Wind Power Pattern Forecasting Based on Projected Clustering and Classification Methods

Heon Gyu Lee, Minghao Piao, and Yong Ho Shin

A model that precisely forecasts how much wind power is generated is critical for making decisions on power generation and infrastructure updates. Existing studies have estimated wind power from wind speed using forecasting models such as ANFIS, SMO, *k*-NN, and ANN. This study applies a projected clustering technique to identify wind power patterns of wind turbines; profiles the resulting characteristics; and defines hourly and daily power patterns using wind power data collected over a year-long period. A wind power pattern prediction stage uses a time interval feature that is essential for producing representative patterns through a projected clustering technique along with the existing temperature and wind direction from the classifier input. During this stage, this feature is applied to the wind speed, which is the most significant input of a forecasting model. As the test results show, nine hourly power patterns and seven daily power patterns are produced with respect to the Korean wind turbines used in this study. As a result of forecasting the hourly and daily power patterns using the temperature, wind direction, and time interval features for the wind speed, the ANFIS and SMO models show an excellent performance.

Keywords: Renewable energy, wind energy, wind power pattern forecasting, projected clustering, classification.

I. Introduction

Wind power is the generation of electric power using mechanical energy converted through a wind turbine [1]. A wind generator can theoretically convert 59.3% of wind energy at maximum, but it is only practically possible to convert from 20% to 40% of wind energy owing to existent loss factors such as the wing shape, mechanical friction, and generator efficiency. In addition, the production level of electric power generated from wind is irregular and sometimes cannot meet the required power supply. To make matters worse, the power may change on a large scale. Since each problem is caused from the wind power source, it is very hard to forecast an accurate quantity of wind power generation. An accurate analysis model for predicting wind power generation can reduce the cost needed to maintain the equilibrium of supply and demand of electric power and help in the decision-making of timely infrastructure updates for the wind power industry. Therefore, more exact prediction techniques of power patterns are indispensable for the efficient operation and planning of wind power generation, and mathematical methods such as data mining are used in such prediction techniques [2]–[3].

In general, wind power generation prediction builds a power pattern model from related data and forecasts power patterns by applying a built model [4]. A variety of prediction models for wind power generation use wind speed as input data. At the learning stage of the prediction model, the relation between wind speed and wind power is learned, and we can predict the amount of wind power for a specified wind speed. The difference between the predicted wind power and actual measurement becomes the prediction error. There have been many comparative studies on predicting wind power based on wind speed. Li and others [5] applied regression and an

Manuscript received Aug. 4, 2014; revised Jan. 22, 2015; accepted Feb. 3, 2015.

This work was supported by the Postal Technology R&D program of MSIP (10039146, Development of Implementation Technology for SMART Post).

Heon Gyu Lee (hg_lee@etri.re.kr) is with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

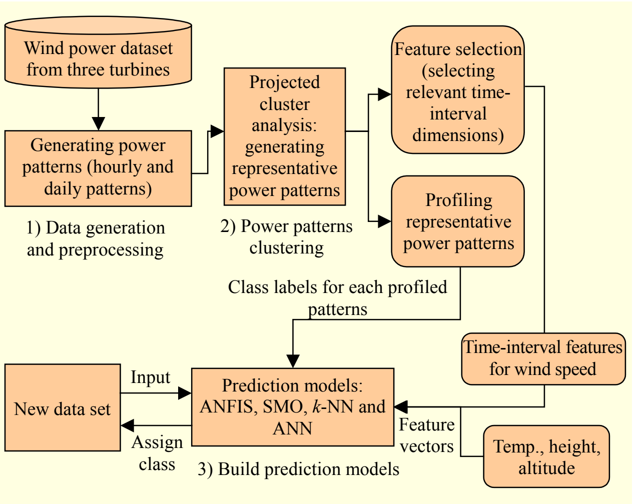
Minghao Piao (bluemhp@cbnu.ac.kr) is with the Database Laboratory, Chungbuk National University, Cheongju, Rep. of Korea.

Yong Ho Shin (corresponding author, yhshin@ynu.ac.kr) is with the School of Business, Yeungnam University, Gyeongsan, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.15.2314.0070

artificial neural network (ANN) model and showed that the ANN performed better than regression. Üstüntas and Sahin [6] estimated the power curve using a cluster center fuzzy logic (CCFL) model, which demonstrated the lowest prediction errors. Kusiak and others [2] suggested five non-parametric models for monitoring wind farm power; that is, a neural network (NN), M5 tree, representative tree, bagging tree, and *k*-nearest neighbor (*k*-NN), the last of which showed the best performance. Unlike existing studies, Schlechtingen and others [7] added ambient temperature and wind direction in addition to wind speed as prediction model inputs, as well as applying four data mining techniques. The applied prediction models were CCFL, NNs, *k*-NN, and an adaptive neuro-fuzzy interference system (ANFIS). Their test results showed that the prediction model errors were reduced when the temperature and wind direction were applied instead of the wind speed only. When the four prediction models were compared with each other, ANFIS showed the lowest prediction errors and the possibility of early detection of abnormal power output. Rahmani and others [8] applied a hybrid technique of ant colony optimization and particle swarm optimization on wind speed and environment temperature for short-term wind energy forecasting. The mean absolute percentage error was used to assess the accuracy of the model. As a result of the aforementioned related studies, it can be summarized that ANN, *k*-NN, and ANFIS are good techniques for power generation and power curve estimations. However, existing studies applying these techniques did not analyze generation patterns varying with time (hour, day, and month). They estimated the power generation output (kW) with respect to wind speed input (m/s) by building a power curve based on a prediction model. This is the main difference between them and our suggested model, of which this study forecasts power generation for each time stamp. Moreover, since they only estimate the power output based on wind speed, temperature, and direction, there exists a limitation in that most of them are only short-term forecasting models. Recently, Azad and others [9] proposed statistical-based and NN-based approaches to predict the hourly wind speed data of the subsequent year in their long-term wind speed forecasting study. The proposed approaches exhibited small error rates, with values occurring in the range 0.8 m/s to 0.9 m/s. Even though wind speed and output power of a wind generator have a proportional relation (Schlechtingen [7] insisted that temperature and wind direction affect wind power), our study predicts not wind power but long-term hourly wind speed.

