

Robust Big Data Analytics for Electricity Price Forecasting in the Smart Grid

Kun Wang¹, Member, IEEE, Chenhan Xu², Yan Zhang³, Senior Member, IEEE, Song Guo⁴, Senior Member, IEEE, and Albert Y. Zomaya⁵, Fellow, IEEE

Abstract—Electricity price forecasting is a significant part of smart grid because it makes smart grid cost efficient. Nevertheless, existing methods for price forecasting may be difficult to handle with huge price data in the grid, since the redundancy from feature selection cannot be averted and an integrated infrastructure is also lacked for coordinating the procedures in electricity price forecasting. To solve such a problem, a novel electricity price forecasting model is developed. Specifically, three modules are integrated in the proposed model. First, by merging of Random Forest (RF) and Relief-F algorithm, we propose a hybrid feature selector based on Grey Correlation Analysis (GCA) to eliminate the feature redundancy. Second, an integration of Kernel function and Principle Component Analysis (KPCA) is used in feature extraction process to realize the dimensionality reduction. Finally, to forecast price classification, we put forward a differential evolution (DE) based Support Vector Machine (SVM) classifier. Our proposed electricity price forecasting model is realized via these three parts. Numerical results show that our proposal has superior performance than other methods.

Index Terms—Big data, price forecasting, classification, feature selection, smart grid

1 INTRODUCTION

1.1 Background

ONE of the main goals of smart grid is to reduce power peak load and to balance the gap between power supply and demand [1]. Customers are able to partake in the operations of smart grid, where the energy cost can be reduced by energy preservation and load shifting. In this case, dynamic pricing is a key indicator of users' switching load [2].

Generally, accurate point price forecasting is expected because of the requirement of economy and industry [3]. As for customers, they are actually eager to know whether the electricity price exceeds the specific customer-defined

thresholds, which they used to decide to turn the load on or off. Under this circumstance, customers require the electricity price classification. Hence, some specific thresholds based on point price forecasting algorithms are used to classify the electricity price. Function approximation techniques are the fundamental of point price forecasting algorithms, in which the basic process of price formation is imitated by a price model [4]. Moreover, price classification requires lower accuracy. Thus, electricity price classification becomes a key priority in the price forecasting.

The electricity price is influenced by various factors, such as fuel price, electricity requirement, renewable energy supply, etc., and it varies hourly. Since the electricity price changes frequently and large amounts of smart meters monitor the environment, such as fuel generation, wind generation, and transmission, in real-time, the amount of historical data is quite large [5], [6], [7], [8].

1.2 Related Works

We review the related works in electricity price forecasting and feature engineering in this part.

Machine learning and time-series model are two main ways for electricity price forecasting. Varshney et al. [9] developed a hybrid model to predict day ahead electricity market according to temperature and load information, with the utilization of neural network structure and analysis of singular spectrum. Mousavian et al. [10] put forward a probabilistic methodology to forecast per hour electricity price, where the bootstrapping technology is utilized for studying uncertainty and a generalized extreme learning machine method is proposed for wavelet neural networks. Kobayashi et al. [11] developed a switched Markov chain model for solving optimal electricity pricing problem in real-time based on a welfare function, which considers a

- K. Wang is with the Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing 210028, China, and the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China. E-mail: kwang@njupt.edu.cn.
- C. Xu is with the National Engineering Research Center of Communications and Networking, Nanjing University of Posts and Telecommunications, Nanjing 210028, China. E-mail: xchank@outlook.com.
- Y. Zhang is with the University of Oslo, Oslo 0313, Norway, and the Simula Research Laboratory, Fornebu 1364, Norway. E-mail: yanzhang@ieee.org.
- S. Guo is with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China. E-mail: song.guo@polyu.edu.hk.
- A. Y. Zomaya is with the School of Information Technologies, The University of Sydney, Sydney, NSW 2006, Australia. E-mail: albert.zomaya@sydney.edu.au.

Manuscript received 14 Oct. 2016; revised 23 May 2017; accepted 30 June 2017. Date of publication 4 July 2017; date of current version 7 Mar. 2019. (Corresponding author: Kun Wang.)

Recommended for acceptance by B. He, Y. Chen, and J. Zhou.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TBDATA.2017.2723563

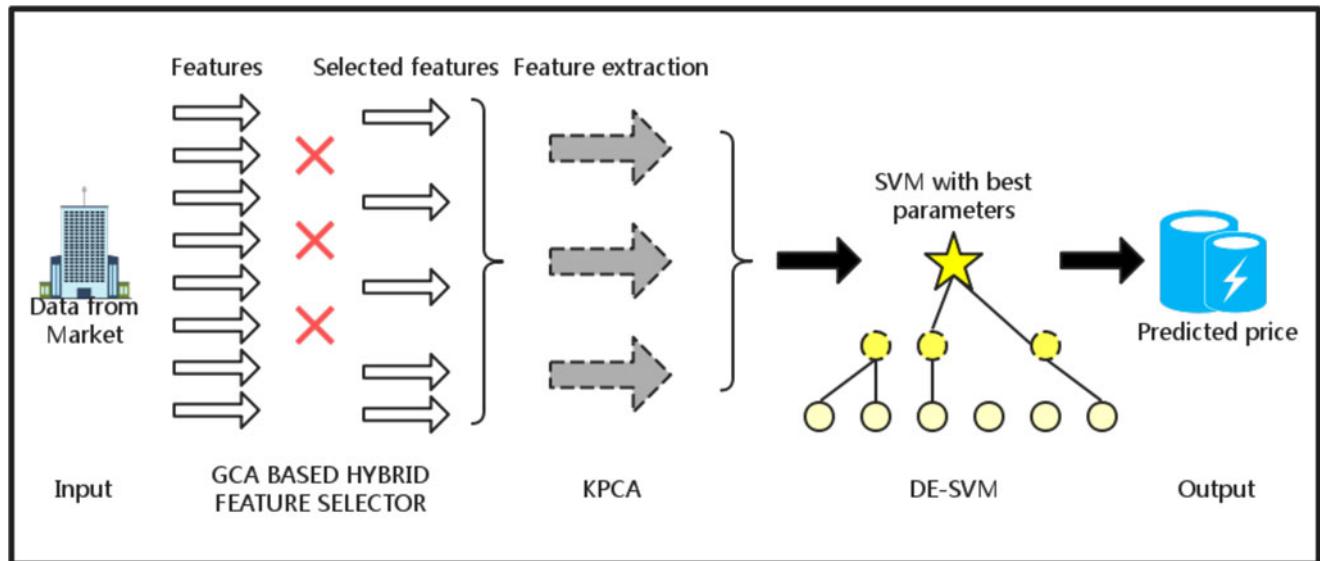


Fig. 1. Our proposed approach.

trade-off between users' utility and power conservation. Mosbah et al. [12] used multilayer neural networks in composite topology to enhance per hour electricity price forecasting accuracy.

Time series analysis is also widely used in electricity price forecasting, wherein auto-regressive integrated moving average (ARIMA) has good performance in stable electricity market [13]. Ozozen et al. [14] proposed an ARIMA based algorithm to predict electricity price in Turkey markets. Because of the existence of various outliers in ARIMA, building a model with raw data from market makes the forecasting accuracy unstable. Portela et al. [15] developed a seasonal autoregressive moving average Hilbertian (AMAH) model to estimate the mobile average values in functional time series, which can be applied to electricity price forecasting.

Feature engineering is fundamental to the application of classifier. Selection and extraction are two common operations in feature engineering. In electricity price, various methods are used for feature engineering. Zhao et al. [16] conducted a study of existing feature engineering methods and discussed how to obtain suitable features in price forecasting. Qiu et al. [17] applied multi-variable mutual information to feature selection. Abedinia et al. [18] extended the Mutual Information (MI) and Interaction Gain (IG) to measure the relevance of features. Qian et al. [19] used C4.5 algorithm for feature selection in price forecasting and the result shows that C4.5 performs better than Iterative Dichotomiser 3 (ID3) in decision tree building. Mori et al. [20] developed a feature extraction algorithm based on Symbolic Aggregation Approximation (SAX) to process time-series data.

