Multivariate Electricity Consumption Prediction with Extreme Learning Machine

Hui Song
Computer Science and
Information Techonology
RMIT University
Melbourne, VIC 3000, Australia
Email: hui.song@rmit.edu.au

A. K. Qin
Computer Science and
Information Techonology
RMIT University
Melbourne, VIC 3000, Australia
Email: kai.qin@rmit.edu.au

Flora D. Salim
Computer Science and
Information Techonology
RMIT University
Melbourne, VIC 3000, Australia
Email: flora.salim@rmit.edu.au

***Abstract*—In this paper, Extreme Learning Machine (ELM) is demonstrated to be a powerful tool for electricity consumption prediction based on its competitive prediction accuracy and superior computational speed compared to Support Vector Machine (SVM). Moreover, ELM is utilized to investigate the potentials of using auxiliary information such as electricity-related factors and environmental factors to augment the prediction accuracy obtained by purely using the electricity consumption factors. Furthermore, we formulate a combinatorial optimization problem of seeking an optimal subset of auxiliary factors and their corresponding optimal window sizes using the most suitable ELM structure, and propose a Discrete Dynamic Multi-Swarm Particle Swarm Optimization (DDMS-PSO) to address this problem. Experimental studies on a real-world building dataset demonstrate that electricity-related factors improve accuracy while environmental factors further boost accuracy. By using DDMS-PSO, we find a subset of electricity-related and environmental factors, their respective window sizes, and the number of hidden neurons in ELM which leads to the best prediction accuracy.**

I. Introduction

The rapidly growing world energy usage has caused issues of supply difficulties, exhaustion of energy resources and heavy environmental impact. Electricity consumption as a form of energy consumption, with rapid expansion of residential and commercial areas, has grown rapidly, which is a threat for sustainable development. The prediction of electricity consumption in buildings does not only help to improve energy monitoring and usage in buildings, but plays a vital role in improving the electricity performance, with the aim of achieving energy consumption conservation and reducing the environmental impact [1]. Also, electricity consumption prediction plays a significant role in decision-making and future planning that rely on prediction accuracy.

Electricity consumption prediction has a history of more than 20 years. It has been a way to measure the characteristics of buildings and aid the development of electric power plants, therefore electricity prediction has been an application of time series analysis [2]. Electricity consumption prediction is an indispensable part of managing and researching power systems, and it can make full use of electricity and ease the conflict between supply and demand based on the analysis of the existing electric energy [3]. Electricity consumption is highly dependent on electric power, economic, social and meteorological factors. The precision of prediction is very important for informing the analyses of electric power exchange, trading evaluation, network function, security and trends, and the safety strategy of reduction load [4].

Existing electricity consumption prediction research always involves multi-variate historical data. Different researchers use different attributes to analyze this issue. Thermal comfort is a criterion to evaluate the environment and can be regarded as a factor which will affect electricity usage [5]. Moreover, some other factors such as temperature, events, wind speed or features in the buildings are always taken into consideration for improving the prediction accuracy. In [6], the area of windows, walls, partitions and floors, the type of windows and walls are regarded as factors which have direct or indirect relationship with electricity consumption. In [7], temperatures, space heating demand, water heating demand and energy demand are applied to help improve the performance of prediction.

In this project, we aim to predict electricity usage of a university building. In this scenario, ELM is used for electricity consumption prediction through exploring the predictive performance of historical electricity consumption logs along with electricity-related and environmental data. Furthermore, a Discrete Dynamic Multi-Swarm Particle Swarm Optimization (DDMS-PSO) is proposed to address the optimization problem of discrete values in order to find a subset of all factors, regardless of the heterogeneity of their feature space, with their respective window sizes and the number of hidden neurons which can lead to the best prediction performance.

The dataset is collected from the smart meters of the buildings in the city campus of RMIT University, Melbourne. We mainly focus on understanding the trend of electricity consumption under the influence of electricity-related and environmental factors on energy usage. Auxiliary environmental data is crawled from an online weather station that broadcasts periodic readings from every 20-minutes to 1-hour. The output of prediction is useful to help event planning and resource management. The following aspects are explored in this paper.

- Historical smart meter data is used to evaluate and forecast the buildings' future electricity consumption. Then the result is regarded as the baseline to evaluate if the

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

auxiliary information can help improve the prediction accuracy.

- Electricity-related factors are added to identify if they have effect on improving the prediction performance and which factors can increase the accuracy the most.
- Based on the best combination of the electricity-related factors, explore if environmental factors actually influence the prediction accuracy and the optimal subset of the environmental factors.
- Evolutionary algorithm is applied to explore if there is a subset of electricity-related and environmental factors, their respective window sizes and a suitable number of hidden neurons in ELM to generate the best prediction accuracy.

The rest of this paper is organized as follows. Section 2 presents the related work on electricity consumption prediction. Problem definition and proposed solutions to address the problems in this paper will be introduced in section 3. Section 4 mainly focuses on experimental settings and related results. Conclusions and future work will be given in section 5.