Wind power patterns have significantly varying power generation depending on the season and time. Therefore, it is necessary to create accurate wind power patterns under different time conditions and to analyze their characteristics.



Wind power dataset from three turbines
Generating power patterns (hourly and daily patterns)
1) Data generation and preprocessing
Projected cluster analysis: generating representative power patterns
2) Power patterns clustering
Feature selection (selecting relevant time-interval dimensions)
Profiling representative power patterns
Class labels for each profiled patterns
Time-interval features for wind speed
New data set
Input
Assign class
Prediction models: ANFIS, SMO, k-NN and ANN
Feature vectors
Temp., height, altitude
3) Build prediction models

Fig. 1. Wind power pattern profiling and forecasting framework.

For this study, we generate wind power patterns of wind turbines and use a projected clustering technique to profile the resulting characteristics. Traditional clustering algorithms discover clusters using all data dimensions. Most of the time, however, time-series data, such as power patterns, are characterized based on time-interval features in the subset of the dimensions. As the number of dimensions increases, distance measures of the clustering algorithms become more and more meaningless [10]. Thus, the application of projected clustering methods allows selecting of the cluster composition of similar power patterns and feature vectors that are the subset of all dimensions used for that cluster composition.

Figure 1 shows the framework of wind power pattern profiling and forecasting suggested in this study. From Fig. 1, the following steps can be identified:

1. Data generation and preprocessing.
   a) The recorded wind power value based on time of use of the wind turbines is calculated using the values of different time granularities, such as hour and day.
   b) Temperature, wind speed, and direction data are induced to generate the training and testing datasets; the wind speed is the value measured at the same time as the wind power.
2. Power patterns clustering.
   a) From the wind power patterns of the wind turbines, the patterns with high similarity are grouped through a projected cluster analysis.
   b) Representative power patterns are then created from each group, and class labels are created for the groups.
3. Building the prediction models.
   a) As a result of the cluster analysis, we can generate the wind speed values corresponding to the time intervals by detecting those intervals that belong to subsets of the time dimensions applied to the algorithms.

b) These time-interval features for the wind speed are employed as inputs for prediction model learning along with the temperature and wind direction.

c) As the output of the prediction model, class labels corresponding to the representative power patterns of each group are assigned to the new data.

This study has three contributions. Firstly, representative power patterns of wind turbines operated in Korea are found, and these features are then profiled by subspace projection methods. Secondly, this study proposes a way to select task-relevant features instead of wind speed in a whole time dimension (selecting time-interval features, temperature, and wind direction through subspace discovery over a whole time dimension) so as to build a highly accurate prediction model. Finally, we apply the existing techniques in the wind power prediction research and suggest the most appropriate method, upon assessment, for predicting the profiled representative power patterns.

The rest of this paper is organized as follows. In Section II, we review previous studies of projected clustering approaches and point out their characteristics. In Section III, we introduce state-of-the-art classifiers for predicting power patterns. In Section IV, we present the experimental results and discuss issues. Finally, in Section V, we provide some concluding remarks.

II. Projected Clustering Methods

In this study, we use projected clustering approaches for discovering representative power patterns. Projected clustering is a method for detecting clusters with the highest similarity from the subsets of all data dimensions. The biggest difference between projected clustering and traditional clustering methods is that in a projected clustering approach, the detection of various subsets is carried out based on the fact that subsets differ from each other and that they include meaningful clusters, rather than considering all dimensions given during the clustering process [11]. For example, if time is a dimension and power generation is an object in the data where generated wind power values are recorded, through the projected clustering shown in Fig. 2, then time intervals (t_7 through t_{15}) can be detected, which are subsets having discriminating power among different clusters. Projected clustering approaches can be divided into three paradigms based on the detection methods of the subsets [12].

The first paradigm is to divide a data space into grid-cells (cell-based) and form clusters of sufficient density from the cells. The basic concept is to first define grid-cell sets before assigning objects to suitable cells and to then calculate the density of each cell. Next, cells with a density of a certain threshold or lower are removed and clusters are built from a series of cells with high density. A popular cell-based method, CLIQUE [13], is a grid-based clustering algorithm that detects clusters of subsets through certain procedures. When multi-dimensional data points of large capacity are given, the data space, in general, is not uniformly occupied by the data points. The clustering of this approach discriminates sparse and crowded regions in space (or unit) and detects the entire distribution type of the datasets. Clustering of CLIQUE is defined as the biggest group of connected dense units. SCHISM [14] finds subsets by using a "support" and Chernoff–Hoeffding bound concept and determines the interesting subsets using a depth-first search and backtracking.

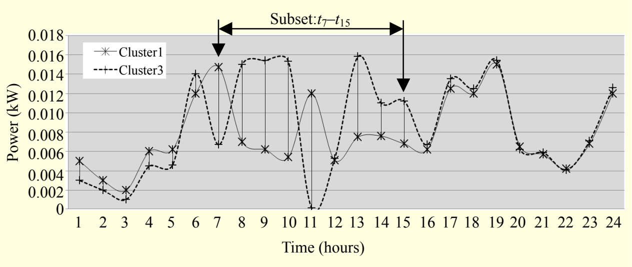


Fig. 2. Different power patterns of different clusters.

