Electrical Power and Energy Systems 129 (2021) 106855

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

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

Long-term planning of integrated local energy systems using deep learning algorithms

Saman Taheri [a], Mohammad Jooshaki [b], Moein Moeini-Aghtaie [a,*]

[a] Department of Energy Engineering, Sharif University of Technology, Iran
[b] Department of Electrical Engineering, Sharif University of Technology, Iran

ARTICLE INFO

Keywords:
Energy consumption prediction
Integrated local energy systems
Deep learning
Deep recurrent neural network
Gaussian process

ABSTRACT

Optimal investment and operations of integrated local energy systems (ILESs) require medium to long-term prediction of energy consumption. To forecast load profiles, deep recurrent neural networks (DRNNs) are becoming increasingly useful due to their capability of learning uncertainty and high variability of load profiles. However, to explore and choose a DRNN model, out of conceivably numerous configurations, depends entirely on the performing task. In this regard, we tune and compare seven DRNN variants on the task of medium and long-term predictions for heating and electricity consumption. The ultimate DRNN model outperforms two state-of-the-art machine learning techniques, namely gradient boosting (GB) regression and support vector machine (SVM) in terms of accuracy. Also, developing optimization frameworks that can employ deep learning algorithms to alleviate the significant computational burden associated with EH planning is neglected in previous literature. Thus, we aim at improving the efficiency of the optimization process through a strong coupling of deep learning algorithms and conventional optimization methods. In this regard, a step-by-step algorithm is formulated to facilitate the use of Gaussian Process (GP) regression within the optimization problem and obtain optimal results without explicitly performing the optimization problem. The overall computational time is decreased by a factor of 0.47 when the proposed optimization framework is implemented for a practical EH system.

1. Introduction

In recent years, significant steps have been taken to optimize multi-carrier local energy systems consisting of natural gas and electricity networks. As such, the EH concept, a centralized unit in which different energy carriers are transformed, stored, and converted [1], provides an interface to model integration of various energy systems, e.g., electricity and natural gas networks. Enabling new design approaches for integrated systems, increasing efficiency, addressing environmental issues, providing higher flexibility in the operation of the multi-carrier energy systems, and reducing costs are reasons behind using the EH models. However, determining the optimum profiles of EH inputs has remained challenging since. Price volatility of energy carriers, optimal penetration of renewable energy sources, and incorporation of physics-based models of conversion units and storage devices are among the most critical challenges–along with prediction of load profiles [2]. In this context, day-ahead scheduling of power-to-gas (P2G) storage and gas load management in electricity and gas markets are conducted in [3] to minimize the cost of natural gas consumption. A novel framework for optimizing the profits of wind farms and P2G facilities is presented in [4]. Different P2G processes and their operational impacts on both electricity and gas networks are analyzed in [5].

Energy loads were traditionally treated to be passive and static in classic formulations [6]. With the proliferation of distributed generation and energy storage systems to supply electricity and heat demands in the EHs, however, a growing need for enhanced load prediction techniques has emerged [7]. Meanwhile, load forecasting is challenging because of the nonlinear and complex patterns of the loads, which had otherwise been assumed to be given and fixed across various time horizons [8]. Such time intervals include: (I) short-term (hours or days ahead) that concerns unit commitment and economic dispatch of generating units and storage systems as well as control/maintenance of energy equipment [9], and (II) medium-to-long term ($\geqslant$1 week up to 20 years ahead) that is useful for proper scheduling, system planning, and installation of new generation capacity [10]. The majority of previous research has focused on the short-term time horizon, whereas we investigate the application of long-term load prediction in this study.

Most forecasting methods are based on either statistical techniques or machine learning algorithms (MLAs) [11]. While a large variety of

* Corresponding author.
E-mail address: Moeini@sharif.edu (M. Moeini-Aghtaie).

https://doi.org/10.1016/j.ijepes.2021.106855
Received 16 August 2020; Received in revised form 26 November 2020; Accepted 24 January 2021
Available online 25 February 2021

Nomenclature

Indices

d	Indicating the day within a year 1, 2,…, N_d
i	Indicating the energy hub candidate equipment; $i \in$ CE
j	Indicating nodes in networks 1, 2, … , N_l
k	Indicating the kth iteration of training 1, 2, …, K
l	Index for the layer in the neural network $l \in \Gamma$
m	Index for nodes in networks 1, 2, … , N_{l-1}
p	Indicating the prediction of GP regression 1, 2, … , N_p
t	Indicating the hour of the day 1, 2, … , N_t
y	Index for the year in the planning horizon 1, 2, …, N_y
ts	Index for the training set entries 1, 2, …, N_{ts}

Variables and Parameters

β	Learning rate
Γ	Set of hidden layers in a neural network
CE	Candidate equipment for energy hub
σ	Variance of prediction from the GP model
a	The hyperbolic tangent activation function
d_t	Predicted value of consumption
$D^{e}_{t,d,y}$	Electricity demand
$D^{h}_{t,d,y}$	Heat demand
dr	Discount rate
e	Mean squared average error of demand
E^{emax}	Maximum connection capacity to the electricity grid
E^{gmax}	Maximum connection capacity to the natural gas network
e_E	Mean squared average error of the electricity demand
e_H	Mean squared average error of the heat demand
$E^{e}_{t,d,y}$	Electricity purchased from grid
f_p	Predictions of new input features calculated by GP regression
h^{l}_{j}	Output of hidden node j of layer l
h^{l-1}_{m}	Output of hidden node m of layer $l-1$
$K(X_1, X_2)$	Kernel Function indicating the proximity between X_1 and X_2
N_d	Number of days in year y
N_k	Number of iterations of the training process
N_l	Number of nodes in layer l
N_p	Number of GP predictions
N_y	Number of years in the planning horizon
N_{ts}	Number of training set entries
N_t	Number of hours in a day
$tanh$	Tangent hyperbolic activation function
X_p	New input features to the GP model
X_{ts}	Training inputs of GP model
Y_{ts}	Targets of training data for GP regression
η^{eCHP}	CHP gas to electricity efficiency
η^{hCHP}	CHP gas to heat efficiency
η^{hF}	Thermal efficiency of the furnace
η^{TR}	Efficiency of the transformer
IC_i	Investment cost for the candidate equipment
OP_y	Operating cost of EH in year y
$PC^{e}_{t,d,y}$	Electricity price
$PC^{g}_{t,d,y}$	Natural gas price
σ_s	Sigmoid utility as a gating function
τ_y	Present value factor for operational costs in year y
d^{a}_{t}	Actual value of consumption
d^{ts}_{t}	training set values of energy consumption
$w^{l}_{j,m}$	weight of the connection from node m in layer $l-1$ to node j in layer l
X^{GP}	Prediction of the GP regression model
y_t	Output values of a feed-forward neural network
$\mathbf{a}_t$	Vector of the outputs of the hyperbolic tangent activation function
$\mathbf{b_{a,t}}$	Vector of bias values for activation function
$\mathbf{b_{c,t}}$	Vector of bias values for candidate unit
$\mathbf{b_{f,t}}$	Vector of bias values for forget gate
$\mathbf{b_{o,t}}$	Vector of bias values for output gate
$\mathbf{b_{u,t}}$	Vector of bias values for update gate
$\mathbf{b_{Y,t}}$	Vector of bias values for prediction
$\mathbf{c}_t$	Vector of candidate values for replacing the vector of values in memory cell
$\mathbf{f_p}$	Vector of the predictions calculated by GP regression
$\mathbf{f_t}$	Vector of the outputs from forget gate
$\mathbf{o}_t$	Vector of the output from output gate
$\mathbf{u}_t$	Vector of the output from update gate
$\mathbf{w_{a,t}}$	Vector of weightening values for activation function
$\mathbf{w_{c,t}}$	Vector of weighting values for candidate unit
$\mathbf{w_{f,t}}$	Vector of weighting values for forget gate
$\mathbf{w_{o,t}}$	Vector of weighting values for output gate
$\mathbf{w_{u,t}}$	Vector of weighting values for update gate
$\mathbf{w_{Y,t}}$	Vector of weighting values for prediction
$\mathbf{X_p}$	Vector of the new input features to the GP
$\mathbf{X}_t$	Input features at time step t
$\mathbf{X_{ts}}$	Vector of the training inputs of GP model
$\mathbf{Y_{ts}}$	Vector of the targets of training data for GP regression
$\mathbf{Z}$	Non-dimensional heat and electricity demands used in GP
$\mathbf{Z_p}$	Non-dimensional heat and electricity demands used as prediction entries for GP
$\mathbf{Z_{ts}}$	Non-dimensional heat and electricity demands used in GP as training data sets
$E^{gCHP}_{t,d,y}$	Gas purchased from the grid for CHP
$E^{gF}_{t,d,y}$	Gas purchased from the grid for furnace
$E^{g}_{t,d,y}$	Total gas purchased from grid
P_i	Capacity of hub equipment
P_{CHP}	Installed capacity of CHP
P_F	Installed capacity of furnace
P_{TR}	Installed capacity of transformer