II. Related Work

A. Electricity Consumption Prediction

Many methodologies have been employed for electricity consumption prediction, which include Aritifical Neural Networks (ANNs) [8], Fuzzy Inference System (FIS) [9] and Support Vector Machines (SVMs) [10]. Moreover, Zhao and Magoulès [2] categorize the methodologies into five different kinds based on the reviewed paper: engineering methods, statistical methods, ANN, Support Vector Machines (SVM) [11] and Grey Models. Foucquier et al. [1] summarize the state of the art methods as stastical methods which include multiple Linear Regression or conditional analysis, Genetic Algorithm (GA), ANN and SVM as well as some hybrid models which combine two or three of the above methods. Among all the methodologies, ANN and Support Vector Machine for regression, so called Support Vector Regression (SVR) are widely used in electricity consumption [12]. However, it is very difficult to say which one outperforms others without complete comparison under the same circumstances because each of them is still being developed [2].

B. ELM and Its Applications

A new learning method which was proposed by Huang et al. [13], called Extreme Learning Machine (ELM), mainly focuses on solving the drawbacks caused by gradient descent based algorithms such as Back Propagation (BP). ELM is based on Single hidden Layer Feedforward Neural Network architecture and includes three different layers, input layer, hidden layer and output layer. The hidden bias and the weight for connecting input layer and hidden layer are generated randomly and maintained through the whole training process.

Assuming dataset $(\mathbf{x}_i, \mathbf{y}_i)$ with a set of M distinct samples, satisfy $\mathbf{x}_i \in \mathcal{R}^{d1}$ and $\mathbf{y}_i \in \mathcal{R}^{d2}$, so a SLNF with N hidden neurons can be mathematically formulated as:

$$\sum_{i=1}^{N} \beta_i \mathbf{f}(\mathbf{w}_i^T \mathbf{x}_j + b_i), 1 \leq j \leq M \tag{1}$$

Where f is the activation function; $\mathbf{w}_i$ represents the weight for connecting input layer and hidden layer; b_i is bias and β_i is the output weight.

In ELM, the structure perfectly approximates to the given output data:

$$\sum_{i=1}^{N} \beta_i \mathbf{f}(\mathbf{w}_i^T \mathbf{x}_j + b_i) = \mathbf{y}_j, 1 \leq j \leq M \tag{2}$$

Which can be written as $\mathbf{HB} = \mathbf{Y}$, the matrix $\mathbf{H}$ can be represented as:

$$\mathbf{H} = \begin{pmatrix} f(\mathbf{w}_1^T \mathbf{x}_1 + b_1) & ... & f(\mathbf{w}_N^T \mathbf{x}_1 + b_N) \\ ... & ... & ... \\ f(\mathbf{w}_1^T \mathbf{x}_M + b_1) & ... & f(\mathbf{w}_N^T \mathbf{x}_M + b_N) \end{pmatrix} \tag{3}$$

$\mathbf{B} = (\beta_1^T, \beta_2^T, ..., \beta_N^T)^T$ and $\mathbf{Y} = (y_1^T, y_2^T, ..., y_M^T)^T$.

The obvious difference between ELM and other Neural Networks is from the hidden layer to output layer. In ELM, in order to calculate the output weight $\mathbf{B}$ from the knowledge of the hidden layer output matrix $\mathbf{H}$ and the target values, the matrix $\mathbf{H}$ which is a Moore-Penrose generalized inverse is proposed and denoted as $\mathbf{H}^+$ [14]. Theoretical proofs and a more thorough presentation of the ELM algorithm are detailed in the original paper [13].

ELM does not have so many parameters to be adjusted except for the number of hidden neurons, which makes it easier to be applied in regression [15] or classification [16] issues and very low computational cost during the process of training. Recently, ELM has been gradually gained much attention for its application in time series prediction, such as predicting sales in fashion retailing in [17]. In [18], ELM is utilized for electricity price forecasting and has demonstrated its fast computational ability. ELM is used for wind power density prediction in [19] and compared with ANN and SVM. In [20], ELM has been successfully applied for daily dew point temperature prediction.

C. Evolutionary Algorithms for Energy Consumption Prediction

Evolutionary Algorithm (EA) [21] is being widely used to handle large-scale, non-differentiable and complex multi-mode problem without any information about optimized problems for its global convergence ability and strong robustness.

Recently, many optimization algorithms have been successfully applied in time series prediction. In [22], Particle Swarm Optimization (PSO) is used to optimize the input subset for SVM in time series predicton. Shafie-Khah et al. [23] utilize PSO to optimize RBFN for obtaining a robustness prediction structure for price forecasting of electricity markets. Azadeh et al. [24] use GA to tune all parameters for NN applied in predicting electrical energy consumption. In [25], almost all methods such as GA, PSO, Ant colony optimization

(ACO), Differential Evolution (DE) applied for renewable and sustainable energy are reviewed. Therefore, the optimization methodologies are popular for solving parameters adjustment in energy consumption prediction.

III. Problem Definition

A. Scenario Assumption

Assuming T which represents the length of time series, is expressed as $T = \{t_1, t_2, ..., t_q\}$, q means the number of sample points, therefore our related time series dataset include 3 aspects:

- Electricity consumption x_T
- Electricity-related factors r_T, defined as $r_T = \{r_T^1, r_T^2, ..., r_T^n\}$, where n is the number of the internal factors.
- Environmental factors z_T, defined as $z_T = \{z_T^1, z_T^2, ..., z_T^m\}$, where m is the number of environmental time series, including temperature, dew point(a measure of atmospheric moisture), humidity(the amount of water vapor in the air), wind speed(caused by air moving from high pressure to low pressure, usually due to changes in temperature), sea level(offer insights into ongoing climate change) etc.