The second paradigm is density-based projected clustering, which utilizes an algorithm to identify clusters. In this algorithm, a cluster is defined to be a dense area (that is, a group of points that are closely packed together, whereby each point has many nearby neighbors) separated by sparsely populated areas (that is, low-density areas). Though the overall clustering concept is based on DBSCAN [15], the density calculation here considers only the relevant dimensions. The representative algorithm is FIRES [16], and it applies an efficient filter-refinement method. Above all, the existing base-clusters are created, and those that fail to meet the given density conditions are removed in the filtering stage. Next, the base-clusters are merged to create the maximal dimensional projected cluster approximations. Lastly, the final refined clusters are built during the refinement stage. The SUBCLU [17] is a DBSCAN-based greedy algorithm for projected clustering. Unlike grid-based approaches, it can detect clusters with arbitrary shapes.

The third paradigm is a clustering-oriented approach. As the data dimension increases in the clustering for high-dimensional data, clustering that considers all dimensions can hinder the performance remarkably owing to the presence of sparse data. PROCLUS [18], a famous algorithm, starts from a single dimensional space. Instead, the algorithm of the third paradigm begins by searching the initial estimation regarding clusters in a high-dimensional space. Weight is provided for each cluster per each dimension, and the renewed weight is used to create

Table 1. Properties of the three paradigms.

Paradigm	Algorithm	Properties
Cell-based	CLIQUE	Fixed threshold and grid size, pruning by monotonicity property
	SCHISM	Enhanced CLIQUE by variable threshold, using heuristics for pruning
Density-based	FIRES	Variable density threshold, based on filter-refinement architecture to drop irrelevant base-clusters
	SUBCLU	Fixed density threshold, pruning by monotonicity property
Clustering-oriented	PROCLUS	Fixed cluster number, iteratively improving result like *k*-means, partitioning
	STATPC	Statistical tests, reducing result size by redundancy elimination

clusters again for the next iteration. STATPC [19] detects relevant subsets based on objects and builds candidate subspaces, which are refined to build local optimal projected clusters. Finally a greedy search algorithm is used to review all subspaces and build optimal clusters.

Table 1 shows the properties of the clustering algorithms used in our study (all properties of the clustering algorithms are stated in [12]). The important parameter settings and performance evaluation results of the algorithms are described in detail in Section IV.

III. Classification Model for Predicting Representative Power Patterns

Feature vectors for the classifier's supervised learning include prior information such as the temperature, wind direction, and wind speed, which are time-interval features; class labels are representative power patterns built through clustering. Among all the features used for the supervised learning, the wind speed affects the power output the most, while the wind speed and wind power are measured for the same hour. Therefore, the model considers only the wind speed values, which is a time-dimensional subset (time-intervals features) selected during the clustering stage. This has the effect of relevant feature selection and a dimensionality reduction to build accurate and fast classifiers.

For instance, Fig. 3 describes the time dimension involved in building three clusters, C_0, C_1, and C_2. Time intervals (f_1, f_2, f_3, f_4); that is, the subsets of all time dimensions that are applied to the clustering of the wind power patterns, are applied to the wind speed equally, and only the wind speed value corresponding to these time intervals is drawn as a feature vector. The classifiers used in the study are the sequential minimal optimization (SMO) algorithm, which shows an excellent performance, and AFNIS, *k*-NN, and ANN, which were all evaluated in related papers [8]–[9].

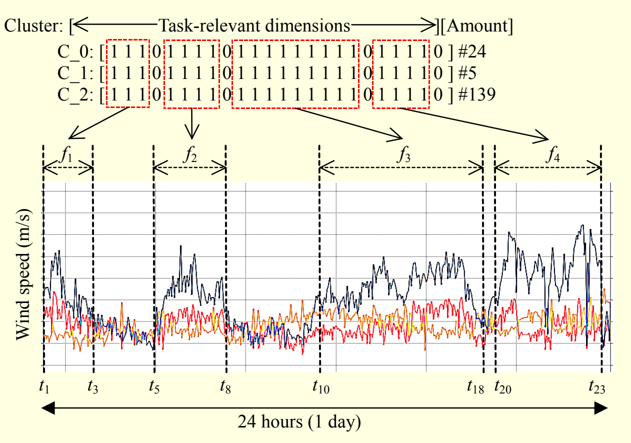


Fig. 3. Example of time-interval features for wind speed.

1. ANFIS

ANFIS [20] is a kind of ANN based on a Takagi–Sugeno fuzzy inference system. Since it integrates both NNs and fuzzy logic principles, it has the potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have the learning capability to approximate nonlinear functions. As wind power prediction includes the uncertainty of the input/output variables, power generation is determined through learning. Therefore, employing ANFIS can help with the accuracy of prediction, as modeling a prediction system mathematically is difficult, and the nonlinearity is contained.

Figure 4 shows the ANFIS structure utilized in this study, and the layer properties and learning procedure are as follows:

- *Layer 1.* A given node, *i*, has $O_{1,i} = \mu_{A_i}(x)$, $i = 1, 2$, and $O_{1,i} = \mu_{B_{i-2}}(y)$, $i = 3, 4$, where x is the input value of node *i*, A_i indicates the fuzzy set related to the function of the node, and $O_{1,i}$ is the membership function that represents the membership degree of the input value x for A_i; and μ_{A_i} is put into (1) in various ways and can be written as (2) through a parameter adjustment.

