Evolutionary Multi-objective Ensemble Learning for Multivariate Electricity Consumption Prediction

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

A. K. Qin
Computer Science and Software Engineering, School of Software and Electrical Engineering, Swinburne University of Technology,
Hawthorn, VIC 3122, Australia
Email: kqin@swin.edu.au

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

Abstract*—Energy consumption prediction typically corresponds to a multivariate time series prediction task where different channels in the multivariate time series represent energy consumption data and various auxiliary data related to energy consumption such as environmental factors. It is non-trivial to resolve this task, which requires finding the most appropriate prediction model and the most useful features (extracted from the raw data) to be used by the model. This work proposes an evolutionary multi-objective ensemble learning (EMOEL) technique which uses extreme learning machines (ELMs) as base predictors due to its highly recognized efficacy. EMOEL employs evolutionary multi-objective optimization to search for the optimal parameters of the model as well as the optimal features fed into the model subjected to two conflicting criteria, i.e., accuracy and diversity. It leads to a Pareto front composed of non-dominated optimal solutions where each solution depicts the number of hidden neurons in the ELM, the selected channels in the multivariate time series, the selected feature extraction methods and the selected time windows applied to the selected channels. The optimal solutions in the Pareto front stand for different end-to-end prediction models which may lead to different prediction results. To boost ultimate prediction accuracy, the models with respect to these optimal solutions are linearly combined with combination coefficients being optimized via an evolutionary algorithm. We evaluate the proposed method in comparison to some existing prediction techniques on an Australian University based dataset, which demonstrates the superiority of the proposed method.

I. Introduction

In smart buildings, electricity consumption prediction plays a key role in the energy supply plan making process with the aim to save energy. However, it is a non-trivial task because electricity consumption can be influenced by a range of factors such as environmental and socioeconomic factors [1], [2] which should be properly considered by the prediction model, and meanwhile the prediction model per se may involve various configurations and parameters to be determined.

In electricity consumption prediction, the auxiliary data (e.g., environmental and socioeconomic data) may provide important information in addition to the historical consumption data. For example, some previous work [1] had leveraged on the environmental factors including temperature, humidity and air pressure to help much improve the prediction performance. Moreover, the ultimate prediction performance may heavily depend on the powerfulness of the features extracted from the raw data. Accordingly, choosing appropriate feature extraction methods to apply and determining the optimal parameter settings (e.g., the time window size) of the selected methods play a significant role [1], [3], [4]. Furthermore, model configurations may significantly influence the prediction performance. For example, different applications usually prefer different parameter settings, and accordingly how to obtain an optimal model configuration for a specific application has been widely studied [5], [6].

To build a good electricity consumption prediction model, we need to decide which channels in multivariate time series (representing both the consumption data and the auxiliary data) should be used, which feature extraction techniques should be applied to the selected channels, the parameter settings of the selected feature extraction techniques (e.g., the time window size) and the parameter settings of the prediction model. Existing works have addressed some of these aspects. For example, feature selection techniques have been studied for channel selection [7]. Feature extraction methods such as the discrete wavelet transform (DWT) [4], the principal component analysis (PCA) [8] and some statistical methods [3] have been applied to energy consumption prediction. In [9], the parameters of the prediction model, i.e., support vector machine, are optimized via an evolutionary algorithm (EA) to seek the best prediction performance. However, none of them has addressed all the aspects as one optimization problem involving both discrete and continuous decision variables such as method indexes and parameter settings.

EAs, as a family of nature-inspired optimization techniques [10]–[12], is highly competent to solve complex optimization problems involving the mixed types of decision variables, e.g., continuous and discrete. Therefore, they provide a potential solution to solve such a problem. In fact, to obtain an optimal prediction model by using the EA, we may need to consider more than just training accuracy as a single objective to be optimized. Model complexity, for example, is often regarded as a key factor related to the generalization performance. Accordingly, we may resort to multi-objective EAs (MOEAs),

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

leading to multiple non-dominated optimal solutions (models). Instead of selecting the best one among these models, ensemble learning [12], [13] provides an effective way to establish a robust and accurate model by combing all of these models.

In this work, we propose an evolutionary multi-objective ensemble learning (EMOEL) technique with extreme learning machines (ELMs) being used as base predictors due to its highly-recognized efficacy. The EMOEL employs a powerful multi-objective EA named MOEA/D [14] to search for both the optimal model parameters and the optimal features to be fed into the model subject to prediction accuracy and model diversity as two conflicting objectives. The Pareto front (PF) eventually obtained by MOEA/D is composed of non-dominated optimal solutions. Each of them encodes the selected auxiliary factors (channels), the selected feature extraction methods, the sizes of the time windows, and the number of hidden neurons in the ELM, and thus corresponds to an end-to-end predictor. All the solutions in the PF are linearly combined to build an ensemble, where combination coefficients are optimized via a single-objective DE algorithm. The proposed EMOEL technique can relieve labour-intensive model tunning and produce more accurate and robust results via the ensemble. Experimental results on a real-world electricity consumption data set from RMIT University (Australia) demonstrate the superiority of EMOEL over several other methods in comparison.

Next section describes the background of this work, followed by the details of the proposed method in Section III. Experimental results are reported and discussed in Section IV. Then the introduction to the related work will be introduced in Section V. Section VI concludes the paper with some future work being mentioned.

II. Background

A. Electricity consumption prediction

Electricity consumption as the mainly energy consumed in smart buildings, plays a significant role in energy saving and efficiency. Electricity consumption prediction contributes to effective building energy management and conservation, building energy control and operation, and help manager to better schedule and plan the operations of the supply system, therefore accurate prediction of electricity consumption in smart buildings is vital important. However, as electricity consumption is inevitably affected by various time series data, features extracted from the auxiliary data and the respective time window sizes selection, and prediction model configuration. Therefore, constructing efficient and effective prediction model by considering all aspects is still a challenge for accurate electricity consumption prediction with various auxiliary time series data.

