $I = 3$, $J = 5$, $K' = 1$, $K = 12$, the dimension of the input data is 132 and the dimension of the output data is 219. Our DNN model is simply composed of 1 fully connected hidden layers, 1 batch normalization layer [35] and 1 rectified linear unit (ReLU) layer [36]. As described in section IV, we use Eq.(29) as the customized loss function for our unsupervised DL solver. The maximum epochs for DNN training is empirically set as the fixed value 400. Although the network depth level is not high, via the experimental results, we found that the trained DNN model draws the desirable solutions under various renewable power capacity conditions (refer to Fig.13).

Fig. 9, 10, amd 11 show the prediction results for normalized solar radiation, OAT, and wind speed during 2018.10.01-2018.10.08 (8days) at DC1-3 by using the LSTM

TABLE 3. Solar power capacity (kW), wind power capacity (kW), and amount of user requests ($\times 10^3$) at 06:00am - 05:00pm, 2018.10.10 (case1), 2018.10.15 (case2), and 2018.10.20 (case3) in DC1 (Univ of Nevada), DC2 (Oak Ridge National Lab), and DC3 (NREL National Wind Technology Center, M2).

case1	6	7	8	9	10	11	12	13	14	15	16	17
spc^1	1.527	126.527	360.413	569.747	610.532	835.034	858.660	813.360	734.051	576.706	272.166	95.3713
spc^2	0	1.923	36.864	158.964	406.702	606.504	606.337	294.938	349.724	294.810	156.204	90.665
spc^3	0	2.555	33.290	65.668	95.149	139.913	158.996	81.163	56.9876	126.449	24.522	5.511
wpc^1	0	0	38.15	22.04	0	0	0	0	0	33.2	60.12	85.92
wpc^2	0	0	0	0	0	0	0	0	0	0	0	0
wpc^3	0	0	4.54	0	0	0	4.67	6.47	0	0	0	0
Λ^1	5.655	10.889	7.744	18.185	0.925	12.786	5.655	11.493	15.214	8.335	7.808	9.372
Λ^2	14.500	15.914	13.603	12.477	5.983	13.882	12.553	12.034	11.493	7.105	10.39	6.55
Λ^3	12.553	9.4584	10.5838	15.4448	10.0884	6.4708	9.4944	8.5887	5.249	4.925	10.494	5.389
Λ^4	9.494	11.179	11.219	8.382	7.713	10.707	9.744	8.822	10.262	9.142	17.829	8.595
Λ^5	9.744	11.164	6.437	4.272	11.499	7.211	14.5	10.209	12.186	7.227	12.766	7.171
case2	6	7	8	9	10	11	12	13	14	15	16	17
spc^1	0	122.428	383.696	608.429	782.346	886.969	922.833	874.476	753.805	567.274	334.968	93.354
spc^2	0	1.033	60.392	218.703	338.902	282.429	325.549	469.593	157.523	101.514	90.71	77.106
spc^3	0	42.644	271.412	508.292	692.262	820.888	886.987	849.992	743.096	565.099	337.979	106.925
wpc^1	159.6	76.62	134.68	222.18	244.81	218.23	480.14	510.41	438.01	345.99	292.15	232.73
wpc^2	0	0	0	0	0	0	0	0	0	0	0	0
wpc^3	0	0	0	0	18.17	11.87	53.05	0	0	0	0	0
Λ^1	9.251	6.941	5.513	10.553	15.371	7.213	10.985	9.088	13.352	9.660	10.106	8.5
Λ^2	8.781	8.867	10.16	8.753	10.157	6.923	7.367	12.068	13.783	11.128	7.815	15.594
Λ^3	8.881	9.993	7.567	8.433	14.225	11.127	13.666	14.091	15.111	10.629	14.913	4.71
Λ^4	10.756	7.767	16.871	6.753	7.385	9.235	12.802	10.654	5.96	12.786	7.855	5.342
Λ^5	11.447	11.484	8.565	10.978	7.621	6.601	8.768	14.574	11.3	12.438	14.168	6.991
case3	6	7	8	9	10	11	12	13	14	15	16	17
spc^1	0	97.717	325.938	526.574	676.233	774.136	808.664	703.192	357.659	473.933	205.607	43.374
spc^2	0	0.0204	5.164	25.314	50.640	139.909	272.479	371.189	504.172	366.566	369.647	153.665
spc^3	0	30.742	210.576	405.228	562.219	689.989	758.275	717.89	615.738	461.738	258.707	56.626
wpc^1	0	0	0	0	21.48	214.7	150.08	136.7	60.4	0	0	0
wpc^2	0	0	0	0	0	0	0	0	0	4.78	36.81	7.29
wpc^3	0	0	0	0	0	4.43	46.85	84.68	93.01	88.86	78.81	0
Λ^1	12.306	14.409	5.998	10.317	11.181	11.52	10.676	6.705	7.951	12.484	9.657	9.32
Λ^2	8.849	8.228	10.739	2.817	11.203	12.632	16.033	11.095	10.35	7.535	14.933	1.946
Λ^3	6.859	10.288	12.342	10.88	6.381	7.633	10.148	9.898	10.186	6.383	13.807	9.663
Λ^4	12.616	8.304	9.508	10.653	8.489	9.111	7.091	9.361	7.545	15.032	11.155	9.586
Λ^5	7.658	11.549	13.161	8.791	6.224	10.198	11.33	10.83	12.8	14.626	11.987	8.72

