

# A self-adaptive deep learning model for building electricity load prediction with moving horizon

Xiaojun Luo, Lukumon O. Oyedele \*

Big Data Enterprise and Artificial Intelligence Lab (Big-DEAL), University of the West of England, Bristol, United Kingdom



## ARTICLE INFO

### Keywords:

Moving horizon  
Self-adaptability  
Prediction  
Deep learning  
Deep neural network  
Particle swarm optimisation

## ABSTRACT

A self-adaptive deep learning model powered by ranking selection-based particle swarm optimisation (RSPSO) is developed to predict electricity load in buildings with moving horizons. The main features of the load prediction model include its self-adaptability, repeatability, robustness and accuracy. In real-world building applications, the relationship among weather data, time signature and electricity load is quite complicated. In the proposed self-adaptive deep learning model, a deep learning model with multiple hidden layers is implemented to improve prediction precision. Meanwhile, RSPSO is implemented to select the network's optimum architecture, which involves discrete variables (i.e. the quantity of neurons in each layer and the quantity of hidden layers) and categorical variables (i.e. activation function in each layer and learning approach). Moreover, the moving horizon approach is adopted to update the architecture and structure of the dynamic deep learning model while enabling its capability in capturing the latest featuring patterns in the electricity load of the building. The proposed load prediction model is tested with the local meteorological profile and electricity load of an educational building. The self-adaptive load prediction model is identified to be the most effective at forecasting the next horizon's energy consumption, while its prediction performance would deteriorate with the increase of time. The mean squared error, mean absolute error, and coefficient of determination of the proposed prediction model are within the range of 4.48 kW–11.23 kW, 1.28 kW–2.31 kW and 97.52%–98.92%, respectively, demonstrating its prediction accuracy and repeatability. When Gaussian white noise is added to meteorological data, the increase in mean absolute error is within the range of 2.08%–15.33%, demonstrating the robustness of the proposed prediction model in overcoming uncertainty in the weather forecast. Therefore, the proposed accurate, robust, repeatable and self-adaptive load prediction model can be rooted in practical energy management systems thus facilitate building operation and system control.

## 1. Introduction

In recent years, the global energy demand has been primarily boosted owing to population growth, economic development and climate change. During 2016–2040, the primary energy consumption is expected to rise at a multifactorial yearly growth rate of one percent (Sieminski, 2017). As human beings generally spend a considerable quantity of time indoor, building energy consumption is growing swiftly. The quantity of electricity fed into the electricity grid must always equal the quantity consumed to keep its frequency and voltage stable (Kennedy & Eberhart, 1995, 1997). Therefore, accurate estimation of electricity consumption for each building is essential, while the grid-scale electricity should be able to satisfy the electricity consumption of each building connected to the electricity grid. Underestimation of building electricity consumption may cause frequency and voltage drop, power plants switch off and even system collapse. On the other hand, overestimation leads to power plants disconnection

and unnecessary operating costs (Luo & Fong, 2019). Hence, a highly accurate, robust, repeatable and adaptive electricity load prediction model will support building energy management and fault detection of building appliances and power network infrastructure (Roman Cardell, 2020).

### 1.1. Related works

In recent years, there have been many deep learning and machine learning-based building energy prediction models. Sheraz et al. (Aslam et al., 2021) presented an overview of deep learning-based methods for the prediction of electrical power production from solar panels and wind turbines, as well as the electrical power consumption of buildings. The reviewed deep learning models include artificial neural network (ANN), convolutional neural network (CNN), deep neural network (DNN), autoencoder, deep belief network, long short term memory (LSTM) and recurrent neural network (RNN). It points out that lowering

\* Corresponding author.

E-mail addresses: [Xiaojun.luo@uwe.ac.uk](mailto:Xiaojun.luo@uwe.ac.uk) (X. Luo), [l.oyedele@uwe.ac.uk](mailto:l.oyedele@uwe.ac.uk), [Ayolook2001@yahoo.co.uk](mailto:Ayolook2001@yahoo.co.uk) (L.O. Oyedele).

computational cost and complexity as well as uncertainty quantification will be significant future research directions. Liang et al. (Zhang et al., 2021) presented an overview of machine learning-based electricity load prediction methods. It covered support vector machine (SVM), regression models, neural network, tree-based algorithm, deep learning, extreme learning, Bayesian networks, ensemble methods and fuzzy time series methods. It points out that robust autotuning algorithms will be crucial to vast industrial applications of electricity load prediction. From a systematic review of electrical load forecasting models, Corentin et al. (Kuster et al., 2017) revealed that ANN, SVM, and autoregressive integrated moving average are more appropriate for short-term energy prediction.

From the above literature review of deep learning and machine learning methods for predicting building energy consumption, it is seen that neural network-based models are the most effective ones for building electricity consumption prediction. It is mainly because of its capability of adequately, arbitrarily and precisely approximating the comprehensive nonlinear association among the output and input datasets of a complex system (Wang & Yin, 2008). As a result, various ANN-based predictive models have been developed for building energy consumption thanks to its capability of self-learning. Luo et al. (Luo & Fong, 2019) developed an ANN forecasting model for office building's heating and cooling loads. Historical weather profile, historical heating and cooling load, along with temperature sensors measurements from various rooms of the building, are adopted as input datasets to the ANN prediction model. The optimal neuron numbers were verified and chosen for different sub-zones of the building. Kusiak et al. (Kusiak et al., 2010) proposed a multi-layer perceptron-based ANN prediction. Different quantity of neurons and activation functions were tried in a single hidden layer. Deb et al. (Deb et al., 2016) proposed a feedforward structure-based ANN prediction model for building cooling load. The trial-and-error approach was adopted to choose the optimum quantity of neurons in the hidden layer. Ahmad et al. (Ahmad et al., 2019) proposed three ANN models to predict the cooling load of the building. Gaussian process regression approach, linear regression approach, and Levenberg–Marquardt backpropagation approach were adopted separately in each ANN model, while the different number of neurons was also tested in each prediction model. Yang et al. (Yang, Hugues et al., 2005; Yang, Rivard et al., 2005) and Wang et al. (Wang et al., 2018) also evaluated the performance of ANN models for forecasting building heating and cooling loads. In these two ANN models, the quantity of neurons was selected by two different empirical equations (Hecht-Nielsen, 1989 and NeuroShell, 1993). Moon et al. (Moon et al., 2019) constructed diverse ANN models for electrical energy consumption. The ANN-based prediction models were designed with a different combination of the activation functions and quantity of hidden layers. It was concluded that better performance could be achieved when scaled exponential linear units were used as the activation function and when five hidden layers were adopted. Luo et al. (Luo, Oyedele, Ajayi et al., 2020; Luo, Oyedele, Ajayi, Akinade, Delgado et al., 2020; Luo, Oyedele, Ajayi, Akinade, Owolabi et al., 2020; Luo, Oyedele, Akinade et al., 2020) constructed an ANN model for predicting building cooling demand. The year-round cooling demand was grouped by k-means clustering, while several ANN sub-models were trained for each group. The quantity of neurons in each sub-model was decided by a trial-and-error process. Kim et al. (Kim et al., 2020) identified that the ANN model is more precious and steady than linear regression methods in forecasting electricity load. Renzhi et al. (Lu & Hong, 2019) developed a DNN model to forecast price and load in smart grids. The difference between DNN and ANN models is that there are multiple hidden layers in the former, while there is generally one single hidden layer in the latter. The quantity of neurons and hidden layers was decided by the accuracy test. It was found that the multiple layered-DNN models had greater prediction precision than a single layered-ANN model.

In the above-mentioned ANN prediction models, the model architecture is mainly determined through experiences, trial-and-error

processes and experiments. To enhance computational efficiency, grid search, random search and Bayesian optimisation approaches were adopted to tune the neural network's architecture using cross-validation. For example, Gustavo et al. (Paneiro Gustavo et al., 2020) developed an ANN model to analyse environmental effects caused by blast-induced ground vibrations, in which grid search and cross-validation are adopted to choose the optimal number of neurons, activation function, optimiser and epochs. Kalliola et al. (Kalliola et al., 2021) developed an ANN model to predict real estate prices, in which the hyper-parameters were fine-tuned by random search. Zhou et al. (Zhou et al., 2021) developed an adaptive hyper-parameter tuning model to forecast ship fuel consumption. Bayesian optimisation was implemented to choose the optimum architecture of ANN, SVM, least absolute shrinkage, random forest, and selection operator. However, in practical application, the limitation of computational time in predicting real-time electricity consumption makes it challenging to find the optimal set of hyper-parameters by sweeping through the parameter space. On the other hand, the Gaussian distribution of the optimisation function is the essential requirement of Bayesian optimisation, which is not the case in many prediction datasets.

Some popular evolutionary optimisation algorithms were adopted to improve the convergence rate and computational efficiency of neural network-based prediction models. The evolutionary optimisation algorithms mainly include particle swarm optimisation (Li et al., 2015; Muralitharan et al., 2018), genetic algorithm (Li et al., 2018; Ruiz et al., 2018) and teaching–learning algorithm (Li et al., 2018). However, they were primarily used to amend threshold values and weighting factors of neural networks. Our previous study (Luo, Oyedele, Ajayi, Akinade, Owolabi et al., 2020) is the first paper to adopt an evolutionary algorithm in choosing DNN architecture to the best of our knowledge. Historical weather data and time signatures were adopted as input datasets to the DNN model, while the historical energy consumption profile was not accounted. The number of neurons in each hidden layer was chosen between 5 and 10, which might not be sufficient to disclose the complex relationship between historical weather data, time variables, past energy consumption data and future energy consumption. Moreover, the robustness of the prediction model in overcoming uncertainty in weather forecast data was not tested (Weather.com, 2021; Weatheronline, 2021). More importantly, the self-adaptive ability in capturing the latest featuring patterns in building energy consumption for practical online energy prediction was not investigated. Last but not least, the categorical variables such as activation function and learning approach were treated as integer variables during GA optimisation, which is not close to the reality since there is no numerical relationship among different activation functions and learning approaches. Meanwhile, the number of decision variables was constant. For example, when there are fewer than 4 hidden layers, there are still 4 decision variables for the number of neurons while 5 decision variables for activation function. The extra decision variables were designed for the GA algorithm but actually optimised without any purpose.

In order to enhance the accuracy for online day-ahead heating demand forecast, Felix et al. (Bünning et al., 2020) suggested integrating the historical data with newly collected data at each day to train the DNN model. However, the architecture of the DNN model was fixed. Jin et al. (Yang, Hugues et al., 2005; Yang, Rivard et al., 2005) proposed two adaptive ANN models which can adjust themselves to unforeseen pattern changes in the input dataset using accumulative and sliding window techniques, respectively. Although the input datasets were updated along with the time being, the architecture of ANN was fixed.

### 1.2. Identification of knowledge gaps

From the above literature review, the following knowledge gaps are identified from the state-of-art research works regarding building energy prediction:

- **Lack of effective approach to finding optimal architecture of neural network:** Neural network's optimal architecture is dependent on the features of input and output datasets. Inappropriate architecture may deteriorate the accuracy of the prediction model. In neural network-based energy prediction models, no matter with a single hidden layer or multiple hidden layers, the neural network's architecture was decided by experience or trial-and-error process. For example, the quantity of neurons in each layer was decided by empirical equations (Lu & Hong, 2019; Wang et al., 2018; Yang, Hugues et al., 2005; Yang, Rivard et al., 2005) or through the tedious trial-and-error process (Deb et al., 2016; Kusiak et al., 2010; Lu & Hong, 2019; Luo, 2020; Moon et al., 2019; Ruiz et al., 2018), numerous experiments (Ahmad et al., 2019; Luo et al., 2019), grid search (Paneiro & Rafael, 2021), random search (Kalliola et al., 2021) and Bayesian optimisation (Zhou et al., 2021). Levenberg–Marquardt back-propagation was generally used as the learning process, while different activation functions for hidden and output layers were chosen through experience or trial-and-error process. However, empirical equations are not suitable for multiple hidden layers, heuristic trial-and-error approach, grid search, and random search is time-consuming and may still not be able to reach the global optimum, while Gaussian distribution of optimisation function is the essential requirement of Bayesian optimisation, which is not the case in many prediction datasets.
- **The state-of-the-art neural network models lack self-adaptivity:** Most energy load prediction models were trained using fixed-term historical data. The quantity of hidden layers and neurons, activation function, weighting factors of interconnected neurons of the prediction models did not change afterwards. The quantity of hidden layers and neurons determine the dimension and length of the proposed DNN model. The activation function affects the comprehensive relationship between output and input datasets. The optimisation approach decides the training process of weighting factors. Therefore, the prediction model lacks self-adaptive capability, while the accuracy may decrease when new features occur in building energy consumption.
- **Lack of evolutionary algorithm for optimising discrete and categorical variables in the neural network.** In previous hybrid evolutionary optimisation and machine learning prediction models, evolutionary algorithms were generally adopted to assist the training process of the DNN weighting coefficients. The weighting factors could be treated as continuous variables while the total quantity of design variables was constant due to the fixed architecture of the DNN model. Therefore, the conventional and basic versions of evolutionary algorithms were commonly adopted.
- **Lack of robustness in tackling uncertainties in weather forecast:** Most of the energy consumption prediction models were assessed by accuracy and repeatability. The prediction performance was generally evaluated by applying the prediction model on training and testing datasets collected from weather stations and energy management systems. However, the building energy prediction relies on the weather forecast, which there may exist uncertainty. The robustness of the prediction model in dealing with such forecast uncertainty was generally not considered.