mathematical methods have been employed for long-term forecasts, deep learning is an attractive option as it is able to automatically learn complex behaviour of loads [12]. General regression neural networks, either elaborate [13] or simplified [14], non-linear regression [15], gradient boosting machine [16], multi-layered perceptron networks [17], GP [18], SVM [19], have been used to forecast electricity consumption. While a wide range of MLAs has been utilized to predict electricity consumption, their performance in forecasting heating load has not been thoroughly investigated as of yet. Nonetheless, heat demand prediction is of great importance when it comes to the EH optimization [20].

In this regard, the authors in [21] employed a reinforcement learning algorithm to solve an EH optimal dispatch problem when electricity and heating loads are stochastic. A hybrid algorithm is developed in [22,23] to forecast electricity, cooling, and heating demands by merging deterministic thermal networks with MLAs. Reference [24] optimized the operational cost of a smart EH by using a reinforcement learning algorithm. More recently, the authors in [11] predicted mid-long term electricity and heat consumption via supervised-based machine learning models with limited data information. In [25], the field of Bootstrap

aggregating (Bagging) and forecasting methods are combined to obtain more accurate electricity demand forecasts. A detailed review of the relative performance and application of different MLAs is accessible in the existing literature [26,27]. In this study, we aim to develop a DRNN model since it is capable of capturing the sequential nature of the demands [28]. Besides, DRNNs can model the implicit temporal dependencies that exist in the load time series [29]. To the best of our knowledge, developing a DRNN model, tuned exclusively on the task of heat and electricity demand prediction for an EH, is not addressed in the literature.

Furthermore, we developed a general structure to use GP regression to compute optimal solutions to the EH problem without explicitly executing the optimization problem. By incorporating the GP regression in the optimization process, the authors in [30] were able to drastically reduce the overall computational burden.GP is also being used in the optimization of energy systems since it is capable of learning highly complex functions. For instance, a stratified thermal storage is sized using GP regression as an optimizer in [31]. Inspired by this paper, we thought that GP regression can be used similarly to reduce the computational burden of EH planning problems as thermal energy storage is a commonly used equipment in the EH systems. As such, GP is utilized in cases that a certain threshold number of circumstances are provided. Accordingly, the proposed optimization framework facilitates linking demand forecasts, required to be predicted at a one-hour resolution, to the design optimization process that occurs over a remarkably longer timescale.

Based on the above-mentioned remarks, there is a need for the evaluation of MLAs that can illuminate long-term temporal dependencies in predicting energy loads, particularly to forecast heating demand. In addition, developing novel frameworks capable of incorporating these long-term predictions in the optimization process is a critical research gap, which is addressed in this paper. Hence, the fundamental objective of this paper is to propose an optimization scheme, which uses DRNN and GP regression to aid the optimal design of EHs. On these bases, the main contributions of the study are summarized in the following terms:

- We show that the proposed DRNN configuration performs better than other machine learning techniques such as SVM and GB regression in terms of accuracy. Also, the potential benefit of implementing the introduced optimization framework is demonstrated in terms of computational time (Section 3).

2. Proposed technique for energy hub planning

The proposed optimization framework is implemented on an EH with a general configuration illustrated in Fig. 1. According to this figure, the EH has a general layout structure in which natural gas and electricity are the input energy carriers, technologies like combined heat and power (CHP), natural gas furnace (F), and transformer (TR) are EH equipment to supply heat and electricity demands at the output port. It is worth mentioning that the cooling demand is not included because of the lack of representative data. For the sake of simplicity, we consider a simplified model for the EH in which factors such as equipment life cycle and maintenance are disregarded. Nonetheless, more comprehensive models of EHs which have been proposed in [32–34] can be readily incorporated into the proposed framework. In this study, a linear EH model is assumed for the EH configuration [35]. The mathematical formulation of the model is derived in the following subsections.

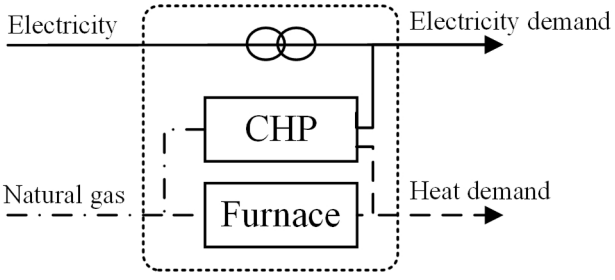

Fig. 1. Schematic of the EH configuration.

2.1. Objective function

The objective of the planning model is to minimize the total net present value of the investment and operating costs of the EH as formulated in (1):

$$\mathrm{Min}\left(\sum_{i\in \mathrm{CE}} P_i \mathrm{IC}_i + \sum_{y=1}^{N_y} \tau_y \mathrm{OP}_y\right) \tag{1}$$

here, τ_y is the present-worth coefficient of the operational costs in stage y of the planning horizon, which is calculated using (2):

$$\tau_y = (1+dr)^{1-y}; \forall y \in \{1,\ldots,N_y\} \tag{2}$$

The operational cost of the EH is a function of consumed electricity and natural gas and their prices, as expressed by (3):

$$\mathrm{OP}_y = \sum_{d=1}^{N_d}\sum_{t=1}^{N_t}\left(\mathrm{PC}^{\mathrm{e}}_{t,d,y} E^{\mathrm{e}}_{t,d,y} + \mathrm{PC}^{\mathrm{g}}_{t,d,y} E^{\mathrm{g}}_{t,d,y}\right) \forall y \in \{1,\ldots,N_y\} \tag{3}$$

2.2. Constraints

The EH planning problem is subject to the input–output balance as follows:

$$D^{\mathrm{e}}_{t,d,y} = \eta^{\mathrm{TR}} E^{\mathrm{e}}_{t,d,y} + \eta^{\mathrm{eCHP}} E^{\mathrm{gCHP}}_{t,d,y}; \forall t \in \{1,\ldots,N_t\}, \forall d \in \{1,\ldots,N_d\}, \forall y \in \{1,\ldots,N_y\} \tag{4}$$

$$D^{\mathrm{h}}_{t,d,y} = \eta^{\mathrm{hCHP}} E^{\mathrm{gCHP}}_{t,d,y} + \eta^{\mathrm{hF}} E^{\mathrm{gF}}_{t,d,y}; \forall t \in \{1,\ldots,N_t\}, \forall d \in \{1,\ldots,N_d\}, \forall y \in \{1,\ldots,N_y\} \tag{5}$$

$$E^{\mathrm{g}}_{t,d,y} = E^{\mathrm{gF}}_{t,d,y} + E^{\mathrm{gCHP}}_{t,d,y}; \forall t \in \{1,\ldots,N_t\}, \forall d \in \{1,\ldots,N_d\}, \forall y \in \{1,\ldots,N_y\} \tag{6}$$

The amounts of the input electricity and natural gas power are restricted by their network connection capacities as formulated in (7) and (8).