Previous studies mainly focus on feature selection algorithms or classifiers design, where traditional classifiers, e.g., Decision Tree (DT) and Artificial Neural Network (ANN) are very popular [21], [22]. However, DT usually faces the overfit problem, which means the DT performs well in training but not in prediction [23], and ANN has a limited generalization capability and its convergence cannot be easily controlled [24]. Also, these learning based methods do not take the big data into consideration, and the evaluation of performance is

only conducted on the price data, which is not quite large. Hence, the price forecasting accuracy could still be improved with the help of big data.

1.3 Contributions

In this paper, we investigate the electricity price forecasting issue. Our objective is to predict the electricity price accurately by using the big data from grid. To overcome this challenging obstacle, we propose a Support Vector Machine (SVM) [25] underpinned framework that can predict the price efficiently. SVM is a classifier that try to find a hyperplane which can divide data into the correct classes. The support vector is a part of data that could help to determine the hyperplane. Although SVM is a promising approach, the following challenges need to be addressed on making the accurate electricity price forecasting.

- *High computational complexity.* According to Hu's work [26], SVM is weak in processing uncertain information and has high computational complexity. In the electricity price forecasting, irrelevant and redundant features take great computation complexity to the training process of SVM and also decrease the forecasting accuracy.
- *Hard to tune parameters.* There are three super parameters which are cost penalty, insensitive loss function parameter, and kernel parameter. These super parameters affect the performance of SVM in forecasting. It's hard to tune these parameters for higher accuracy and more efficiency. The common method used to adjust SVM super parameters is gradient descent (GD) algorithm or cross validation [25]. However, these two methods bring much computational complexity and may be unable to converge [27].

To address the challenges mentioned above, we propose a parallelized electricity forecasting framework, called Hybrid Selection, Extraction and Classification (HSEC), as shown in the Fig. 1. The three components of HSEC are parallelized Hybrid Feature Selector (HFS) based on Grey Correlation Analysis (GCA), feature extraction process based

on Kernel Principle Component Analysis (KPCA) and differential evolution (DE) based SVM classifier. The HSEC performs feature engineering by selecting features corresponding to the time sequence and the dimensional reduction of electricity price data features. The HFS uses the fusion of two feature selectors based on GCA rather than one to give an appropriate selection of features. Different from Principle Component Analysis (PCA) which is not suitable for high dimensional non-linear data [28], KPCA uses kernel function to deal with this dilemma. The main contributions of this paper are summarized as follows:

- We propose an integrated electricity price forecasting framework to make accurate big data forecasting in smart grid. To the best of our knowledge, it is the first attempt in this paper that feature selection, extraction and classification are integrated in this framework design for the studied problem.
- To achieve this framework, we first propose a GCA-based HFS, combining Relief-F algorithm and Random Forest (RF), to calculate the feature importance and control the feature selection. For feature extraction, we use KPCA to further reduce the redundancy among the selected features. We are the first to study the redundancy among the selected features in the electricity price forecasting field. We also design a DE-SVM algorithm to tune the super parameters of SVM, which has a higher accuracy than existing classifiers.
- The performance of our proposal is evaluated by several extensive simulations that based real world data traces of grid price and workload. The numerical results show that our proposal has better performance than benchmark approaches.

The rest of this paper are organized as follows. Section 2 surveys the proposed price forecasting framework. Sections 3 and 4 describe the feature selection and feature extraction, respectively. In Section 5, the enhanced SVM classifier based on DE is demonstrated. Section 6 shows the experimental results for verifying our proposed framework. The paper is concluded in Section 7 finally.

2 SYSTEM FRAMEWORK

Fig. 1 depicts the system framework of HSEC. The modules in this framework are made up with three parts, i.e., feature engineering (feature selection, feature extraction) and classification.

2.1 Design Goals

The goal of our framework is to do efficient and accurate forecasting of electricity price. To achieve this, we need to process the raw data, figure out the selected features and carefully tune the classifier. Thus, the following metrics are important for the processing performance of our proposed framework.

- *Accuracy of classification*: This is the core goal of our framework design.
- *Dimensional reduction rate*: In this framework, the performance of feature engineering influences the accuracy of classification directly.

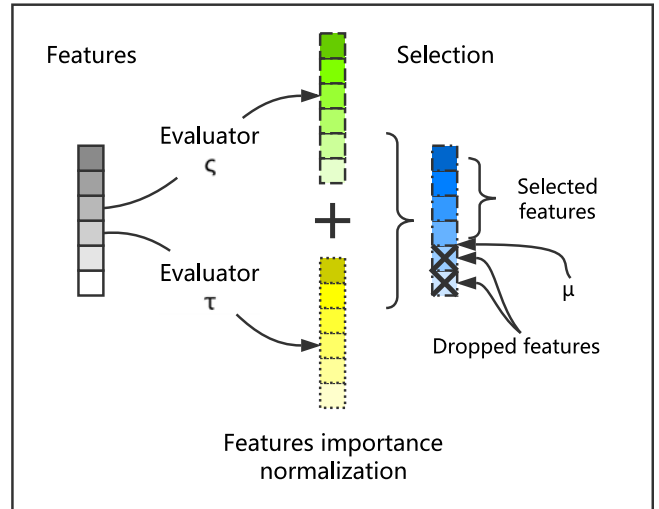


Fig. 2. HFS structure.

- *Time-efficiency*: Applied in electricity price forecasting, the framework should run fast.

2.2 Framework Overview

The primary issue in electricity price forecasting is accuracy. However, various factors influence the electricity, which makes the classifier training difficult. To enhance the accuracy of the proposed framework, we develop a parallelized HFS, a KPCA-based feature extraction, and a DE-SVM based classifier.

The HSEC begins with standardizing the raw data, which corresponds to the first part in Fig. 1. This standardization process is crucial for the implementation of the whole framework. Second, data flow into the GCA based HFS, where data will be used to train Relief-F and RF in parallel. Fig. 2 illustrates the details of HFS. This feature selector decides whether a feature is reserved by an index which is given by Relief-F and RF and is called feature importance. Due to the decoupling design of this selection algorithm, this process could execute distributively. Third, KPCA will be performed in the selected features for further removal of redundant features. In our proposed framework, factors incorporate depending on feature importance and redundancy. For example, the weather condition may affect the generation of solar and wind energy, this constrain will be reflected in the redundancy among weather, solar, and wind. Finally, the processed data is sent to build SVM. Since SVM is controlled by several super parameters, we use DE algorithm to tune these parameters. Table 1 introduces the major notations used in this paper. We will describe the details of these modules in the next three sections.

3 GCA BASED HYBRID FEATURE SELECTOR

This section describes the process of features selection. We propose a new parallel HFS based on GCA by fusing RF and Relief-F, and it is controlled by a new proposed threshold μ . The fusion of RF and Relief-F brings a feature selection that is more accurate. The Relief-F and RF can give feature importance, respectively. These two approaches are both efficient. Features are first roughly selected by GCA,

TABLE 1
Symbols and Variables

Notation	Description
A	a matrix of electricity price
$\lambda_i^*(k)$	the normalized component of data sequence
$\gamma(\lambda_o^*(k), \lambda_i^*(k))$	the grey coefficient between sequences λ_o^* and λ_i^*
$\Gamma_i(\lambda_o^*, \lambda_i^*)$	the grey correlation grade between sequences λ_o^* and λ_i^*
$W^R[T_k]$	the feature importance given by evaluator ζ
$W^F[T_k]$	the feature importance given by evaluator τ
μ	a threshold controlling the feature selection
X	the output matrix of HFS
C	the covariance matrix of X
F^*	the feature space used in KPCA
$\phi()$	a nonlinear function represents the mapping from the input vectors to feature space F^* in KPCA
p_k	the principle component
$K(x, y)$	the kernel function which has a form $\langle \phi(x), \phi(y) \rangle$
$\omega(c)$	the regularized risk function
N_p	the population size in DE-SVM
X_i^g	the i th individual of the g th generation in DE-SVM
V_i^{g+1}	the i th mutant individual of the $(g+1)$ th generation in DE-SVM
F_i	the mutation scale factor of the i th iteration in DE-SVM
F_{max}	the maximum mutant scale factor in DE-SVM
F_{min}	the minimum mutant scale factor in DE-SVM
g_{max}	the total number of the algorithm iterations in DE-SVM
CR	the crossover probability in DE-SVM
$U_{i,j}^{g+1}$	the intermediate of the crossing in DE-SVM

and the HFS performs the further selection by μ . We assume a matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad (1)$$

to denote the electricity price data. The rows represent the time stamps and columns denote the feature index, i.e., a_{ij} is the j th component of the data that i hour ahead of the price that is to be predicted. The matrix can be also formulated as

$$A = \begin{bmatrix} \vec{t}_1 \\ \vec{t}_2 \\ \vdots \\ \vec{t}_m \end{bmatrix}, \quad (2)$$

where

$$\vec{t}_k = [a_{k1}, a_{k2}, \dots, a_{kn}] \quad k \in [1, m]. \quad (3)$$

3.1 Preliminaries: Grey Correlation Analysis

Since various features have various degrees of influence on the final predicted electricity price, we use GCA to determine the importance of each feature. GCA calculates the correlation between each feature and the final electricity

price. Via GCA, the depth of RF could be effectively controlled.