B. Problem Definition

In time series prediction, there are one-step-ahead prediction and multi-step-ahead prediction. It is very clear that one-step-ahead mainly focuses on the next single value ahead while multi-step-ahead prediction takes multiple future values into consideration. As it is known that multi-step-ahead prediction is much more complex because of the accumulation of errors and increasing uncertainties, we are focusing on exploring an optimal subset of auxiliary factors for the best prediction accuracy, so in order to decrease the influence of other uncertainties for this problem, we only focus on one-step-ahead time series prediction.

We have three different types of dataset, electricity consumption x_T, electricity-related factors $r_T, r_T \subset \mathcal{R}^n$, environmental factors $z_T, z_T \subset \mathcal{R}^m$. In order to explore the influence of all these factors, two general problems which should be solved are defined as follows:

- Explore if electricity-related factors and environmental factors improve prediction accuracy.
- Explore if there is an optimal subset of all auxiliary factors, their respective window sizes, and the number of hidden neurons in ELM which can obtain the best prediction performance.

C. Proposed Solutions

1) An Approach for Solving The First Problem: ELM has been proven to be capable of universal approximation in a satisfied sense, and it has been shown to have good generalization capabilities and extremely fast speed [16]. The only task for applications is to select a suitable activation function and set the number of hidden neurons. Moreover, compared with conventional learning approaches, it avoids many difficulties such as learning rates, learning epochs, stop criteria and local optima [15]. All the advantages are motivations for us to utilize it as a basic prediction model for electricity consumption prediction.

There are two different kinds of factors to be taken into consideration as well as historical electricity data. In order to solve the first problem, the factors are added gradually with several different predefined window sizes, three subproblems for addressing the first problem are defined as follows:

- Only use the historical electricity consumption to realize prediction in order to further identify if others factors have effect on improving prediction performance.
  With the aim of finding out which factors will improve the prediction accuracy, a baseline is necessary for a further comparison. All prediction results, which will be generated with different combinations of factors must be compared with the result from only historical electricity consumption.
  Assuming $H = H_1, H_2, ..., H_D, d = 1, 2, ..., D$ is the set of different time series, then we have:
  $$x_{t+1} = f(x_t, x_{t-1}, ..., x_{t-H_d-1}) \quad (4)$$
  Where H_d is the length of d^{th} time window used for prediction and $f()$ is the prediction function of a certain predictor.
- Find which factors influence the prediction accuracy the most among all electricity-related factors.
  The electricity-related factors are added to the historical electricity consumption dataset to explore their single and overall performances with the predefined window sizes and compared with the result from purely electricity consumption.
  $$\begin{aligned} x_{t+1} = f(x_t, x_{t-1}, ..., x_{t-H_d-1}; \\ r_t^i, r_{t-1}^i, ..., r_{t-H_d-1}^i) \\ i = 1, 2, ..., m \end{aligned} \quad (5)$$
- Among the environmental factors, find which of them help to improve the prediction accuracy for this problem with the influence of electricity-related factors.
  Based on the electricity-related factors, a further exploration about the optimal combination of environmental factors is necessary.
  $$\begin{aligned} x_{t+1} = f(x_t, x_{t-1}, ..., x_{t-H_d-1}; \\ r_t^c, r_{t-1}^c, ..., r_{t-H_d-1}^c; \\ z_t^j, z_{t-1}^j, ..., z_{t-H_d-1}^j) \\ j = 1, 2, ..., n \end{aligned} \quad (6)$$
  Where c represents the electricity-related factors which have an effect on improving the prediction accuracy.

The first problem is addressed through investigating the potential influence of electricity-related factors and environmental factors step by step, which means the next step is always based on the previously selected factors, hence it is necessary to find an optimal subset among all auxiliary factors. Then it comes to the second problem, with a larger set of factors, there are many different combinations, which will cause highly computational cost, therefore a new approach is necessary to address the problem.

2) Discrete Dynamic Multi-Swarm Particle Swarm Optimization for Addressing The Second Problem: PSO is a suitable method to solve optimization problems [26], including local PSO and global PSO. For the global version of PSO, each particle updates its velocity and position according to the best solution found so far by itself and the best solution found so far by the whole population. In the local version of PSO, each particle adjusts its velocity and position through its personal best and the best solution achieved so far within its neibourhood. Compared with global PSO, local PSO has better global search ability [27] because global PSO is easier to be trapped into local optimum. For the second poblem, there are many different combinatorial possibilities with not only one optimal solution, which means it is a multimodal problem, therefore a local version of PSO is necessary.

Moreover, we use binary to describe whether a factor is selected or not. 0 means a factor that is not selected while 1 is the opposite. Also, window sizes and the number of hidden neurons parts are discrete. Therefore, this problem becomes a discrete and combinatorial optimization problem.