B. Feature extraction

Features extracted in time series can be categorized into time series features, frequency related features and statistical features. Time series feature extraction approaches include Piecewise aggregated approximation (PAA) [15], Piecewise linear approximation (PLA) [16] etc., extracting local features by segmenting time windows into different parts. Statistical feature, as a local and global feature representation, is performed by summarizing the statistics trend in different time periods, which has been successfully applied in daily energy consumption profiles [3]. DWT and Discrete Fourier transforms (DFT) as frequency related features extraction methods are two popular time series representation in the transformation domain [17]. DWT as a very popular multi-resolution method has been used in [4], [18] because DWT does not only have the ability to extract the rising trend and periodic waves, but it can also distinguish stochastic behavior [4]. PLA, statistical feature extraction and DWT with different coefficients or different number of segments will be used to extract useful features from historical electricity data and various auxiliary time series.

C. Multi-objective optimization

For many real-world applications, there are always more than one conflicting objectives to be optimized simultaneously. Problems like this are called multi-objective optimization problems (MOPs), defined as follows:

$$\begin{cases} \text{Minimize:} & \mathbf{F}(\mathrm{x}) = (f_1(\mathrm{x}), f_2(\mathrm{x}), ..., f_m(\mathrm{x}))^T \\ \text{Subject to:} & x \in \Omega \end{cases} \quad (1)$$

where $\Omega \subset \mathbb{R}^n$ represents the decision space and $\mathrm{x} = (x_1, x_2, ..., x_n) \in \Omega$ is a decision variable with respect to a solution for a specific MOP. $\mathbf{F}(\mathrm{x}) : \Omega \rightarrow \mathbb{R}^m$ denotes the m-dimensional objective vector of the solution x.

For a specific MOP, assuming x_A and x_B are its two solutions, if and only if $f_i(\mathrm{x}_A) \leq f_i(\mathrm{x}_B), \forall i \in \{1, ..., m\}$ and there is a $j \in \{1, ..., m\}$ satisfied $f_j(\mathrm{x}_A) < f_j(\mathrm{x}_B)$, x_A dominates x_B (i.e., $\mathrm{x}_A \prec \mathrm{x}_B$). A solution $\mathrm{x}^* \in \Omega$ is called Pareto optimum if there is no other solution that dominates x^*. The set of all Pareto optimal solutions is called Pareto optimal set (PS) and Pareto optimal front is defined as the corresponding objective vectors of the solutions in the PF. Therefore, to solve a MOP is to find its PS.

D. Differential evolution

Differential evolution (DE) was proposed by Storn and Price in 1995 [11]. It has four main operations: initialization, mutation, recombination and selection [11], [19], [20]. It generates offspring by using differences of randomly sampled pairs of individual vectors from the population. Its offspring will compete with their parents inside the population and the winner will be kept as the parents of the next generation. The population is randomly generated with Eq. 2.

$$\mathbf{p}_i^j = U_{min} + rand(U_{max} - U_{min}) \quad (2)$$

where ps is the population size and D is the dimension, $i = 1, ..., ps, j = 1, ..., D$; U_{min} and U_{max} are lower and upper bounds. $rand$ is a random number with uniform distribution.

For each individual $\mathbf{p}_i$, the standard DE can be mutated as:

$$\mathbf{v}_i = \mathbf{p}_{r_1} + F(\mathbf{p}_{r_2} - \mathbf{p}_{r_3}) \quad (3)$$

where r_1, r_2 and r_3 are randomly generated series number and they are different from i; F is mutation factor.

After that, the crossover operation will be performed to generate new individuals with Eq. 4:

$$\mathbf{u}_i^j = \begin{cases} \mathbf{v}_i^j & \text{if} \quad rand^j \leq CR \quad \text{or} \quad j = i_{rand} \\ \mathbf{p}_i^j & \text{otherwise} \end{cases} \tag{4}$$

where $rand^j$ is a randomly generated number in the uniform distribution $[0, 1]$; i_{rand} is randomly generated index of the element in $[1, D]$; CR is crossover rate.

Finally, a greedy search is adopted by DE to select the better individuals, as shown in Eq. 5. where f is the fitness function, such as prediction error function.

$$\mathbf{p}_i = \begin{cases} \mathbf{u}_i & \text{if} \quad f(\mathbf{u}_i) < f(\mathbf{p}_i) \\ \mathbf{p}_i & \text{otherwise} \end{cases} \tag{5}$$

E. Extreme learning machine

ELM [6] is based on Single hidden Layer Feedforward (SLNF) and has input layer, hidden layer and output layer. The hidden bias and the weight for connecting the 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 formulated as:

$$\sum_{i=1}^{N} \beta_i \mathbf{f}(\mathbf{w}_i^T \mathbf{x}_j + b_i), 1 \leq j \leq M \tag{6}$$

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:

$$\sum_{i=1}^{N} \beta_i \mathbf{f}(\mathbf{w}_i^T \mathbf{x}_j + b_i) = \mathbf{y}_j, 1 \leq j \leq M \tag{7}$$

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

$$\mathbf{H} = \begin{pmatrix} f(\mathbf{w}_1^T \mathbf{x}_1 + b_1) & \cdots & f(\mathbf{w}_N^T \mathbf{x}_1 + b_N) \\ \cdots & \cdots & \cdots \\ f(\mathbf{w}_1^T \mathbf{x}_M + b_1) & \cdots & f(\mathbf{w}_N^T \mathbf{x}_M + b_N) \end{pmatrix} \tag{8}$$

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

The output weight $\mathbf{B}$ is calculated by $\mathbf{B} = \mathbf{H}^{+}\mathbf{Y}$, and $\mathbf{H}^{+}$ is a Moore-Penrose generalized inverse of $\mathbf{H}$ [21]. The only task for ELM applications is to select a suitable activation function and set the number of hidden neurons, making it easier applied to electricity prediction issues [7] .