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}}, \tag{1}$$

$$\mu_{A_i}(x) = \exp\left[-\left(\frac{x - c_i}{a_i}\right)^2\right]. \tag{2}$$

- *Layer 2.* A T-norm computation is conducted, and each

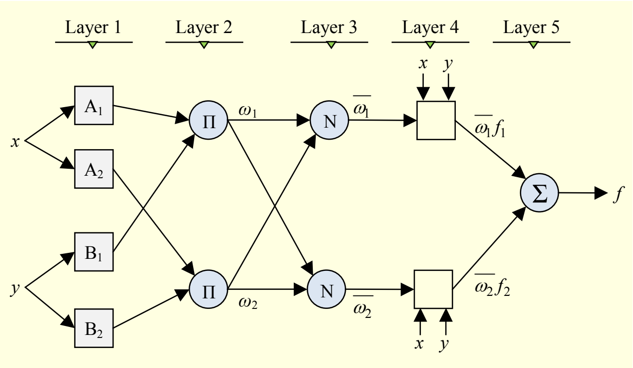


Fig. 4. Structure of ANFIS.

membership function is multiplied and presented as (3) below.

$$O_{2,i} = \omega_i = \mu_{B_i}(y) \quad \text{for } i = 1, 2. \qquad (3)$$

▪ *Layer 3*. The *i* rule is normalized to perform computations such as (4) below.

$$O_{3,i} = \overline{\omega_i} = \frac{\omega_i}{\omega_1 + \omega_2} \quad \text{for } i = 1, 2. \qquad (4)$$

▪ *Layer 4*. The output function of each rule is multiplied by the compatibility obtained from layer 3, such as (5). The parameters p_i, q_i, and r_i are determined in such a way as to minimize errors.

$$O_{4,i} = \overline{\omega_i} f_i = \overline{\omega_i}(p_i x + q_i y + r_i) \quad \text{for } i = 1, 2. \qquad (5)$$

▪ *Layer 5*. The output is calculated through the above process.

$$O_{5,i} = \sum \overline{\omega_i} f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i}. \qquad (6)$$

2. SVM by SMO

The SMO algorithm [21] is appropriate to realize the optimization of the support vector machine (SVM), which is offered by a different normalization value to the class for imbalanced learning. SMO is an algorithm for solving the quadratic programming problem that arises during the training of support vector machines and is widely used for training such machines. The learning stage of the SMO algorithm detects an optimal hyperplane using training data and classifies it using test data. Although the SMO algorithm provides the Poly and radial basis function (RBF) kernels, the RBF kernel is generally used in many cases for the following reasons. The RBF kernel can handle nonlinear relationships between classes and attributes and has fewer hyper-parameters that influence the complexity of the model selection than the Poly kernel. The RBF kernel function is as follows:

$$K(x, x') = \exp\left(-\frac{\| x - x' \|^2}{\sigma^2}\right). \qquad (7)$$

For the test, the RBF kernel was chosen as the kernel function. A grid-search approach using a 10-fold cross validation was carried out to determine the optimal value for each dataset of parameters C and γ, and as a result, the parameter range was determined as $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ to $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$.

3. *k*-NN

k-NN is a basic instance-based learner that finds the training instance closest in Euclidean distance to the given test instance and predicts the same class as this training instance [22]. This paper used the IBk algorithm; that is, the *k*-NN classifier, which is provided by Java WEKA [23]. The number of nearest neighbors can be specified explicitly in an object editor or determined automatically using a leave-one-out cross validation, subject to an upper limit given by the specified value. The predictions from more than one neighbor can be weighted according to their distance from the test instance, and two different formulas are implemented for converting the distance into a weight. The number of training instances maintained by the classifier can be restricted by setting the window size option. As new training instances are added, the oldest ones are removed to maintain the number of training instances at this size. Parameter *k* is attested by setting a default value of 1 to each class.

4. ANN

ANN is useful to consider complicated nonlinearity, while a multilayer perceptron (MLP) NN is currently utilized for time-series forecasting. An ANN model consists of learning, parameter coordination, verification, and forecasting steps. At the learning step, the structure of the NN is determined by learning the nonlinear relationship between input and output variables using the backpropagation algorithm. The verification stage attempts to predict using the structure determined by learning and minimizes the error with ANN model learning. The accuracy of forecasted wind power patterns is verified by analyzing the performance error with mean absolute error (MAE). In the study, an MLP model provided by Java WEKA [23] was used. The nodes in this network are all sigmoid (except for when the class is numeric, in which case the output nodes become *unthreshold* linear units).

IV. Experimental Results and Discussion

1. Datasets and Data Preprocessing

Wind power generation data were collected from three wind

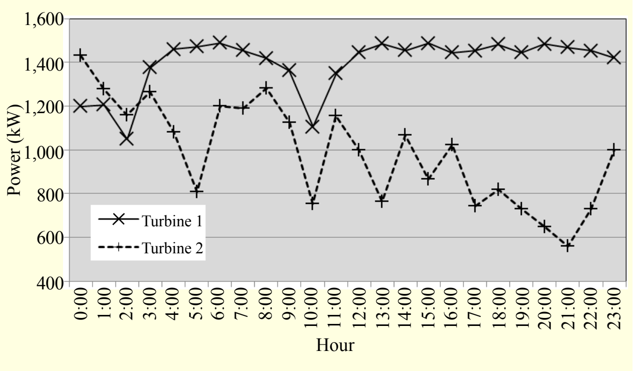


Fig. 5. Hourly power patterns of two turbines.

Table 2. Extracted features from raw data.