We propose a Discrete Dynamic Multi-Swarm Particle Swarm Optimization (DDMS-PSO) to solve the multimodal discrete and combinatorial optimization problem. DDMS-PSO is improved from the algorithm of Dynamic Multi-Swarm Particle Swarm Optimizer (DMS-PSO) [28] and especially used for solving our problem. DMS-PSO is based on the local version of PSO with a periodically dynamic neighborhood topology for solving continuous optimization problem. The search process of the Dynamic Multi-Swarm Particle Swarm Optimizer is as follows: A whole swarm is divided into several sub-swarms randomly, each with the number of particles. Then each sub-swarm searches for its best solution according to its historical information and the best solutions obtained so far in its group. After some generations, all the particles are mixed together and divided again (This procedure is always defined as Regroup period, denoted as R). Then this process is repeated until the stop criterion is satisfied. In this way, each sub-swarm's information has the chance to be exchanged with others'. The Dynamic Multi-Swarm Particle Swarm Optimizer has been proven to perform better than some other PSO variants on many complex issues [29]. DDMS-PSO extends DMS-PSO to be applied in discrete optimization problem with obeying the same search process of DMS-PSO. It deals with the solutions before calculating the fitness for discrete problem in a different way.

In order to update the velocity and position of each particle synchronously, all parameters that need to be optimized are mapped to $[0, 1]$. The position updating equations of DDMS-PSO can be described as follows:

$$\begin{aligned} v_{id}^{k+1} &= \omega * v_{id}^{k} + c_1 * r_1 * (pbest_{id}^{k} - X_{id}^{k}) \\ &\quad + c_2 * r_2 * (lbest_{pd}^{k} - X_{id}^{k}) \\ v_{id}^{k+1} &= min(V_{max}^{id}, max(-V_{max}^{id}, v_{id}^{k+1})) \qquad (7) \\ x_{id}^{k+1} &= x_{id}^{k} + v_{id}^{k+1} \qquad (8) \end{aligned}$$

Where $x_{max} = 1$, $x_{min} = 0$ and $V_{max} = 0.2 * (x_{max} - x_{min})$. ω is an inertia weight which plays a significant role in balancing the global and local search ability. $lbest = (lbest_{p1}, lbest_{p2}, ..., lbest_{pD})^T$ is the best position achieved within its neighborhood and p represents the number of sub-swarms. $d = 1, 2, ..., d_1, d_1 + 1, ..., 2 * d_2 + 1, D$; and d_1 is the number of factors and there are $d_1 + 1$ window sizes for these corresponding d_1 factors and historical electricity consumption respectively. The last dimension represents the number of hidden neurons. $i = 1, 2, ..., n$, n is population size, k is the number of the iteration; v_{id} is the velocity of the i^{th} particle; c_1 and c_2 are acceleration factors used to represent the weighting of stochastic acceleration terms that pull each particle towards $pbest$ and $lbest$. r_1 and $r_2 \in [0, 1]$ are two random numbers.

Out of bound issues need to be addressed, given that all the factors have different ranges in the feature space. To deal with the particle out of range problem, we constrain the bounds to $[0, 1]$. The values of 0 and 1 will determine the selection of the factor x_d, which will decrease the diversity of the solutions. Therefore when the particles' positions are out of bound, they are set randomly in $(0, 1)$. For Eq. (8), it will be further constrained as:

$$\begin{aligned} x_{id}^{k+1} = (x_{id}^{k+1} > x_{max}) * r_1 + (x_{id}^{k+1} < x_{min}) * r_2 \\ r_1, r_2 \in (0, 1) \qquad (9) \end{aligned}$$

After updating the position of each particle, in order to calculate the fitness, the factors selection part will be reflected to only 0 and 1 while window sizes and the number of hidden neurons in ELM will be mapped to their real values as follows:

$$\begin{aligned} x_{id} &= round(x_{id}), 0 < d =< d_1 \\ x_{id} &= (X_{Trmax} - X_{Trmin}) * x_{id} + X_{Trmin}, d_1 < d < D \\ x_{id} &= 50 * fix(((X_{rmax} - X_{rmin}) * x_{id} + X_{rmin})/50), \\ &\qquad d = D \quad (10) \end{aligned}$$

Where X_{Trmax} and X_{Trmin} are the maximum and minimum values of the window sizes. X_{rmax} and X_{rmin} are the maximum and minimum number of hidden neurons in ELM.

Several necessary notations for the pseudo code are the following:

ns: the number of particles in a sub-swarm
p: the number of sub-swarms
n: population size, $n = ns * p$

R: Regrouping Period

Max_FEs: Max fitness evaluations, stop criterion

Pseudo code used for describing the procedure of DDMS-PSO is as follows:

Algorithm 1 DDMS-PSO

Initialize a population of n particles with random values positions **x** and velocities **v** in the range of $[0, 1]$ from D dimensions in the search space
Map each particle's position to its real range with (10)
Evaluate the fitness
Divide the population into p sub-swarms randomly, with ns particles in each sub-swarm, find each particle's local best **lbest** and set **pbest** $=$ **x**
while $t < Max\_FEs$ **do**
    for Each particle i **do**
        Adapt velocity of each particle using Eq. (7)
        Update the position of each particle with Eq. (8)
        Bound the constraint of each particle as Eq. (9)
        Map each particle's position to its real range with Eq. (10)
        Evaluate the fitness $f(x_i)$
        if $f(x_i) < pbest_i)$ **then**
            $pbest_i \leftarrow x_i$
        end if
        if $pbest_i < lbest_i$ **then**
            $lbest_i \leftarrow pbest_i$
        end if
        if $mod(t, R) = 0$ **then**
            Regroup the sub-swarms randomly
        end if
    end for
end while