III. The Proposed Method

A. Motivations

For this prediction task involving n different time series, m different feature extraction approaches (assuming different time series probably prefer different feature extraction methods), t different time window sizes and h different parameter settings for prediction model, there is not a general model considering them work simultaneously. Also, the existing work corresponding to multivariate time series does not consider the influence of different feature extraction methods assigned to different time series along with different time windows. Moreover, the combinatorial possibilities will be $mth(1 + mt)^n$, making it difficult to find the optimal combination for the best prediction performance with greed search. Motivated by all the mentioned challenges, EA is applied to perform channels selection of multivariate time series, feature extraction methods selection and respective time windows selection, and the parameter in prediction model selection simultaneously.

Furthermore, a single prediction model can hardly promise good performance across different applications in time series domain. Ensemble learning is considered given its good generalization and strong robustness. As the most important aspect in ensemble learning is to keep the members diversity, multi-objective optimization subject to prediction error and diversity among individuals as two conflicting objectives will be applied to obtain the members for ensemble learning. The last challenge is the combination rule for ensemble learning. Combination coefficients among the members will be taken into account as it quantifies the contribution of each member. Therefore, EMOEL is proposed for solving electricity consumption prediction with multiple auxiliary time series data by considering channels selection, feature extraction methods selection and time windows selection for the selected channels, and prediction model configuration and obtaining the members for ensemble learning subject to accuracy and diversity.

B. Framework

1) Encoding and decoding in EA: The selection for auxiliary time series has two states, selected (0) and not selected (1). For each selected auxiliary time series, the feature extraction methods (denoted with indexes 1, 2,..., m, m is the number of feature extraction methods) and time windows (denoted with indexes 1, 2,..., t, t is the number of time window settings) to be selected for obtaining the optimal features are all integers. For example, in order to predict next hour electricity usage with time window $T = 24$, the statistical features can be extracted by segmenting the sequence into 4 parts and then mean value, maximum value, minimum value and standard deviation will be summarized from each part, i.e., 0:00 am to 6:00 am, 7:00 am to 12:00 pm, 13:00 pm to 18:00 pm and 19:00 pm to 24:00 am. The sequence can be split into different segments, representing different feature extraction methods. Also, the parameters in prediction model are discrete values. Here, we integrate the search space of each auxiliary time series selection with respective feature extraction methods selection, i.e., 0 (not selected), 1, 2,..., m. When using an EA to address this optimization issue, the solution space is encoded as continuous values between 0 and 1. Every time before evaluating the candidate solution, it will be decoded according to the real solution space. In this way, the prediction result obtained by each combination of the selected channels, the selected feature extraction methods and time windows applied to the selected channels and parameters in prediction model is regarded as a member of ensemble learning.

2) Objectives in MOEAs: Given a training set generated from the optimal features with s different samples, the accuracy and diversity will be optimized by MOEAs with the following objective functions.

Accuracy: To maximize the average accuracy of an ensemble member on the training set means to minimize the average prediction error, defined as:

$$\text{Minimize:} \qquad \text{Err}_k = \frac{1}{s}\sum_{i=1}^{s}(p^i - \hat{p}_k^i)^2 \tag{9}$$

p^i is the real value of i^{th} training sample and $\hat{p}_k^i$ represents the estimated prediction result obtained by k^{th} predictor for the i^{th} training sample.

Diversity: The second objective is to maximize the diversity (minimize the correlation) between the outputs of different ensemble members. Negative correlation learning (NCL) [22] will be used to define the diversity as follows:

$$\text{Minimize:} \qquad \text{Div}_k = \sum_{i=1}^{s}(\hat{p}_k^i - \hat{p}^i)\sum_{j\neq k, j=1}^{M}(\hat{p}_j^i - \hat{p}^i) \tag{10}$$

where $\hat{p}_k^i$ and $\hat{p}_j^i$ represent the outputs of the k^{th} and j^{th} base predictor for the i^{th} training sample, respectively. $\hat{p}^i$ is the average output of all base predictors on i^{th} training sample.

3) Combination rule for ensemble learning: Assuming an ensemble with M different members (M different solutions from MOEAs), the ensemble output is defined as $P = \sum_{k=1}^{M} w_k \hat{p}_k$. The combination coefficients satisfied $\sum_{k=1}^{M} w_k = 1$ and w_k means how much the k^{th} base predictor $\hat{p}_k$ contributes to ensemble output P.

C. Implementation

1) Multi-objective optimization algorithm: MOEAs based on decomposition (MOEA/D) [14] has achieved a great success in the filed of MOPs and has attracted a lot of attention. MOEA/D explicitly decomposes a MOP into M scalar optimization subproblems. These M subproblems will be addressed simultaneously by involving a population of solutions. At each generation, the population is composed of the best solution found so far for each subproblem. Each subproblem has a uniformly distributed vector as its weight. By calculating Euclidean distance between two weights, each subproblem will have T closest neighborhoods. Therefore, the optimal solutions to two neighborhoods will be similar. Each subproblem will be solved by only using its neighborhoods' information, which makes it much more efficient in solving MOP compared with other MOEAs. In this task, all the objectives are to be minimized, and the details of MOEA/D integrated with ELM are illustrated in Algorithm. 1.