### 1.3. Innovation and contribution

To overcome the above-mentioned knowledge gaps, the innovation and contribution of this study are identified as follows:

- **Effective evolutionary algorithm to optimising the architecture of neural network:** To help decrease computational time as well as improve prediction accuracy and repeatability, ranking selection-based PSO (RSPSO) is used to optimise DNN's architecture, including the discrete variables (i.e. the quantity of hidden layers and quantity of neurons in each hidden layer) and the categorical variables (i.e. activation function and the learning).

- **RSPSO to optimise discrete and categorical variables in a neural network:** To facilitate the optimisation process of both discrete and categorical variables, the revised RSPSO is adopted. For discrete variables, the feasible positions are ranked based upon the fitness value of its objective function. For categorical variables, the historical best value is adopted. The different selection probability schemes are adopted for discrete and categorical variables, respectively.
- **Moving horizon to improve self-adaptability:** To enable the proposed predictive model to be self-adaptive to online and dynamic electricity load patterns, the moving horizon approach is adopted. At the beginning of each month, the new DNN model's optimal architecture is decided by PSO. Meanwhile, the DNN model with the new optimal architecture is trained using the latest 12 months' meteorological and energy data to determine corresponding weighting factors.
- **Improved robustness in tracking uncertainties in weather forecast:** To evaluate the impacts of uncertainty in weather forecast profile on building energy consumption prediction, Gaussian white noises are added on the input datasets to exhibit the robustness of the proposed prediction model.

The rest of this paper is organised as follows. Section 2 introduces the data collection and pre-processing approach using a case study. Section 3 illustrates the methodology for the proposed self-adaptive prediction model. Section 4 discusses the prediction performance, with the focus on its accuracy, repeatability, robustness and self-adaptive capability. Section 5 illustrates how the proposed model can be used in practical moving horizon energy prediction. The last section summarises the contributions of this study, as well as discusses limitations and possible follow-up work.

## 2. Data collection and pre-processing

The proposed self-adaptive prediction model is utilised at a real-life campus building to assess its effectiveness. Upon collection and pre-processing of raw data, it serves as the historical database for training, testing and evaluation purposes.

### 2.1. Collection of raw data

To afford sufficient data for training, testing and evaluating for the proposed self-adaptive prediction model, the building electricity load profile  $E(t)$  is collected from the Northavon House over one and a half years (i.e. Jul. 2018–Dec. 2019) at each hour. The collected electricity load profile is presented in Fig. 1. The Northavon House is a campus building in University of the West of England, Bristol, the United Kingdom, which mainly contains both scheduled classrooms, non-scheduled meeting rooms and scheduled office rooms.

Building thermal performance is generally affected by meteorological data and time index (Luo, Oyedele, Ajayi, Akinade, Owolabi et al., 2020, Luo XJ et al., 2020e). The meteorological profile involves:

- dry-bulb temperature
- dew-point temperature
- cloud ratio
- wind velocity
- and solar radiation

Meanwhile, time index stands for the stamps series of time, including:

- hour
- day
- and month

Therefore, during the same period of 1 Jul 2018–31 Dec 2019, the meteorological profile measured at the Bristol weather station is used as a weather database [Weatheronline].

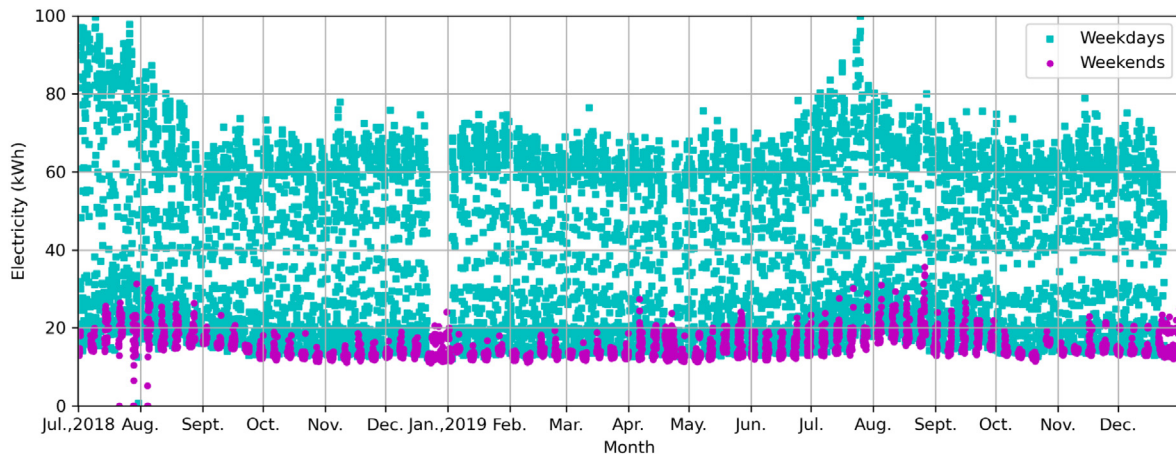


Fig. 1. Electricity consumption profile.

**Table 1**  
One-hot encoding results of time variables.

Time variable	Results of one-hot encoding
Day ( $d$ )	$d'$
Mon.	00000001
Tue.	00000010
Wed.	00000100
Thu.	00001000
Fri.	00010000
Sat.	00100000
Sun.	01000000
Bank holiday	10000000
Month of the year ( $m$ )	$m'$
January	000000000001
February	000000000010
March	000000000100
April	000000001000
May	000000010000
June	000000100000
July	000001000000
August	000010000000
September	000100000000
October	001000000000
November	010000000000
December	100000000000

## 2.2. Pre-processing of raw data

Due to the value range of different meteorological data, the min-max scaling approach (Luo, Oyedele, Akinade et al., 2020) is adopted to normalise dry-bulb temperature  $\alpha_1$ , dew-point temperature  $\alpha_2$ , wind velocity  $\alpha_3$ , cloud ratio  $\alpha_4$  and solar radiation  $\alpha_5$  into the range of [0-1]:

$$\alpha'_j(t) = \frac{\alpha_j(t) - \min_{1 \leq t \leq 365 \times 24} \alpha_j(t)}{\max_{1 \leq t \leq 365 \times 24} \alpha_j(t) - \min_{1 \leq t \leq 365 \times 24} \alpha_j(t)} \quad (1)$$

Owing to the cyclical nature of hour  $h(t)$ , it is converted into the corresponding sine value  $h_s(t)$  and cosine value  $h_c(t)$ . For example, in fact, there is only one hour difference between 23:00 and 0:00. However, there would be 23 h difference if categorical or original hour variables are adopted (Mena et al., 2014).

$$h_s(t) = \sin \frac{2\pi h(t)}{24} \quad (2)$$

$$h_c(t) = \cos \frac{2\pi h(t)}{24} \quad (3)$$

Bank holidays are regarded as an extra day type owing to their different occupancy behaviour and corresponding operating performance of lighting, office equipment and heating-and-cooling systems. Meanwhile, day  $d(t)$  and month  $m(t)$  is converted into binary variables

using a one-hot encoding approach (Roman Cardell 2020), as shown in Table 1. The min-max scaling approach is also adopted to convert the energy data into the range of [0-1]:

$$E'(t) = \frac{E(t) - \min_{1 \leq t \leq 365 \times 24} E(t)}{\max_{1 \leq t \leq 365 \times 24} E(t) - \min_{1 \leq t \leq 365 \times 24} E(t)} \quad (4)$$

## 2.3. Composition of input datasets

In summary, the input datasets consist of meteorological profile (i.e. dry-bulb temperature, wet-bulb temperature, wind velocity, cloud ratio and solar radiation), time index (i.e. hour, day, and month) as well as energy data over the past 24 h. Therefore, the input dataset  $X_t$  at time step  $t$  is composed of 33 elements:  $X_t = \{x_{i,t} | i = 1, 2, \dots, 33\}$ ,  $x_{1,t} = T'_{db}(t)$ ,  $x_{2,t} = T'_{dew}(t)$ ,  $x_{3,t} = V'(t)$ ,  $x_{4,t} = \xi'(t)$ ,  $x_{5,t} = R'(t)$ ,  $x_{6,t} = h_s(t)$ ,  $x_{7,t} = h_c(t)$ ,  $x_{8,t} = d'(t)$ ,  $x_{9,t} = m'(t)$ ,  $x_{9+j,t} = E'(t-j)$ . The output dataset  $z_t$  is the electricity consumption at the corresponding time step, and  $z_t = E(t)$ .

## 3. Self-adaptive deep learning model powered by RSPSO

The RSPSO-powered self-adaptive deep learning model is proposed to enhance the accuracy, robustness and repeatability of building electricity consumption prediction. The main priority of the RSPSO-powered DNN model is its advanced searchability, in which RSPSO is collaboratively hybrid with the DNN model to figure out its optimal architecture. Moreover, the moving horizon is adopted to catch up with the latest patterns in building energy consumption. Simultaneously, the architecture and structure of the deep learning model are determined using the updated training datasets and RSPSO.

### 3.1. Theory of the DNN

The DNN models consist of an input layer, multiple hidden layers and an output layer. The number of neurons in each hidden layer indicates the length of the network, while the number of hidden layers determines its dimension. In general, with a larger quantity of hidden layers and corresponding neurons, DNN is able to deliver a multi-level depiction of the dataset, thus alleviate the local-optimum problem in data approximating (Deng & Yu, 2013). The graph of the DNN model is illustrated in Fig. 2.

#### 3.1.1. DNN algorithm

As discussed in Section 2.2, the database is composed of input dataset  $X$  and desired output dataset  $Z$ , and  $X = \{x_{i,t} | i = 1, 2, \dots, 33; t = 1, 2, \dots, T\}$ ,  $Z = \{z_t | t = 1, 2, \dots, T\}$ , where  $T$  is the total quantity

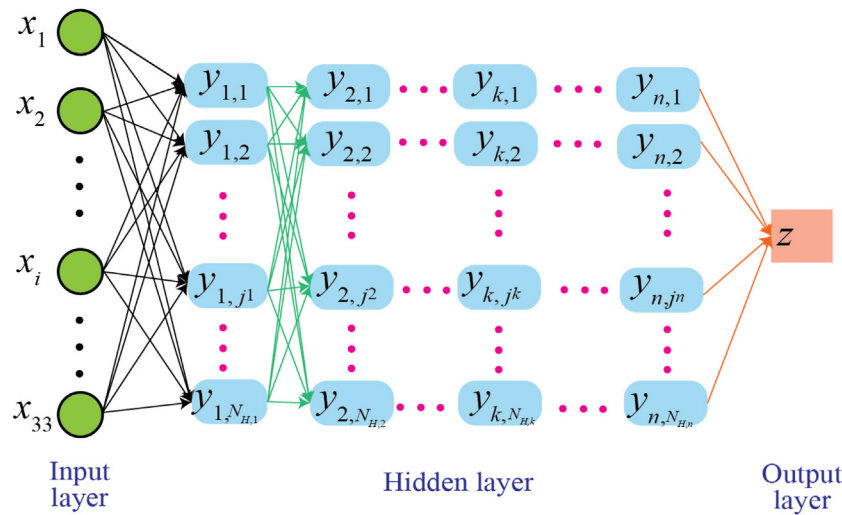


Fig. 2. Structure of DNN model.

of training time steps. In the first hidden layer, the  $j_1^{th}$  ( $j_1 | j_1 = 1, 2, \dots, N_{H,1}$ ) neuron can be calculated as:

$$y_{1,j_1} = f\left(\sum_{i=1}^{33} \sum_{t=1}^T (w_{IN,i,1,j_1} x_{i,t})\right) \quad (5)$$

while the  $j^k$  neuron in the  $k^{th}$  ( $n \geq k \geq 2$ ) the hidden layer can be obtained as:

$$y_{k,j_k} = f\left(\sum_{j_{k-1}=1}^{N_{H,k-1}} (w_{k-1,j_{k-1},k,j_k} y_{k-1,j_{k-1}})\right) \quad (6)$$

The neuron  $z$  in the output layer is calculated as:

$$z_t = f\left(\sum_{j_n=1}^{N_{H,n}} (w_{n,j_n,O} y_{n,j_n,t})\right) \quad (7)$$

where  $w$  is the weighting coefficients among connected neurons,  $i$  is the index of neurons in the input layer,  $k$  is the index of hidden layers,  $j_k$  is the index of neurons in the  $k^{th}$  hidden layer, while  $f$  is the activation function.