$$E^{\mathrm{e}}_{t,d,y} \leqslant E^{\mathrm{emax}}; \forall t \in \{1,\ldots,N_t\}, \forall d \in \{1,\ldots,N_d\}, \forall y \in \{1,\ldots,N_y\} \tag{7}$$

$$E^{\mathrm{g}}_{t,d,y} \leqslant E^{\mathrm{gmax}}; \forall t \in \{1,\ldots,N_t\}, \forall d \in \{1,\ldots,N_d\}, \forall y \in \{1,\ldots,N_y\} \tag{8}$$

The electricity and natural gas from the grids should be also constrained based on the installed capacities of the candidate equipment as below:

$$\eta^{\mathrm{eTR}} E^{\mathrm{e}}_{t,d,y} \leqslant P_{\mathrm{TR}}; \forall t \in \{1,\ldots,N_t\}, \forall d \in \{1,\ldots,N_d\}, \forall y \in \{1,\ldots,N_y\} \tag{9}$$

$$\eta^{\mathrm{eCHP}} E^{\mathrm{gCHP}}_{t,d,y} \leqslant P_{\mathrm{CHP}}; \forall t \in \{1,\ldots,N_t\}, \forall d \in \{1,\ldots,N_d\}, \forall y \in \{1,\ldots,N_y\} \tag{10}$$

$$\eta^{\mathrm{hF}} E^{\mathrm{g}}_{t,d,y} \leqslant P_{\mathrm{F}}; \forall t \in \{1,\ldots,N_t\}, \forall d \in \{1,\ldots,N_d\}, \forall y \in \{1,\ldots,N_y\} \tag{11}$$

2.3. Embedding deep learning in the optimization process

As mentioned before, determining the optimal decision variables in an EHOD problem requires developing optimization frameworks that utilize load prediction models. In this respect, a novel framework is introduced to develop a model for EH studies and to execute the optimization subjected to the medium-long term load prediction. The flowchart of the proposed framework is depicted in Fig. 2.

As shown in the first stage, the historical electricity and heat loads data are gathered. These data can be collected by either the energy audit

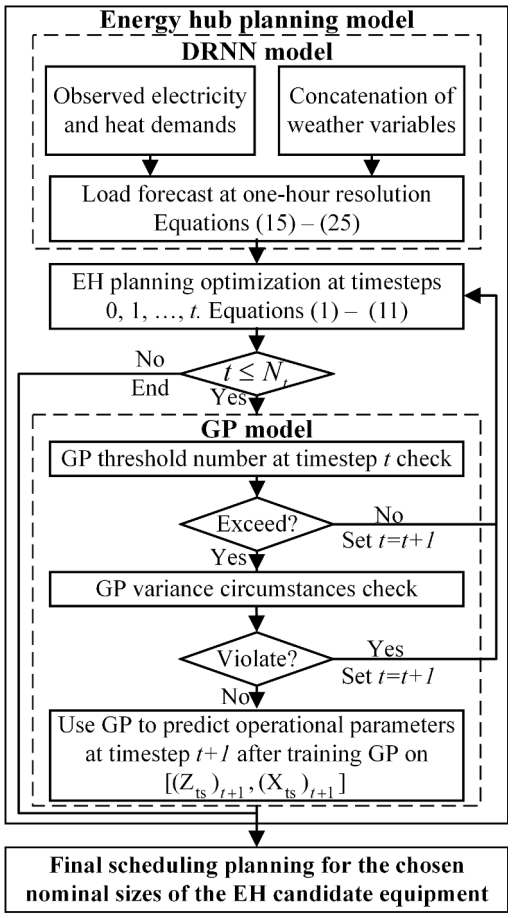

Fig. 2. EH planning model using DRNN and GP models.

programs executed for any specific case studies or using energy simulation software tools with different levels of complexity, such as EnergyPlus and TRNSYS - used to evaluate the of buildings' energy demand over a longer time frame [36]. However, these tools require physical principles to compute energy demands, and hence, detailed information on operational schedules, construction, and therefore, greater expertise are needed [31]. In addition to the historical energy consumption values, a concatenation of weather variables is also needed. Accordingly, these are the two sets of data used as inputs to the DRNN model; date-related variables and a concatenation of weather variables. The weather variables are air pressure, temperature, and humidity at the predetermined location; while the date-related inputs are (I) hours of the day (from 1 to 24), (II) days of a year, and (III) years in the economic life of the project. The day of the year is supposed a concatenation of 365 binary variables; each variable signifies the numerical value of the item in the dataset. This representation of categorical variables is known as one-hot encoding [37], where all bits are zero except the one that acts as a binary flag for each day of a year. A similar implementation is considered for the other date-related variables.

Secondly, we implement the DRNN model to perform the EH electricity and heat load prediction at one-hour resolution. A wide range of long-term load forecast models with different levels of elaboration can be used in this stage as discussed before. In this study, we used a deep learning algorithm applied to the collected data to predict the loads within the economic life (EL) of the project. The objective behind implementing the DRNN model in this step is to use the predicted load values in the optimization process.

In the next step, the optimization problem is solved to attain the optimal investment plan as well as operation schedule of the EH facilities. Given that the size of the EH components are known, the objective of EH problem is to minimize the operational costs of the EH over the planning horizon. Since we have considered the EH equipment to take nominal sizes available in the market, we should solve the optimization problem for different possible capacities' combinations of the EH candidate equipment, including CHP, furnace, and transformer. The optimal sizes of the equipment are then chosen by comparing the value of the objective function for different cases. Multiple alternatives for CHP and furnace capacities are taken from those presented by [38,39]. It should be mentioned that considering discrete capacity options for the candidate equipment does not limit the application of the proposed framework, and continuous optimal capacities can also be employed. The next step, which is embedding the GP in the optimization problem, is worthy of more discussion that is provided below.

2.3.1. Gaussian Process Regression

Teaching a function that makes predictions for all possible input values from a training set of data can be cast directly into the GP framework. A GP is a generalization of Gaussian distribution that specifies a distribution over functions [40]. Given a training set with entries (X_{ts}, Y_{ts}) for $ts \in \{1, 2, ..., N_{ts}\}$, we want to make predictions for new inputs X_p for $p \in \{1, 2, ..., N_p\}$ that have not been included in the training set. In this regard, the prediction f_p is made through the following:

$$f_p = K^{\mathrm{T}}(X_p)(\mathbf{K_{ts}} + \sigma^2\mathbf{I})^{-1}\mathbf{Y_{ts}}; \forall p \in \{1, 2, \ldots, N_p\} \tag{12}$$

where $\mathbf{K_{ts}}$ denotes the $N_{ts} \times N_{ts}$ matrix of covariance (or kernel function) evaluated at training points with entries (X_1, X_2). In this study, the Kernel function is considered as a squared exponential function indicating the proximity between X_1 and X_2 [41]. $K^T(X_p)$ is the transpose of the vector of covariances $K(X_{ts}, X_p)$, $\mathbf{Y_{ts}}$ is a $N_{ts} \times 1$ vector containing the training targets, and σ^2 is the noise variance on the observations. In addition to point estimation, GP can also be used to calculate the variance associated with the predictions as follows:

$$\sigma^2(f_p) = K(\mathbf{X_p}, \mathbf{X_p}) - K(\mathbf{X_p}, \mathbf{X_{ts}})[K(\mathbf{X_{ts}}, \mathbf{X_{ts}})]^{-1}K(\mathbf{X_p}, \mathbf{X_{ts}}); \forall p \in \{1, 2, \ldots, N_p\} \tag{13}$$

In this study, the Bayesian optimization package (*bayesopt*) in Python is employed as the optimization solver. The *bayesopt* package is a compelling option since it uses the Bayesian Optimization scheme in which computationally efficient algorithms for solving non-convex problems are engaged [37]. This stage is integrated with the GP model to improve the computational time of the overall optimization process, as we need to execute the optimization at each time-step (i.e., the first hours up to EL of the project). Therefore, it is assumed that the optimal operational variables $\mathbf{X^{GP}} = [E^{\mathrm{e}}, E^{\mathrm{gCHP}}, E^{\mathrm{gf}}]$ at each time interval are functions of electricity and heating demands $f_p([D^{\mathrm{e}}, D^{\mathrm{h}}])$. We use $\mathbf{Z}$ here to express that the entries to the GP are matrices consisting of non-dimensional heating and electricity demand. $\mathbf{Z}$ has also been divided to the test $\mathbf{Z_{ts}}$ and prediction $\mathbf{Z_p}$ sets. As such, the normal distribution over the obtained results $f_p(\mathbf{Z})$ from GP regression can be expressed as:

$$\mathbf{X^{GP}} = f_p(\mathbf{Z}) = N[\mu(\mathbf{Z}),\ K(\mathbf{Z_{ts}}, \mathbf{Z_p})] \tag{14}$$

where $\mathbf{X^{GP}}$ is the prediction from the GP regression process. The results of the GP model $\mathbf{X^{GP}} = [E^{\mathrm{e}}, E^{\mathrm{gCHP}}, E^{\mathrm{gf}}]$, as well as the associated standard deviations $[\sigma(E^{\mathrm{e}}), \sigma(E^{\mathrm{gCHP}}), \sigma(E^{\mathrm{gf}})]$, are obtained at a given time-step by implementing the Python *Scikit* package.