IV. Experimental Result

The proposed problems will be solved through addressing the following sub-problems step by step:

- Experiment 1
  Use only historical electricity consumption data to evaluate electricity prediction (Part D is the result);
- Experiment 2
  Add electricity-related factors to explore if they can improve the prediction accuracy and find the subset which influences the most (Part E is the result);
- Experiment 3
  Based on the optimal subset of electricity-related factors, explore the environmental factors which can further improve the prediction performance (Part F is the result);
- Experiment 4
  Apply DDMS-PSO to find the optimal subset of all auxiliary information, their corresponding window sizes and the number of hidden neurons in ELM which can lead to the best prediction accuracy (Part G is the result).

Experiment 1 will identify if ELM is comparative with SVR, then the result will be regarded as the baseline to be compared with experiment 2. Furthermore, experiment 3 will be compared with the result of experiment 2 to identify that environmental factors can further improve the prediction accuracy. Subproblem 1 is solved by experiment 1, 2 and 3 while experiment 4 addresses the subproblem 2.

A. Data Description

Datasets of electricity consumption, electricity-related factors and environmental factors are from 01.06.2014 to 31.05.2015 with 12 points in one day and normalized to $[0, 1]$. The dataset is sampled according to the number of points in one day with different window sizes for one-step-ahead prediction. The window sizes are set from 5 to 60 with an interval of 5. Since one year data is used, there will be 365 samples for each point. However, considering the maximum window size is 60, which means for a point to be predicted, there is at least 60 points before it, therefore there will be 360 samples at most for each one of these 12 points under the maximum window size. In order to compare the influence of the window sizes under the same output, we divide the dataset into 360 samples for each of the 12 points.

When ELM is used as the basic predictor, as we know that the weight and the bias are generated randomly, it means the weight and bias are different every time when ELM is run, which will cause the result unstable. In order to address this issue, cross validation is applied.

B. Evaluation Method

Validation techniques are proposed for model selection and performance evaluation. Error rate or accuracy is used to evaluate the performance. The next issue to be considered is how to apply the dataset for training and testing. Using the whole dataset as training data will cause overfitting of the final model for the training data. Another validation technique is the holdout method, which has the issue that a sparse dataset may not be able to set aside a part of the dataset for testing. Cross validation as a resampling method which is always carried out by splitting the dataset into a fixed number of examples without replacement randomly, can overcome the drawbacks of the previous methods. For each data split, the model is retained from scratching with the training examples, then it is evaluated through the test examples.

K-fold cross-validation is to divide the dataset into K partitions. For each K experiments, $K - 1$ folds are used for training and the rest for testing, then the mean value of these K results is regarded as the final result. In order to make the result much more stable and convincing, our result is evaluated by 10 10-fold cross-validation with Root Mean Square Error (RMSE). For each folder, the result is an average value of running 10 times. The final result is the average of the 10 10-fold. Since there are several different window sizes, in order to compare the influence of window sizes, for each 10-fold with different window sizes, the random number for training and testing is the same, which means although the window sizes are different, the output of training and testing are the same under the same 10-fold.

TABLE I

THE COMPARISON OF ELM AND SVR WITH ONLY HISTORICAL ELECTRICITY CONSUMPTION

			RMSE result under different window sizes											
			5	10	15	20	25	30	35	40	45	50	55	60
ELM	Number of hidden neurons	50	0.0698	0.0645	0.0632	0.0633	0.0639	0.0645	0.0651	0.0658	0.0668	0.0679	0.0689	0.0700
		100	0.0702	0.0644	0.0624	**0.0621**	0.0627	0.0630	0.0631	0.0633	0.0636	0.0640	0.0646	0.0649
		200	0.0835	0.0663	0.0629	**0.0622**	0.0630	0.0630	0.0631	0.0632	0.0634	0.0633	0.0637	0.0638
		300	0.1211	0.0697	0.0648	0.0637	0.0643	0.0638	0.0641	0.0640	0.0642	0.0641	0.0643	0.0642
		400	0.2316	0.0757	0.0680	0.0657	0.0663	0.0654	0.0657	0.0652	0.0655	0.0655	0.0654	0.0651
		500	0.4544	0.0860	0.0725	0.0684	0.0690	0.0674	0.0675	0.0670	0.0670	0.0670	0.0668	0.0665
		800	1.3918	0.1491	0.0956	0.0833	0.0807	0.0764	0.0752	0.0739	0.0736	0.0732	0.0725	0.0719
		1000	1.9661	0.2206	0.1202	0.0994	0.0920	0.0855	0.0823	0.0802	0.0792	0.0788	0.0774	0.0767
SVR	Different parameters settings	H_1	0.0735	0.0671	0.0643	0.0639	0.0640	0.0642	0.0640	0.0638	0.0639	0.0638	0.0638	0.0638
		H_2	0.0726	0.0662	0.0638	0.0633	0.0636	0.0636	0.0634	0.0633	0.0633	0.0632	0.0632	0.0633
		H_3	0.0721	0.0656	0.0635	0.0632	0.0632	0.0632	0.0631	0.0629	0.0630	0.0628	0.0629	0.0630
		H_4	0.0721	0.0652	0.0635	0.0631	0.0632	0.0633	0.0630	0.0630	0.0630	0.0631	0.0633	0.0633
		H_5	0.0774	0.0686	0.0669	0.0662	0.0662	0.0662	0.0657	0.0657	0.0657	0.0657	0.0660	0.0659
		H_6	0.0828	0.0728	0.0708	0.0702	0.0701	0.0702	0.0692	0.0699	0.0696	0.0694	0.0691	0.0693
		H_7	0.1051	0.0836	0.0851	0.0825	0.0825	0.0834	0.0809	0.0812	0.0816	0.0819	0.0822	0.0827
		H_8	0.1197	0.0980	0.0979	0.0956	0.0974	0.0967	0.0967	0.0968	0.0973	0.0974	0.0986	0.0992
		H_9	0.1306	0.1043	0.1096	0.1100	0.1114	0.1086	0.1096	0.1091	0.1098	0.1099	0.1117	0.1133