As the objective fitnesses have different magnitudes, where some magnitudes are much larger than others. In order to overcome the problems caused by undesirably bias during the search direction, we apply adaptive normalization for each of

Algorithm 1 MOEA/D integrated with ELM

Input:
A MOP with its solution space Ω and F objectives
M (also ps_1): population size (i.e. subproblems in MOEA/D)
T: neighborhood size
$maxFEs_1$: maximal number of evaluations in MOEA/D

Output:
Selected channels, selected feature extraction methods and time windows for the selected channels, the number of hidden neurons in ELMs with the eventually evolved population $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$

Step 1) Initialization:
Step 1.1) Randomly generate an initial population $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$ with uniform distribution in search space Ω, then for each individual $\mathbf{x}_i, i = 1, ..., M$, generate the training dataset i by decoding each dimension into its real space (i.e., indexes for feature configuration and the number of hidden neurons in ELM), $\#FEs = 0$
Step 1.2) Evaluate each candidate solution $\mathbf{x}_i, i = 1, ..., M$ via the i^{th} generated training dataset and i^{th} ELM to obtain its objective fitness $\mathbf{Z}(\mathrm{x}_i)$, $\#FEs = \#FEs + ps_1$
Step 1.3) Randomly generate M uniformly distributed weights $\{\lambda_1, ..., \lambda_M\}$, where $\lambda_i = \lambda^1, ..., \lambda^F$ with respect to i^{th} individual, its T closest weight vectors $\{\lambda_{i_1}, ..., \lambda_{i_T}\}$ are obtained to form its neighborhoods $B(i) = \{i_1, ..., i_T\}$
Step 1.4) Initial reference point $\mathbf{Z}^* = \{z^{1^*}, ..., z^{F^*}\}$, satisfied $z^{f^*} = \min_{i=\{1,...,M\}} z^f(\mathbf{x}_i), f = 1, ..., F$
Step 1.5) Initialize normalization factors as $\tilde{\mathbf{Z}} = \{\tilde{z}^1, ..., \tilde{z}^f\}$, satisfied $\tilde{z}^f = \max_{i=\{1,...,M\}} |z^f(\mathbf{x}_i)|, f = 1, ..., F$

Step 2) Evolution:
while $FEs_1 < maxFEs_1$ **do**
Step 2.1) Adaptive normalization: In the current population, for each individual, apply adaptive normalization to its fitness on each objective, i.e., $z^f(\mathbf{x}_i) = \frac{z^f(\mathbf{x}_i)}{\tilde{z}^f}, i = 1, ..., M, f = 1, ..., F$
for $i = 1, ..., M$ **do**
Step 2.2) Reproduction: Randomly generate two indexes k, l from $B(i)$, and then generate a new solution i' from $\mathbf{x}_k$ and $\mathbf{x}_l$ by using the DE operator along with a Gaussian mutation applied under probability of 0.5:

$$\mathbf{x}_{i'} = \begin{cases} \mathbf{x}_i + 0.5\cdot(\mathbf{x}_k - \mathbf{x}_l) + rnd(0,\sigma) & \text{if } rnd(0,\sigma) \leq 0.5 \\ \mathbf{x}_i + 0.5\cdot(\mathbf{x}_k - \mathbf{x}_l) & \text{otherwise} \end{cases}$$

Each element in σ is set to one twentieth of the corresponding decision variable's range
Step 2.3) Repairing: Apply a problem-specific repair on the newly generated $\mathbf{x}_{i'}$ to limit each of its elements in lower (upper) bound
Step 2.4) Evaluation: Decode the newly generated $\mathbf{x}_{i'}$ into its real space (i.e., indexes for feature configuration and the number of hidden neurons in ELM), generate i'^{th} training dataset according to the new feature parameters, and evaluate $\mathbf{x}_{i'}$ via i'^{th} ELM using corresponding training dataset and obtain $\mathbf{Z(x)}_{i'}$, $\#FEs = \#FEs + 1$
Step 2.5) Adaptive normalization: Normalize $\mathbf{x}_{i'}$, i.e., $z^f(\mathbf{x}_{i'}) = \frac{z^f(\mathbf{x}_{i'})}{\tilde{z}^f}, i = 1, ..., M, f = 1, ..., F$
Step 2.6) Replacement: For each $i_s \in B(i)$, if $\max_{f\in\{1,...,F\}} \lambda_{i_s}^f \cdot |z^f(\mathbf{x}_{i'}) - z^{f^*}| \leq \max_{f\in\{1,...,F\}} \lambda_{i_s}^f \cdot |z^f(\mathbf{x}_{i_s}) - z^{f^*}|$, set $\mathbf{x}_{i_s} = \mathbf{x}_{i'}$ and $z^f(\mathbf{x}_{i_s}) = z^f(\mathbf{x}_{i'})$
Step 2.7) Update reference point: If $z^f(\mathbf{x}_{i'}) < z^{f^*}$, set $z^{f^*} = z^f(\mathbf{x}_{i'})$
end for
Step 2.8) Update normalization factors: $\tilde{\mathbf{Z}} = \{\tilde{z}^1, ..., \tilde{z}^f\}$, satisfied $\tilde{z}^f = \max_{i=\{1,...,M\}} |z^f(\mathbf{x}_i)|, f = 1, ..., F$
end while