As stated, the GP regression predictions are assumed as potential solutions for the problem. Subsequently, the GP regression is exclusively utilized just after the length of the training data for a given time-step exceeds a minimum threshold. Besides, the predictions from the GP regression process are only used when the standard deviations associated with the GP obtained solutions are not outside an appropriate limit. In this paper, the GP results are considered acceptable estimates if $\sigma(E^{\mathrm{e}})$

$, \sigma(E^{gCHP}), \sigma(E^{gf}) \leqslant 5$kWh. Otherwise, the *bayesopt* is run, and the relative data corresponding to that time-step is kept for following GP regression.

3. Basic Concept and Different Configurations of a DRNN

In this section, we briefly discuss the mathematical background of DRNNs utilized for energy consumption prediction. Then, we explore different types of depth of the DRNN by plugging intermediate nonlinear layers. We aim to develop a DRNN configuration that yields the best performance on the task of electricity and heat demand prediction. The proposed DRNN structures are adaptive (i.e., it learns without intervention), and capable of modeling non-linear complexities and patterns in load profiles, as well as modeling the transient nature of energy consumption profiles without having explicit information about schedule variables.The associated codes and configuration files are publicly available for all our experiments in a GitHub Repository [42].

3.1. Mathematical Background

In general, a standard neural network is used to learn the mapping from an input vector to an output vector via several interconnected layers of processing elements. Fig. 3 represents a schematic of a feed-forward neural network. In a standard neural network, the relationship between the output and input(s) of each node of a layer, which is called a neuron, is determined by an activation function, as expressed in (15):

$$h_j^l = a\left(\sum_{m=1}^{N_{l-1}} w_{j,m}^l h_m^{l-1}\right); \forall j \in \{1,2,\ldots,N_l\}, \forall l \in \Gamma \tag{15}$$

where a(.) stands for activation function between the inputs feeding neuron j in layer l and the output heading for the next layer. The hyperbolic tangent, denoted by *tanh*, is a compelling activation function for neural network applications [43]. Proper values for the weighting factors $w_{j,m}^l$, at k^{th} training step, are determined based on a process named *training*. During the training process, the weights are updated using stochastic gradient descent as the partial derivatives of the set of parameters concerning its inputs and a learning rate β [43]:

$$\left(w_{j,m}^l\right)_k = \left(w_{j,m}^l\right)_{k-1} - \beta \frac{\partial e}{\partial\left(w_{j,m}^l\right)_{k-1}}; \forall j \in \{1,2,\ldots,N_l\}, \forall m \in \{1,2,\ldots,N_{l-1}\}, \forall l \in \Gamma, \forall k \in \{1,2,\ldots,N_k\} \tag{16}$$

The partial derivatives can be defined based on backpropagation approach explained in [44]. However, standard neural networks do not share features learned across different sequences of the input vector

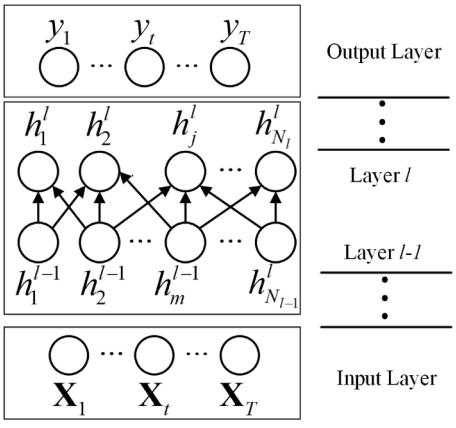

Fig. 3. Schematic diagram of a Feed-forward neural network.

[31]. As an extension to the standard neural networks, recurrent neural networks (RNNs) address this shortcoming by modeling temporal dependencies of targeted data. Fig. 4 shows a simplified scheme of applying RNN to sequential data. As illustrated, the value of activation function from previous time-steps is passed on to the future time intervals. Vector of activation function value at time interval t ($\mathbf{a}_t$), which accounts for temporal dependencies concerning previous time-step t–1, and output vector $\mathbf{Y}_t$, can be evaluated as [43]:

$$\mathbf{a}_t = \tanh\left(\mathbf{w}_{\mathbf{a},\mathbf{t}}^\top \odot [\mathbf{a}_{t-1}, \mathbf{X}_t] + \mathbf{b}_{\mathbf{a},\mathbf{t}}\right); \forall t \in \{1,2,\ldots,T\} \tag{17}$$

$$y_t = \sigma_s\left(\mathbf{w}_{\mathbf{Y},\mathbf{t}}^\top \odot \mathbf{a}_t + \mathbf{b}_{\mathbf{Y},\mathbf{t}}\right); \forall t \in \{1,2,\ldots,T\} \tag{18}$$

As shown in Eq. (17), the weighted input and weighted activation function from previous time step are activated in the network with the *tanh* activation function. In Eq. (18), σ_s is a sigmoid function applied to all the elements including the weighted activation function and the bias vector.

Nonetheless, typical RNNs do not embed long-term temporal dependencies because of an issue also called the *vanishing gradients* problem [31]. To deal with this problem, the long-short term memory (LSTM) unit is accommodated in RNNs, converting them from a standard form of network to a deep recurrent neural network [45]. A schematic of an LSTM network is given in Fig. 5, where the DRNN has a new vector $\mathbf{c}_t$ that stands for a memory cell. The memory cell provides a feature to remember short and long-term temporal dependencies of target data by deciding when to update associated values of the memory cell. In this regard, the cell consists of three different gates, namely update, forget, and output gates. An input enters separately into the forget gate, update gate, and output gate once it passes through the LSTM unit, and respectively yields the following vectors:

$$\mathbf{f}_t = \sigma_s\left(\mathbf{w}_{\mathbf{f},\mathbf{t}}^\top [\mathbf{a}_{t-1}, \mathbf{X}_t] + \mathbf{b}_{\mathbf{f},\mathbf{t}}\right); \forall t \in \{1,2,..,T\} \tag{19}$$

$$\mathbf{u}_t = \sigma_s\left(\mathbf{w}_{\mathbf{u},\mathbf{t}}^\top [\mathbf{a}_{t-1}, \mathbf{X}_t] + \mathbf{b}_{\mathbf{u},\mathbf{t}}\right); \forall t \in \{1,2,..,T\} \tag{20}$$

$$\mathbf{o}_t = \sigma_s\left(\mathbf{w}_{\mathbf{o},\mathbf{t}}^\top [\mathbf{a}_{t-1}, \mathbf{X}_t] + \mathbf{b}_{\mathbf{o},\mathbf{t}}\right); \forall t \in \{1,2,..,T\} \tag{21}$$

According to Fig. 5, the input goes also through activation function tanh to create $\widetilde{\mathbf{c}}_t$ based on (22), which is used to further update the value of the cell memory. At every timestep, the value of memory cell $\widetilde{\mathbf{c}}_\mathbf{t}$ is calculated as a candidate for replacing $\mathbf{c}_t$. Then, the LSTM unit calculates the element-wise multiplication between $\mathbf{u}_t$ and $\widetilde{\mathbf{c}}_t$ as well as $\mathbf{f}_t$ and $\mathbf{c}_{\mathbf{t-1}}$, to decide if the value of the unit must be updated or not. If the value of update gate is 1, then the candidate value $\widetilde{\mathbf{c}}_\mathbf{t}$ will be used as the new value of $\mathbf{c}_\mathbf{t}$.