based predictor. The amount of solar radiation shows the regular patterns over distributed DC regions, as shown in Fig.9-(a), Fig.10-(a), and Fig.11-(a). The OAT curves of DC1 and DC3 have somewhat irregular patterns (Fig.9-(b), Fig.11-(b)), while the DC2 has the regular OAT pattern (Fig.10-(b)). The wind speed curves of DC1 and DC3 also show aperiodic patterns (Fig.9-(c), Fig.11-(c)), while the DC2 has the periodic wind speed patterns (Fig.10-(c)). Note that the LSTM predictor achieves the high prediction accuracy under both the regular weather conditions and the irregular ones. However, for some control time points (e.g., 10/7/2018, 08:00 a.m in Fig.9-(a)), the transient prediction error for the renewable power capacity is identified. For such cases, the MICRO time scale management of the MAMI optimizer copes with the changing capacity patterns.

Fig.12 shows the results of DNN training progress for the customized loss function Eq.(29). The service response latency bound is $\bar{d} = 0.7sec$. The violation of box constraints $\underline{m}, \overline{m}, \underline{f}, \overline{f}$ is diminished along the progress of DNN training, as shown in Fig.12-(a), (b), (c), and (d). Interestingly, in Fig.12-(d), the violation of constraint for CPU frequency upper bound $\bar{f}$ is consistently increased until the epoch $= 250$, and after then, it is drastically decreased. This is because, as shown in Fig.12-(a), the DNN training is firstly performed aims to increase CPU frequency f, in order to mitigate the violation of the service response latency bound caused by the insufficient initial m. When m reaches a sufficient size at epoch $= 250$ however, the DNN training again performs weight parameter updating toward reducing the violation of CPU frequency upper bound. Fig.12-(e), (f), and (g) show the degrees of constraint violation for service response latnecy bound, the positivity of queue time, and the request dispatching, respectively. In Fig.12-(e) and (g), the associated constraint violation curves fall down during epoch $= 0 - 60$, increase during epoch $= 60 - 120$, and decrease consistently after epoch $= 120$. This is because, both the amount of the rejected requests and the service response latency are decreased, due to the increasing negativity of queue time $m * f - \lambda$ (the excessive request dispatching to DCs) until epoch $= 60$, as shown in Fig.12-(f). However, after epoch $= 60$, updating the weight parameters in the direction of reducing the positivity violation of queue time is performed. Consequently, as shown in Fig.12-(h), our DL solver results in the network weight parameters which draw the minimized energy cost without any serious constraint violation.

In order to investigate the performance of our proposed DL based MAMI optimizer, we test three real cases based on renewable power capacity trace data during 12 hours: case1 (2018.10.10, 06:00 am - 05:00 pm at DC1-DC3); case2 (2018.10.15, 06:00 am - 05:00 pm at DC1-DC3); and case3 (2018.10.20, 06:00 am - 05:00 pm at DC1-DC3). The amount of user requests are randomly generated by using normal distribution with average 10^4 $(reqs/sec)$ and standard deviation 9×10^3. The associated trace data is shown in Table 3.