Algorithm 2 Combination coefficients optimization with DE

Input:
A single-objective optimization with its solution space Ω
ps_2: population size in DE
F: mutation factor (0.5)
CR: crossover rate (0.5)
$maxFEs_2$: maximal number of evaluations in DE
D_2: dimension (same with the population size ps_1 in MOEA/D)
Output:
Optimized coefficients $\mathbf{p}_{i^*}$, $i^* = \arg_i \max f(\mathbf{p}'_i), i = 1, ..., ps_2$
Step 1) Initialization: Randomly generate an initial population with ps_2 individuals $\{\mathbf{p}_1, ..., \mathbf{p}_{ps_2}\}$ using Eq. 2, $\#FEs_2 = 0$
Step 2) Evaluation: Normalize each individual $\mathbf{p}_i, i = 1, ..., ps_2$ as $\mathbf{p}'_i = \frac{\mathbf{p}_i}{\sum_{j=1}^{D_2} \mathbf{p}_i^j}$ and using $\mathbf{p}'_i$ to valuate each individual $\mathbf{p}_i$ to obtain its objective fitness $f(\mathbf{p}'_i)$, $\#FEs_2 = \#FEs_2 + ps_2$
Step 3) Evolution:
while $\#FEs_2 < maxFEs_2$ **do**
  for $i = 1, ..., ps_2$ **do**
    Step 3.1) Mutation: Randomly generate series number r_1, r_2, r_3 different from i and perform mutation with Eq. 3
    Step 3.2) Crossover: Crossover operation using Eq. 4 to generate new individual $\mathbf{u}_i$
    Step 3.3) Evaluation: Normalize $\mathbf{u}_i$ as $\mathbf{u}'_i = \frac{\mathbf{u}_i}{\sum_{j=1}^{D_2} \mathbf{u}_i^j}$ and evaluate $\mathbf{u}_i$ with $f(\mathbf{u}'_i)$, $\#FEs_2 = \#FEs_2 + 1$
    Step 3.4) Selection: By comparing $f(\mathbf{u}'_i)$ and $f(\mathbf{p}'_i)$ with Eq. 5 to decide which one of $\mathbf{u}_i$ and $\mathbf{p}_i$ to be kept as the parent in next generation
  end for
end while

the objective fitness $z^f, f = 1, ..., F$ (F is the number of objectives and F=2 in this task) as follows:

$$z^f(\mathbf{x}) = \frac{z^f(\mathbf{x})}{\tilde{z}^f} \tag{11}$$

where $\tilde{z}^f$ is the normalization factor updated at beginning of every generation, obtained by $\tilde{z}^f = \max_{i=\{1,...,M\}} |z^f(\mathbf{x}_i)|$.

2) Combination coefficients optimization: Each obtained solution from MOEA/D with ELM corresponding to the combination of selected channels, the selected feature extraction methods and respective time windows applied to the selected channels, and the number of hidden neurons in ELM in the final population of MOEA/D, will be combined by combination coefficients to form the ensemble learning. The coefficients (summed to 1) will be optimized by a single-objective DE using Mean Square Error (MSE) from ensemble learning as the objective function. The combination coefficients optimization with DE is presented in Algorithm. 2.

IV. Experiments

A. Data description

The data includes historical electricity data, two electricity-related factors (apparent power and power factor) and environmental factors (temperature, dew point, humanity, wind speed, sea level). The previous two parts are collected from the smart meters of the buildings in the city campus of RMIT University, Melbourne, with 15-minutes to 1-day. The environmental factors are crawled from an online weather station that broadcasts periodic readings from every 20-minutes to 1-hour.

The dataset is from 21.03.2014 to 18.12.2015 and 18.01.2016 to 19.04.2016, as the building is shut down during 19.12.2015 and 17.01.2016, with 12 points in one day and normalized to $[0.15, 1]$ except for the prediction targets in the training and testing sets. The dataset is sampled according different time windows (8 to 28 with an interval of 2) for one-step-ahead prediction. Here, we mainly focus on prediction on 'Holidays' according to RMIT calendar. There are 3180 samples in total, which will be randomly created into training (occupy 2/3) and testing (occupy 1/3) datasets 3 times.

B. Experimental setup

In the proposed EMOEL model, as large neighborhood will cause the same solutions for all neighborhoods for this prediction task, we set neighborhood size $T = 5$. Population sizes are $ps_1 = 50, 80, 100, 120$ in MOEA/D and $ps_2 = 50, 100, 120$ in DE. Dimension is $D_2 = ps_1$ in DE. With feature extraction, each dimension in MOEA/D is described as: d_1~$d_7 \in [0, 10]$ are channel selection and respective feature extraction methods selection; $d_8 \in [1, 10]$ is feature extraction method selection for electricity data; d_9~$d_{16} \in [1, 11]$ represent time windows selection applied to the selected channels and electricity data; $d_{D_1} \in [20, 400]$ with intervals 20 is the number of hidden neurons in ELM and $D_1 = 17$. Without feature extraction, each dimension in MOEA/D is described as: d_1~$d_7 \in \{0, 1\}$ are channel selection; d_8~$d_{14} \in [1, 11]$ represent time windows selection applied to the selected channels; $d_{15} \in [1, 11]$ is time window selection for electricity data; $d_{D_1} \in [20, 400]$ with intervals 20 is the number of hidden neurons in ELM and $D_1 = 16$. $maxFEs_1 = 120000$ and $maxFEs_2 = 200000$ in MOEA/D and DE, respectively.