In principle, GCA determines the correlation by quantifying the degree of ‘‘closeness’’ between two different data sequences. The closer two data sequence are, the greater correlation is. Thus, GCA can provide a quantitative measure of the closeness between the electricity prices.

Since the physical meaning of each feature in the framework is different, the dimension of data is not necessarily the same. Therefore, when the grey correlation grade analysis is carried out, the non-dimensional data processing will be executed accordingly. In this processing, each feature of data is nondimensionalized by dividing the data via their average value. Given two sequences $\lambda_0(k), i \in [1, n]$ and $\lambda_i(k) = a_{ki}, k \in [1, m]$, where n and m are the count of features and the length of time sequence, respectively, non-dimensional data processing makes the data sequences comparable. Since the original sequence has the characteristic of ‘‘the-larger-the-better’’ [29], it can be normalized as

$$\lambda_i^*(k) = \frac{\lambda_i(k) - \min \lambda_i(k)}{\max \lambda_i(k) - \min \lambda_i(k)}. \quad (4)$$

After the normalization, the gray coefficient [29] is calculated as

$$\begin{aligned} \gamma(\lambda_o^*(k), \lambda_i^*(k)) &= \frac{\Delta_{min} + \zeta \Delta_{max}}{\Delta_{oi}(k) + \zeta \Delta_{max}}, \quad \zeta \in (0, 1) \\ \Delta_{oi}(k) &= |\lambda_o^*(k) - \lambda_i^*(k)|, \\ \Delta_{max} &= \max_{i,k} |\lambda_o^*(k) - \lambda_i^*(k)|, \\ \Delta_{min} &= \min_{i,k} |\lambda_o^*(k) - \lambda_i^*(k)|, \end{aligned} \quad (5)$$

where ζ is named as distinguishing coefficient, and is usually set as 0.5 [30].

The grey correlation grade is denoted by

$$\Gamma_i(\lambda_o^*, \lambda_i^*) = \frac{\sum_{k=1}^m \gamma(\lambda_o^*(k), \lambda_i^*(k))}{m}. \quad (6)$$

By sorting the features with descending grey correlation grade order, several low correlated features can be removed. After ϵ features are dropped, the time sequence becomes

$$\vec{t}_i = [a_{k1}, a_{k2}, \dots, a_{k(n-\epsilon)}]. \quad (7)$$

3.2 Hybrid Feature Selector

The HFS consists of two paratactic feature importance evaluators, which are evaluators ζ and τ . As shown in Fig. 2, these two evaluators evaluate features’ importance independently. The features’ importance that generated by these two evaluators are jointly considered when conducting the features selection among features.

Algorithm 1 describes how HFS works on the features in detail. The evaluators and selector are indicated independently in pseudo code. Evaluator ζ is based on RF, which is an ensemble learning algorithm. RF grows on Bootstrap AGGREGATING (Bagging) samples [31]. Bagged data is divided into training and Out of Bag (OOB) data sets. In evaluator ζ , we start with initializing all weights as 0, then we train the RF, and use OOB data with noise to calculate

the feature importance. Evaluator τ is based on Relief-F, which is a widely adaptable filter-based feature evaluating algorithm. In evaluator τ , we update weights from the distance among near hits and misses. These two evaluators are designed in a decoupled approach, i.e., they can give feature importance independently. Hence, they can be deployed on a distributed system and are able to make full use of computing resources. Gathering the feature importance generated by two evaluators, selector performs threshold-controlled feature selection.

Algorithm 1. Hybrid Feature Selector

Input: $W^R[T_k] \leftarrow 0.0, W^F[T_k] \leftarrow 0.0, A[], R[n]$
Output: $W^R[T_k], W^F[T_k]$

1. **begin**
2. **initialization:** set all weight
 $W^R[T_k] \leftarrow W^F[T_k] \leftarrow 0.0$, read data from $A[]$
3. **Evaluator ζ :**
4. **begin**
5. **for** k **from** 1 **to** m **do**
6. **for** i **from** 1 **to** n **do**
7. calculate $errOBB1_i$ using corresponding OBB data set of decision tree[i]
8. randomly add noise to all OBB data on feature T_k
9. calculate $errOBB2_i$ using corresponding OBB data set of decision tree[i]
10. **end**
11. calculate the importance of feature
 $W^R[T_k] \leftarrow \frac{\sum_{i=1}^n (errOBB2_i - errOBB1_i)}{n}$
12. **end**
13. **end**
14. **Evaluator τ :**
15. **begin**
16. **for** k **from** 1 **to** m **do**
17. select an item in $class(C_i)$ by random
18. find k nearest hits item $H_j(C_i)$
19. for each $class(C_j) \neq class(C_i)$
20. find k nearest miss item $M_j(C_j)$
21. **end**
22. **for** i **from** 1 **to** m **do**
23. update $W^F[T_i]$
24. **end**
25. **end**
26. **Selector:**
27. **begin**
28. normalize W^R, W^F
29. perform feature selection
30. **end**
31. **end**

In Algorithm 1, $R[n]$ represents a RF that consists of n decision trees, $W^R[T_k]$ is the feature importance given by evaluator ζ , and $W^F[T_k]$ is the feature importance given by τ . These two parameters are updated as

$$W^F[T_k] = W^F[T_k] - \frac{\sum_{j=1}^k diff(A, a^*, H_j)}{m * k} + \frac{\sum_{C \neq class(\lambda)} diff(A, a^*, M_j(C))}{m * k}, \quad (8)$$

where a^* is a randomly selected item in class C_i , the function $diff(A, R_1, R_2)$ calculates the difference between two items R_1 and R_2 in the attribute A , i.e.,

$$diff = \begin{cases} 0 & \text{(Discrete attributes, the values are different)} \\ 1 & \text{(Discrete attributes, the values are the same)} \\ \text{Actual difference normalized to the interval } [0, 1]. & \text{(Continuous attributes)} \end{cases}$$

Feature selection is based on W^R and W^F , which can be normalized by

$$\begin{aligned} W^R &= W^R / \max(W^R), \\ W^F &= W^F / \max(W^F). \end{aligned} \quad (9)$$

Then, the feature selection performs as

$$T_k = \begin{cases} \text{reserve} & W^R[T_k] + W^F[T_k] > \mu \\ \text{drop}, & W^R[T_k] + W^F[T_k] \leq \mu, \end{cases}$$

where μ is a threshold controlling the feature selection.

The features selected by HFS also have redundancy among them. Thus, they will be sent to KPCA for further feature extraction.

4 FEATURE EXTRACTION-KPCA

In this section, we propose feature extraction for this framework. The features selected by HFS can be considered that have no irrelevant features, but also have redundant features. PCA is the most common method for feature extraction which reduce the redundancy among features, but it assumes a linear mapping from high to low dimension space [32]. In electricity price forecasting, data requires non-linear mapping to find an appropriate low dimensional embedding [33]. Thus, Kernel Principle Component Analysis is applied to perform the nonlinear dimension reduction.