$$\widetilde{\mathbf{c}}_t = \tanh\left(\mathbf{w}_{\mathbf{c},\mathbf{t}}^\top [\mathbf{a}_{t-1}, \mathbf{X}_t] + \mathbf{b}_{\mathbf{c},\mathbf{t}}\right); \forall t \in \{1,2,\ldots,T\} \tag{22}$$

$$\mathbf{c}_t = \mathbf{u}_t \odot \widetilde{\mathbf{c}}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1}; \forall t \in \{1,2,\ldots,T\} \tag{23}$$

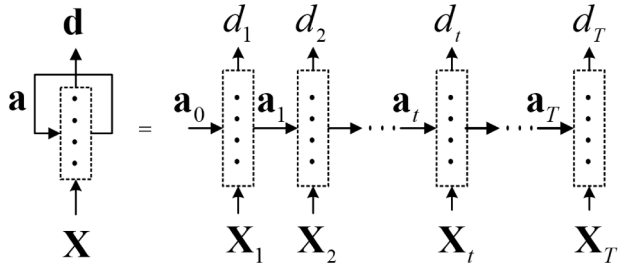

Fig. 4. A standard recurrent neural network structure (packed and unfolded in time).

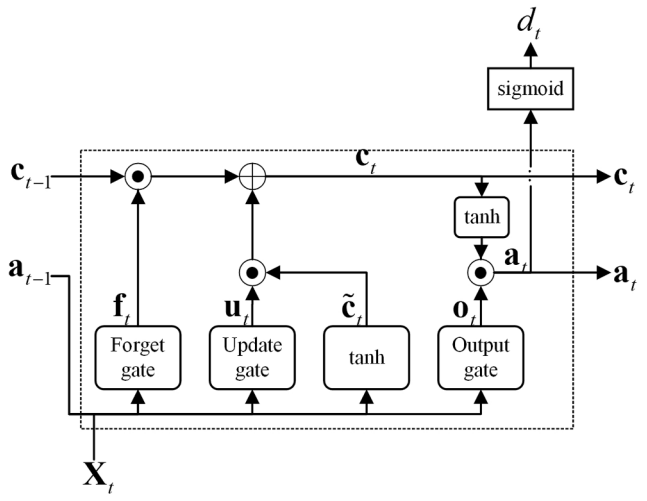

Fig. 5. Schematic of the LSTM unit.

Finally, the activation function value of the DRNN at time-step t can be computed as:

$$\mathbf{a}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t); \forall t \in \{1, 2, \ldots, T\} \tag{24}$$

In the proposed DRNN network, weighting vectors, namely $\mathbf{w_f}, \mathbf{w_u}, \mathbf{w_o}$, and $\mathbf{w_c}$, as well as bias vectors $\mathbf{b_f}, \mathbf{b_u}, \mathbf{b_o}$, and $\mathbf{b_c}$ are learned through backpropagation approach [44].

3.2. Different Configurations of a DRNN

In a typical DRNN structure, the relationship between energy consumption and the input vector is determined by a LSTM unit. However, a typical DRNN can be expanded through plugging intermediate layers in between input-to-memory, memory-to-memory, and memory to output layer. Multiple types of depth for a DRNN can now be reached by combining these transitions, each configuration with a different implication. We can also use several intermediate layers instead of one to reach even higher depth. A wide range of DRNN network configurations, from a conventional model to deeper variants, can be utilized on the task of load prediction. Generally speaking, the more complex the machine learning algorithm is, the more it will be able to represent a highly nonlinear function like a load profile. Yet, adding more layers might hurt because the optimization of hyperparameters becomes harder, and also the prediction function might overfit to the training data. In this paper, we train a standard DRNN model (Fig. 6(a)), a deep transition from input to memory cell (DTI-DRNN- Fig. 6(b)), a deep transition from memory to output (DTO-DRNN- Fig. 6(c)), and several stacked DRNN model (S-DRNN model- Fig. 6(d)) starting with two LSTM units of up to 4 layers. We also combine a DTO-DRNN with an S-DRNN with two LSTMs (S,DTO-DRNN- Fig. 6(e)). Altogether, 7 DRNN variants will be optimized and compared with SVM and GB regression in terms of performance.

3.3. Evaluation Metric

The root of mean squared error (RMSE) divided by the mean squared average of the demand, as presented below, is used to evaluate the accuracy of the proposed models.

$$e = \sqrt{\frac{\sum_{t=1}^{T} \left\| d_t^{a} - d_t \right\|^2}{\sum_{t=1}^{T} \left\| d_t^{ts} \right\|^2}} \tag{25}$$

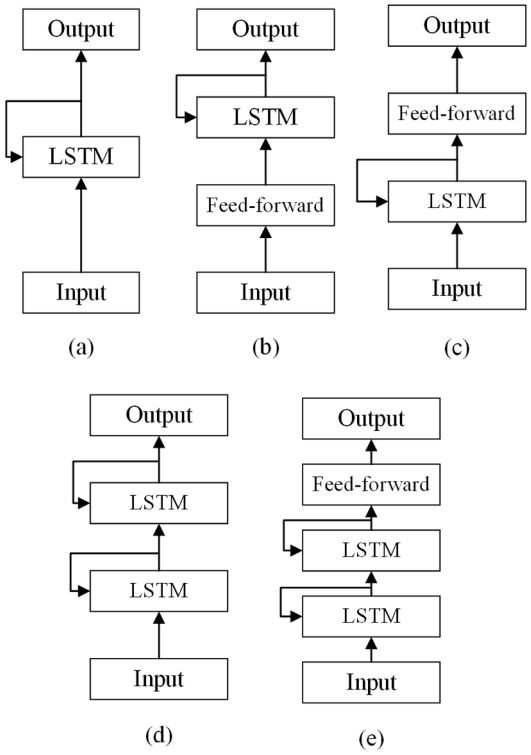

Fig. 6. Schematic of four DRNN. (a) A conventional DRNN. (b) Deep transition input (DTI) DRNN. (c) Deep Transition output (DTO) DRNN. (d) Stacked (S) DRNN. (e) Stacked and deep transition output combined (S,DTO).

where d_t^{a}, d_t, d_t^{ts} are the actual, predicted, and training set values of demands, respectively.

It is also worth to mention that we explored the empirical aspects of tuning the proposed configurations by searching through all the associated hyper-parameters. These hyperparameters include number of hidden units, optimization methods, model regularization, and batching. For a detailed outline of conducted hyper-parameter optimization, interested readers are invited to see the supplementary information available on [42].

4. Results and discussion

The proposed DRNN architectures are trained and tested on datasets of heating and electricity consumption profiles, aggregated from multiple load profiles of a residential customer. The datasets are gathered between December 2010 and November 2018 with a one-hour timestep resolution, thereby containing 140,160 measurements- from which 80% are used as the training set. In addition to the historical energy consumption values, a concatenation of meteorological variables is also needed. The weather variables are air pressure, temperature, and humidity plus wind speed and solar irradiation at the predetermined location. Another set of data used as an input to the DRNN models is the date-related variables which include (I) hours of the day (from 1 to 24), (II) days of a year, (III) the day in a given month and the month number and (IV) years in the time horizon of the project. A day in a given year is supposed as a concatenation of 365 binary variables; each variable signifies the numerical value of the item in the dataset. This representation of categorical variables is known as one-hot encoding [46], where all bits are zero except the one that acts as a binary flag for each day of a year. A similar implementation is considered for the other date-related variables. Further details about data's characteristics and preprocessing can be found in [47].