We firstly investigate the performance of EMOEL between with feature extraction and without feature extraction. Then EMOEL will be compared with several state-of-the-art models. A multilayer perceptron (MLP) with three layers will be compared, where the number of hidden neurons are set from 10 to 100 with 10 intervals. ELM with different number of hidden neurons (from 20 to 400 with interval 10) will be compared. ϵ-SVR model is implemented using the LibSVM library [5] and using the same parameters setting with [7]. H-ELM, proposed in [23], has unsupervised and supervised stages, where unsupervised stage focuses on feature extraction and supervised stage aims for classification or regression. A slightly more dramatic variation on the Long Short Term Memory (LSTM) is the Gated Recurrent Units (GRUs) [24]. The parameters in H-ELM and GRUs are tuned according to [13]. Moreover, an ensemble model which uses 8 different DWTs and ELMs (the number of hidden neurons change from 20 to 400 with interval 10) to generate multiple members, then Partial Least Square regression (PLSR) will be used for ensemble prediction [12]. Here, all comparison models will be tested on electricity data with and without auxiliary time series and the best performance for each model will be selected except for GRUs and PLSR. The data for PLSR is processed in the same way with the original paper [12]. All results will be evaluated on average values based on 10 runs and Wilcoxon signed-rank test is performed to show the significant difference.

C. Results

The proposed EMOEL with (W) and without (WO) feature extraction will be compared under different population sizes in MOEA/D and DE, where MOEA/D focuses on obtaining the PF for balancing objectives corresponding to prediction error and negative correlation and a single-objective DE is applied to optimize combination coefficients for combining the members in PF. Then the procedure from performing Pareto front to coefficients optimization with DE will be presented. Finally, the results from the proposed EMOEL_{W} and EMOEL_{WO} will be compared with several state-of-the-art models.

TABLE I: Performance comparison of EMOEL_{W} and EMOEL_{WO} under different population sizes in MOEA/D and DE, represented with ps_1 and ps_2, respectively

Population size		MSE (Mean)	
		EMOEL_{W}	EMOEL_{WO}
ps_1=50	PF_{Best}	**1517.4679**	1931.9540
	ps_2=50	**1094.9611**	1282.8072
	ps_2=100	**1094.9611**	1282.8075
	ps_3=150	**1094.9618**	1282.8046
ps_1=80	PF_{Best}	**1224.0288**	1892.2791
	ps_2=50	**849.2124**	1248.3400
	ps_2=100	**849.2117**	1248.3400
	ps_3=150	**849.1552**	1248.3356
ps_1=100	PF_{Best}	**1078.7979**	1882.9249
	ps_2=50	**694.0135**	1237.4441
	ps_2=100	**694.0012**	1237.4341
	ps_3=150	**694.3698**	1237.5032
ps_1=120	PF_{Best}	**1235.1595**	1845.0009
	ps_2=50	**816.6883**	1194.4143
	ps_2=100	**816.6599**	1194.3667
	ps_2=150	**816.9114**	1194.4779

1) Comparison between EMOEL_{W} and EMOEL_{WO}: The results will be reported with the best single prediction result from Pareto front (denoted as PF_{Best}) and the ensemble performance with different population sizes in MOEA/D and DE from the view of with and without feature extraction. EMOEL_{W} and EMOEL_{WO} represent prediction results with and without feature extraction, respectively. ps_1 is the population size in MOEA/D and ps_2 is the population size in DE for optimizing the weights for ensemble learning. The result comparison will be performed with Wilcoxon signed-rank test for EMOEL_{W} and EMOEL_{WO} and labeled bold to show the significant better.

Tab. I presents the performance of the proposed EMOEL with and without feature extraction under different population sizes in MOEA/D and DE. Comparing EMOEL_{W} to EMOEL_{WO} under PF_{Best} (which has large negative correlation in PF) with different ps_1, the prediction accuracy with feature extraction is better than without feature extraction whenever ps_1 =50, 80, 100, 120. Moreover, when comparing the ensemble prediction results between EMOEL_{W} and EMOEL_{WO} under the same ps_1, the prediction accuracy will improve significantly after using feature extraction. Also, ps_1=100 will lead to the best prediction performance for single prediction in PF with PF_{Best}=1078.7979. Therefore, multiple feature extraction approaches allocation for different selected auxiliary time series will further improve prediction accuracy by comparing the results of EMOEL_{W} and EMOEL_{WO} under each combination of ps_1 and ps_2 setting. Moreover, the result obtained demonstrates that the proposed EMOEL will improve the prediction accuracy significantly compared to single predictor.

TABLE II: Performance comparison of the proposed EMOEL under different population sizes in MOEA/D and DE (ps_1 and ps_2, respectively) with feature extraction

ps_1 \ ps_2	MSE (Mean)		
	50	100	150
50	1094.9611	1094.9611	1094.9618
80	849.2124	849.2117	849.1552
100	694.0135	**694.0012**	694.3698
120	816.6883	816.6599	816.9114

Tab. II further shows the comparison of the ensemble performance under different ps_1 and ps_2 using the statistical test. When ps_1=100, the ensemble learning performance is better for all ps_2 settings. Even though the performance on ps_2=50, 100, 150 are very similar, the result from ps_2=100 is statistical better than ps_2=50 and 150. Therefore, the population sizes in MOEA/D and DE are sensitive and will actually influence the performance of ensemble learning prediction accuracy. When ps_1=100 and ps_2=100, the ensemble learning performance reaches the best.