The features selected via HFS are used as the input of KPCA, and the output matrix of HFS can be rewritten as

$$X = (x_1, x_2, x_3, \dots, x_N)^T, \quad (10)$$

where x_i is the i th variable related to electricity price.

The correlation between features and eigen values can be formulated as

$$\lambda v = C^{F^*} v, \lambda \geq 0 \text{ and } v \in F^*, \quad (11)$$

where the covariance matrix of X is denoted by C , F^* represents the feature space, and λ is eigenvalue. Here, $C^{F^*} v$ is given as

$$\begin{aligned} C^{F^*} v &= \frac{1}{N} \sum_{i=1}^N \langle \phi(x_i), v \rangle \phi(x_i), \\ \sum_{k=1}^N \phi(x_k) &= 0, \end{aligned} \quad (12)$$

where $\phi()$ maps the input data to feature space, and $\langle x, y \rangle$ calculates the inner product of x and y . Therefore, Eq. (11) could be rewritten as

$$\lambda \langle \phi(x_k), v \rangle = \langle \phi(x_k), C^{F^*} v \rangle, \quad k = 1, \dots, N. \quad (13)$$

When $\lambda \neq 0$, v has the form as

$$v = \sum_{i=1}^N \beta_i \phi(x_i), \quad i = 1, \dots, N, \quad (14)$$

where β_i are the corresponding coefficients of x_i .

Generally, the specific form of $\phi()$ is unknown. Hence, we introduce the kernel function [34] as

$$K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle, i, j \in [1, N]. \quad (15)$$

Combining Eqs. (13) and (14), we have

$$\lambda \sum_{i=1}^N \beta_i K_{ki} = \frac{1}{N} \sum_{i=1}^N \beta_i \sum_{j=1}^N K_{kj} K_{ij}. \quad (16)$$

Since $k \in [1, N]$, we set $\beta = [\beta_1, \dots, \beta_N]^T$, then Eq. (16) can be rewritten as

$$\lambda N K \beta = K^2 \beta. \quad (17)$$

To perform the dimensional reduction, eigenvectors $\beta_1, \beta_2, \dots, \beta_n$ with $\lambda_1 \geq \lambda_2 \dots \geq \lambda_n$ are picked, and these eigenvectors need to be normalized.

We have

$$\langle v_k, v_k \rangle = 1, \quad k \in [1, N]. \quad (18)$$

Substituting Eq. (14) into Eq. (18), we can get

$$\begin{aligned} 1 &= \left\langle \sum_{i=1}^N \beta_i^k \phi(x_i), \sum_{j=1}^N \beta_j^k \phi(x_j) \right\rangle \\ &= \sum_{i=1}^N \sum_{j=1}^N \beta_i^k \beta_j^k \langle \phi(x_i), \phi(x_j) \rangle \\ &= \sum_{i=1}^N \sum_{j=1}^N \beta_i^k \beta_j^k K_{ij} \\ &= \langle \beta_k, K \beta_k \rangle \\ &= \lambda_k \langle \beta_k, \beta_k \rangle. \end{aligned} \quad (19)$$

Then the principal component can be extracted as

$$p_k = \langle v_k, \phi(x) \rangle = \sum_{i=1}^N \beta_i^k \langle \phi(x_i), \phi(x) \rangle, \quad (20)$$

where p represents principal component, $v_k \in F$, $k = 1, \dots, n$.

Generally, kernel functions have the following forms: linear kernel

$$k(x, y) = \langle x, y \rangle, \quad (21)$$

sigmoid kernel

$$k(x, y) = \tanh(\alpha_0 \langle x, y \rangle^d + \alpha_1), \quad (22)$$

and radial basis kernel

$$k(x, y) = \exp(-\theta \|x - y\|^2), \quad (23)$$

where d, α_0, α_1 , and θ are specified by user [32]. We will give the numerical results in Section 6 to show the performance comparisons among these kernel functions.

Based on the feature selection and feature extraction illustrated above, we propose the improved SVM classifier in the next section to perform the final electricity price forecasting.

5 CLASSIFIER ADJUSTMENT

After the two-stage feature selection and extraction, unimportant and redundant features have been dropped. This section describes our proposed approach that accomplishes the final electricity price forecasting via the processed data. Since SVM is robust and efficient enough in electricity price data, we choose SVM as the classifier. In this section, the formulated classification problem are investigated first. After that, the DE based SVM is proposed to optimize this problem.

5.1 Problem Formulation

In this paper, we model the classification problem as

$$f(x, c) = \sum_{i=1}^D c_i \lambda_i(x) + b, \quad (24)$$

where b depends on the data distribution and $c_i^\infty (i = 1, 2, \dots)$ are the parameter of the classifier to be determined. Eq. (24) defines a hyperplane in the D -dimensional feature space. We then define the regularized risk function given by

$$w(c) = \frac{\sum_{i=1}^D |y_i - f(x_i, c)|_\epsilon + \mu c^2}{D}, \quad (25)$$

where ϵ is the insensitive loss function parameter, μ is a constant, and y_i is the actual price. To obtain the parameter c , minimization of this regularized risk function should be performed. The robust error function is denoted by

$$x = \begin{cases} 0 & \text{if } |y_i - f(x_i, c)| < \epsilon \\ |y_i - f(x_i, c)| & \text{otherwise.} \end{cases} \quad (26)$$

This function used to minimize Eq. (25) can be formulated as

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^N (\alpha^* - \alpha) K^*(x, x_i) + b, \quad (27)$$

where $\alpha^* \geq 0$ for $i \in [1, N]$. Note that $K^*(x, y)$ is a kernel function of SVM, and it represents the dot product in the F^*

$$K^*(x, y) = \sum_{i=1}^D \lambda_i(x) \lambda_i(y). \quad (28)$$

In fact, a symmetric function corresponding a positive-semidefinite kernel matrix can also be used as the kernel function [35]. The application of kernel function makes the feature λ_i never need to be calculated in infinite dimensional feature space. To obtain α and α^* , we can maximize the quadratic form as

$$\begin{aligned} R(\alpha^*, \alpha) &= -\epsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) + \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) \\ &\quad - \frac{1}{2} \sum_{i,j=1}^N (\alpha_i^* + \alpha_i) (\alpha_j^* - \alpha_j) K^*(x_i, x_j). \end{aligned} \quad (29)$$

5.2 Optimal Classification

As previously discussed, our goal is to minimize the regularized risk function. However, there is a strong relationship between the regularized risk function of SVM and the value of super parameters, which are c (cost penalty), ε (insensitive loss function parameter) and σ (kernel parameter). However, how to tune these three basic super parameters for higher accuracy and more efficiency is still a critical issue. The basic method used to adjust super parameters of SVM is gradient descent algorithm or cross validation [24]. However, these two methods bring much computational complexity and may be unable to converge. In HSEC, therefore, a reliable Differential Evolution algorithm is applied to tune the super parameters.

In essence, DE mainly consisting of four procedures to optimize the target, which are initialization, crossover, mutation and selection of the super parameters. Every different vector consists of the values of these super parameters can be a component in the population. The current population, represented by P_c , consists of D -dimensional vectors

$$X_i^g = \{X_{1i}^g, X_{2i}^g, \dots, X_{Di}^g\}, i \in [1, N_p], \quad (30)$$

where g is an index indicating the generation a vector belongs to, i is a population index and N_p represents the population size. Every component in X_i^g is called an individual. Once initialized, DE mutates individuals to generate new individuals from one generation to the next randomly. The four main procedures run repeatedly as follows until the optimal parameters are obtained:

- 1) *Initialization*: This stage forms the first population randomly. We make the first population obey the uniform distribution.
- 2) *Mutation*: The target of mutation operation is to generate new individuals. Generally, the i th mutant individual of the $(g+1)$ th generation is determined by

$$V_i^{g+1} = X_{r1}^g + F(X_{r2}^g - X_{r3}^g), F \in (0, 2), \quad (31)$$

where $X_{r1}^g, X_{r2}^g, X_{r3}^g$ are individuals selected from the g -generation by selection operation. F is a mutation scale factor [36]. We improve the mutation operation as

$$V_i^{g+1} = X_*^g + F(X_{r1}^g - X_{r2}^g), \quad (32)$$

$$F_i = F_{max} - \frac{g(F_{max} - F_{min})}{g_{max}}, \quad (33)$$

where X_*^g represents the most suitable individual in the g th generation, F_i controls the mutation scale of the i th iteration, F_{max} and F_{min} are the maximum and the minimum F_i , and g_{max} is the total number of the algorithm iterations. Different from traditional DE, the mutation scale factor dynamic adjusts in our approach, which is able to accelerate the process of optimization.