Feature	Description
Timestamp	<year / month / day / hour>
Wind turbine ID	Turbine ID code
Wind power	Wind power generation measured by hour
Wind speed	Wind speed measured by hour
Ambient temp.	Ambient temperature of turbines measured
Wind direction	Wind direction measured

turbines with different regional characteristics in Korea for a year throughout the four seasons in 2010. Two of them (WT1 and WT2) were operated on land, and the other on an island (WT3). As ten minutes of saved wind power can be changed into a value of various time granularities (day, week, and month), wind power patterns of different time units can be built.

Definition 1. A time schema (TS) is defined as the time granularity and its domain. The form of the schema is as follows:

$$\mathrm{TS} = (G : D), \tag{8}$$

where G is a time granularity, and D is the domain value of time granularity G as a set of positive integer numbers. In the case of TS = (day:{1–30}), <20> is valid for expressing the twentieth day, whereas <31> is not valid.

Definition 2. The power pattern of each turbine can be described as below for the given TS

$$p^w_{\mathrm{TS}} = \{p^w_1, \dots, p^w_{|D|}\}, \tag{9}$$

where w is the turbine identifier used to measure the power generation, and p^w is the total power generation during the given TS. If TS = (hour:{1–24}), then the power patterns are illustrated using hour units and one-day power patterns that have a total of 24 dimensions. Figure 5 shows an example of hourly power patterns.

Table 3. Data distribution after data preprocessing.

Data type	Wind turbine ID	Training (80%)	Testing (20%)	Total
Type 1: TS = (hour:{1–24})	WT1	5,792	1,448	7,240
	WT2	5,768	1,442	7,210
	WT3	4,832	1,208	6,040
Type 2: TS = (day:{1–30})	WT1	285	71	356
	WT2	280	70	350
	WT3	216	54	270

The above hourly power patterns show the change in power generation of two turbines for a day-long period. The wind energy data used in this paper are shown in Table 2, while the distribution of the training dataset and test dataset preprocessed by the two types of time units is shown in Table 3.

2. Projected Clustering for Profiling Power Patterns

Building representative power patterns through a clustering analysis of the wind power patterns can characterize the change in wind power generation patterns depending on the time of objects inside the clusters. The cluster analysis stage describes the cluster analysis results of only SCHISM, FIRES, and PROCLUS, which showed excellent performance among the six algorithms introduced in Table 1. The projected clustering algorithm uses Java WEKA's [24] OpenSubpace, which is a data mining tool. OpenSubspace supports up-to-date performance evaluators to facilitate studies on projected clustering. For unsupervised learning methods such as clustering, it is difficult to provide appropriate parameter settings without prior knowledge on the data. In the case of the *k*-means algorithm, for instance, users face a difficulty in defining the appropriate number of clusters in advance. OpenSubspace supports parameter bracketing for selecting the most appropriate parameter values. However, as most parameters are supposed to be set as a range and not as a specific value, more repeated works are needed than in traditional methods to determine the optimal range. For example, the PROCLUS algorithm uses parameter C as the number of clusters and parameter D as the number of dimensions. If the range of these two parameters is set at C = {1–5}, D = {5–8} by the user, then a total of 20 individual results need to be analyzed to find the optimal parameter pair of (C, D). The optimal parameter settings of each algorithm for the two types of data corresponding to the hour and day in the test are given in Tables 4 and 5, respectively.

Because the projected clustering algorithm groups similar power generation patterns for the time dimensions in the

http://dx.doi.org/10.4218/etrij.15.2314.0070

Table 4. Parameter bracketing for Type 1 (TS = (hour:{1–24})).

Algorithm	Parameter	From	Offset	Op	Steps	To
Cell-based (SCHISM)	TAU	0.1	0.1	+	10	1.0
	XI	1	1	+	24	24
	U	0.05	0	+	1	0.05
	Total number of experiments: 240 (steps: 10×24×1)					
Density-based (FIRES)	BASE_DBSCAN_EPSILON	1.0	0	+	1	1.0
	BASE_DBSCAN_MINPTS	100	0	+	1	100
	GRAPH_K	15	1	+	4	18
	GRAPH_MIN CLU	1	1	+	4	4
	GRAPH_MU	1	1	+	4	4
	GRAPH_SPLIT	0.66	0	+	1	0.66
	POST_DBSCAN_EPSILON	300	0	+	1	300
	POST_DBSCAN_MINPTS	24	0	+	1	24
	PRE_MINIMUM PERCENT	10	00	+	1	10
	Total number of experiments: 64 (steps: 1×1×4×4×4×1×1×1×1)					
Clustering-oriented (PROCLUS)	average Demensions	1	1	+	24	24
	numberOfClusters	2	1	+	8	9
	Total number of experiments: 192 (steps: 24 × 18)					

Table 5. Parameter bracketing for Type 2 (TS = (day:{1–30})).

Algorithm	Parameter	From	Offset	Op	Steps	To
Cell-based (SCHISM)	TAU	0.1	0.1	+	10	1.0
	XI	1	1	+	30	30
	U	0.05	0	+	1	0.05
	Total number of experiments: 300 (steps: 10×30×1)					
Density-based (FIRES)	BASE_DBSCAN_EPSILON	1.0	0	+	1	1.0
	BASE_DBSCAN_MINPTS	100	0	+	1	100
	GRAPH_K	15	1	+	4	18
	GRAPH_MIN CLU	1	1	+	4	4
	GRAPH_MU	1	1	+	4	4
	GRAPH_SPLIT	0.66	0	+	1	0.66
	POST_DBSCAN_EPSILON	300	0	+	1	300
	POST_DBSCAN_MINPTS	30	0	+	1	30
	PRE_MINIMUM PERCENT	10	0	+	1	10
	Total number of experiments: 64 (steps: 1×1×4×4×4×1×1×1×1)					
Clustering-oriented (PROCLUS)	average Demensions	1	1	+	30	30
	numberOfClusters	2	1	+	9	10
	Total number of experiments: 270 (steps: 30 × 9)					

training datasets and classifies which group the test data objects belong to out of the defined clusters, it includes the clustering and classification methods together. Therefore, an evaluation measure such as the sum of the squared error or normalized mutual information [25] for a traditional clustering method is inappropriate. The present study used evaluation measures such as the precision, recall, F_1-value, and accuracy to evaluate the three clustering algorithms. Formal definitions of these measures are given below.