We adopt the rank based genetic algorithm (GA) [24] based DC management scheme as the conventional approach to compare with our proposed one. This scheme uses the adaptive penalty method by which the penalty weight values are iteratively updated during evolution phase according to the degree of constraint violation. It can be used to solve the constrained non-linear programming problem. For the GA scheme, we set the population size $=$ 1000 in which each chromosome has 3177-bits binary genes. In addition, we set the mutation probabilty $=$ 0.3, the selection pressure $=$ 3 for roulette wheel procedure [9], and the evolution steps $=$ 12000, respectively. Especially, we use the elitisim approach to guarantee the stable convergence to the optimal solution [43].

Fig.13-(a), (b) and (c) show the DC energy cost of the proposed MAMI optimizer and the rank based GA scheme. Note that the MAMI optimizer outperforms the GA scheme for every case regardless of the renewable power capacity condition. The MAMI optimizer achieves in average the reduction of energy cost about 25% compared to the GA scheme. The experimental results of service response latency of DC1-DC3 for every case are shown in Fig.13 (d)-(f). Note that the rank based GA scheme does not satisfy service response latency bound even for rather loosed values $\bar{d} = 0.7s, 0.9s$. This indicates that the GA scheme is not suitable for resolving the complex constrained problem. Contrary, even for the tight constraint value $\bar{d} = 0.3sec$, the MAMI optimizer consistently satisfies the service response latency bound constraint for all cases. This indicates that the MAMI optimizer can generate the DNN weight parameters deriving feasible solution under all cases. We argue that the MAMI optimizer can be a promising alternative for the conventional metaheuristics so as to solve the complex constrained optimization problem.

Furthermore, once the DNN model training is completed, the computation time for DNN inferencing is generally negligible. We found that the average inference time of our trained DNN model is less than $1s$. Moreover, the DNN training time is acceptable. We found that the average training time for our DNN model is less than $30min$. However, the rank based GA scheme needs the unacceptable re-computation time at each control time step. The average computation time required for the GA scheme appeared in Fig.13, is long than $10hr$.

VI. CONCLUSION

In this paper, we proposed a deep learning (DL) based novel optimizer using long short term memory (LSTM) prediction in order to achieve the energy cost efficient geo-distributed sustainable data centers. To the best of our knowledge, this is the first study to propose the data center energy cost optimization by using unsupervised DL method as the constrained non-convex problem solver. We proposed the temporal MACRO/MICRO (MAMI) time scale management technique in order to efficiently achieve the coordinated decision making of dynamic right sizing (DRS) and the CPU frequency scaling (FS), and the request dispatching (RD). This technique minimizes the energy cost of data centers without the unacceptable DRS wake-up transition overhead. To demonstrate the performance of our proposed DL based MAMI optimizer, we used $7 * 10^5$ real trace data of renewable power capacity from U.S. Measurement and Instrumentation Data Center (MIDC). The adopted LSTM predictor showed the accurate prediction results for fluctuated solar and wind power capacity. The DL solver shows that it gradually approaches the feasible optimal solution for constrained non-convex optimization problem. Compared to the conventional rank based genetic algorithm (GA), on average, the proposed DL based MAMI optimizer reduces the DC energy cost about 25% for real trace data while gauranteeing the service quality requirement.

REFERENCES

[1] J. Qiu, J. Zhao, H. Yang, and Z. Y. Dong, "Optimal scheduling for prosumers in coupled transactive power and gas systems," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1970–1980, Mar. 2018.

[2] Y. Xiao, X. Wang, P. Pinson, and X. Wang, "A local energy market for electricity and hydrogen," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3898–3908, Jul. 2018.

[3] D. Cheng, X. Zhou, Z. Ding, Y. Wang, and M. Ji, "Heterogeneity aware workload management in distributed sustainable datacenters," *IEEE Trans. Parallel Distrib. Syst.*, p. 1, Aug. 2018, doi: 10.1109/TPDS.2018.2865927.