TABLE II

SUBSET OF ELECTRICITY-RELATED FACTORS WITH THE BEST PREDICTION PERFORMANCE

		Electricity-related factor		**Number of hidden neurons**	**RMSE**
		A_1	A_2		
Window sizes	15	1	1	200	**0.0536**
	20	1	1	200	**0.0536**

C. Experimental Settings

- ELM parameter settings
  The parameter in ELM is only the number of hidden neurons, set as 50, 100, 200, 300, 400, 500, 800, 1000.
  Activation function: Sigmoid function
- SVR parameter settings
  There are two different kinds of SVRs, called ϵ-SVR model and nu-SVR model. In this paper, ϵ-SVR model is implemented using the LibSVM library [30] for SVR.
  Kernel type: Radial Basis Function
  C (Cost): 2^{-5} 1 2^{5} 2^{10} 2^{15}
  ϵ: 0.01 0.03 0.05 0.08 0.1 0.15 0.18 0.2
- DDMS-PSO parameter settings
  $\omega = 0.729$, $c_1 = c_2 = 1.49445$, $R = 10$, $n = 30$, $ns = 10$, $p = 3$, $Max\_FEs = 1000$
  $D = 16$, the previous 7 dimensions are all auxiliary factors; followed it 8 dimensions are their respective window sizes and the window size for electricity factor; the last deimension represents the number of hidden neurons.
  X_{Trmax} and X_{Trmin} are 5 and 35 respectively, therefore the final optimal window size will be the values between 5 and 35.
  X_{rmax} and X_{rmin} are 50 and 600, hence the optimized number of hidden neurons in ELM will be the values between 50 and 600 with an interval of 50.
- Factors expression
  Electricity usage: E
  Electricity-related factors: A_1 (apparent power), A_2 (power factor)
  Environmental factors: F_1 (temperature), F_2 (dew point), F_3 (humidity), F_4 (wind speed), F_5 (sea level)
  In the following results, 0 means the factor is not selected while 1 is selected. For the electricity usage, it is always selected.

D. Experimental Result with Only Historical Electricity Consumption

Although there are 5 and 8 different settings for C and ϵ respectively, some of the results are same because ϵ plays a dominant role in the prediction result. Therefore, in most cases when ϵ is the same, the results stay the same. For SVR, only the different results of the parameter settings are listed, denoted as H_1, H_2, H_3, H_4, H_5, H_6, H_7, H_8 and H_9.

In order to identify that ELM performs comparably with SVR, the statistical method Signrank is applied to demonstrate the difference between ELM and SVR. For Signrank, the returned result $h = 1$ indicates a rejection of the null hypothesis, while $h = 0$ indicates a failure to reject the null hypothesis at the 5% significance level. Moreover, in order to present the result much more clearly, the best result averaged from 10 10-fold cross validation will be labeled bold and compared with others using Signrank. The results which have no obvious difference with the best one will be labeled bold as well. Tab. 1 is the result for ELM and SVR seperately.

From Tab.1 we can see when the window size is set as 20 and the number of hidden neurons is 200, ELM obtains the best average result 0.0621. Then the best result in ELM is used to compare with all the rest results, including the results from different parameter settings of SVR. From the labeled data, it can be clearly seen that ELM absolutely performs better than SVR. Also, when the window size is set as 20, the number of hidden neurons in ELM are set as 100 or 200, the results have no differences.