4.1. Selecting the best DRNN model

The configuration files were developed/run on top of a Tensorflow-based API called Keras. To evaluate the performance of the proposed models, RMSE relative to mean square average of electricity and heat demand (e_E, e_H) was used on TS as well as a detached cross-validation (CV) set. For all experiments, we train 35 epochs. The evaluation metrics are reported on epochs 5, 10, 20, 30, and 35. The epoch of the smallest error on the CV is stated/selected in Table 1, which gives an insight into the convergence speed and problem of overfitting. For instance, while deeper DRNNs ought to have a higher accuracy theoretically, they do not, possibly because of a more difficult optimization problem. Adam [48] was used as the default optimization method with an initial β of 0.001, multiplied with a gradient clipping of 10. The dropout criterion was also applied with a fixed rate of 0.1 + L2. We usually get a slower model using dropout; yet, the model improves significantly in terms of accuracy. We also implement a mini-batch gradient descent, with a size of 5600, to calculate gradients on random sets of instances. The feed-forward network used for transitions between input-to-hidden and hidden-to output is a 3-layer multi-layer perceptron with a fixed layer size of 15.

As represented in Table 1, two configurations perform better than other networks: a stacked DRNN with 3 LSTM units, together with an S-DRNN combined with DTO-DRNN. The S-DRNN network has the smallest CV-error both in electricity and heat prediction. Yet, the S,DTO-DRNN is selected for further experiments. This is because of our experiment with hidden layer sizes. The S-DRNN network is optimal when using 500 neurons, while the S-DRNN combined with DTO-DRNN performs at its best with a hidden layer size of 400. On the other hand, the number of parameters grow quadratically by increasing the size of the hidden layers. Since the S,DTO-DRNN network is much smaller and not so much worse, it is selected as the ultimate model.

4.2. Comparison study with SVM and GB Machine

Fig. 7 shows the hourly predictions of SVM, GB regression, and the S, DTO-DRNN model, along with the actual values of normalized electricity and heat demand for two days within the forecast period. Predicting one step ahead of demands with all three models is verified to be reasonably accurate, given that the prediction model is executed over a long-term horizon. Nevertheless, the introduced LSTM-based unit is more capable of fitting the load curves, as it can share features across frames in the time series.

To further demonstrate the potential ability of the models, the associated average errors corresponding to the electricity and heat demand forecast for six months in the prediction period are represented in Fig. 8. For electricity demand prediction, the monthly average errors associated with S,DTO-DRNN, SVM, and GB are calculated 21.1%, 26.5%, and 28.1%, respectively. For heat demand prediction, the reliability of the prediction obtained by SVM is rather weak. Also, the GB regression has difficulty to learn high variability and uncertainty of load profiles. That means the GB model is likely to stick in local optima when it is used for long-term energy consumption prediction, and the results might not be replicable. By and large, the introduced S, DT0-DRNN outperforms the alternatives as they fail to capture the time-dependent details of the load time series.

Table 1
Comparison of different configurations in terms of e_E, e_H on CV and TS–neurons fixed to 300 for each hidden layer, dropout 0.1 + L2, Adam, number of chunks = 10, reported on the best epoch.

Configuration	e_E-TS[%]	e_H-TS[%]	e_E-CV [%]	e_H-CV [%]	Epoch
DRNN	20.3	24.5	23.9	28.6	35
DTI-DRNN	19.6	22.8	24.9	27.6	35
DTO-DRNN	19.3	23.5	27.3	28.1	30
S-DRNN, 2LSTM	18.8	21.3	25.6	26.5	23
S-DRNN, 3LSTM	**18.3**	**19.2**	**21.5**	**23.6**	**19**
S-DRNN, 4LSTM	19.0	20.1	22.9	26.4	35
S,DTO-DRNN	**18.6**	**20.0**	**22.4**	**25.9**	**25**

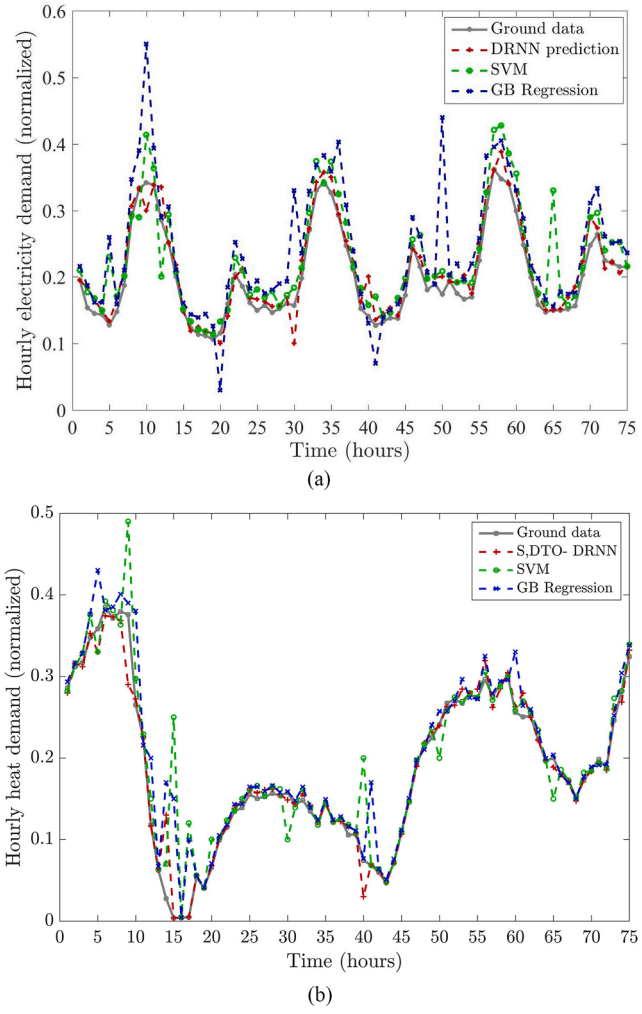

Fig. 7. Hourly predictions of energy profiles between February 22, 2017-February 24, 2017. (a) Electricity demand and (b) Heat demand.

4.3. Optimization of EH Using DRNN and GP Regression

Using coefficients adapted from [49] for the associated parameters in the EH model 2, the optimal sizes provided in Table 2 are obtained.

The final scheduling of the EH is then obtained using the combined algorithm of the GP regression, optimization scheme, and DRNN. Operation of the EH components are presented in Table 3, demonstrating the EH performance on a typical day in winter. Table 4 represents EH individual costs pertain to investment and operation as well as overall costs of the planning.

4.4. Performance of the proposed optimization method

To investigate the potential benefits of utilizing the introduced framework using GP regression model, computational time has been compared between two Cases: 1) with the introduced optimization scheme executed and 2) when the GP regression model is skipped. Fig. 9 checks the number of calls to the "*bayseopt*" package for Case 1 and Case 2 through the interested time horizon. As the figure represents, the number of calls to the "*bayesopt*" package in Case 1 is significantly less than that of Case 2 as we go ahead in time. This reduction in calls reduces the portion of time-steps in which the optimization process is

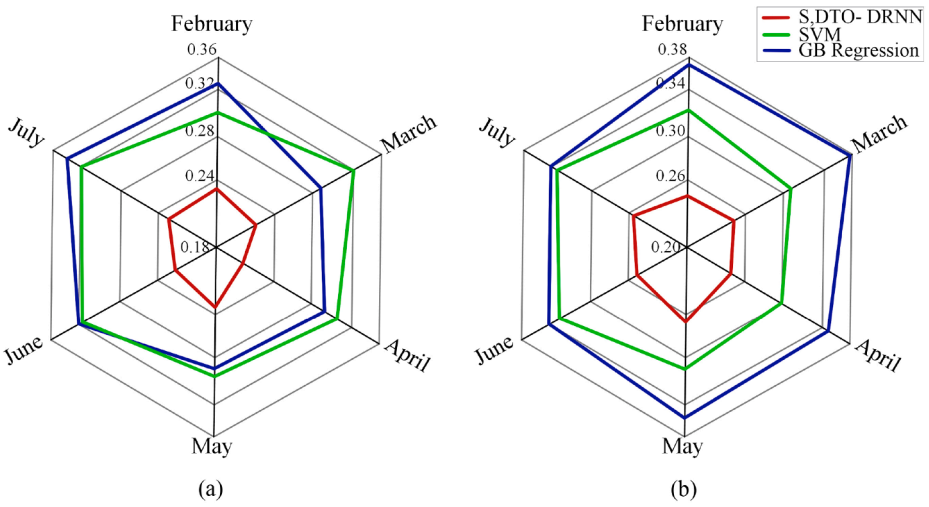

Fig. 8. Monthly average Error with respect to different months within the forecast period (a) Electricity demand and (b) Heat demand.