[4] *Adoption of the Paris Agreement*, UNFCC, 2015. [Online]. Available: http://unfccc.int/resource/docs/2015/cop21/eng/l09.pdf

[5] Facebook. *Adding Clean and Renewable Energy to the Grid.* Accessed: Dec. 2018. [Online]. Available: https://sustainability.fb.com/clean-and-renewable-energy/

[6] Apple. *To Ask Less of the Planet, We Ask More of Ourselves.* Accessed: Dec. 2018. [Online]. Available: https://www.apple.com/lae/environment/climate-change/

[7] S. X. Chen, H. B. Gooi, and M. Q. Wang, "Sizing of energy storage for microgrids," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 142–151, Mar. 2012.

[8] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. Peng, "Green data center with IoT sensing and cloud-assisted smart temperature control system," *Comput. Netw.*, vol. 101, pp. 104–112, Jun. 2016.

[9] Y. Peng, D.-K. Kang, F. Al-Hazemi, and C.-H. Youn, "Energy and QoS aware resource allocation for heterogeneous sustainable cloud datacenters," *Opt. Switching Netw.*, vol. 23, pp. 225–240, Jan. 2017.

[10] H. Nazaripouya, B. Wang, Y. Wang, P. Chu, H. R. Pota, and R. Gadh, "Univariate time series prediction of solar power using a hybrid wavelet-ARMA-NARX prediction method," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Exposit. (T&D)*, May 2016, pp. 1–5.

[11] B. Aksanli, J. Venkatesh, L. Zhang, and T. Rosing, "Utilizing green energy prediction to schedule mixed batch and service jobs in data centers," in *Proc. ACM 4th Workshop Power-Aware Comput. Syst.*, 2011, Art. no. 5.

[12] J. Zhang, G. Welch, G. Bishop, and Z. Huang, "A two-stage Kalman filter approach for robust and real-time power system state estimation," *IEEE Trans. Sustainable Energy*, vol. 5, no. 2, pp. 629–636, Apr. 2014.

[13] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1378–1391, Oct. 2013.

[14] X. Mei, X. Chu, H. Liu, Y.-W. Leung, and Z. Li, "Energy efficient real-time task scheduling on CPU-GPU hybrid clusters," in *Proc. INFOCOM IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.

[15] D. Cheng, X. Zhou, P. Lama, M. Ji, and C. Jiang, "Energy efficiency aware task assignment with DVFS in heterogeneous Hadoop clusters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 1, pp. 70–82, Jan. 2018.

[16] M. R. V. Kumar and S. Raghunathan, "Heterogeneity and thermal aware adaptive heuristics for energy efficient consolidation of virtual machines in infrastructure clouds," *J. Comput. Syst. Sci.*, vol. 82, no. 2, pp. 191–212, 2016.

[17] Q. Fang, J. Wang, Q. Gong, and M. Song, "Thermal-aware energy management of an HPC data center via two-time-scale control," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2260–2269, Oct. 2017.

[18] Z. Liu, M. Lin, A. Wierman, S. Low, and L. L. H. Andrew, "Greening geographical load balancing," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 657–671, Apr. 2015.

[19] D. Cheng, Y. Guo, C. Jiang, and X. Zhou, "Self-tuning batching with DVFS for performance improvement and energy efficiency in Internet servers," *ACM Trans. Auto. Adapt. Syst.*, vol. 10, no. 1, 2015, Art. no. 6.

[20] Z. Liu *et al.*, "Renewable and cooling aware workload management for sustainable data centers," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 175–186, 2012.

[21] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[22] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[23] *Measurement and Instrumentation Data Center (MIDC)*, Accessed: Dec. 2018. [Online]. Available: https://midcdmz.nrel.gov/

[24] R. de Paula Garcia, B. S. L. P. de Lima, A. C. de Castro Lemonge, and B. P. Jacob, "A rank-based constraint handling technique for engineering design optimization problems solved by genetic algorithms," *Comput. Struct.*, vol. 187, pp. 77–87, Jul. 2017.

[25] L. Peng, A. R. Dhaini, and P.-H. Ho, "Toward integrated Cloud–Fog networks for efficient IoT provisioning: Key challenges and solutions," *Future Gener. Comput. Syst.*, vol. 88, pp. 606–613, Nov. 2018.

[26] L. Peng, "On the future integrated datacenter networks: Designs, operations, and solutions," *Opt. Switching Netw.*, vol. 19, pp. 58–65, Jan. 2016.