### 3.1.2. Activation function

Four typical activation functions (i.e. sigmoid, hyperbolic tangent, rectified linear unit and exponential linear unit) are tested to achieve better prediction performance for the DNN model.

### 3.1.3. DNN training algorithm

The objective of training DNN is to minimise the mean absolute error  $MAE$  between the predicted  $\hat{z}$  and the actual measurement  $z$ . The training process is conducted by fine-tuning the weighting factors set  $W = \{w_{IN,i,1,j_1} | N_{IN} \geq i \geq 1, N_{H,1} \geq j_1 \geq 1\} \cup \{w_{k-1,j_{k-1},k,j_k} | n \geq k \geq 2, N_{H,k-1} \geq j_{k-1} \geq 1\} \cup \{w_{n,j_n,O} | N_{H,n} \geq j_n \geq 1\}$ .

Four different training algorithms are used to find out the optimal weighting coefficients of the neural network, such as adaptive moment estimation (ADAM), stochastic gradient descent (SGD), ADAMAX, and Nesterov-accelerated adaptive moment estimation (NADAM).

### 3.1.4. Performance indicator

As mentioned in Section 3.1.3, mean absolute error ( $MAE$ ) is adopted to evaluate the prediction performance:

$$MAE = \frac{1}{T} \sum_{t=1}^T (z_t - \hat{z}_t) \quad (8)$$

Moreover, the mean squared error ( $MSE$ ) is adopted to verify the prediction accuracy further:

$$MSE = \frac{1}{T} \sum_{t=1}^T (z_t - \hat{z}_t)^2 \quad (9)$$

Furthermore, the coefficient of determination  $R^2$  is adopted to assess the repeatability of the proposed prediction model:

$$R^2 = \frac{\left[\sum_{t=1}^T \left(z_t - \frac{\sum_{t=1}^T z_t}{T}\right) \left(\hat{z}_t - \frac{\sum_{t=1}^T \hat{z}_t}{T}\right)\right]^2}{\sum_{t=1}^T \left(z_t - \frac{\sum_{t=1}^T z_t}{T}\right) \sum_{t=1}^T \left(\hat{z}_t - \frac{\sum_{t=1}^T \hat{z}_t}{T}\right)} \quad (10)$$

### 3.2. Selection of decision variables

The architecture of the DNN prediction model is generally problem-dependent, which will have significant implications on DNN performance. Selection-based PSO would be adopted to enable the automatic process of finding the optimal architecture from a multidimensional space. The whole-set hyperparameters of the DNN model can be selected using RSPSO algorithms to originate a well-generalised architecture, even though the searching process would be considerably complex. On the other hand, if only a small part of optimal hyperparameters are selected, the performance improvement would be quite slight compared to the existing fixed-architecture models. Under-fitting may be caused by small quantity of hidden layers and small quantity of neurons in each hidden layer. On the other hand, over-fitting will be resulted from large quantity of neurons and hidden layers. Connecting performance among each neuron and layer is determined by the corresponding activation function as it reveals the relationship between each neuron. The convergence performance of DNN is affected by the learning algorithm, as it decides how weighting factors are founded. Therefore, these four hyper-parameters are crucial in determining the overall performance of DNN model. As a result, the total quantity of hidden layers, the total quantity of neurons and activation function in each hidden layer, as well as the learning algorithm, are considered as critical parameters for the proposed ranking selection-based PSO approach.

The decision variables are summarised in Table 2, while two features are identified:

- The number of decision variables that need optimisation depends on the value of one of the decision variables (i.e. quantity of hidden layers), which is changeable. For example, if there are 2 hidden layers, there would be 1 (quantity of hidden layers)

**Table 2**  
Decision variables of PSO for DNN.

Quantity of hidden layers	{2, 3, 4, 5}
Quantity of neurons in each hidden layer	{5, 10, 15, 20, 25, 30, 35, 40, 45, 50}
Activation function	{tanh, sigmoid, ReLU, ELU}
Learning approach	{ADAM, NADAM, ADAMAX, SGD}

+ 2 (quantity of neurons in each hidden layer) + (2 + 1) (each layer's activation function) + 1 (learning algorithm) = 7 decision variables in determining the DNN architecture. If there are 5 hidden layers, there would be  $1 + 5 + (5 + 1) + 1 = 13$  decision variables.

- The quantity of neurons in each hidden layer and the quantity of hidden layers can be considered as discrete variables, while activation function and learning approaches are regarded as categorical variables. As a result, the type of decision variables is a mix of discrete and categorical.

### 3.3. Ranking selection based PSO algorithm

PSO is a robust evolutionary algorithm that has been widely adopted in handling various highly nonlinear and sophisticated engineering problems. The first version of PSO was developed by Kennedy and Eberhart (J. Kennedy & R. C. Eberhart, 1995) to solve continuous problems. Later on, they also proposed a binary version of PSO for discrete optimisation problems (J. Kennedy & R. C. Eberhart, 1997). The key benefits of the PSO are its less computational time and swift learning speed. In PSO, each particle, with its position and velocity, represents a single solution. Generally, the fundamental PSO is only suitable for either continuous or discrete problems. According to the analysis in Section 3.2, the optimisation problem requires the changeable number and mixed types of decision variables. Although categorical variables, like activation function and learning approach, can be transformed into discrete variables, there is no relationship between the adjacent numbers and no velocity for the categorical variables. Therefore, ranking selection is integrated into fundamental PSO algorithm to consider both continuous and discrete design variables. Additionally, the ranking selection can adjust the search behaviour of a swarm at various searching phases of categorical variables. The optimisation of DNN architecture can be regarded as a mixed variable optimisation problem as:

$$\text{Minimise } f(\Phi) \quad (11)$$

where  $f(\Phi)$  is the objective function (i.e. MAE of the prediction value from the DNN model),  $\Phi$  is a vector made up of design variables.  $\Phi = \begin{pmatrix} \Phi^D \\ \Phi^C \end{pmatrix}$ , while  $\Phi^D = [\phi_1, \phi_2, \dots, \phi_{\phi_1+1}]^T$  and  $\Phi^C = [\phi_{\phi_1+1}, \dots, \phi_{2\phi_1+1}, \phi_{2\phi_1+2}]^T$ . The number of decision variables in  $\Phi^D$  and  $\Phi^C$  depends on the value of  $\phi_1$ . Compared to the basic PSO algorithm, RSPSO has two prior processes. At first, a population of individuals is ranked. Secondly, each individual is chosen with a certain probability distribution according to its ranks.

The optimal selection scheme used in the evolutionary algorithm depends on the optimisation task and the type of problem (Blickle & Thiele, 1996). In the proposed RSPSO, two types of ranking schemes are developed to rank the population of individuals:

- For discrete variables, the feasible positions are ranked according to the objective function's fitness value, while the infeasible positions are ranked in accordance with their nondomination levels. Furthermore, the infeasible positions at the same nondomination level are graded based upon the fitness value of the objective function. After the ranking of each population, a sequence *PbestSeq* is formulated to store the topmost  $N$  historical positions with superior ranking values. The elements in *PbestSeq* are indicated by *Pbest*:

$$PbestSeq^{(k)} = \begin{cases} Rank(P^1), k = 1 \\ Rank(PbestSeq^{(k-1)} \cup P^{(k)})[1 : N], k > 1 \end{cases} \quad (12)$$

- For categorical variables, the allowable value of each categorical variable memorises its historical best value, which refers to the fitness value of the objective function. If there is the historical best value for a permissible value yet, then the permissible value would be regarded as the historical best value. If the existing historical best value and the corresponding permissible values are all the objective function values, the smaller one of the existing historical best value and permissible values would be regarded as the new historical best value. When all allowable values of a categorical variable have their historical best values, it would be graded based upon the fitness value of the objective function,  $DCV_i = RankV(DC_i)$

For the discrete variables, the selection probability distribution is formulated, while the selected element is determined as:

$$g = \text{ceil} \left( \text{ceil} \left( M \times (rand_1)^{\frac{\log(ER)}{\log(0.5)}} \times rand_2 \right), 0 < ER \leq 1, \right. \\ \left. \text{if } g = 0 \text{ then set } g = 1 \right) \quad (13)$$

where  $M$  is the total quantity of particles in a sequence.  $ER$  is the utilisation-regulated parameter, which indicates the anticipated ratio of the total quantity of the elements to be designated to  $M$ . If  $ER < 1$ , the probability of selecting the  $i^{\text{th}}$  element is

$$P_{i,d}^{(k)} = \sum_{k=i}^N \frac{k^{1/s} - (k-1)^{1/s}}{M^{1/s} \times k}, i = 1, 2, 3, \dots, M \quad (14)$$

$$s = \frac{\log(ER)}{\log(0.5)} \quad (15)$$

For the categorical variables, its value would be arbitrarily chosen from its allowable values until each value has been explored:

$$g = \text{ceil}(N_C \times rand), \text{if } g = 0 \text{ then set } g = 1 \quad (16)$$

$$P_{i,d}^{(k)} = C_{d,g} \quad (17)$$

$N_C$  is the total number of the particles in  $C_d$ . Once each permissible value of the categorical variables is visited at least once, the categorical variables will be arbitrarily chosen from the ranked sequence:

$$g = \text{ceil} \left( \text{ceil} \left( M \times (rand_1)^{\frac{\log(ER)}{\log(0.5)}} \times rand_2 \right), 0 < ER \leq 1, \right. \\ \left. \text{if } g = 0 \text{ then set } g = 1 \right) \quad (18)$$

$$P_{i,d}^{(k)} = DCV_{d,g} \quad (19)$$

$$V_{i,d}^{(k+1)} = INTR \gamma_1 \times V_{i,d}^{(k)} + \gamma_2 \times c_1 \times (Pbest_{n,d}^{(k)} - P_{i,d}^{(k)}) \\ + \gamma_3 \times c_2 \times (Pbest_{g,d}^{(k)} - P_{i,d}^{(k)}) \quad (20)$$

$$P_{i,d}^{(k+1)} = P_{i,d}^{(k)} + V_{i,d}^{(k+1)} \quad (21)$$

where  $\gamma_1$  is the inertia weight,  $\gamma_2$  is the cognitive coefficient,  $\gamma_3$  is the social coefficient,  $c_1$  and  $c_2$  are random values in range [0, 1].  $Pbest_{n,d}$  is the  $d^{\text{th}}$  the dimension of the particle  $Pbest_n$ . Operation INTR is formulated to deal with the design variables with integer values:

$$INTR(r) = \begin{cases} \text{floor}(r), & \text{if } \text{frand} > r - \text{floor}(r) \\ \text{ceil}(r), & \text{otherwise} \end{cases} \quad (22)$$

### 3.4. Framework of rspso-powered DNN prediction model

According to Eq. (23) given in Box I, there are a total of  $4.201 \times 10^8$  types of DNN architectures. Depending on the complexity of the DNN, it could take between 1 and 30 s to simulate each type of architecture. The optimisation of DNN architecture is a complex mixed discrete and categorical variables optimisation problem. If the grid search optimisation approach is adopted, this could take one week to simulate all possible types of architecture. Due to its robustness in handling a wide variety of highly nonlinear and complex engineering problems, PSO is adopted to select the optimal architecture of DNN in this study.

$$4 \times (10^2 \times 4^2 + 10^3 \times 4^3 + 10^4 \times 4^4 + 10^5 \times 4^5) = 4.201 \times 10^8 \quad (23)$$

When number of hidden layers is 2
When number of hidden layers is 3
When number of hidden layers is 4
When number of hidden layers is 5

Learning approach
Number of neurons in each layer
Activation function

Box I.

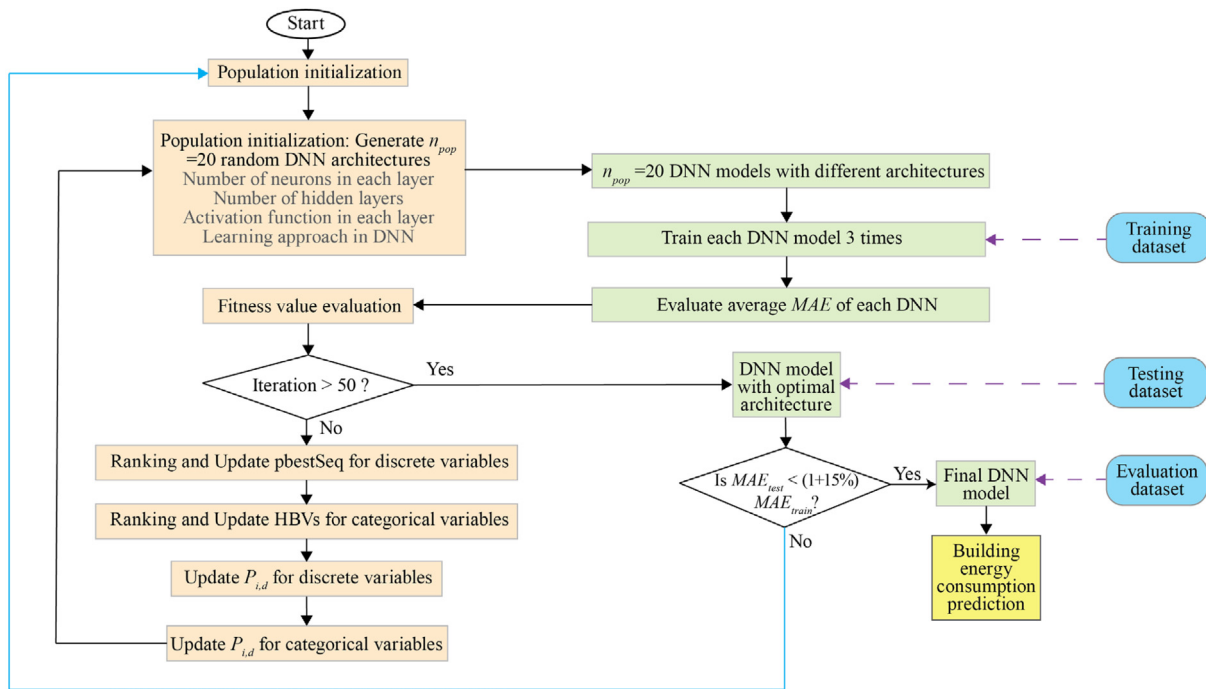


Fig. 3. Flowchart of interaction between RSPSO and DNN.