- 3) *Crossing*: Crossing is to increase the variety of generation and mix the mutant individuals via the origin individuals in every dimension with a certain probability. The crossing operation performs as

$$U_{i,j}^{g+1} = \begin{cases} |V_{i,j}^g| & r_{i,j} \leq CR \\ X_{i,j}^g & otherwise, \end{cases} \quad (34)$$

where $j \in [1, D]$, D is the dimensions of the space, and CR is used to control the crossover probability. $r_{i,j}$ is a random factor uniformly distributed between $[0, 1]$. $U_{i,j}^{g+1}$ is called the intermediate of the crossing.

- 4) *Selection*: Selection can be formulated as

$$X_i^{g+1} = \begin{cases} |U_i^{g+1}| & f(U_i^{g+1}) < f(X_i^g) \\ X_i^g & otherwise, \end{cases} \quad (35)$$

where $f(U_i^{g+1})$ and $f(X_i^g)$ represent the fitness function of U_i^{g+1} and X_i^g , respectively. In other words, the selection operation selects the individuals that make SVM more accurate.

The pseudo code of DE-SVM is demonstrated as Algorithm 2. This algorithm starts with initializing the population randomly. After that, it executes the procedures mentioned above repeatedly to obtain the optimal super parameters of SVM.

Algorithm 2. Differential Evolution-Based SVM

Input: $X_i \leftarrow (0, 0, 0)$
Output: $F(y) = \{C_1, C_2, \dots, C_k\}$

1. **begin**
2. set g_{max}, F, CR, N_p ,
3. randomly set $X_i \leftarrow \{c, \sigma, \varepsilon\}, i \in [1, N_p]$
4. **for** i from 1 to N_p **do**
5. $f_i(X_i) \leftarrow \frac{1}{TN} \sum_{i=1}^T N(\tilde{y}_i - y_i)^2$
6. **if** $f_i(X_i) < f_{i+1}(X_{i+1})$ **then**
7. reserve $f_i(X_i)$
8. compare $f_i(X_i)$ with $f_{i+2}(X_{i+2})$
9. **else**
10. reserve $f_{i+1}(X_{i+1})$
11. compare $f_{i+1}(X_{i+1})$ with $f_{i+2}(X_{i+2})$
12. **end**
13. obtain $f_{min}(X_i)$ and denote the X_i as X_*
14. **end**
15. $V_i^{g+1} \leftarrow X_*^g + F(X_{r1}^g - X_{r2}^g), u_{i,j}, X_i^{g+1}$
16. $X_* \leftarrow (c_*, \sigma_*, \varepsilon_*)$
17. solve classification function, output $F(y)$
18. **end**

According to the integration of HFS, KPCA and DE-SVM, our electricity price forecasting framework can predict the electricity price accurately. In the next section, we will give experiments and analysis based on the real-world price data.

6 NUMERICAL RESULTS

6.1 Analysis

The complexity of Algorithm 1 could be estimated as follows: the line 7 of the algorithm takes $O(n)$ time, the line 8 takes $O(m)$ time, and the line 9 takes $O(n)$ time. Consider the complexity of line 11 as $O(n)$, the evaluator ζ takes $O(m^2n + 3mn^2)$ time. Accordingly, lines 7-9 can be executed parallel to decrease the complexity to $O(m^2 + 3mn)$. As the result of $n \gg m$, the complexity of

TABLE 2
Attributes of Data

Attributes	Description
Hour	The index of hours
DA_DEMD	The day-ahead demand, which consists of fixed demand bids, price sensitive demand bids, decrement bids, and increment offers.
DEMAND	The Non-PTF Demand for ISO-NE CA and the load zones as measured by metering.
DA_LMP	The locational marginal price one day ahead.
DA_CC	The congestion part of the day ahead price.
DA_EC	The energy part of the day ahead price.
DA_MLC	The marginal loss part of the day ahead price.
RT_LMP	The dynamic locational marginal price.
RT_EC	The energy part of the dynamic price.
RT_CC	The congestion part of the dynamic price.
RT_MLC	The marginal loss part of the dynamic price.
DryBulb	The dry bulb temperature ($^{\circ}$ F).
DewPnt	The dew point temperature ($^{\circ}$ F).
SYSLoad	The actual system load (MW) measured by metering.
COAL	The energy supplied by coal.
GAS	The electricity supplied by natural gas.
HYDRO	The electricity supplied by pondage, pumped Storage, reservoir, and run of River.
NUCLEAR	The nuclear electricity.
OIL	The electricity supplied by diesel Oil, jet Fuel Oil, and other oil.
REFUSE	The electricity generated by burning refuse.
SOLAR	The electricity generated by solar energy.
WIND	The electricity supplied by wind power.
OTHER	Other electricity sources.
TOTAL	The total energy generation.
WEATHER	The weather condition.

evaluator ζ is $O(mn)$. It is straightforward that the lines 18 and 20 take $O(n)$ time, and line 17 takes $O(1)$ time. Hence, the complexity of evaluator τ is $O(mn)$ time. These two evaluators could run synchronously by Apache Spark [37]. To this end, the complexity of Algorithm 1 is $O(mn)$.

Obviously, the complexity of Algorithm 2 could be generated by multiplying the complexity of SVM by the generation limitation N_p . In the worst situation, the complexity of SVM is $O(SV^3)$ time. Hence, the complexity of Algorithm 2 is $O(N_pSV^3)$.

In fact, the ANN based learning method is extremely complex. The complexity of ANN is several times that of our proposal due to the forward and back propagation operations through hidden layers.

6.2 Simulation Setup

In order to investigate the capability of our proposed framework, we develop a simulator with Python according to the system framework designed in Section 2. During the simulation, the simulator is running on the platform with Intel Core i5, 4 GB RAM, and 500G hard disk. Hourly electricity price data and energy generation data of the ISO New England Control Area (ISO NE-CA) from 2010 to 2015 are taken as the input for this framework [7], which consists of over 50,000 real-world electricity price records. The data includes attributes shown in Table 2. Simulation results are organized as follows:

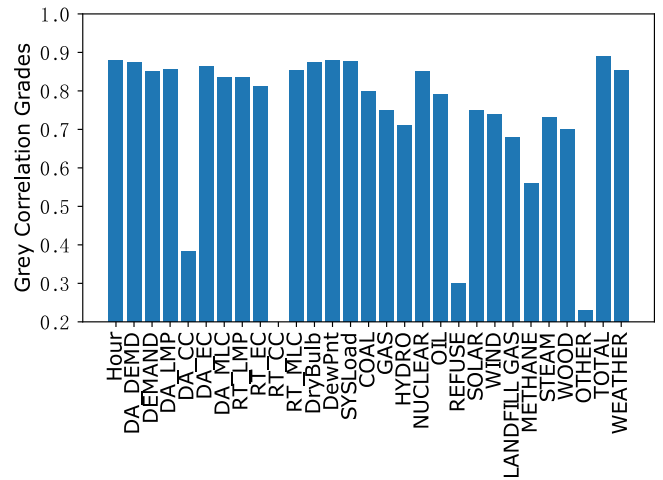


Fig. 3. Grey correlation grades of each feature (DA_CC, RT_CC, REFUSE, and OTHER have low grades obviously).

- (1) Feature selection performance of GCA based HFS;
- (2) KPCA performance in various kernel functions compared with PCA;
- (3) DE-SVM performance compared with benchmark algorithms.
- (4) The HSEC performance with the benchmark frameworks.
- (5) DE-SVM robustness compared with benchmark algorithms.

6.3 Simulation Results

1) Feature Selection Performance of GCA Based HFS.