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP})\,, \tag{10}$$

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN})\,, \tag{11}$$

$$F_1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}\,, \tag{12}$$

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}\,. \tag{13}$$

In (10)–(13), we use the following abbreviations: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

All tests for the three clustering measures use 10-fold cross validation. In addition, the original classes used to evaluate the algorithms in the cluster analysis stage are the wind turbine IDs (WT1, WT2, WT3) of the three regions. Table 6 shows the result of the evaluators based on the clustering methods for the two datasets.

In both data types, the test results show that the PROCLUS method, which is a clustering-oriented approach, achieves a good performance. In addition, Fig. 6 shows an accuracy comparison between SCHISM, FIRES, and PROCLUS and illustrates that PROCLUS outperforms the other clustering methods.

Based on the results in Fig. 4, the PROCLUS algorithm was used, and the clustering results using different time units are presented in Table 7. PROCLUS has two parameters; that is, averageDimension-and numberOfClusters. Data type 1 was applied using fourteen time dimensions (hour) depending on the parameter bracketing set using the range shown in Table 4,

Table 6. Description of the summary results.

Data	Algorithm	Precision	Recall	F_1	Class
TS = (hour: {1–24})	Cell-based (SCHISM)	0.748	0.873	0.805	WT1
		0.688	0.550	0.611	WT2
		0.647	0.423	0.512	WT3
	Density-based (FIRES)	0.762	0.912	0.830	WT1
		0.655	0.475	0.551	WT2
		0.824	0.538	0.651	WT3
	Clustering-oriented (PROCLUS)	**0.809**	**0.873**	**0.881**	**WT1**
		0.769	**0.750**	**0.937**	**WT2**
		0.579	**0.423**	**0.900**	**WT3**
TS = (day: {1–30})	Cell-based (SCHISM)	0.748	0.873	0.805	WT1
		0.688	0.550	0.611	WT2
		0.647	0.423	0.512	WT3
	Density-based (FIRES)	0.787	0.833	0.810	WT1
		0.694	0.625	0.658	WT2
		0.500	0.462	0.480	WT3
	Clustering-oriented (PROCLUS)	**0.88**	**0.863**	**0.871**	**WT1**
		0.882	**0.925**	**0.871**	**WT2**
		0.565	**0.500**	**0.531**	**WT3**

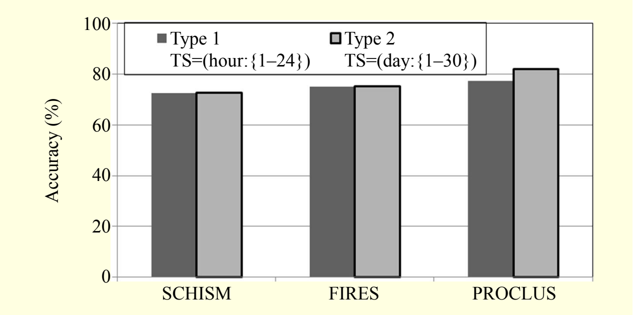


Fig. 6. Accuracy comparison between SCHISM, FIRES, and PROCLUS for two different datasets.

Table 7. Confusion matrix of original groups vs. discovered clusters.

Data type	Parameter	Discovered clusters				
Type 1: TS = (hour: {1–24})	D=14, C=5	C1	C2	C3	C4	C5
	WT1	5	455	1053	4481	0
	WT2	4801	115	0	807	45
	WT3	560	3517	9	743	1
Type 2: TS = (day: {1–30})	D=9, C=4	C1	C2	C3	C4	—
	WT1	234	50	0	1	
	WT2	0	32	83	180	
	WT3	0	0	51	229	

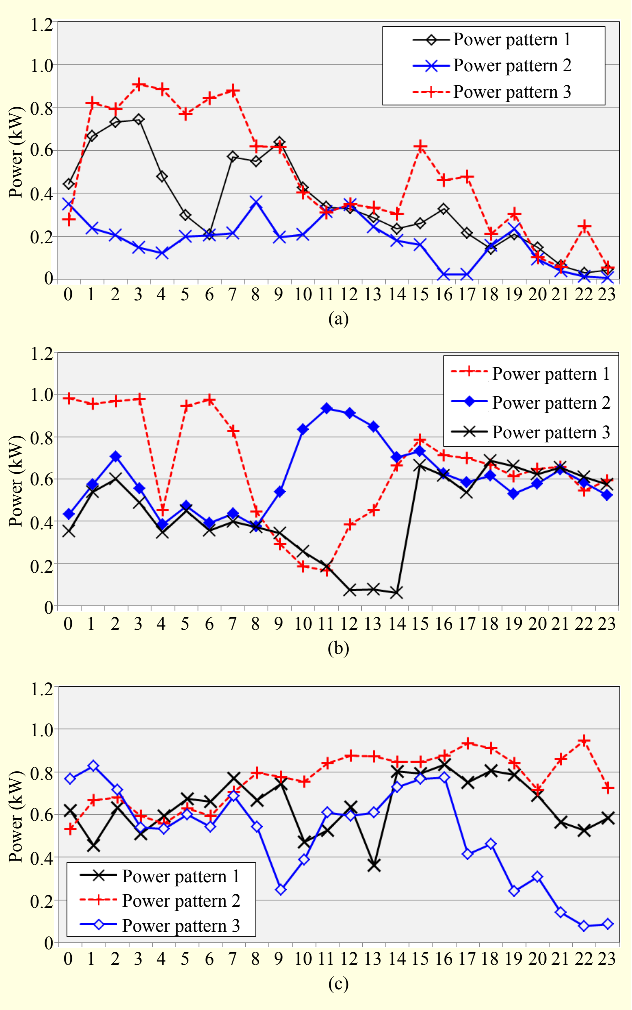


Fig. 7. Hourly representative power patterns: (a) normalized power patterns for WT1 dataset, (b) normalized power patterns for WT2 dataset, and (c) normalized power patterns for WT3 dataset.