The framework of the proposed RSPSO-powered DNN model is demonstrated in Fig. 3. The DNN model, with its architecture, is regarded as a particle of the PSO algorithm. At the start,  $n_{pop} = 20$  DNN models are arbitrarily assigned with different architectures. Since training DNN is a random process, there would be slight differences in the resulted MAE while the training process is repeated. Therefore, each DNN model is trained three times, while the mean value of MAE is adopted as the fitness value of PSO objective function. Thereafter, *pbestSeq* and HBVs are ranked and selected to update the discrete and categorical variables of each particle (i.e. architecture of each DNN model). This iteration process is repetitive for 50 times. After obtaining the optimal architecture for the DNN model, the testing dataset is adopted to validate the well-trained DNN model. If the MAE value of the testing dataset is 15% larger than that resulted from the training dataset, the above-mentioned RSPSO optimisation process would be repeated to select the optimal architecture for the DNN model. Otherwise, the DNN model would be considered effective and treated as the optimal DNN model. Thereafter, the evaluation dataset can be adopted to simulate the energy prediction process for the future coming months.

Meanwhile, according to Table 3, there would be 5 hidden layers and 50 neurons in each hidden layer at most, this would result in  $33 \times 50 + 4 \times 50 \times 50 + 50 = 11700$  weighting factors to optimise. Therefore, the performance of 4 different learning approaches, including ADAM, NADAM, ADAMAX, SGD is investigated.

### 3.5. Moving horizon

Yearly update of DNN architecture may not reflect the latest feature in energy consumption profile, while weekly or daily update may be too time consuming. Therefore, monthly update of DNN architecture is adopted. In other words, the moving horizon is set at 1 month. As illustrated in Section 2, the historical meteorological and energy data is available from the past one and a half years in this study. To enable the training dataset considers the seasonal change of weather and energy profile, one year is set as the base timeline of training dataset. To enable the RSPSO-powered deep learning model to be self-adaptive, six groups of the historical datasets are constructed as there are one and a half year's historical data available. In practical application, this is a continuous process and there will be more than 6 DNN models with the time being. As demonstrated in Fig. 4, to enhance the robustness of the DNN model, in each one-year set, 80% of the randomly chosen datasets and the remaining 20% of datasets are adopted to train and test the neural network, respectively. Meanwhile, the following one-month datasets are treated as future prediction scenarios to evaluate the well-trained prediction model. In the following description, DNN1, 2, 3, 4, 5 and 6 represents the DNN model for predicting energy consumption in July, August, September, October, November and December 2019, respectively. The following one-month datasets are named as evaluation datasets in the following discussion. Afterwards, the prediction horizon moves one month forward.

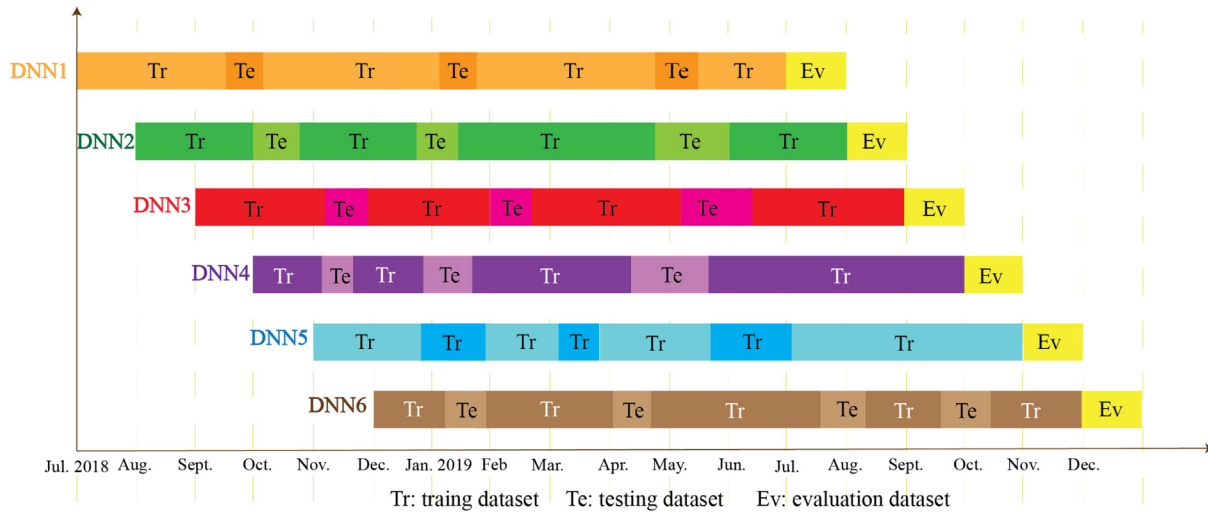


Fig. 4. Diagram of moving horizon.

Table 3  
Optimisation result at different RSPSO parameters.

RSPSO parameters	$\gamma_1 = 0.5$	$\gamma_1 = 0.5$	$\gamma_1 = 0.5$	$\gamma_1 = 0.7$	$\gamma_1 = 0.9$
	$\gamma_2 = 1$	$\gamma_2 = 3$	$\gamma_2 = 2$	$\gamma_2 = 2$	$\gamma_2 = 2$
	$\gamma_3 = 3$	$\gamma_3 = 1$	$\gamma_3 = 2$	$\gamma_3 = 2$	$\gamma_3 = 2$
Average MAE value (kW)	1.16	1.14	1.12	1.15	1.17

In practical application, the training and testing datasets are still selected from the past year’s historical data, while the evaluation datasets consist of the future one month’s weather forecast, time variables and predicted electricity consumption. In other words, at the beginning of each month, the past year’s historical data is split randomly as training and testing datasets with the ratio of 80%:20%. Thus the well-trained neural network can be adopted to forecast the future one-month electricity load. This process is repeated at the beginning of each month to ensure the DNN have the optimal architecture to catch the latest changes in the energy consumption patterns.

#### 4. Performance evaluation of the proposed RSPSO-powered self-adaptive deep learning model

To guarantee the optimal DNN architecture is obtained, the performance of RSPSO is investigated. To test the self-adaptive capability, the performance of DNN is evaluated using the moving horizon approach. To precision and repeatability of the prediction model is explored by its MAE, MSE and  $R^2$  values, while the robustness of the prediction model is tested by adding Gaussian white noises to input datasets.

##### 4.1. Performance evaluation of RSPSO

To improve the effectiveness of RSPSO algorithm in architecture optimisation, its parameters were selected through enumerate approach. The convergence performance of RSPSO algorithm was evaluated with 6 DNN model architecture optimisation. To assess the optimisation ability of RSPSO, the performance of the reference GA-enhanced DNN model is studied.

##### 4.1.1. RSPSO parameters selection

As shown in Eq. (14), RSPSO parameters generally include inertia weight  $\gamma_1$ , cognitive parameter  $\gamma_2$  and social parameter  $\gamma_3$ . Different RSPSO parameters are selected using the enumerate approach to prevent the optimisation from being converged to a local optimum. The

performance of the RSPSO algorithm is evaluated according to the average MAE value of training and testing datasets in DNN1. As summarised in Table 3, the DNN1 model results in the smallest MAE value with  $\gamma_1 = 0.5, \gamma_2 = 2, \gamma_3 = 2$ . Therefore, this set of RSPSO parameters is adopted in determining the optimal architecture of prediction models DNN 1–6.

##### 4.1.2. Comparison of RSPSO-powered DNN and reference GA-powered DNN prediction model

As discussed in Section 3.3, DNN architecture optimisation consists of hybrid continuous and discrete design variables, while fundamental PSO is not able to solve such comprehensive problem. That is also one of the reason why RSPSO is proposed in this study. To assess the optimisation ability of RSPSO, a GA-powered DNN model is designed as the reference case. In this study, perfect knowledge of weather profile for the entire period, to avoid uncertainty and noises in the data measurements when comparing the RSPSO-powered DNN and GA-powered DNN prediction model. The same pre-processed training and testing datasets are adopted as described in Section 4.1, while the architecture of the DNN model is determined by GA optimisation. Through the same approach of sensitivity analysis as adopted in (Luo, Oyedele, Ajayi et al., 2020), the optimal GA parameters are found to be *retain probability* = 0.8, *selection probability* = 0.2, *mutation probabilities* = 0.2. The optimal architecture of the DNN1 model from RSPSO and GA is summarised in Table 4. From both RSPSO-powered DNN and GA-powered DNN, 3 hidden layers with exponential linear activation function and ADAM learning algorithm would result in the optimal performance. Meanwhile, GA-DNN and PSO-DNN would result in a different optimal number of neurons and activation functions in each hidden layer. The MAE, MSE,  $R^2$  and computational time of both models are illustrated in Table 5. The GA-DNN model is adopted as the baseline, while performance improvement is calculated as in Box II: The most significant advantage of RSPSO-DNN over GA-DNN is its faster convergence speed in solving the optimisation problem of DNN architecture. The computational time of the RSPSO-DNN model is 39.69% shorter than that of GA-DNN. Moreover, when conducting energy consumption prediction for the evaluation dataset, the MAE and MSE of RSPSO-DNN is 4.94% and 7.72% smaller than that of GA-DNN. Although there is only a 0.2 kW difference in MAE between GA-DNN and RSPSO-DNN, there would result in a 1752 kWh difference in one year. The prediction results of the first week of July is summarised in Fig. 5. It is found that the prediction values from PSO-DNN are closer to the real measurement most of the time.



$$\frac{\text{Performance indicator of RSPSO powered DNN model} - \text{Performance indicator of GA\_DNN model}}{\text{Performance indicator of GA\_DNN model}} \times 100\%$$

Box II.

**Table 4**  
The architecture of optimal DNN architecture determined by GA and RSPSO.

DNN architecture	Quantity of neurons in each hidden layer			Quantity of layer (s)	Activation function				Optimisation approach
	1	2	3		Hidden layer 1	Hidden layer 2	Hidden layer 3	Output layer	
GA-optimised	55	25	20	3	ReLU	sigmoid	ReLU	ELU	ADAM
RSPSO-optimised	55	35	15	3	ReLU	ReLU	ReLU	ELU	ADAM

**Table 5**  
Prediction performance of RSPSO-DNN and GA-DNN.

	MAE (kW)		MSE (kW)		R <sup>2</sup> (%)		Computational time
	Train&Test	Evaluation	Train&Test	Evaluation	Train&Test	Evaluation	
GA-DNN	1.13	2.43	3.31	12.17	99.24	97.70	9 h 32 min
RSPSO-DNN	1.12	2.31	3.11	12.23	99.29	97.89	5 h 45 min
Improvement	↓0.89%	↓4.94%	↓6.04%	↑0.49%	↑0.05%	↑0.19%	↓39.69%

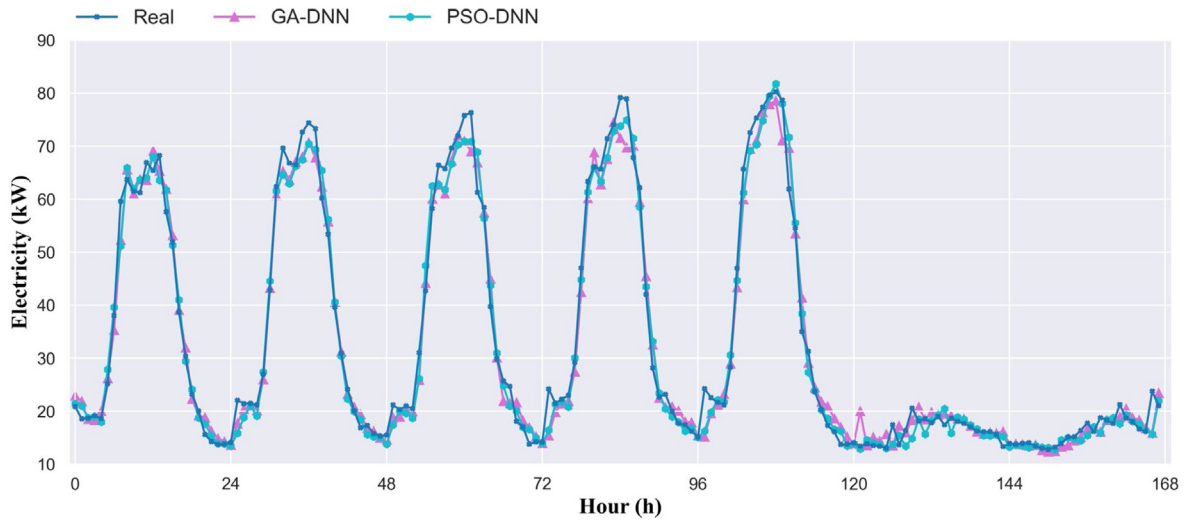


Fig. 5. Prediction results of GA-DNN and PSO-DNN.