[27] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. Leung, "SOVCAN: Safety-oriented vehicular controller area network," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 94–99, 2017.

[28] A. Yona, T. Senjyu, A. Y. Saber, T. Funabashi, H. Sekine, and C.-H. Kim, "Application of neural network to 24-hour-ahead generating power forecasting for PV system," in *Proc. IEEE Power Energy Soc. Gen. Meeting-Conversion Del. Electr. Energy 21st Century*, 2008, pp. 1–6.

[29] D. T. Nguyen and L. B. Le, "Optimal bidding strategy for microgrids considering renewable energy and building thermal dynamics," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1608–1620, Jul. 2014.

[30] J. Aho *et al.*, "A tutorial of wind turbine control for supporting grid frequency through active power control," in *Proc. IEEE Amer. Control Conf. (ACC)*, Jun. 2012, pp. 3120–3131.

[31] T. Horvath and K. Skadron, "Multi-mode energy management for multi-tier server clusters," in *Proc. IEEE Int. Conf. Parallel Archit. Compilation Techn. (PACT)*, Oct. 2008, pp. 270–279.

[32] Y. Zhang, Y. Wang, and X. Wang, "Greenware: Greening cloud-scale data centers to maximize the use of renewable energy," in *Proc. ACM/IFIP/USENIX Int. Conf. Distrib. Syst. Platforms Open Distrib. Process.* Berlin, Germany: Springer, 2011, pp. 143–164.

[33] W. Zhu *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. AAAI*, vol. 2, no. 5, 2016, p. 6.

[34] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep features learning for medical image analysis with convolutional autoencoder neural network," *IEEE Trans. Big Data*, p. 1, Jun. 2017, doi: 10.1109/TBDATA.2017.2717439.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[36] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[37] S. Wan, Y. Liang, Y. Zhang, and M. Guizani, "Deep multi-layer perceptron classifier for behavior analysis to estimate parkinson's disease severity using smartphones," *IEEE Access*, vol. 6, pp. 36825–36833, 2018.

[38] *NVIDIA Developer*. Accessed: Dec. 2018. [Online]. Available: https://developer.nvidia.com/

[39] *Keras: The Python Deep Learning Library*. Accessed: Dec. 2018. [Online]. Available: https://keras.io/

[40] *TensorFlow: An Open Source Machine Learning Library for Research and Production*. [Online]. Available: https://www.tensorflow.org/

[41] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[42] S. Xiao, H. Yu, Y. Wu, Z. Peng, and Y. Zhang, "Self-evolving trading strategy integrating Internet of Things and big data," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2518–2525, Aug. 2018.

[43] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

DONG-KI KANG received the M.S. degree in computer engineering from Chonbuk National University, Jeonju, South Korea, in 2011. He is currently pursuing the Ph.D. degree in electrical engineering with the Korea Advanced Institute of Science and Technology, Daejeon, South Korea. His research interests include cloud computing, deep learning, and GPU power control.

EUN-JU YANG received the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2018, where she is currently pursuing the Ph.D. degree in electrical engineering. Her research interests include deep learning, machine learning distributed processing, and multi-access edge computing.

CHAN-HYUN YOUN (S'84–M'87) received the B.Sc. and M.Sc. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 1981 and 1985, respectively, and the Ph.D. degree in electrical and communications engineering from Tohoku University, Sendai, Japan, in 1994. Before joining the University, from 1986 to 1997, he was the Leader of the High-Speed Networking Team, KT Telecommunications Network Research Laboratories, where he had been involved in the research and developments of centralized switching maintenance system, maintenance and operation system for various ESS's system, high-speed networking, and ATM network testbed. Since 1997, he has been a Professor with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. He was an Associate Vice-President of office of planning and budgets in KAIST from 2013 to 2017. He has authored a book on *Cloud Broker and Cloudlet for Workflow Scheduling* (Springer, 2017). His research interests include distributed high performance computing systems, cloud computing systems, edge computing systems, deep learning framework, and others. He received the Best Paper Award from CloudComp 2014. He was a General Chair for the 6th EAI International Conference on Cloud Computing, KAIST, in 2015. He was a Guest Editor of the IEEE WIRELESS COMMUNICATIONS in 2016.