Table 2
Optimal capacity of the EH candidate equipment.

Candidate Equipment	Capacity (kW)
TR	900.00
CHP	300.00
F	1172.28

Table 3
The proposed EH scheduling on a typical winter day.

Time-step	E^{e}	E^{gCHP}	E^{gf}
t1	263.10	69.97	263.44
t2	185.92	91.83	361.12
t3	250.84	79.73	365.72
t4	238.05	81.96	425.55
t5	165.52	68.69	476.45
t6	162.76	75.60	505.28
t7	186.68	11.92	664.82
t8	254.13	65.76	701.83
t9	355.83	110.32	589.29
t10	459.75	111.32	406.48
t11	428.19	159.86	339.80
t12	320.92	54.27	266.85
t13	403.76	77.95	208.76
t14	393.90	75.46	233.68
t15	360.10	81.32	187.55
t16	333.12	71.32	155.43
t17	318.75	77.62	188.76
t18	244.57	78.69	176.45
t19	175.76	40.55	201.00
t20	163.21	51.32	233.54
t21	126.42	61.32	228.90
t22	85.88	71.32	198.72
t23	89.54	80.46	255.68
t24	63.64	95.94	233.73

Table 4
Planning costs of the EH with regard to the investment, operation, and total cost.

Type	Costs(M$)
Investment Cost	3.18
Operation Cost	6.34
Total Cost	9.52

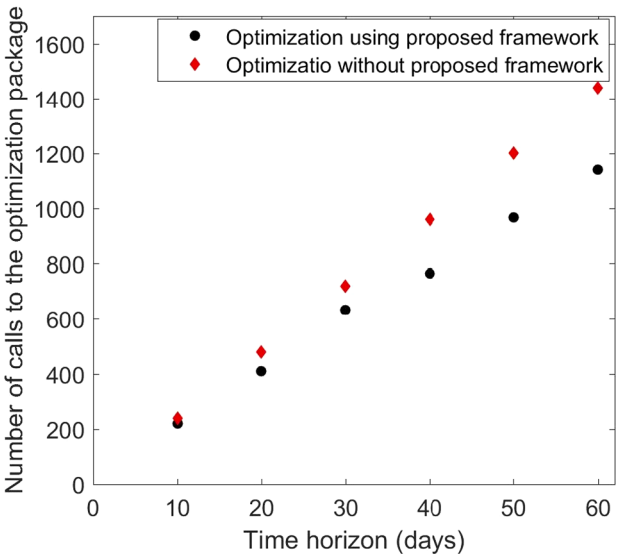

Fig. 9. Number of calls to the optimization package with and without using GP regression model.

actually executed, which in turn, leads to a significant improvement in overall computational efficiency. As the figure shows, the number of *bayesopt* calls over a time horizon of two months is 1,440. Employing the GP using *scikit* package, the number of calls is reduced to 1,140 which shows an appreciated reduction by a factor of 0.792. At the interested EL, the computational time acquired corresponding to Case 1 is 13.43 h compared to 28.41 h in Case 2. This reduction in overall computation by a factor of 0.473 is due to the implementation of the GP regression model.

5. Conclusion

A DRNN model is presented in this paper, which addresses the long-term loads changes to account for the optimal planning of EHs. To cope with this complicated nonlinear program, an algorithm is proposed to solve the resulting model. The simulation results for a predetermined location indicate that the proposed step-by-step framework can improve the overall computational efficiency of the EHOD optimization problem, provide a solution to incorporate long-term load prediction into the optimization process, and use GP regression to solve the problem

without explicitly performing the optimization. Test results for real data reveal the potential benefits of the proposed method in practical operation. The numerical test results also demonstrate the good performance of the introduced algorithm.

Based on the proposed methodology, several exciting directions are worth further investigating. How to solve the complicated/nonlinear EH models, while the convergence is guaranteed, is still an interesting topic. In cases where the specific loads' data is not obtainable while longer-term load forecast is concerned, an equivalent model would be helpful, which is a noteworthy topic that needs more study. Another significant, yet challenging issue is how to manipulate uncertainty of renewable energy resources at the input port of EHs. All these issues are open for future studies.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

Saman Taheri: Conceptualization, Methodology, Software, Writing - original draft. **Mohammad Jooshaki:** Conceptualization, Writing - review & editing. **Moein Moeini-Aghtaie:** Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Taheri S, Ghoraani R, Pasban A, Moeini-Aghtaie M. Stochastic framework for planning studies of energy systems: A case of EHs. IET Renewable Power Gener 2020;14(3). https://doi.org/10.1049/iet-rpg.2019.0642.

[2] Wang Yi, Zhang Ning, Zhuo Zhenyu, Kang Chongqing, Kirschen Daniel. Mixed-integer linear programming-based optimal configuration planning for energy hub: Starting from scratch. Appl Energy 2018;210:1141–50. ISSN 0306-2619.

[3] Khani H, Farag HEZ. Optimal day-ahead scheduling of power-to-gas energy storage and gas load management in wholesale electricity and gas markets. IEEE Trans Sustain Energy 2018;9(2):940–51. ISSN 1949-3037.

[4] Zhang R, Jiang T, Li F, Li G, Chen H, Li X. Coordinated bidding strategy of wind farms and power-to-gas facilities using a cooperative game approach. IEEE Trans Sustain Energy 2020;11(4):2545–55. ISSN 1949-3037.

[5] Clegg S, Mancarella P. Integrated modeling and assessment of the operational impact of power-to-gas (p2g) on electrical and gas transmission networks. IEEE Trans Sustain Energy 2015;6(4):1234–44. ISSN 1949-3037.

[6] Batić Marko, Tomašević Nikola, Beccuti Giovanni, Demiray Turhan, Vraneš Sanja. Combined energy hub optimisation and demand side management for buildings. Energy Build 2016;127:229–41. ISSN 0378-7788.

[7] Luo XJ, Fong KF. Development of integrated demand and supply side management strategy of multi-energy system for residential building application. Appl Energy 2019;242:570–87. ISSN 0306-2619.

[8] Khodaei Hossein, Hajiali Mahdi, Darvishan Ayda, Sepehr Mohammad, Ghadimi Noradin. Fuzzy-based heat and power hub models for cost-emission operation of an industrial consumer using compromise programming. Appl Therm Eng 2018;137:395–405. ISSN 1359-4311.

[9] Cau Giorgio, Cocco Daniele, Petrollese Mario, Knudsen Kær Søren, Milan Christian. Energy management strategy based on short-term generation scheduling for a renewable microgrid using a hydrogen storage system. Energy Convers Manage 2014;87: 820–831, 2014. ISSN 0196-8904.

[10] Ahmadi A, Mohammadi-Ivatloo B. Long-term wind power forecasting using tree-based learning algorithms. IEEE Access 2020;8:151511–22. https://doi.org/10.1109/ACCESS.2020.3017442. 19975025.

[11] Ahmad Tanveer, Chen Huanxin, Huang Ronggeng, Yabin Guo, Wang Jiangyu, Shair Jan, Akram Hafiz Muhammad Azeem, Mohsan Syed Agha Hassnain, Kazim Muhammad. Supervised based machine learning models for short, medium and long-term energy prediction in distinct building environment. Energy 2018;158:17–32. ISSN 0360-5442.

[12] Goodfellow Ian, Bengio Ian, Courville Aaron. Deep learning. MIT press; 2016. ISBN 0262337371.

[13] Paudel Subodh, Elmitri Mohamed, Couturier Stéphane, Nguyen Phuong H, Kamphuis René, Lacarrière Bruno, Le Corre Olivier. A relevant data selection method for energy consumption prediction of low energy building based on support vector machine. Energy Build 2017;138: 240–256. ISSN 0378-7788.