TABLE III
SUBSETS OF ENVIRONMENTAL FACTORS FOR IMPROVING PREDICTION ACCURACY MOSTLY

		Environmental factors					Number of hidden neurons	RMSE
		F_1	F_2	F_3	F_4	F_5		
Window	15	1	1	0	1	0	200	**0.0491**
sizes	15	1	1	0	1	0	300	**0.0492**

TABLE IV
OPTIMAL SUBSET OF ALL FACTORS, RESPECTIVE WINDOW SIZES AND NUMBER OF HIDDEN NEURONS BASED ON DDMS-PSO

	Auxiliary factors							Window sizes								Number of hidden neurons	RMSE
	F_1	F_2	F_3	F_4	F_5	A_1	A_2	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8		
1	1	1	0	1	0	1	0	21	9	0	15	0	19	0	12	250	**0.0483**
2	1	1	0	1	0	1	0	22	8	0	13	0	19	0	12	250	**0.0483**
3	1	1	0	1	0	1	0	21	6	0	16	0	19	0	12	250	**0.0483**
4	1	1	0	1	0	1	0	28	6	0	15	0	19	0	12	250	**0.0483**
5	1	1	0	1	0	1	0	21	11	0	15	0	19	0	12	250	**0.0483**
6	1	1	0	1	0	1	1	20	8	0	10	0	18	23	12	250	**0.0483**
7	1	1	0	1	0	1	0	20	7	0	14	0	19	0	12	250	**0.0483**
8	1	1	0	1	0	1	0	23	5	0	14	0	19	0	12	250	**0.0483**
9	1	1	0	1	0	1	0	21	6	0	21	0	19	0	13	250	**0.0483**
10	1	1	0	1	0	1	0	26	19	0	14	0	19	0	12	250	**0.0484**

E. Experimental Result with Electricity-Related Factors

Since there are two electricity-related factors, there are three different combinations for these two factors, which are A_1, A_2 and A_1 combined with A_2. Comparing all the results with each other, the best combination is presented in Tab.2.

Tab.2 tells us when A_1 and A_2 are combined together, the effectiveness is better. Furthermore, when it is compared with Tab. 1, the accuracy improves significantly, which further demonstrates that A_1 and A_2 play a significant role in improving the prediction accuracy.

F. Experimental Result with Environmental Factors

Since there are 5 different environmental factors, so 31 different combinations need to be tested. Moreover, because we have proven that the electricity-related factors can improve the prediction accuracy, for each combination, we add A_1 and A_2 in it to explore the influence. Comparing all the obtained results, the best combination and their parameters settings are presented in Tab.3.

Tab.3 shows that when environmental factors F_1, F_2 and F_4 are combined based on A_1 and A_2, the prediction accuracy will be improved the most. Moreover, under the same window size and optimal subset of environmental factors, although the number of hidden neurons can be different, the results have no obvious difference with each other.

G. DDMS-PSO-Based Experimental Result

In the previous experiments, the influence of each kind of factors are explored step by step for addressing the first problem. T_1 to T_5 are the window sizes of F_1 to F_5, respectively. The window sizes of A_1 and A_2 are T_6 and T_7. T_8 represents the window size of the electricity factor. The experiment result with running 10 times for the second problem is presented in Tab.4.

Tab.4 shows that when the subsets of factors are F_1, F_2, F_4, A_1, A_2 and F_1, F_2, F_4, A_1 with their corresponding window sizes and an optimized number of hidden neurons in ELM, it will lead to the best prediction accuracy compared with other experimental results. Also, the number of hidden neurons are stable at 250. Moreover, from the last column, it can be seen that the prediction results are quite stable at 0.0483, which indicates that DDMS-PSO has good capability for finding the best and stable solution for this prediction problem.

From Tab.1 to Tab.4, the following conclusions are derived:

- Compared with SVR, ELM shows a better prediction ability on this prediction problem.
- Each of the electricity-related factors can help improve the prediction accuracy but when they are combined, it will lead to better performance.
- Under the influence of electricity-related factors, a subset of environmental factors can further and mostly improve the prediction performance.
- DDMP-PSO presents a strong local search ability. By using DDMP-PSO, an optimal subset of electricity-related factors and environmental factors, their respective window sizes and the number of hidden neurons in ELM which can lead to the highest prediction accuracy for this problem is found.

V. Conclusion and Future Work

ELM was compared to SVR for energy consumption prediction and showed superiority over SVR. Moreover, electricity-related and environmental factors was experimentally demonstrated to improve prediction accuracy obtained by purely using consumption data. Furthermore, DDMS-PSO was proposed to seek the optimal subset of electricity-related factors and environmental factors, their respective window sizes and the number of hidden neurons in ELM, aiming at best prediction accuracy. The results show that DDMS-PSO successfully found a subset leading to the overall best predication accuracy. Our future work includes the incorporation of suitable feature extraction techniques [31], the use of clustering [32] to pre-partition training samples and establish a prediction model per cluster, and the implementation of the proposed method on GPUs [33] to accelerate computation speed.

ACKNOWLEDGMENT

This work is supported by Buildings Engineered for Sustainability research project, funded by RMIT Sustainable Urban Precincts Program.

REFERENCES

[1] A. Foucquier, S. Robert, F. Suard, L. Stéphan, and A. Jay, "State of the art in building modelling and energy performances prediction: A review," *Renewable and Sustainable Energy Reviews*, vol. 23, pp. 272–288, 2013.

[2] H. X. Zhao and F. Magoulès, "A review on the prediction of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, pp. 3586–3592, 2012.

[3] M. Yalcintas and S. Akkurt, "Artificial neural networks applications in building energy predictions and a case study for tropical climates," *International journal of energy research*, vol. 29, pp. 891–901, 2005.

[4] X. L. Yuan, X. Wang, and J. Zuo, "Renewable energy in buildings in chinała review," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 1–8, 2013.