4.1.3. The convergence of RSPSO in DNN architecture optimisation

The convergence performance of the proposed self-adaptive deep learning model for six sets of training datasets is shown in Fig. 6. The optimisation of DNN architecture reaches convergent after 22, 27, 12, 36, 39 and 33 iterations for DNN1, DNN2, DNN3, DNN4, DNN5 and DNN6, respectively. It is also identified that the MAE values of the 6 DNN prediction models are smaller than 1.70 kW.

The optimal architecture for the 6 DNN models is summarised in Table 6. The quantity of neurons in each hidden layer, the quantity of layers, activation function, the learning approach is different among the 6 DNN models. It also demonstrates its self-adaptive ability in chasing the latest featuring patterns for online building energy consumption prediction. The various DNN architecture indicates the varying featuring patterns of meteorological and energy consumption data during the different periods of the year.

4.1.4. Performance assessment of the determined optimal architecture

To demonstrate DNN's optimal architecture in energy consumption prediction, four reference DNN models are introduced, with each changing one parameter from the optimal DNN1 architecture. The architecture of each DNN reference model is summarised in Table 7.

- In DNN\_R1, there is only one hidden layer with 55 neurons in that hidden layer, while activation function and learning algorithm is kept the same as those in RSPSO-powered DNN1.
- In DNN\_R2, there are 35 neurons in each hidden layer, while the total quantity of neurons is kept the same as that in the RSPSO-powered DNN1. The number of layers, activation and learning approaches is also kept the same as those in the RSPSO-powered DNN1.
- In DNN\_R3, the sigmoid activation function is adopted in hidden layers, while the quantity of neurons and layers, activation function, and learning algorithm are kept the same as that in the RSPSO-powered DNN1.
- In DNN\_R4, SGD is adopted as the learning approach, while the quantity of neurons in each layer, number of layers, activation function, and learning algorithm is kept the same as that in the RSPSO-powered DNN1.

The MAE, MSE, R<sup>2</sup> and computational time of both models are summarised in Table 8, while the prediction results of the first week of July is summarised in Fig. 7. Compared to the four reference DNN models, the RSPSO-powered DNN has a 7.63% and 17.67% reduction

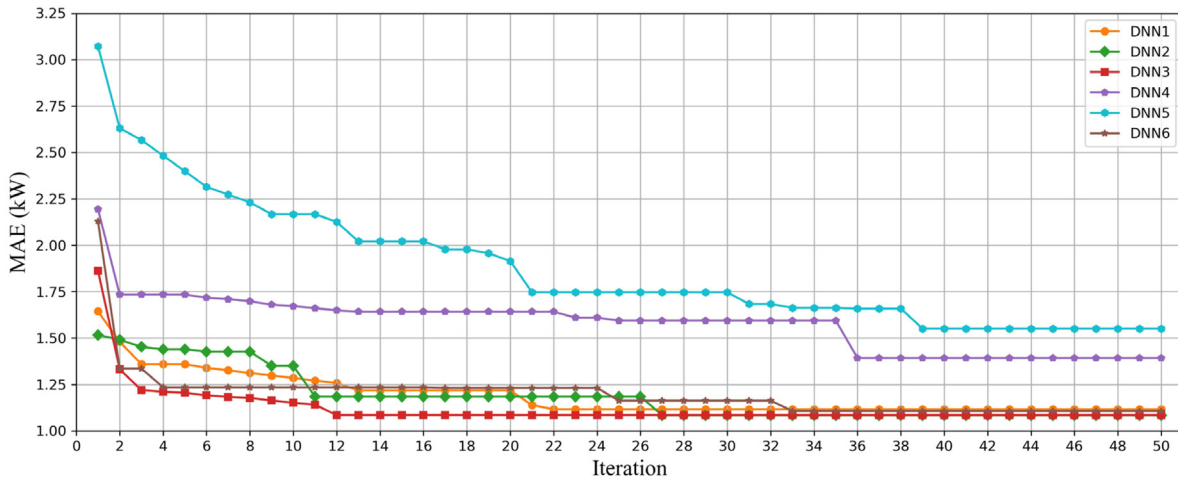


Fig. 6. The convergence of MAE in RSPSO optimisation.

Table 6  
Optimal architecture of each DNN model.

DNN architecture	Quantity of neurons in each hidden layer				Quantity of layer	Activation function					Learning approach
	1	2	3	4		Hidden layer 1	Hidden layer 2	Hidden layer 3	Hidden layer 4	Output layer	
DNN1	55	35	15	3	3	ReLU	ReLU	ReLU		ELU	ADAM
DNN2	25	50	40	50	4	sigmoid	ReLU	sigmoid	sigmoid	ELU	NADAM
DNN3	40	25	50	20	4	sigmoid	ELU	sigmoid	ReLU	ELU	ADAMAX
DNN4	5	45	50		3	ELU	ELU	ELU		ReLU	ADAM
DNN5	5	45			2	tanh	ELU			ELU	NADAM
DNN6	30	5	45		3	ELU	ReLU	sigmoid		ELU	NADAM

Table 7  
The architecture of each DNN reference model.

DNN architecture	Quantity of neurons in each layer			Number of layer (s)	Activation function				Learning approach	
	1	2	3		Hidden layer 1	Hidden layer 2	Hidden layer 3	Output layer		
DNN_R1	55			1	ReLU				ELU	ADAM
DNN_R2	35	35	35	3	ReLU	ReLU	ReLU		ELU	ADAM
DNN_R3	55	35	15	3	sigmoid	sigmoid	sigmoid		ELU	ADAM
DNN_R4	55	35	15	3	ReLU	ReLU	ReLU		ELU	SGD
DNN1	55	35	15	3	ReLU	ReLU	ReLU		ELU	ADAM

Table 8  
Prediction performance of each DNN reference model.

	MAE (kW)	MSE (kW)	R <sup>2</sup> (%)
DNN_R1	2.47	13.64	97.43
DNN_R2	2.37	12.76	97.60
DNN_R3	2.49	12.68	97.61
DNN_R4	2.49	12.40	97.67
PSO-DNN	2.30	11.23	97.89
Biggest performance improvement	7.63%	17.67%	0.47%

in MAE and MSE, respectively. The prediction values from PSO-DNN are closer to the real measurement than those reference DNN models during most of the time.

#### 4.2. Analysis of the effectiveness of moving horizon

Moving horizon building energy prediction as described in Fig. 4 should be adopted owing to the varying featuring patterns of meteorological and energy consumption data during the different periods of the year. Therefore, the DNN model should be trained using the past year’s historical database and adopted to predict next month’s building energy consumption. In other words, the prediction models of DNN1, DNN2, DNN3, DNN4, DNN5 and DNN6 should be adopted to forecast

the building energy consumption in July, August, September, October, November and December 2019, respectively. Two sets of DNN reference models are adopted to evaluate the effectiveness of moving horizons in improving the prediction accuracy.

The specifications of the first set of DNN reference models are illustrated in Table 9. The 6 DNN models are also adopted to forecast the electricity load of the remaining months. For example, DNN1 is still trained using the database during July 2018–June 2019. However, the meteorological and energy consumption datasets from each month from July–December 2019 are adopted as evaluation datasets. Namely, the DNN model with the same architecture and structure (i.e. weighting factors) is adopted for building energy consumption prediction of the following months. The RMSE, MAE and R<sup>2</sup> of the reference DNN networks are illustrated in Table 10, while the prediction results of the first week of July is summarised in Fig. 8. It is seen that the smallest MAE, MSE and the biggest R<sup>2</sup> value of July, August, September, October, November, December 2019 is obtained using DNN1, DNN2, DNN3, DNN4, DNN5 and DNN6, respectively. However, the prediction accuracy might be affected by the month. It is seen that for the same month, the prediction model with the moving horizon approach (the method proposed in this paper) has higher prediction accuracy and repeatability compared to that without moving horizon (reference model).

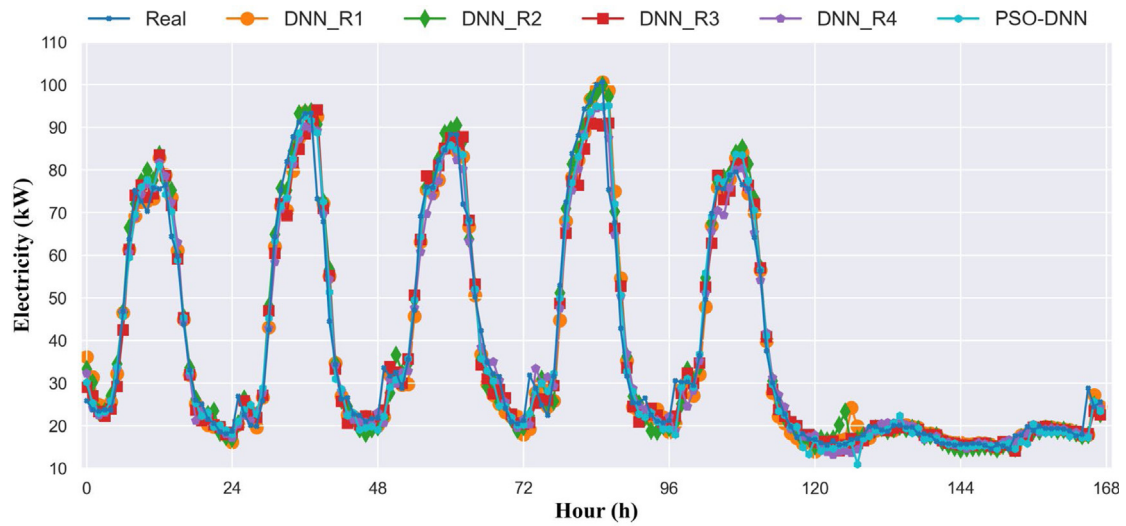


Fig. 7. Prediction results of each DNN reference model.

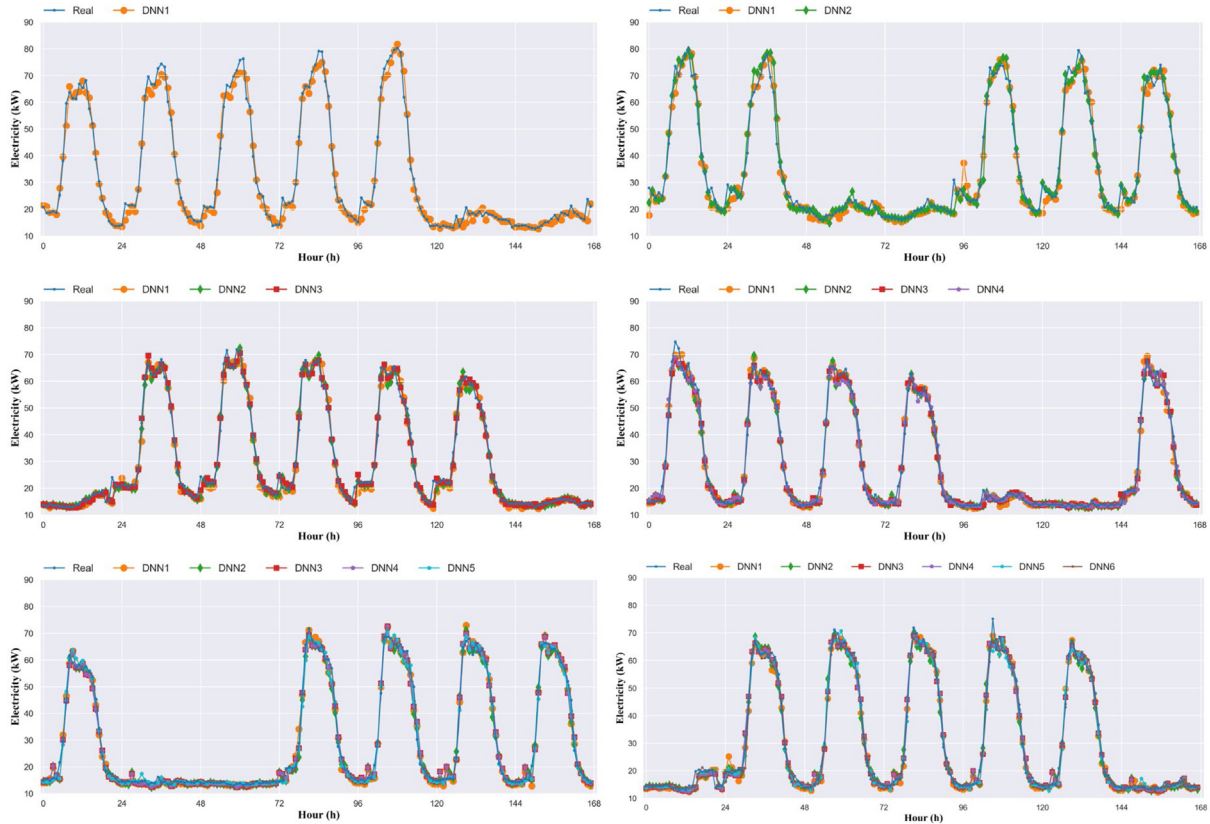


Fig. 8. Prediction results of the proposed PSO-DNN and each DNN reference model.