[14] Rafe Biswas MA, Robinson Melvin D, Fumo Nelson. Prediction of residential building energy consumption: A neural network approach. Energy 2016;117: 84–92. ISSN 0360-5442.

[15] Cui Can, Wu Teresa, Hu Mengqi, Weir Jeffery D, Li Xiwang. Short-term building energy model recommendation system: A meta-learning approach. Appl Energy 2016;172:251–63. ISSN 0306-2619.

[16] Touzani Samir, Granderson Jessica, Fernandes Samuel. Gradient boosting machine for modeling the energy consumption of commercial buildings. Energy Build 2018;158:1533–43. ISSN 0378-7788.

[17] Cheng Fan Fu, Xiao Chengchu Yan, Liu Chengliang, Li Zhengdao, Wang Jiayuan. A novel methodology to explain and evaluate data-driven building energy performance models based on interpretable machine learning. Appl Energy 2019;235:1551–60. ISSN 0306-2619.

[18] van der Meer Dennis W, Shepero Mahmoud, Svensson Andreas, Widén Joakim, Munkhammar Joakim. Probabilistic forecasting of electricity consumption, photovoltaic power generation and net demand of an individual building using Gaussian Processes. Appl Energy 2018;213: 195–207. ISSN 0306-2619.

[19] Xiaodong Xu, Wang Wei, Hong Tianzhen, Chen Jiayu. Incorporating machine learning with building network analysis to predict multi-building energy use. Energy Build 2019;186:80–97. ISSN 0378-7788.

[20] Fakhari I, Ahmadi P, Behzadi AB, Gholamian E. Comparative double and integer optimization of low-grade heat recovery from PEM fuel cells employing an organic Rankine cycle with zeotropic mixtures. Energy Conver Manage 2021;228. https://doi.org/10.1016/j.enconman.2020.113695 [In this issue].

[21] Sheikhi Aras, Rayati Mohammad, Ranjbar Ali Mohammad. Energy Hub optimal sizing in the smart grid; machine learning approach. In: 2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). IEEE; 2015. p. 1–5.

[22] Goudarzi Shidrokh, Anisi Mohammad Hossein, Kama Nazri, Doctor Faiyaz, Soleymani Seyed Ahmad, Sangaiah Arun Kumar. Predictive modelling of building energy consumption based on a hybrid nature-inspired optimization algorithm. Energy Build 2019. ISSN 0378-7788.

[23] Perera ATD, Wickramasinghe PU, Nik Vahid M, Scartezzini Jean-Louis. Machine learning methods to assist energy system optimization. Appl Energy 2019;243: 191–205. ISSN 0306-2619.

[24] Rayati Mohammad, Sheikhi Aras, Ranjbar Ali Mohammad. Optimising operational cost of a smart energy hub, the reinforcement learning approach. Int J Parallel Emergent Distrib Syst 2015;30(4):325–41. ISSN 1744-5760.

[25] de Oliveira Erick Meira, Cyrino Oliveira Fernando Luiz. Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods. Energy 2018;144: 776–788. ISSN 0360-5442.

[26] Fan Cheng, Sun Yongjun, Zhao Yang, Song Mengjie, Wang Jiayuan. Deep learning-based feature engineering methods for improved building energy prediction. Appl Energy 2019;240:35–45. ISSN 0306-2619.

[27] Ahmad Tanveer, Chen Huanxin, Guo Yabin, Wang Jiangyu. A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. Energy Build 2018;165:301–20. ISSN 0378-7788.

[28] Mason Karl, Duggan Jim, Howley Enda. Forecasting energy demand, wind generation and carbon dioxide emissions in Ireland using evolutionary neural networks. Energy 2018;155:705–20. ISSN 0360-5442.

[29] Agrawal Rahul Kumar, Muchahary Frankle, Tripathi Madan Mohan. Long term load forecasting with hourly predictions based on long-short-term-memory networks. In: 2018 IEEE Texas Power and Energy Conference (TPEC). IEEE; 2018. p. 1–6.

[30] Rajabi Mohammad Mahdi, Ketabchi Hamed. Uncertainty-based simulation-optimization using gaussian process emulation: application to coastal groundwater management. J Hydrol 2017;555:518–34.

[31] Rahman Aowabin, Smith Amanda D. Predicting heating demand and sizing a stratified thermal storage tank using deep learning algorithms. Appl Energy 2018;228:108–121. ISSN 0306-2619.

[32] Olsen Daniel Julius, Zhang Ning, Kang Chongqing, Ortega-Vazquez Miguel A, Kirschen Daniel S. Planning Low-Carbon Campus Energy Hubs. IEEE Trans Power Syst 2018. ISSN 0885-8950.

[33] Bahrami Shahab, Aminifar Farrokh. Exploiting the potential of energy hubs in power systems regulation services. IEEE Trans Smart Grid 2018. ISSN 1949-3053.

[34] Majidi Majid, Zare Kazem. Integration of smart energy hubs in distribution networks under uncertainties and demand response concept. IEEE Trans Power Syst 2018;34(1):566–74. ISSN 0885-8950.

[35] Zhang Xiaping, Shahidehpour Mohammad, Alabdulwahab Ahmed, Abusorrah Abdullah. Optimal expansion planning of energy hub with multiple energy infrastructures. IEEE Trans Smart Grid 2015;6(5):2302–11. ISSN 1949-3053.

[36] Shabunko V, Lim CM, Mathew S. EnergyPlus models for the benchmarking of residential buildings in Brunei Darussalam. Energy Build 2018;169:507–16. ISSN 0378-7788.

[37] Cassel Maico, Lima F. Evaluating one-hot encoding finite state machines for SEU reliability in SRAM-based FPGAs. In: 12th IEEE International on-line testing symposium (IOLTS'06), pages 6—-pp. IEEE; 2006a. ISBN 0769526209.

[38] Mone CD, Chau DS, Phelan PE. Economic feasibility of combined heat and power and absorption refrigeration with commercially available gas turbines. Energy Convers Manage 2001;42(13):1559–73. ISSN 0196-8904.

[39] CHP Market Directory, Overview of companies in the CHP sector. www.energieagentur.nrw; 2017.

[40] Rasmussen Carl Edward, Nickisch Hannes. Gaussian processes for machine learning (GPML) toolbox. J Mach Learn Res 2010;11(Nov):3011–5.

[41] Williams Christopher KI, Rasmussen Carl Edward. Gaussian processes for machine learning, volume 2. MA: MIT press Cambridge; 2006.
[42] Taheri Saman, Moeini-aghtaie Moein, Joshaki Mohammad. A GitHub repository with configuration files for DRNN experiments. https://github.com/samantaheri71/LSTM-PAPER; 2020.
[43] Glorot Xavier, Bengio Yoshua. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics; 2010. p. 249–56.
[44] Andrychowicz Marcin, Denil Misha, Gomez Sergio, Hoffman Matthew W, Pfau David, Schaul Tom, et al.. Learning to learn by gradient descent by gradient descent. In: Advances in neural information processing systems; 2016. p. 3981–9.
[45] Graves Alex, Mohamed Abdel-rahman, Hinton Geoffrey. Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE; 2013. p. 6645–9.
[46] Cassel Maico, Lima F. Evaluating one-hot encoding finite state machines for SEU reliability in SRAM-based FPGAs. In: 12th IEEE International on-line testing symposium (IOLTS'06), pages 6—-pp. IEEE; 2006b. ISBN 0769526209.
[47] Moeini-Aghtaei M, Taheri S, Joshaki M. Characteristics of the Datset used in the Deep Recurrent Neural Network & Long-Short Term Memory. http://sina.sharif.edu/ moeini/wp-content/uploads/2020/07/Data-characteristics.pdf; 2020.
[48] Kingma Diederik P, Ba Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980; 2014.
[49] Pazouki Samaneh, Haghifam Mahmoud Reza. Optimal planning and scheduling of energy hub in presence of wind, storage and demand response under uncertainty. Int J Electrical Power Energy Syst 2016;80: 219–239. ISSN 01420615.