and the optimal number of clusters was determined to be five for each wind turbine, while data type 2 was applied using nine time dimensions (day) and four clusters for each wind turbine.

For WT1, WT2, and WT3, the original group in Table 7, specific clusters detected from the PROCLUS analysis results are profiled as representative power patterns. The profiling method is simple — the representative patterns are built by calculating the mean value of the power patterns included in the specific groups.

Figure 7 shows the representative power patterns by clusters after the application of the projected clustering algorithm for hourly preprocessed data for the three turbines. Three representative power patterns were built for the original groups, WT1, WT2, and WT3. Although clustering was conducted through the algorithms, groups with a relatively small number of members are excluded from the profiling. For example, the

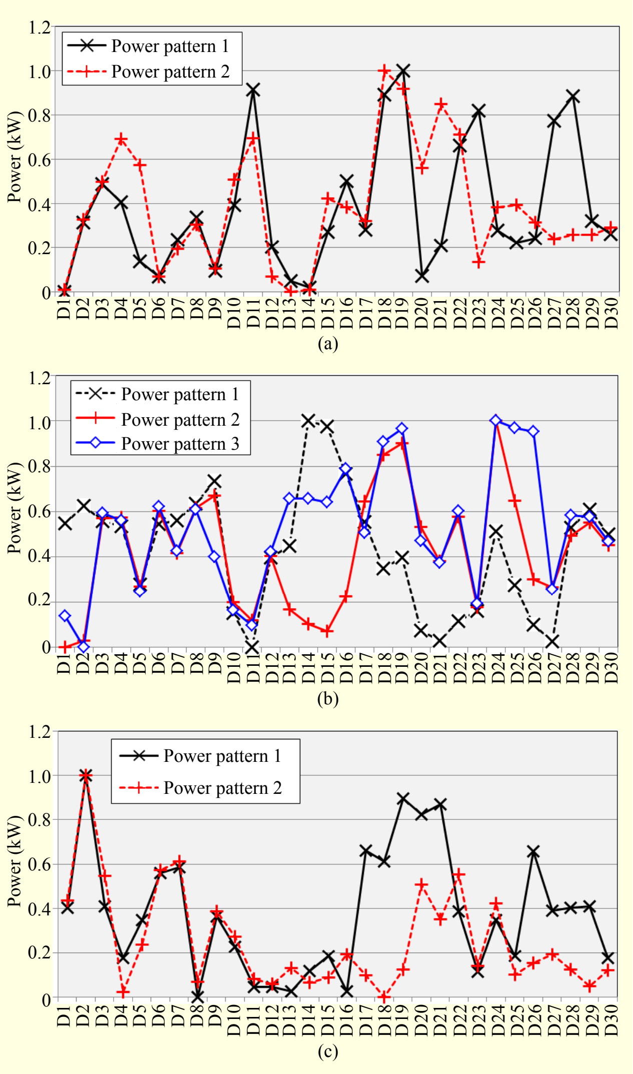


Fig. 8. Daily representative power patterns: (a) normalized power patterns for WT1 dataset, (b) normalized power patterns for WT2 dataset, and (c) normalized power patterns for WT3 dataset.

WT1 dataset of type 1 in Table 7 comprises four clusters, C1 through C4. However, C1 is considered as a minor pattern or outlier because there are only five members in C1; hence, it can be removed. WT2's C5 and WT3's C3 and C5 are also removed for the same reason; thus, nine clusters are built for the representative patterns of type 1. Two daily representative power patterns are built from WT1 and WT3 each, and three from WT2, as shown in Fig. 8. New specific class labels are built for the three turbines of the original groups. For example, as WT1 comprises two different clusters in the daily power pattern analysis, WT1 is segmented into WT1-1 and WT1-2. The same method can be applied to WT2, for example, WT2-1, WT2-2, and WT2-3. Therefore, there are seven groups in total to predict the daily power patterns and nine groups to predict the hourly power patterns. These specific groups are considered as class labels for supervised learning in the classification stage.

Table 8. Description of summary results for hourly power patterns.