GCA based HFS is applied to roughly select features from hourly electricity price data during 2010-1-1 to 2015-12-31 in ISO NE-CA. In feature selection, every feature sequence has a form as a vector. The components of this sequence represent the feature values in different time-stamps. Since our goal is to predict the electricity price, which is named regulation clearing price (RegCP) in the data, features that have little effect on the price can be removed. Before HFS, GCA performs the correlation calculation between features and RegCP, as shown in Fig. 3. The grey correlation grade shown in Fig. 4 indicates that most features' grade is above 0.5. We drop four features with

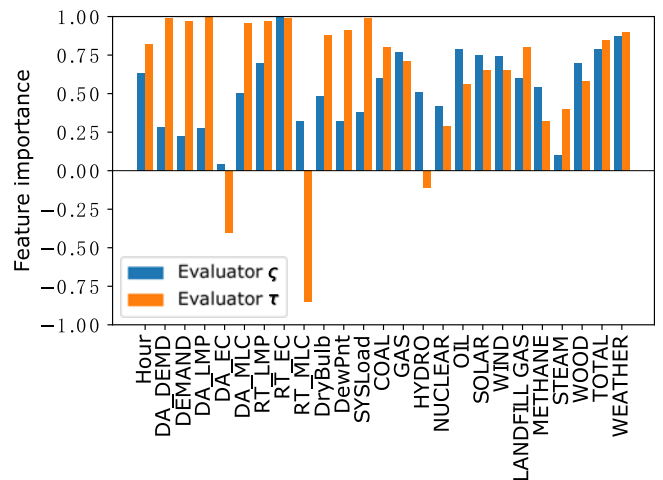


Fig. 4. Feature importance given by Evaluator ζ and τ .

TABLE 3
Performance of DE-SVM in Different Threshold

μ	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
Error	4.6%	4.8%	4.8%	10.2%	13.6%	20.4%	45.8%	45.8%	45.8%	NaN
Time	0.139	0.121	0.121	0.517	0.422	0.216	0.038	0.038	0.038	NaN
Dropped features	DA_EC, RT_MLC, HYDRO, NUCLEAR, METHANE, STEAM	DEMAND	DA_DEMD, DA_LMP, DewPnt, WOOD	DryBulb, SYSLoad, OIL, WIND	Hour, DA_MLC, COAL, GAS, SOLAR, LANDFILL GAS		RT_LMP, TOTAL	WEATHER		RT_EC

obvious low grade, i.e., features DA-CC (0.8543), RT-CC (NaN), REFUSE (0.2341), and OTHER (0.3049). According to the results generated by GCA, the feature importance of reserved features can be evaluated. We perform Evaluator ζ and τ , described in Algorithm 1, to obtain the importance value of each feature. The results are shown in Fig. 4, where negative values mean that the sum of $diff()$ among nearest hits items is larger than that among the nearest miss items. In order to boost the DE-SVM classifier, we discuss the different values of threshold μ , which controls the feature selection in Table 3. For example, updating μ from 1.1 to 1.2 results in the dropping of DEMAND. With the increase of threshold μ , more features are dropped, resulting in the increase of training speed and the decrease of accuracy.

2) *KPCA Performance in Different Kernel Functions Compared with PCA.*

In order to eliminate the redundant information within the features, KPCA is used to extract the principle components. The cumulative contribution comparison among KPCA with different kernels and PCA is shown in Fig. 5. When the cumulative contribution rate reaches 95 percent, the radial basis KPCA extracts most of the principle components, as shown in Fig. 6. Thus, we select the radial basis function as the kernel of KPCA to guarantee the accuracy of forecasting. The data points of radial basis KPCA distribute along coordinate axes, i.e., radial basis KPCA can extract the principle components that are more representative than the other two kernels.

3) *DE-SVM Performance Compared with Benchmark Algorithms.*

We compare the DE-SVM with two benchmark classifiers, i.e., Naive Bayes (NB) and Decision Tree. The

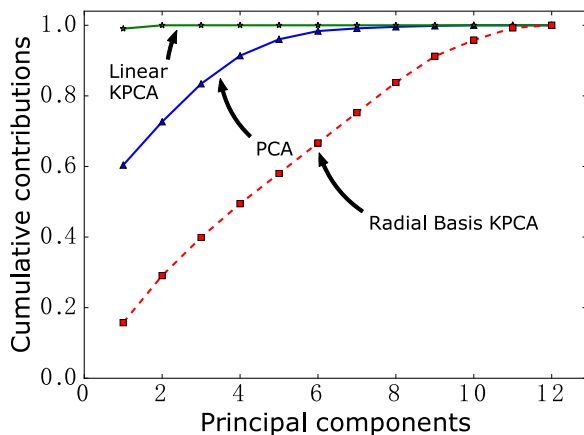


Fig. 5. Performance of PCA, linear KPCA and radial basis KPCA on features.

results are shown in Fig. 7. The accuracy of DE-SVM achieves 98 percent, and its curve fits well with the real value, as NB and DT have some outliers when $timeseries = 11, 12, 14$. The DE optimizes the super parameters of SVM jointly. Therefore, DE-SVM performs better at the accuracy of electricity price forecasting than NB and DT.

4) *The HSEC Performance of Price Forecasting.*

In order to investigate the capability of HSEC, comparisons among different benchmarks are conducted in this part. The benchmarks used in this part are shown in Table 4. As shown in Fig. 8, the HSEC has higher accuracy in electricity price forecasting than all the benchmarks. The

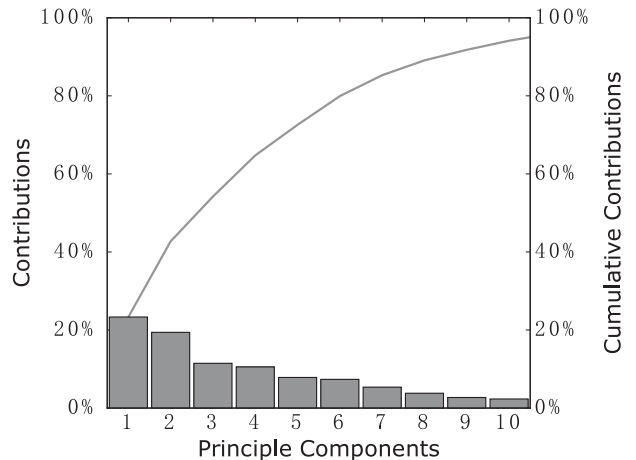


Fig. 6. Cumulative contribution of radial basis KPCA.

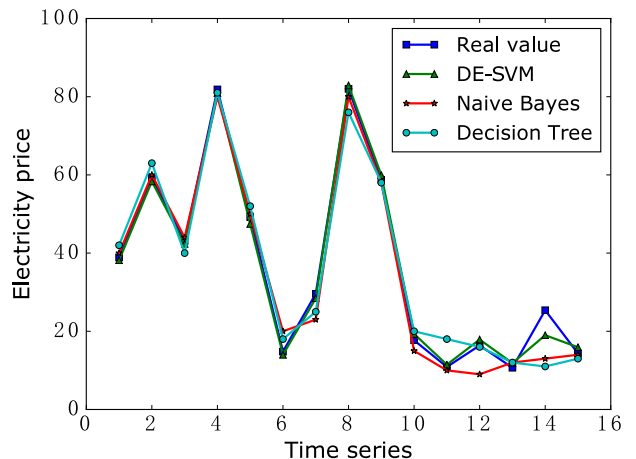


Fig. 7. Comparison on price forecasting among Naive bayes, decision tree and DE-SVM.

TABLE 4
Benchmark Frameworks

Benchmark	Description
A	HFS + DE-SVM
B	KPCA + DE-SVM
C	DE-SVM
D	DT [21]
E	RF [21]

comparison among frameworks A, B, C and HSEC spots that every module in our proposal can improve the accuracy of electricity price forecasting. The HSEC reduces the irrelevance and redundancy among features, and uses DE to tune the super parameters of SVM, which guarantees the accuracy of electricity price forecasting.