**Table 9**  
Training, testing and evaluation datasets information for the first set of reference models.

	Training & Testing	Evaluation (2019)					
DNN1	Jul. 2018–June 2019	Jul.	Aug.	Sept.	Oct.	Nov.	Dec.
DNN2	Aug. 2018–Jul. 2019	Aug.	Sept.	Oct.	Nov.	Dec.	
DNN3	Sept. 2018–Aug. 2019	Sept.	Oct.	Nov.	Dec.		
DNN4	Oct. 2018–Sept. 2019	Oct.	Nov.	Dec.			
DNN5	Nov. 2019–Oct. 2018	Nov.	Dec.				
DNN6	Dec. 2019–Nov. 2019	Dec.					

The specifications of the second set of DNN reference models are summarised in Table 11. For each reference prediction model, although its weighting factors are updated by training with the past 12 months dataset, the architecture of each DNN model is kept the same. For example, the architecture of DNN1 is still determined using the datasets during July 2018–June 2019. However, for each new month, the structure (i.e. weighting factors) are determined using the past year training datasets. The *RMSE*, *MAE* and *R<sup>2</sup>* of the reference DNN networks are illustrated in Table 12. Although the *MAE* and *RMSE* value in each individual case is smaller than those from the first set of reference DNN models, it is still larger than that obtained from the

**Table 10**  
MAE, MSE and  $R^2$  value for DNN models using different evaluation datasets.

	Jul.	Aug.	Sept.	Oct.	Nov.	Dec.		Jul.	Aug.	Sept.	Oct.	Nov.	Dec.
<b>DNN1</b>	2.31	2.24	1.92	1.65	1.66	1.68	<b>DNN1</b>	11.23	10.62	7.97	5.64	5.65	5.75
<b>DNN2</b>	-	2.01	1.70	1.65	1.68	1.58	<b>DNN2</b>	-	8.45	6.50	5.85	5.97	5.43
<b>DNN3</b>	-	-	1.50	1.46	1.61	1.58	<b>DNN3</b>	-	-	5.11	4.69	5.49	5.64
<b>DNN4</b>	-	-	-	1.44	1.61	1.58	<b>DNN4</b>	-	-	-	4.48	5.49	5.64
<b>DNN5</b>	-	-	-	-	1.38	1.39	<b>DNN5</b>	-	-	-	-	4.48	4.74
<b>DNN6</b>	-	-	-	-	-	1.28	<b>DNN6</b>	-	-	-	-	-	4.15

(a) MAE (kW)

(b) MSE (kW)

	Jul.	Aug.	Sept.	Oct.	Nov.	Dec.
<b>DNN1</b>	97.89%	97.21%	97.71%	98.56%	98.64%	98.21%
<b>DNN2</b>	-	97.52%	98.13%	98.51%	98.56%	98.31%
<b>DNN3</b>	-	-	98.53%	98.80%	98.68%	98.24%
<b>DNN4</b>	-	-	-	98.86%	98.68%	98.24%
<b>DNN5</b>	-	-	-	-	98.92%	98.52%
<b>DNN6</b>	-	-	-	-	-	98.71%

(c)  $R^2$

**Table 11**  
Training, testing and evaluation datasets information for the second set of reference models.

	Training & Testing	Evaluation	Training & Testing	Evaluation	Training & Testing	Evaluation	Training & Testing	Evaluation	Training & Testing	Evaluation	Training & Testing	Evaluation
DNN1	Jul. 2018–June 2019	Jul. 2019	Aug. 2018–Jul. 2019	Aug. 2019	Sept. 2018–Aug. 2019	Sept. 2019	Oct. 2018–Sept. 2019	Oct. 2019	Nov. 2019–Oct. 2018	Nov. 2019	Dec. 2019–Nov. 2019	Dec. 2019
DNN2			Aug. 2018–Jul. 2019	Aug. 2019	Sept. 2018–Aug. 2019	Sept. 2019	Oct. 2018–Sept. 2019	Oct. 2019	Nov. 2019–Oct. 2018	Nov. 2019	Dec. 2019–Nov. 2019	Dec. 2019
DNN3					Sept. 2018–Aug. 2019	Sept. 2019	Oct. 2018–Sept. 2019	Oct. 2019	Nov. 2019–Oct. 2018	Nov. 2019	Dec. 2019–Nov. 2019	Dec. 2019
DNN4							Oct. 2018–Sept. 2019	Oct. 2019	Nov. 2019–Oct. 2018	Nov. 2019	Dec. 2019–Nov. 2019	Dec. 2019
DNN5									Nov. 2019–Oct. 2018	Nov. 2019	Dec. 2019–Nov. 2019	Dec. 2019
DNN6											Dec. 2019–Nov. 2019	Dec. 2019

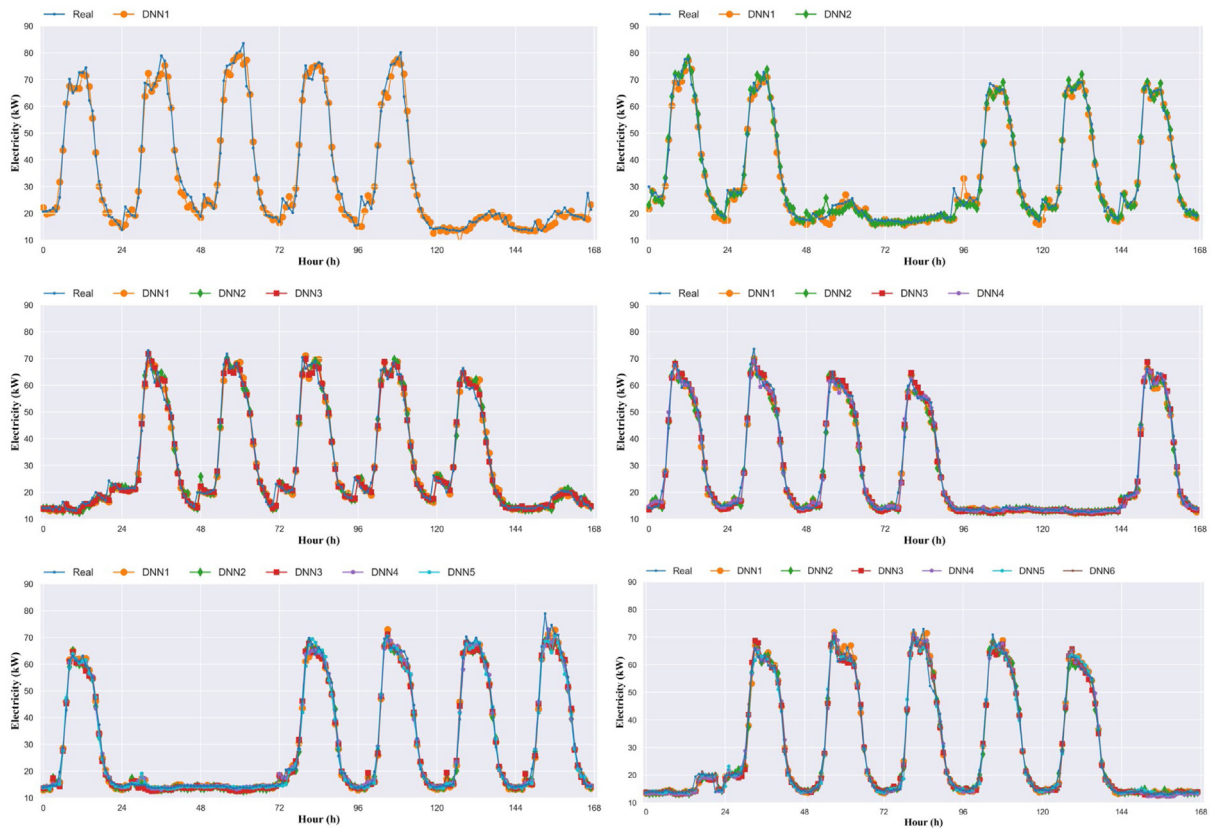
proposed self-adaptive DNN model. The prediction results from the second week of each month are summarised in Fig. 9. It is found that the prediction values from PSO-DNN are closer to the real measurement most of the time. The above two reference cases indicate that the DNN prediction model is the most effective at forecasting next month's energy consumption, while its prediction performance would diminish with the increase of time. It also proves the necessity of adopting a moving horizon in determining the optimal architecture and weighting factors of DNN.

#### 4.3. Performance analysis of robustness

As there generally exists uncertainty in the weather forecast, the prediction model's robustness is assessed. To represent the real situation of weather forecast, the Gaussian noises  $G$  are added to the meteorological data. The Gaussian white noises have a mean of 0, while the standard deviation equals 20% of the maximum value of each type of meteorological data (i.e. dry-bulb temperature, wet-bulb temperature, wind velocity, cloud ratio and solar radiation), respectively.

**Table 12**  
MAE, MSE and  $R^2$  value for DNN models using different evaluation datasets.

	Jul.	Aug.	Sept.	Oct.	Nov.	Dec.		Jul.	Aug.	Sept.	Oct.	Nov.	Dec.
<b>DNN1</b>	2.31	2.04	1.70	1.55	1.55	1.43	<b>DNN1</b>	11.23	8.81	6.22	5.05	5.20	4.94
<b>DNN2</b>	-	2.01	1.69	1.64	1.49	1.30	<b>DNN2</b>	-	8.45	6.37	6.12	4.50	4.33
<b>DNN3</b>	-	-	1.50	1.52	1.54	1.34	<b>DNN3</b>	-	-	5.11	5.02	4.95	4.51
<b>DNN4</b>	-	-	-	1.44	1.42	1.46	<b>DNN4</b>	-	-	-	4.48	4.82	4.86
<b>DNN5</b>	-	-	-	-	1.38	1.42	<b>DNN5</b>	-	-	-	-	4.48	5.33
<b>DNN6</b>	-	-	-	-	-	1.28	<b>DNN6</b>	-	-	-	-	-	4.15
	Jul.	Aug.	Sept.	Oct.	Nov.	Dec.		Jul.	Aug.	Sept.	Oct.	Nov.	Dec.
<b>DNN1</b>	97.89%	97.48%	98.21%	98.71%	98.75%	98.46%	<b>DNN1</b>	97.89%	97.48%	98.21%	98.71%	98.75%	98.46%
<b>DNN2</b>	-	97.52%	98.17%	98.44%	98.92%	98.65%	<b>DNN2</b>	-	97.52%	98.17%	98.44%	98.92%	98.65%
<b>DNN3</b>	-	-	98.53%	98.72%	98.81%	98.59%	<b>DNN3</b>	-	-	98.53%	98.72%	98.81%	98.59%
<b>DNN4</b>	-	-	-	98.86%	98.84%	98.49%	<b>DNN4</b>	-	-	-	98.86%	98.84%	98.49%
<b>DNN5</b>	-	-	-	-	98.92%	98.34%	<b>DNN5</b>	-	-	-	-	98.92%	98.34%
<b>DNN6</b>	-	-	-	-	-	98.71%	<b>DNN6</b>	-	-	-	-	-	98.71%



**Fig. 9.** Prediction results of the proposed RSPSO-DNN and each DNN reference model.

The MAE, MSE and  $R^2$  of these noises added and noise-free prediction models are shown in Table 13, while corresponding prediction results are summarised in Fig. 10. The largest increase of MAE and RMSE is 15.33% and 38.55%, respectively.