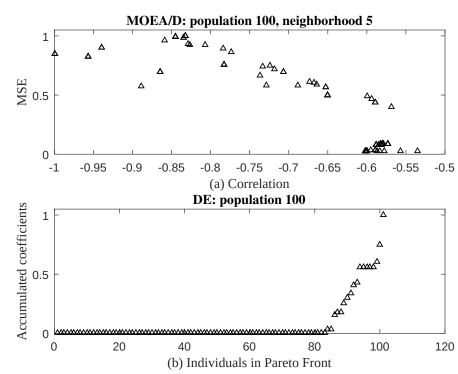

Fig. 1: PF with correlation and MSE and accumulated combination coefficients for each individual in the PF

2) Ensemble learning inside implementation: Fig. (a) in Fig. 1 shows the obtained PF (ps_1=100) with presenting the relationship between prediction error (MSE) and negative correlation after normalization. It can be seen that MSE in the range of $[0.1, 0.4]$ is empty, where we have demonstrated that few solutions distribute for this real-world application and will be dominated by other solutions. In Tab. II, ps_1=100 and ps_2=100 leads to the best ensemble learning performance, therefore Fig. (b) in Fig. 1 presents the accumulated coefficients with increasing trend until 1 under ps_2=100. The accumulated combination coefficients in Fig. (b) from 0 to 1 are corresponding to the single prediction performance MSE from 1 to 0 (after normalization) in PF. Overall, Fig. 1 shows that the members distributed in lower prediction error contribute most to the final ensemble prediction.

TABLE III: Performance of EMOEL compared with the state-of-the-art models

Models	MSE (Mean)
EMOEL_W	**694.0012**
EMOEL_WO	1194.3667
ELM-PLSR	2437.4601
GRUs	2666.5130
HELM	2553.1666
ELM	2694.4811
SVR	2633.5430
MLP	3012.4197

3) Comparison with the state-of-the-art models: The proposed EMOEL will be compared with several state-of-the-art models, such as ELM-PLSR, GRUs, HELM, ELM, SVR and MLP, with mean value presented in Tab. III, among which ELM-PLSR is an ensemble approach designed for electricity consumption prediction with different coefficients of DWT and ELM. EMOEL_W and EMOEL_WO are the ensemble learning performance with and without feature extraction, respectively. By comparing the results with statistical test, the proposed EMOEL outperforms the comparison models and shows the superior in solving the electricity consumption prediction with various auxiliary time series.

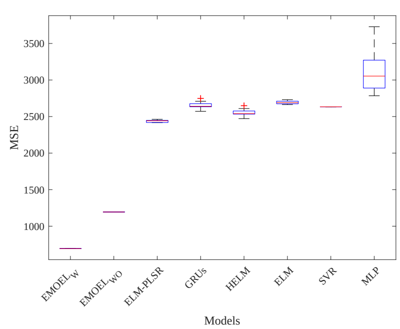

Fig. 2: MSE statistical summary for EMOEL_W, EMOEL_WO, ELM-PLSR, GRUs, HELM, ELM, SVR and MLP

Fig. 2 presents prediction performance with statistical summary of maximal, minimal and median values. The Standard deviation for EMOEL_W, EMOEL_WO and SVR are 0 on their 10 runs. Fig. 2 further identifies the superior of EMOEL on solving this prediction task by comparing to the performance of LM-PLSR, GRUs, HELM, ELM, SVR and MLP.

V. Related Work

Ensemble learning refers to combine the predictions of individual classifiers in order to obtain more robust prediction. Ensemble learning with an optimal population using population-based metaheuristics is first proposed in [25]. After that, multi-objective optimization is applied to evolve population by maintaining accuracy and diversity simultaneously [22]. So far, multi-objective optimization based ensemble learning has been developed in the area of classification and regression.

Ensemble classification design is to produce a diverse set of classifiers by optimizing certain conflicting criteria. A popular approach is to optimize neural network-based classifier or predictive accuracy. In [22], an algorithm called diverse and accurate ensemble learning algorithm (DIVACE) is proposed. DIVACE uses the idea from NCL [26] and memetic Pareto ANN (MPANN) [27], with formulating the ensemble learning problem as a multi-objective problem, and aiming to balance diversity and accuracy to produce an ensemble of neural network classifiers. In [28], multi-objective regularized NCL (MRNCL) is proposed with incorporating an additional regularization term for the ensemble with optimizing neural network structure for obtaining the members of ensemble.

Another popular approach for building MOEAs-based classifier ensemble is to encode a feature subset and other parameters and use a wrapper method to compute the objective function with classification accuracy and feature subset size. This idea is to evolve a set of non-dominated classifiers with respect to the trade-off between accuracy and feature subset size. In [29], MLP is used as the wrapper and classification accuracy and subset feature size are regarded as two objective functions to be optimized. Three classifiers like decision tree, SVM and MLP have been applied to wrappers and two objective functions to be optimized are average accuracy of these three classifiers and their consensus accuracy.

For regression problems with multi-objective optimization ensemble learning, the studies are not as many as classification. In [30], NCL framework is extended to a diversity-encouraging error function to the area of ordinal regression. In [31], Multi-objective optimization is applied to optimize the structure of Recurrent Neural Network and obtain the members of ensemble learning in Pareto front. Different selection methods have been used for pruning the members in Pareto front for the final ensemble learning. However, this approach is only proposed for univariate time series prediction. It does not solve the issues involving multiple factors in time series. In [32], multi-objective is applied for remaining useful life estimation in prognostic with optimizing the structure of deep belief networks. The impact of different time windows as the parameters have been investigated on the simulated data. However, as far as we know, the literatures on regression problem using multi-objective optimization based ensemble learning is relatively sparse. Evolutionary multi-objective ensemble learning for electricity consumption prediction with various auxiliary time series by considering channels selection, features extraction and time windows settings for different selected channels, and optimal prediction model configuration is the first work, which can be generally applied to any real-word multivariate time series prediction.