[5] A. Roetzel and A. Tsangrassoulis, "Impact of climate change on comfort and energy performance in offices," *Building and environment*, vol. 57, pp. 349–361, 2012.

[6] S. Kalogirou, C. Neocleous, and C. Schizas, "Building heating load estimation using artificial neural networks," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, vol. 8, 1997, pp. 1–8.

[7] T. Olofsson, S. Andersson, and R. Östin, "A method for predicting the annual building heating demand based on limited performance data," *Energy and Buildings*, vol. 28, pp. 101–108, 1998.

[8] D. C. Park, M. El-Sharkawi, R. Marks, L. Atlas, M. Damborg *et al.*, "Electric load forecasting using an artificial neural network," *IEEE Transactions on Power Systems*, vol. 6, pp. 442–449, 1991.

[9] L. C. Ying and M. C. Pan, "Using adaptive network based fuzzy inference system to forecast regional electricity loads," *Energy Conversion and Management*, vol. 49, pp. 205–211, 2008.

[10] K. Kavaklioglu, "Modeling and prediction of turkeys electricity consumption using support vector regression," *Applied Energy*, vol. 88, pp. 368–375, 2011.

[11] B. Dong, C. Cao, and S. E. Lee, "Applying support vector machines to predict building energy consumption in tropical region," *Energy and Buildings*, vol. 37, pp. 545–553, 2005.

[12] G. K. Tso and K. K. Yau, "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks," *Energy*, vol. 32, pp. 1761–1768, 2007.

[13] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006.

[14] C. R. Rao and S. K. Mitra, *Generalized inverse of matrices and its applications*. Wiley New York, 1971, vol. 7.

[15] R. Zhang, Y. Lan, G. b. Huang, and Z.-B. Xu, "Universal approximation of extreme learning machine with adaptive growth of hidden nodes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 365–371, 2012.

[16] G. B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, pp. 155–163, 2010.

[17] Z. L. Sun, T. M. Choi, K. F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decision Support Systems*, vol. 46, pp. 411–419, 2008.

[18] X. Chen, Z. Y. Dong, K. Meng, Y. Xu, K. P. Wong, and H. Ngan, "Electricity price forecasting with extreme learning machine and bootstrapping," *IEEE Transactions on Power Systems*, vol. 27, pp. 2055–2062, 2012.

[19] K. Mohammadi, S. Shamshirband, L. Yee, D. Petković, M. Zamani, and S. Ch, "Predicting the wind power density based upon extreme learning machine," *Energy*, vol. 86, pp. 232–239, 2015.

[20] K. Mohammadi, S. Shamshirband, S. Motamedi, D. Petković, R. Hashim, and M. Gocic, "Extreme learning machine based prediction of daily dew point temperature," *Computers and Electronics in Agriculture*, vol. 117, pp. 214–225, 2015.

[21] J. J. Liang, S. Baskar, P. N. Suganthan, and A. K. Qin, "Performance evaluation of multiagent genetic algorithm," *Natural Computing*, vol. 5, pp. 83–96, 2006.

[22] C. k. Zhang and H. Hu, "Using pso algorithm to evolve an optimum input subset for a svm in time series forecasting," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, 2005, pp. 3793–3796.

[23] M. Shafie-Khah, M. P. Moghaddam, and M. Sheikh-El-Eslami, "Price forecasting of day-ahead electricity markets using a hybrid forecast method," *Energy Conversion and Management*, vol. 52, pp. 2165–2169, 2011.

[24] A. Azadeh, S. Ghaderi, S. Tarverdian, and M. Saberi, "Integration of artificial neural networks and genetic algorithm to predict electrical energy consumption," *Applied Mathematics and Computation*, vol. 186, pp. 1731–1741, 2007.

[25] R. Banos, F. Manzano Agugliaro, F. Montoya, C. Gil, A. Alcayde, and J. Gómez, "Optimization methods applied to renewable and sustainable energy: A review," *Renewable and Sustainable Energy Reviews*, vol. 15, pp. 1753–1766, 2011.

[26] J. Kenndy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[27] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the 2002 Congress on Evolutionary Computation,*, vol. 2, 2002, pp. 1671–1676.

[28] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proceedings of IEEE Swarm Intelligence Symposium*, 2005, pp. 124–129.

[29] J. J. Liang, Q. K. Pan, and T. J. Chen, "A dynamic multi-swarm particle swarm optimizer for blocking flow shop scheduling," in *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 2010, pp. 323–327.

[30] C. C. Chang and C. J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, pp. 1–27, 2011.

[31] A. K. Qin, P. N. Suganthan, and M. Loog, "Generalized null space uncorrelated fisher discriminant analysis for linear dimensionality reduction," *Pattern Recognition*, vol. 39, pp. 1805–1808, 2006.

[32] A. K. Qin and P. N. Suganthan, "Enhanced neural gas network for prototype-based clustering," *Pattern Recognition*, vol. 38, pp. 1275–1288, 2005.

[33] A. K. Qin, F. Raimondo, F. Forbes, and Y. S. Ong, "An improved CUDA-based implementation of differential evolution on GPU," in *2012 Genetic and Evolutionary Computation Conference (GECCO'12)*, 2012, pp. 991–998.