**4.4. Performance analysis of input training datasets**

To find out the optimal input training datasets, the rolling datasets and accumulative datasets are adopted to train the 6 DNN models. The

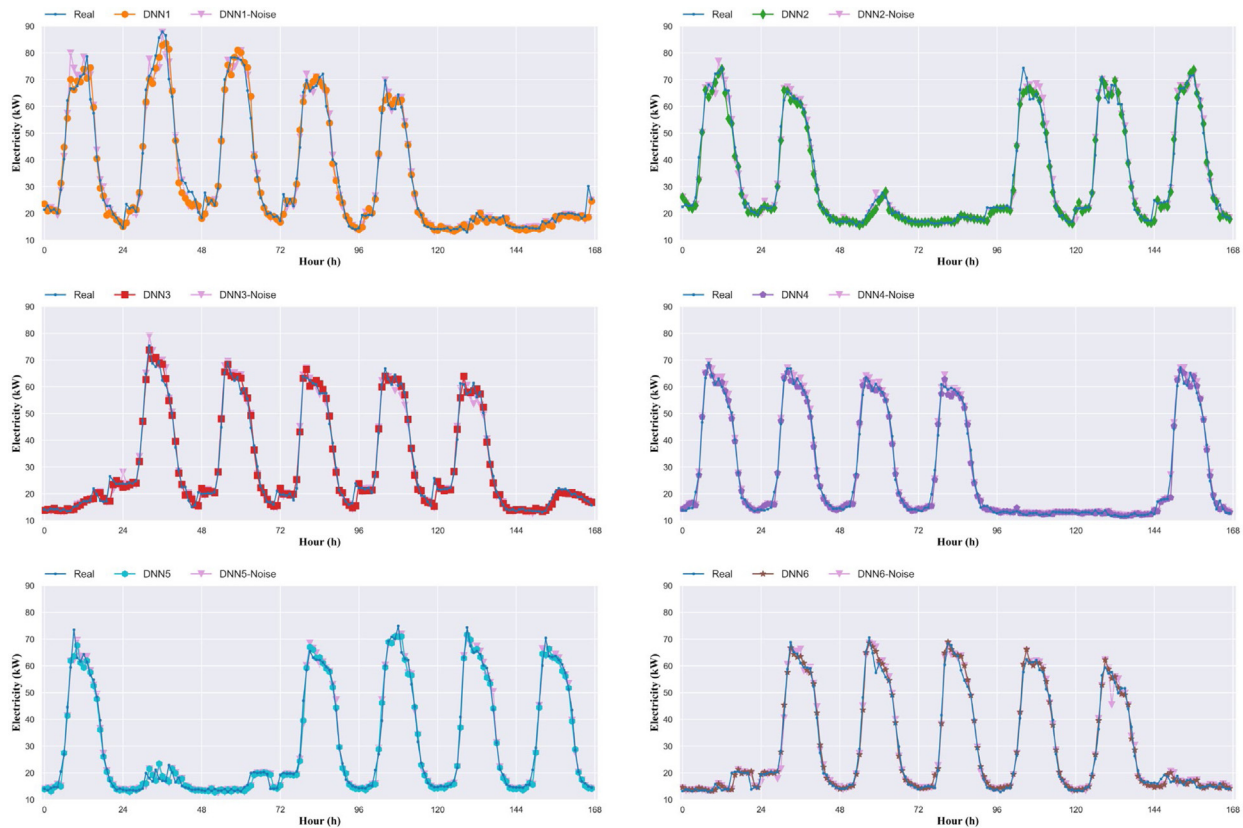


Fig. 10. Prediction results of the proposed PSO-DNN under noise-free and noise-added cases.

Table 13  
Prediction performance under Gaussian white noises.

Performance	Type	Jul.	Aug.	Sept.	Oct.	Nov.	Dec.
MAE (kW)	Noise added	2.49	2.26	1.73	1.47	1.46	1.41
	Noise free	2.31	2.01	1.50	1.44	1.38	1.28
	Increase	7.79%	12.44%	15.33%	2.08%	5.80%	10.16%
$R^2$	Noise added	97.47%	96.80%	97.97%	98.78%	98.81%	98.34%
	Noise free	97.89%	97.52%	98.53%	98.86%	98.92%	98.71%
	Decrease	0.43%	0.74%	0.57%	0.08%	0.11%	0.37%
MSE (kW)	Noise added	13.42	12.15	7.08	4.77	4.95	5.32
	Noise free	11.23	9.45	5.11	4.48	4.47	4.15
	Increase	19.50%	28.57%	38.55%	6.47%	10.74%	28.19%

specifications of the second set of DNN reference models are illustrated in Table 14. The RMSE, MAE and  $R^2$  of the reference DNN models are illustrated in Table 15. It is found that increasing the length of training datasets does not necessarily improve prediction accuracy. It is because that the latest one-year historical data is sufficient to cover the characteristics of building energy consumption. An unbalanced historical database (e.g. two July) may result in biased training. Moreover, a larger database means a longer computational time. Therefore, it is suggested to use the exact past-year historical data as rolling datasets to train the proposed RPSO-DNN model, as illustrated in Fig. 7.

### 5. Summary of the self-adaptive deep learning prediction model and practical implication

The whole process of practical building energy consumption prediction is illustrated in Fig. 11. its architecture should be self-adaptive so that the moving horizon approach can be adopted. At the beginning of each month, last year’s meteorological profile from the local weather station and electricity load profile from the building energy management system is adopted to select the DNN network’s optimal architecture. After the optimal DNN network is well trained

and the weighting coefficients are obtained, it can be utilised to forecast next month’s energy consumption with the up-to-date weather forecast profile from reliable weather forecasting websites (i.e. Accuweather, Weather.com, or Metoffice). Due to the capability of being accurate, robust, repeatable and self-adaptive, the proposed deep learning can be integrated into the building energy management system to forecast day-ahead or week-ahead energy consumption. It can also assist in supply-side management and fault detection.

### 6. Conclusion and future follow-up work

In this research, an RPSO-powered deep learning model with a moving horizon is proposed to enhance the accuracy, robustness, repeatability and self-adaptive capability in moving horizon building electricity load prediction. Although the prediction model is a continuous research of our previous GA-DNN prediction model, its distinct innovation and contribution are summarised as follows:

- A robust evolutionary algorithm to optimise the DNN network’s architecture, thus determine the dimension and length of the DNN model, the comprehensive relationship between output and

**Table 14**  
Training, testing and evaluation datasets information for accumulative and rolling datasets.

	Training & Testing	Evaluation	Training & Testing	Evaluation	Training & Testing	Evaluation	Training & Testing	Evaluation	Training & Testing	Evaluation
DNN1										
DNN2										
DNN3	Jul. 2018–Jul. 2019	Aug. 2019	Jul. 2018–Aug. 2019	Sept. 2019	July. 2018–Sept. 2019	Oct. 2019	Jul. 2018–Oct. 2019	Nov. 2019	Jul. 2018–Nov. 2019	Dec. 2019
DNN4										
DNN5										
DNN6										
Proposed	Aug. 2018–Jul. 2019	Aug. 2019	Jul. 2018–Aug. 2019	Sept. 2019	July. 2018–Sept. 2019	Oct. 2019	Jul. 2018–Oct. 2019	Nov. 2019	Jul. 2018–Nov. 2019	Dec. 2019
			DNN3		DNN4		DNN5		DNN6	

**Table 15**  
MAE, MSE and R<sup>2</sup> value for DNN models using accumulative and rolling datasets.

	Aug.	Sept.	Oct.	Nov.	Dec.		Aug.	Sept.	Oct.	Nov.	Dec.
<b>DNN1</b>	2.28	1.85	1.51	1.46	1.39	<b>DNN1</b>	10.28	7.05	4.70	4.50	4.22
<b>DNN2</b>	2.17	1.64	1.59	1.39	1.42	<b>DNN2</b>	10.12	5.85	5.60	4.10	5.08
<b>DNN3</b>	2.14	1.55	1.43	1.39	1.29	<b>DNN3</b>	10.17	5.61	4.46	4.10	4.34
<b>DNN4</b>	2.04	1.65	1.57	1.33	1.35	<b>DNN4</b>	9.31	6.17	5.21	4.02	4.86
<b>DNN5</b>	2.30	1.64	1.53	1.61	1.50	<b>DNN5</b>	11.50	6.08	5.26	6.06	6.00
<b>DNN6</b>	2.01	1.53	1.51	1.42	1.33	<b>DNN6</b>	8.78	5.31	4.94	4.70	4.32
<b>Proposed</b>	2.01	1.50	1.44	1.38	1.28	<b>Proposed</b>	8.45	5.11	4.48	4.48	4.15

	Aug.	Sept.	Oct.	Nov.	Dec.
<b>DNN1</b>	97.30%	97.98%	98.80%	98.92%	98.69%
<b>DNN2</b>	97.34%	98.32%	98.57%	99.01%	98.42%
<b>DNN3</b>	97.32%	98.39%	98.86%	99.01%	98.65%
<b>DNN4</b>	97.55%	98.23%	98.67%	99.03%	98.49%
<b>DNN5</b>	96.98%	98.25%	98.66%	98.54%	98.13%
<b>DNN6</b>	97.69%	98.47%	98.74%	98.87%	98.65%
<b>Proposed</b>	97.52%	98.53%	98.86%	98.92%	98.71%

input datasets, and the training procedure of various weighting coefficients.

- A revised RSPSO algorithm is adopted to tackle with optimisation problem with a varying number (i.e. the quantity of decision variables is affected by the value of one of the decision variables) and mixed types of decision variables (i.e. quantity of hidden layers and quantity of corresponding neurons in each hidden layer as discrete variables while activation function and learning approach as categorical variables).
- A moving horizon approach is adopted to design the DNN network’s architecture and structure. Thus it has the self-adaptive capability to guarantee high prediction accuracy when new featuring patterns occur in the building energy consumption.
- A Gaussian white noise test to ensure the robustness of the prediction model; thus, it can deal with weather forecast uncertainty and improve the energy prediction performance.

The historical meteorological profile from the local weather station and the recorded electricity load from the energy management system from a real-world campus building is adopted to test the proposed prediction model. Six DNN models are designed with different architectures and structures, with each trained by the past one-year data and being applied to predict next month’s electricity load. The main characteristics of the DNN prediction model are summarised below:

- The most significant advantage of the RSPSO-powered DNN over GA-powered DNN is its faster convergence speed in solving the problem architecture optimisation. The computational time of RSPSO-powered DNN is 39.69% shorter than that of GA-powered DNN. Moreover, when conducting energy consumption prediction for the evaluation dataset, the MAE and MSE of PSO-DNN is 4.94% and 7.72% smaller than that of GA-DNN.
- The different optimal quantity of hidden layers, the quantity of neurons in each layer, activation function in each layer and

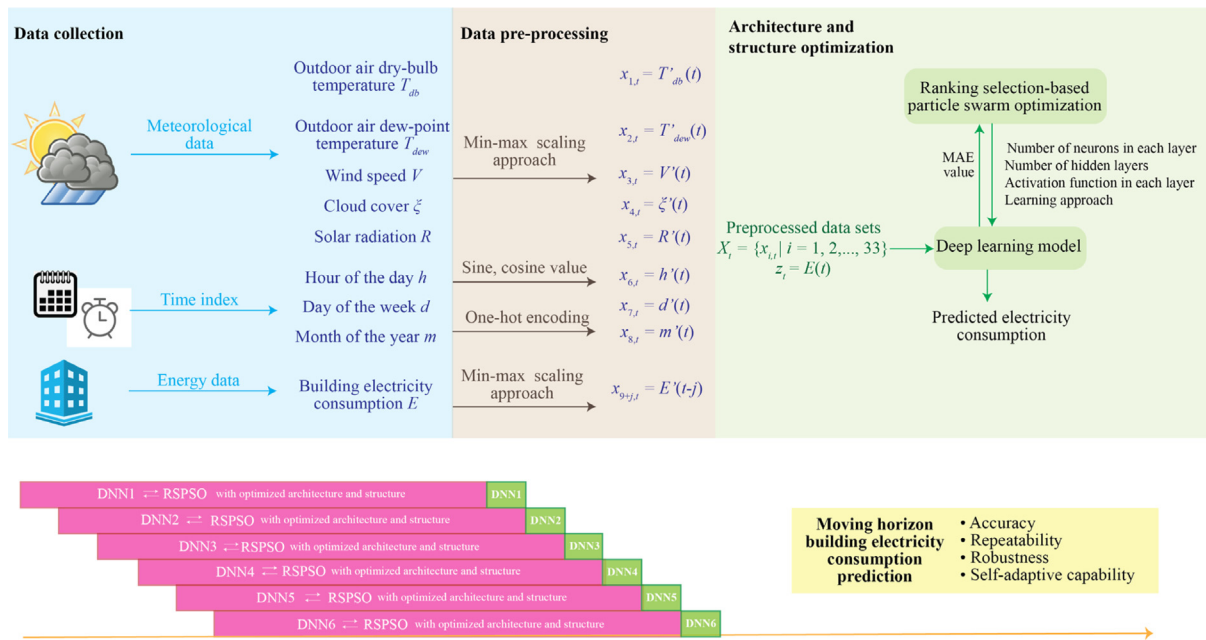


Fig. 11. Summary of the proposed self-adaptive deep learning prediction model.

learning approach for weighting coefficients are determined for the six DNN models. The optimal number of hidden layers is found to be 4 for all the 6 DNN prediction models. It demonstrates that the architecture of DNN models needs to be self-adaptive for moving horizon building energy consumption prediction to chase its latest featuring patterns.