VI. Conclusions and Future Work

We proposed an EMOEL technique for solving electricity consumption prediction problems by considering both the consumption data and the auxiliary data (i.e., environmental information and electricity-related information) in the form of multivariate time series. A multi-objective EA is used to search for optimal configurations in terms of the channels to be used, the feature extraction methods to be applied to the selected channels, the time window sizes to be used by the feature extraction methods and the number of hidden neurons in the ELM (as the prediction model) subject to prediction accuracy and model diversity as two conflicting objectives. The finally obtained Pareto optimal solutions are combined to produce the ensemble prediction, where combination coefficients are optimized via a single-objective DE algorithm. Compared to several state-of-the-art methods, the proposed EMOEL technique demonstrated its superiority.

Our future work includes but is not limited to: (1) a deep analysis on the parameter sensitivity of the proposed method, e.g., population size [33], (2) incorporation of more feature extraction techniques particularly those used for multivariate feature extraction, (3) study on other prediction models and (4) investigation of parallel implementations of the proposed method [34].

REFERENCES

[1] R. E. Edwards, J. New, and L. E. Parker, "Predicting future hourly residential electrical consumption: A machine learning case study," *Energy and Buildings*, vol. 49, pp. 591–603, 2012.

[2] S. Voronin and J. Partanen, "Forecasting electricity price and demand using a hybrid approach based on wavelet transform, arima and neural networks," *International Journal of Energy Research*, vol. 38, no. 5, pp. 626–637, 2014.

[3] C. Fan, F. Xiao, and S. Wang, "Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques," *Applied Energy*, vol. 127, pp. 1–10, 2014.

[4] M. Meng, D. Niu, and W. Sun, "Forecasting monthly electric energy consumption using feature extraction," *Energies*, vol. 4, no. 10, pp. 1495–1507, 2011.

[5] 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.

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

[7] H. Song, A. K. Qin, and F. D. Salim, "Multivariate electricity consumption prediction with extreme learning machine," in *Proc. of the International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 2313–2320.

[8] K. Li, C. Hu, G. Liu, and W. Xue, "Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis," *Energy and Buildings*, vol. 108, pp. 106–113, 2015.

[9] Z. Hu, Y. Bao, and T. Xiong, "Comprehensive learning particle swarm optimization based memetic algorithm for model selection in short-term load forecasting using support vector regression," *Applied Soft Computing*, vol. 25, pp. 15–25, 2014.

[10] J. J. Liang, S. Baskar, P. N. Suganthan, and A. K. Qin, "Performance evaluation of multiagent genetic algorithm," *Natural Computing*, vol. 5, no. 1, pp. 83–96, 2006.

[11] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[12] S. Li, L. Goel, and P. Wang, "An ensemble approach for short-term load forecasting by extreme learning machine," *Applied Energy*, vol. 170, pp. 22–29, 2016.

[13] H. Song, A. Qin, and F. D. Salim, "Multi-resolution selective ensemble extreme learning machine for electricity consumption prediction," in *Proc. of the International Conference on Neural Information Processing*, 2017, pp. 600–609.

[14] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[15] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003, pp. 2–11.

[16] G. Luo, K. Yi, S.-W. Cheng, Z. Li, W. Fan, C. He, and Y. Mu, "Piecewise linear approximation of streaming time series data with max-error guarantees," in *Proc. of the IEEE 31st International Conference on Data Engineering (ICDE)*, 2015, pp. 173–184.

[17] T.-c. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[18] M. Ghofrani, M. Ghayekhloo, A. Arabali, and A. Ghayekhloo, "A hybrid short-term load forecasting with a new input selection framework," *Energy*, vol. 81, pp. 777–786, 2015.

[19] A. K. Qin and X. D. Li, "Differential evolution on the cec-2013 single-objective continuous optimization testbed," in *Proc. of the 2013 IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 1099–1106.

[20] B. Kazimipour, X. D. Li, and A. K. Qin, "Effects of population initialization on differential evolution for large scale optimization," in *Proc. of the 2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 2404–2411.

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

[22] A. Chandra and X. Yao, "Ensemble learning using multi-objective evolutionary algorithms," *Journal of Mathematical Modelling and Algorithms*, vol. 5, no. 4, pp. 417–445, 2006.

[23] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2016.

[24] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[25] X. Yao and Y. Liu, "Making use of population information in evolutionary artificial neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 3, pp. 417–425, 1998.

[26] Y. Liu and X. Yao, "Simultaneous training of negatively correlated neural networks in an ensemble," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 6, pp. 716–725, 1999.

[27] H. A. Abbass, "Pareto neuro-evolution: Constructing ensemble of neural networks using multi-objective optimization," in *Proc. of the 2003 Congress on Evolutionary Computation (CEC)*, 2003.

[28] H. Chen and X. Yao, "Multiobjective neural network ensembles based on regularized negative correlation learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 12, pp. 1738–1751, 2010.

[29] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Feature selection for ensembles: A hierarchical multi-objective genetic algorithm approach," in *Proc. of the Seventh International Conference on Document Analysis and Recognition-Volume 2*, 2003, pp. 1–5.

[30] F. Fernandez-Navarro, P. A. Gutierrez, C. Hervas-Martinez, and X. Yao, "Negative correlation ensemble learning for ordinal regression," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 11, pp. 1836–1849, 2013.

[31] C. Smith and Y. Jin, "Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction," *Neurocomputing*, vol. 143, pp. 302–311, 2014.

[32] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2306–2318, 2017.

[33] B. Kazimipour, X. D. Li, and A. K. Qin, "Initialization methods for large scale global optimization," in *Proc. of the 2013 IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 2750–2757.

[34] A. K. Qin, F. Raimondo, F. Forbes, and Y. S. Ong, "An improved cuda-based implementation of differential evolution on gpu," in *Proc. of the 14th annual conference on Genetic and evolutionary computation*, 2012, pp. 991–998.