5) *DE-SVM Robustness Compared with Benchmark Algorithms.*

The robustness comparison is conducted by two ways. The first way is to add random noise (Error) to each feature and observe the average accuracy of each algorithm, which are drawn by solid lines in Fig. 9. The second way is adding random noise to the selected data, these data may not belong to the same feature (global). The results are illustrated by dashed lines in Fig. 9. In essence, the first noise is usually generated by the malfunction sensors, and transmission under critical environment may bring the second noise. As Fig. 9 depicted, our proposed framework is more robust than the two benchmarks. The noise within one feature has a very limited influence to the accuracy of our proposal, because this less important feature has been dropped during the feature selection and extraction. This framework also strengthens the robustness against the noise over several features.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated the electricity price forecasting problem in smart grid via joint consideration of feature engineering and classifier parameters adjustment. An electricity price forecasting framework which consists of two-stages feature processing and improved SVM classifier has been proposed to solve this problem. Specifically, to

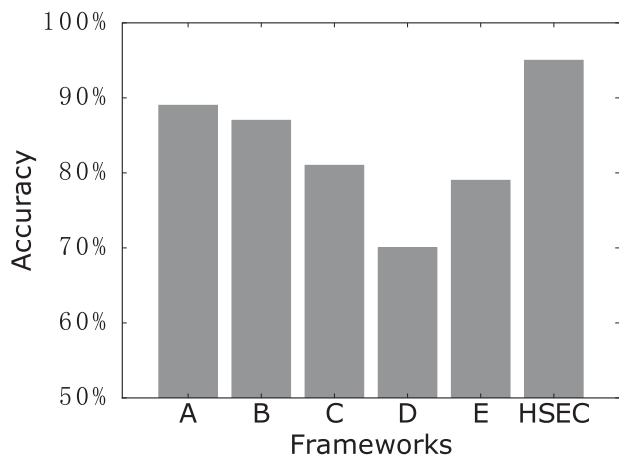


Fig. 8. Comparison on accuracy among HSEC and benchmark frameworks.

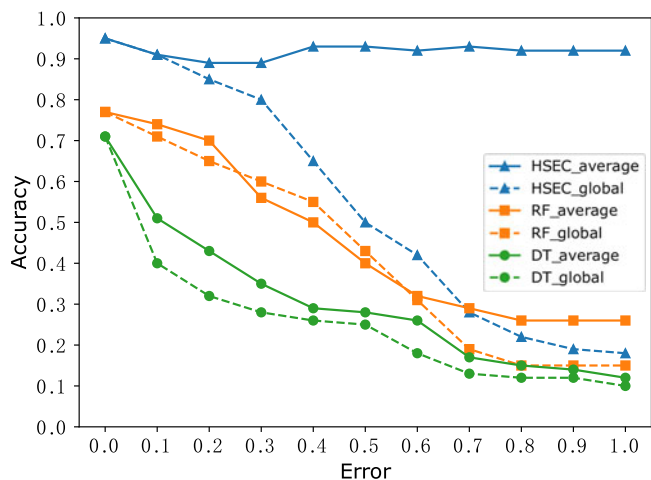


Fig. 9. Comparison on robustness among HSEC and benchmark frameworks.

select those important features, a new hybrid feature selector based on GCA is used to process the n -dimensional time sequence as an input. Additionally, KPCA is applied to extract new features with less redundancy, which boosts SVM classifier in accuracy and speed. Moreover, the DE algorithm obtains the appropriate super parameters for DE-SVM automatically and efficiently. The numerical results have shown that our proposed framework is more accurate than other benchmark algorithms.

With the consideration of the huge amounts of data, it is significant for our framework to adequately utilize computation resources and support the parallel computing. Different from traditional electricity price forecasting approaches where the date is processed sequentially, the proposed framework is easy to implement on a parallelized and distributed system. In the future, the real-time requirement will be considered in this framework.

ACKNOWLEDGMENTS

This work is supported by NSFC (61572262, 61533010, 61373135, 61571233, 61532013); National China 973 Project (2015CB352401); the NSF of Jiangsu Province (BK20141427); China Postdoctoral Science Foundation (2017M610252) and China Postdoctoral Science Special Foundation (2017T100297). The projects 240079/F20 funded by the Research Council of Norway; the project Security in IoT for Smart Grids, with number 248113/O70 part of the IKTPLUSS program funded by the Norwegian Research Council.

REFERENCES

- [1] K. Wang, Y. Wang, X. Hu, Y. Sun, D. J. Deng, A. Vinel, and Y. Zhang, "Wireless big data computing in smart grid," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 2–8, Apr. 2017.
- [2] X. Chen, Y. Zhou, W. Duan, J. Tang, and Y. Guo, "Design of intelligent demand side management system respond to varieties of factors," in *Proc. China Int. Conf. Electricity Distrib.*, 2010, pp. 1–5.
- [3] A. H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 120–133, Sep. 2010.
- [4] M. Erol-Kantarci and H. T. Mouftah, "Energy-efficient information and communication infrastructures in the smart grid: A survey on interactions and open issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 179–197, Jan.–Mar. 2015.