- Take DNN1 as an example: the DNN with optimal architecture determined by PSO shows better performance than the reference DNN models. The architecture of reference DNN models is defined by changing one parameter from the optimal DNN architecture. By comparing with the four reference DNN models, the biggest performance improvement is a 7.63% decrease in MAE, 17.67% in MSE, and 0.47% in R<sup>2</sup>.
- The proposed self-adaptive deep learning model is the most effective at forecasting next month's energy consumption, while its prediction accuracy and repeatability would deteriorate with the increase of timespan from last training. Even if the prediction model is trained using the latest datasets, its prediction performance will fade if its architecture is not adjusted.
- To represent the real situation of weather forecast, the Gaussian noises G are added to the meteorological data. The smallest increase of MAE, RMSE and decrease of R<sup>2</sup> is 2.08%, 6.47% and 0.08%, while the largest ones are 15.33%, 38.55% and 0.74%, respectively. The small changes in accuracy and repeatability demonstrate the robustness of the proposed prediction model in overcoming uncertainty in weather profiles from the forecast website.

Although the proposed self-adaptive prediction model shows excellent performance when applied on the real-world campus building, its limitation and potential future follow-up work are summarised as follows:

- It is found that the optimal DNN architecture is different for various databases. The correlation between the optimal DNN architecture and the changing characteristics of the training database can be investigated.
- Compared to residential, hospital, hotel or other types of buildings, the operating schedules of lighting, office equipment, and the heating-and-cooling system is relatively stable in office buildings. The accuracy, robustness, repeatability and self-adaptive

ability of the proposed prediction model should be further explored when it is adopted in other building types.

- The proposed prediction model is tested on an office building in a temperate oceanic climate, while dry-bulb temperature, wet-bulb temperature, cloud ratio, wind velocity and solar radiation are adopted as meteorological data. It would be interesting to see whether these types of meteorological data also have significant effects on building energy performance in other climate zones.
- It is also essential to explore the accuracy and efficiency of other deep learning algorithms in forecasting electricity load, including the long short-term memory units, recurrent networks, and convolutional neural networks.

**Nomenclature**

<i>c</i>	Random values between 0–1 regenerated for each velocity update
<i>d</i>	Day
<i>E</i>	Electricity load
<i>f</i>	Activation function
<i>h</i>	Hour
<i>m</i>	Month
<i>MAPE</i>	Mean absolute percentage error
<i>N, n</i>	Number
<i>pbest</i>	Particle's best value
<i>T</i>	Temperature
<i>v</i>	Velocity of the particle
<i>V</i>	Speed
<i>w</i>	Weighting coefficient
<i>W</i>	Set of weighting coefficients
<i>x</i>	Neuron in the input layer
<i>X</i>	Input dataset
<i>y</i>	Neurons in hidden layer
<i>z</i>	Neurons in output layer
<i>ẑ</i>	Estimated output of DNN model
<i>Z</i>	Output dataset
<i>θ</i>	Cloud ratio
<i>ε</i>	Solar radiation
<i>γ</i>	PSO parameters



## Subscripts

$db$	Dry-bulb
$dew$	Dew-point
$i$	Index of neurons in input layer
$j$	Index of neurons in hidden layer
$k$	Index of hidden layers
$m$	Particle number
$l$	Iteration number
$H$	Hidden layer
$IN$	Input layer
$O$	Output layer
$pop$	Population
$t$	Time step
$\mu$	Iteration step

## Abbreviations

ADAM	Adaptive moment estimation
ANFIS	Adaptive network-based fuzzy inference systems
ANN	Artificial neural network
ELU	Exponential linear function
DNN	Deep neural network
GA	Genetic algorithm
LM	Levenberg–Marquardt backpropagation
MAE	Mean absolute error
MSE	Mean squared error
NADAM	Nesterov-accelerated Adaptive Moment Estimation
PSO	Particle swarm optimisation
ReLU	Rectified linear unit function
RMSE	Root mean square error
SGD	Stochastic gradient descent approach
tanh	Hyperbolic tangent function

## CRedit authorship contribution statement

**Xiaojun Luo:** Conceptualisation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing, Resources, Data curation, Visualisation. **Lukumon O. Oyedele:** Supervision, Funding acquisition, Project administration.

## 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.

## Acknowledgements

The authors would like to acknowledge and express their sincere gratitude to the Engineering and Physical Sciences Research Council (EPSRC), UK (Grant Reference No. EP/S031480/1) and The Department for Business, Energy & Industrial Strategy (BEIS - grant project number TEIF-101-7025) for providing financial support for this study. Opinions expressed and conclusions arrived at are those of the authors and are not to be attributed to either EPSRC or BEIS.

## Funding

All of the sources of funding for the work described in this publication are acknowledged below: the Engineering and Physical Sciences Research Council (EPSRC), UK (Grant Reference No. EP/S031480/1) and The Department for Business, Energy & Industrial Strategy (BEIS - grant project number TEIF-101-7025).

## Research Ethics

We further confirm that any aspect of the work covered in this manuscript that has involved human patients has been conducted with the ethical approval of all relevant bodies and that such approvals are acknowledged within the manuscript.

Written consent to publish potentially identifying information, such as details or the case and photographs, was obtained from the patient(s) or their legal guardian(s).

## References

- Ahmad, T., Chen, H., Shair, J., & Xu, C. (2019). Deployment of data-mining short and medium-term horizon cooling load forecasting models for building energy optimisation and management. *International Journal of Refrigeration*, *98*, 399–409.
- Aslam, S., Herodotou, H., Mohsin, S. M., Javaid, N., Ashraf, N., & Aslam, S. (2021). A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids. *Renewable and Sustainable Energy Reviews*, *144*, Article 110992.
- Blickle, T., & Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, *4*, 361–394.
- Bünning, F., Heer, P., Smith, R. S., & Lygeros, J. (2020). Improved day ahead heating demand forecasting by online correction methods. *Energy and Buildings*, *211*(2020), Article 109821.
- Deb, C., Eang, L. S., Yang, J., & Santamouris, M. (2016). Forecasting diurnal cooling energy load for institutional buildings using Artificial Neural Networks. *Energy and Buildings*, *121*, 284–297.
- Deng, L., & Yu, D. (2013). *Microsoft research monograph, Deep learning for signal and information processing*.
- Kalliola, Jussi, Kapočiūtė-Dzikienė, Jurgita, & Damaševičius, Robertas (2021). Neural network hyperparameter optimisation for prediction of real estate prices in helsinki. *PeerJ Computer Science*, *7*, Article e444.
- Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimisation. In *Proceedings of the IEEE international conference on neural networks* (pp. 1942–1948).
- Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *IEEE international conference on systems, man, and cybernetics, computational cybernetics and simulation*, Vol. 5 (pp. 4104–4108).
- Kim, M. K., Kim, Y. S., & Srebric, J. (2020). Predictions of electricity consumption in a campus building using occupant rates and weather elements with sensitivity analysis: Artificial neural network vs. linear regression. *Sustainable Cities and Society*, *62*, Article 102385.
- Kusiak, A., Li, M. Y., & Zhang, Z. J. (2010). A data-driven approach for steam load prediction in buildings. *Applied Energy*, *87*, 925–933.
- Kuster, Corentin, Rezgui, Y., & Mourshed, M. (2017). Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society*, *35*, 257–270.
- Li, K., Hu, C., Liu, G., & Xue, W. (2015). Building's electricity consumption prediction using optimised artificial neural networks and principal component analysis. *Energy and Buildings*, *108*, 106–113.
- Li, K., Xie, X., Xue, W., Dai, X., Chen, X., & Yang, X. (2018). A hybrid teaching-learning artificial neural network for building electrical energy consumption prediction. *Energy and Buildings*, *174*, 323–334.
- Lu, R., & Hong, S. H. (2019). Incentive-based demand response for smart grid with reinforcement learning and deep neural network. *Applied Energy*, *236*, 937–949.
- Luo, X. J. (2020). A novel clustering-enhanced adaptive artificial neural network model for predicting day-ahead building cooling demand. *Journal of Building Engineering*, *32*, Article 101504.
- Luo, X. J., & Fong, K. F. (2019). Development of integrated demand and supply side management strategy of multi-energy system for residential building application. *Applied Energy*, *242*, 570–587.
- Luo, X. J., Oyedele, L. O., Ajayi, A. O., & Akinade, O. O. (2020). Comparative study of machine learning-based multi-objective prediction framework for multiple building energy loads. *Sustainable Cities and Society*, *61*, Article 102283.
- Luo, X. J., Oyedele, L. O., Ajayi, A. O., Akinade, O. O., Delgado, J. M. D., Owolabi, H. A., & Ahmed, A. (2020). Genetic algorithm-determined deep feedforward neural network architecture for predicting electricity consumption in real buildings. *Energy and AI*, *2*, Article 100015.
- Luo, X. J., Oyedele, L. O., Ajayi, A. O., Akinade, O. O., Owolabi, H. A., & Ahmed, A. (2020). Feature extraction and genetic algorithm enhanced adaptive deep neural network for energy consumption prediction in buildings. *Renewable and Sustainable Energy Reviews*, *131*, Article 109980.
- Luo, X. J., Oyedele, L. O., Ajayi, A. O., Chukwuka, M., Akinade, O. O., & Lukman, A. (2019). Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands. *Advanced Engineering Informatics*, *41*, Article 100926.
- Luo, X. J., Oyedele, L. O., Akinade, O., & Ajayi, A. O. (2020). Two-stage capacity optimisation approach of multi-energy system considering its optimal operation. *Energy and AI*, Article 100005.
- Mena, R., Rodríguez, F., Castilla, M., & Arahal, M. R. (2014). A prediction model based on neural networks for the energy consumption of a bioclimatic building. *Energy and Buildings*, *82*, 142–155.

- Moon, J., Park, S., Rho, S., & Hwang, E. (2019). A comparative analysis of artificial neural network architectures for building energy consumption forecasting. *International Journal of Distributed Sensor Networks*, 15, 9.
- Muralitharan, K., Sakthivel, R., & Vishnuvarthan, R. (2018). Neural network based optimisation approach for energy demand prediction in smart grid. *Neurocomputing*, 273, 199–208.
- (1993). *NeuroShell 2 manual*. Frederick, MA: Ward System Group, Inc..
- Paneiro, G., & Rafael, M. (2021). Artificial neural network with a cross-validation approach to blast-induced ground vibration propagation modeling. *Underground Space*, 6, 281–289.
- Roman Cardell, J. (2020). *Python-based deep-learning methods for energy consumption forecasting* (Bachelor's thesis), Universitat Politècnica de Catalunya.
- Ruiz, L. G. B., Rueda, R., Cuéllar, M. P., & Pegalajar, M. C. (2018). Energy consumption forecasting based on Elman neural networks with evolutive optimisation. *Expert Systems with Applications*, 92(2018), 380–389.
- Sieminski, A. (2017). *International energy outlook* (pp. 5–30). Energy Information Administration.
- Wang, L., Eric, W. M. L., & Richard, K. K. Y. (2018). Novel dynamic forecasting model for building cooling loads combining an artificial neural network and an ensemble approach. *Applied Energy*, 228, 1740–1753.
- Wang, J., & Yin, Z. (2008). A ranking selection-based particle swarm optimiser for engineering design optimisation problems. *Structural and Multidisciplinary Optimisation*, 37, 131–147.
- <https://weather.com/en-GB/> (last accessed 22 Sept 2021).
- [https://dpds.weatheronline.co.uk/historical\\_data/weather\\_stations\\_download](https://dpds.weatheronline.co.uk/historical_data/weather_stations_download) (last accessed 22 Sept 2021).
- Yang, J., Hugues, R., & Radu, Z. (2005). Online building energy prediction using adaptive artificial neural networks. *Energy and Buildings*, 37, 1250–1259.
- Yang, J., Rivard, H., & Zmeureanu, R. (2005). Online building energy prediction using adaptive artificial neural networks. *Energy and Buildings*, 37, 1250–1259.
- Zhang, Liang, Wen, J., Li, Y., Chen, J., Ye, Y., Fu, Y., & Livingood, W. (2021). A review of machine learning in building load prediction. *Applied Energy*, 285, Article 116452.
- Zhou, Tianrui, Hu, Qinyou, Hu, Zhihui, & Zhen, Rong (2021). An adaptive hyper parameter tuning model for ship fuel consumption prediction under complex maritime environments. *Journal of Ocean Engineering and Science*, Pre-proof.

### Further reading

- <https://www.accuweather.com> (last accessed 22 Sept 2021).
- Close, R. (1989). Theory of the backpropagation neural network. In *Proceedings of IEEE international conference on neural networks, Vol. 1* (pp. 593–605).
- <https://www.metoffice.gov.uk/> (last accessed 22 Sept 2021).