- [5] P. D. Diamantoulakis, V. M. Kapinas, and G. K. Karagiannidis, "Big data analytics for dynamic energy management in smart grids," *Big Data Res.*, vol. 2, no. 3, pp. 94–101, 2015.
- [6] K. Wang, H. Li, Y. Feng, and G. Tian, "Big data analytics for system stability evaluation strategy in the energy internet," *IEEE Trans. Ind. Informat.*, Apr. 2017, doi: 10.1109/TII.2017.2692775.
- [7] ISO new england energy offer data, 2016. [Online]. Available: www.iso-ne.com/isoexpress/web/reports/pricing/-/tree/day-ahead-energy-offer-data
- [8] K. Wang, Z. Ouyang, R. Krishnan, L. Shu, and L. He, "A game theory based energy management system using price elasticity for smart grids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1607–1616, Dec. 2015.
- [9] H. Varshney, A. Sharma, and R. Kumar, "A hybrid approach to price forecasting incorporating exogenous variables for a day ahead electricity Market," in *Proc. IEEE 1st Int. Conf. Power Electron. Intell. Control Energy Syst.*, 2016, pp. 1–6.
- [10] M. Rafiei, T. Niknam, and M. H. Khooban, "Probabilistic forecasting of hourly electricity price by generalization of ELM for usage in improved wavelet neural network," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 71–79, Feb. 2017.
- [11] K. Kobayashi, "Optimal real-time pricing of electricity based on switched markov chain models and welfare maximization," in *Proc. IEEE 4th Int. Conf. Cyber-Phys. Syst. Netw. Appl.*, 2016, pp. 35–39.
- [12] H. Mosbah and M. El-Hawary, "Hourly electricity price forecasting for the next month using multilayer neural network," *Canadian J. Elect. Comput. Eng.*, vol. 39, no. 4, pp. 283–291, 2016.
- [13] A. J. Conejo, M. A. Plazas, R. Espinola, and A. B. Molina, "Day-ahead electricity price forecasting using the wavelet transform and ARIMA models," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1035–1042, May 2005.
- [14] A. Ozozen, G. Kayakutlu, M. Ketterer, and O. Kayalica, "A combined seasonal ARIMA and ANN model for improved results in electricity spot price forecasting: Case study in Turkey," in *Proc. Portland Int. Conf. Manage. Eng. Technol.*, Sep. 2016, pp. 2681–2690.
- [15] J. Portela, A. Munoz, and E. Alonso, "Forecasting functional time series with a new Hilbertian ARMAX model: Application to electricity price forecasting," *IEEE Trans. Power Syst.*, May 2017, doi: 10.1109/TPWRS.2017.2700287.
- [16] J. H. Zhao, Z. Y. Dong, and X. Li, "Electricity price forecasting with effective feature preprocessing," in *Proc. IEEE Power Eng. Soc. General Meeting*, 2006, Art. no. 8.
- [17] Z. W. Qiu, "Multivariable mutual information based feature selection for electricity price forecasting," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2012, pp. 168–173.
- [18] O. Abedinia, N. Amjadi, and H. Zareipour, "A new feature selection technique for load and price forecast of electrical power systems," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 62–74, Jan. 2017, doi: 10.1109/TPWRS.2016.2556620.
- [19] H. Qian and Z. Qiu, "Feature selection using C4.5 algorithm for electricity price prediction," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2014, pp. 175–180.
- [20] H. Mori and Y. Umezawa, "A SAX-based method for extracting features of electricity price in power markets," in *Proc. Transmiss. Distrib. Conf. Expo.: Asia Pacific*, 2009, pp. 1–4.
- [21] K. Wang, J. Yu, Y. Yu, Y. Qian, D. Zeng, S. Guo, Y. Xiang, and J. Wu, "A survey on energy internet: Architecture, approach and emerging technologies," *IEEE Syst. J.*, Jan. 2017, doi: 10.1109/JSYST.2016.2639820.
- [22] H. Jiang, K. Wang, Y. Wang, M. Gao, and Y. Zhang, "Energy big data: A survey," *IEEE Access*, vol. 4, pp. 3844–3861, Aug. 2016.
- [23] A. Shiri, M. Afshar, A. Rahimi-Kian, and B. Maham, "Electricity price forecasting using support vector machines by considering oil and natural gas price impacts," in *Proc. IEEE Int. Conf. Smart Energy Grid Eng.*, 2015, pp. 1–5.
- [24] Anamika and N. Kumar, "Market clearing price prediction using ANN in indian electricity markets," in *Proc. Int. Conf. Energy Efficient Technol. Sustainability*, Apr. 2016, pp. 454–458.
- [25] L. Tian and A. Noore, "A novel approach for short-term load forecasting using support vector machines," *Int. J. Neural Syst.*, vol. 14, no. 5, pp. 329–335, 2004.
- [26] H. Han, J. Dang, and E. Ren, "Comparative study of two uncertain support vector machines," in *Proc. IEEE Int. Conf. Adv. Comput. Intell.*, 2012, pp. 388–390.
- [27] K. Wang, X. Hu, H. Li, P. Li, D. Zeng, and S. Guo, "A survey on energy internet communications for sustainability," *IEEE Trans. Sustainable Comput.*, May 2017, doi: 10.1109/TSUSC.2017.2707122.
- [28] F. Khan, D. Kari, I. A. Karatepe, and S. S. Kozat, "Universal nonlinear regression on high dimensional data using adaptive hierarchical trees," *IEEE Trans. Big Data*, vol. 2, no. 2, pp. 175–188, Apr.-Jun. 2016.
- [29] C. P. Fung, "Manufacturing process optimization for wear property of fiber reinforced polybutylene terephthalate composite with grey relation analysis," *Wear*, vol. 254, no. 3, pp. 298–306, 2003.
- [30] J. L. Deng, "Introduction to the grey system theory," *J. Grey Syst.*, vol. 1, no. 1, pp. 1–24, 1989.
- [31] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
- [32] B. Scholkopf, A. Smola, and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [33] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proc. ACM Int. Conf. Mach. Learn.*, 2004, Art. no. 106.
- [34] Z. Du, X. Wang, L. Zheng, and Z. Zheng, "Nonlinear system modeling based on KPCA and MKSVM," in *Proc. Int. Colloq. Comput. Commun. Control Manage.*, 2009, pp. 61–64.
- [35] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 1996.
- [36] J. Li, L. Ding, and X. Ying, "Differential evolution based parameters selection for support vector machine," in *Proc. IEEE Int. Conf. Comput. Intell. Secur.*, 2013, pp. 284–288.
- [37] M. A. Alsheikh, D. Niyato, S. Lin, H.-P. Tan, and Z. Han, "Mobile big data analytics using deep learning and Apache Spark," *IEEE Netw.*, vol. 30, no. 3, pp. 22–29, May/June 2016.

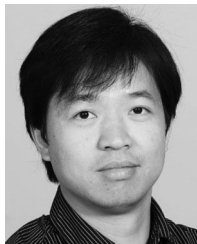


Kun Wang received the BEng and PhD degrees from the School of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2009, respectively. From 2013 to 2015, he was a postdoc fellow in the Electrical Engineering Department, University of California, Los Angeles (UCLA), California. In 2016, he was a research fellow in the School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan.

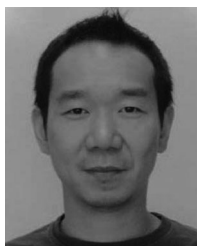
He is currently a research fellow in the Department of Computing, the Hong Kong Polytechnic University, Hong Kong, China, and also an associate professor in the School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China. He has published more than 50 papers in refereed international conferences and journals. He has received Best Paper Award at IEEE GLOBECOM'2016. He serves as associate editor of the *IEEE Access*, the *Journal of Network and Computer Applications*, the *EAI Transactions on Industrial Networks and Intelligent Systems* and editor of the *Journal of Internet Technology*. He was the symposium chair/co-chair of IEEE IECON16, IEEE EEEIC16, IEEE WCSP16, IEEE CNCC17, etc. His current research interests include mainly in the area of big data, wireless communications and networking, smart grid, energy Internet, and information security technologies. He is a member of the IEEE and the ACM.



Chenhan Xu is working toward the undergraduate degree in the School of Internet of Things, Nanjing University of Posts and Telecommunications, China. His current research interests include big data, cloud computing, and machine learning.



Yan Zhang received the PhD degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. He is full professor in the Department of Informatics, University of Oslo, Norway. He is an associate technical editor of the *IEEE Communications Magazine*, an editor of the *IEEE Transactions on Green Communications and Networking*, an editor of the *IEEE Communications Surveys and Tutorials*, an editor of the *IEEE Internet of Things Journal*, and an associate editor of the *IEEE Access*. He serves as chair positions in a number of conferences, including IEEE GLOBECOM 2017, IEEE VTC-Spring 2017, IEEE PIMRC 2016, IEEE CloudCom 2016, IEEE ICC 2016, IEEE CCNC 2016, IEEE SmartGridComm 2015, and IEEE CloudCom 2015. He serves as TPC member for numerous international conference including IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, and IEEE WCNC. His current research interests include: next-generation wireless networks leading to 5G, green and secure cyber-physical systems (e.g., smart grid, healthcare, and transport). He is IEEE VTS (Vehicular Technology Society) Distinguished Lecturer. He is also a senior member of the IEEE, the IEEE ComSoc, the IEEE CS, the IEEE PES, and the IEEE VT society, and a fellow of the IET.



Song Guo received the PhD degree in computer science from the University of Ottawa and was a full professor with the University of Aizu, Japan. He is a full professor in the Department of Computing, The Hong Kong Polytechnic University. His research interests include mainly in the areas of big data, cloud computing, green communication and computing, wireless networks, and cyber-physical systems. He has published more than 350 conference and journal papers in these areas and received five best paper awards from IEEE/ACM conferences. He has served in editorial boards of several prestigious journals, including the *IEEE Transactions on Parallel and Distributed Systems*, the *IEEE Transactions on Emerging Topics in Computing*, the *IEEE Transactions on Sustainable Computing*, the *IEEE Transactions on Green Communications and Networking*, and the *IEEE Communications*. He is an active volunteer as General/TPC Chair for more than 20 international conferences and chair/vice-chair for several IEEE Technical Committees and SIGs. He is a senior member of the IEEE, the ACM, and an IEEE Communications Society Distinguished Lecturer.



Albert Y. Zomaya is currently the chair professor of high performance computing and networking in the School of Information Technologies, The University of Sydney. He is also the director of the Centre for Distributed and High Performance Computing. He published more than 500 scientific papers and articles and is author, co-author or editor of more than 20 books. He is the editor in chief of the *IEEE Transactions on Sustainable Computing* and the Springer's *Scalable Computing* and serves as an associate editor for 22 leading journals, such as, the *ACM Computing Surveys* and the *Journal of Parallel and Distributed Computing*. He is the recipient of the IEEE Technical Committee on Parallel Processing Outstanding Service Award (2011), the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing (2011), and the IEEE Computer Society Technical Achievement Award (2014). Professor Zomaya's research interests include the areas of parallel and distributed computing and complex systems. He is a Chartered Engineer, a fellow of the AAAS, the IEEE, and the IET (UK).

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.