Data	Classifier	Precision	Recall	F_1	Class
TS = (hour: {1–24})	ANFIS	0.824	0.933	0.875	WT1-1
		0.783	0.783	0.783	WT1-2
		0.813	0.765	0.788	WT1-3
		0.870	0.800	0.833	WT2-1
		0.765	0.929	0.839	WT2-2
		0.813	0.897	0.852	WT2-3
		0.857	0.750	0.800	WT3-1
		0.714	0.714	0.714	WT3-2
		1.000	0.800	0.889	WT3-3
	SMO	0.857	0.800	0.828	WT1-1
		0.630	0.739	0.680	WT1-2
		0.778	0.824	0.800	WT1-3
		0.704	0.760	0.731	WT2-1
		0.786	0.786	0.786	WT2-2
		0.769	0.690	0.727	WT2-3
		0.824	0.875	0.848	WT3-1
		0.636	0.500	0.560	WT3-2
		0.929	0.867	0.897	WT3-3
	IBk	0.917	0.733	0.815	WT1-1
		0.556	0.652	0.600	WT1-2
		0.778	0.824	0.800	WT1-3
		0.833	0.600	0.698	WT2-1
		0.714	0.714	0.714	WT2-2
		0.758	0.862	0.806	WT2-3
		0.647	0.688	0.667	WT3-1
		0.643	0.643	0.643	WT3-2
		0.867	0.867	0.867	WT3-3
	MLP	0.813	0.867	0.839	WT1-1
		0.552	0.696	0.615	WT1-2
		0.867	0.765	0.813	WT1-3
		0.727	0.640	0.681	WT2-1
		0.857	0.857	0.857	WT2-2
		0.818	0.931	0.871	WT2-3
		0.938	0.938	0.938	WT3-1
		0.727	0.571	0.640	WT3-2
		1.000	0.800	0.889	WT3-3

Since the representative power patterns (through projected clustering methods) found in this study were calculated from

Table 9. Description of summary results for daily power patterns.

Data	classifier	Precision	Recall	F_1	Class
TS = (day: {1–30})	ANFIS	0.945	0.960	0.952	WT1-1
		1.000	1.000	1.000	WT1-2
		0.778	0.961	0.817	WT2-1
		0.860	0.836	0.848	WT2-2
		0.779	0.698	0.736	WT2-3
		0.926	0.936	0.931	WT3-1
		0.992	0.992	0.992	WT3-2
	SMO	0.932	0.984	0.957	WT1-1
		1.000	1.000	1.000	WT1-2
		0.870	0.934	0.901	WT2-1
		0.914	0.873	0.893	WT2-2
		0.874	0.825	0.849	WT2-3
		0.946	0.936	0.941	WT3-1
		1.000	0.976	0.988	WT3-2
	IBk	0.953	0.976	0.964	WT1-1
		1.000	0.991	0.995	WT1-2
		0.656	0.672	0.664	WT2-1
		0.866	0.882	0.874	WT2-2
		0.594	0.603	0.598	WT2-3
		0.989	0.915	0.950	WT3-1
		0.983	0.967	0.975	WT3-2
	MLP	0.946	0.984	0.965	WT1-1
		0.973	1.000	0.987	WT1-2
		0.886	0.893	0.890	WT2-1
		0.896	0.864	0.880	WT2-2
		0.837	0.817	0.827	WT2-3
		0.969	0.989	0.979	WT3-1
		0.992	0.959	0.975	WT3-2

data obtained over a year-long study, long-term forecasting such as monthly, seasonal, and yearly patterns is impossible. Because Korea has four distinct seasons, if we were to build a10-year, or more than 20-year, dataset of turbine power patterns (Korean meteorological data and wind power generation data), similar in length to that done by Azad and others [9], then more accurate representative power patterns can be generated. Moreover, we can analyze minor patterns in detail, such as those that were eliminated when we used big size data (see Table 7), and generate new representative patterns.

3. Performance Evaluation of Classification Models

The accuracy of the three classification models used to predict the hourly and daily power patterns is evaluated based on the precision, recall, and F_1-values of (10)–(12). All evaluation measures were obtained using a stratified 10-fold cross validation for all classes. First, Table 8 shows the evaluation results for the hourly power pattern prediction model. Table 9 shows the performance evaluation results of the classifiers for seven classes for daily power pattern prediction.

Table 10. Comparison of classifier error rates.

Data type	Classifier	MAE	RMSE
TS = (hour:{1–24})	**ANFIS**	**0.0536**	**0.1744**
	SMO	0.1751	0.2550
	IBk	0.0595	0.2440
	MLP	0.0543	0.2137
TS = (day:{1–30})	ANFIS	0.0319	0.1581
	SMO	**0.0198**	**0.1280**
	IBk	0.0509	0.1749
	MLP	**0.0234**	**0.1378**

Nine classes in respect of hourly power patterns are discovered by the projected clustering algorithm, and ANFIS shows a better performance than the other two as an accurate classifier for each class. In addition, WT1-1, WT3-1, and WT3-3 show better class prediction performances than the other classes. As a result of daily power patterns, the comparison results indicate that the SMO and MLP algorithms perform better than the two classifiers with different algorithms. As this daily power patterns prediction is an evaluation of a smaller amount of data and classes compared with hourly power pattern prediction, all classifiers generally show better results compared to the hourly patterns.

Table 10 compares the error rates of the classifiers for the hourly and daily power pattern predictions, where the measuring methods used were the MAE and root-mean-square error (RMSE). ANFIS also shows the lowest error rate (MAE: about 0.05, RMSE: about 0.18) for the hourly power pattern prediction, whereas SMO and MLP show the lowest error rate (MAE: about 0.2 to 0.3, RMSE: about 0.13 to 0.16) for the daily power pattern prediction.

In this study, we define a class label by generating representative patterns of each turbine using PROCLUS and assign the representative patterns through learning various forecasting models. Overall, the performance of the proposed forecasting model shows a low rate of error. However, the important point in wind power pattern forecasting is the generation of highly reliable and precise representative patterns by clustering; therefore, application of a variety of clustering methods should be pursued. The proposed method in this study enables experts and users in the wind power generation industry to predict in advance the production and variation of electric power by finding the production patterns of wind turbines in diverse time units from weather data.

V. Conclusion

This paper used three projected clustering approaches to discover hourly and daily representative power patterns from data measured from wind turbines. As subsets of all dimensions required for clustering and an appropriate composition of the clusters were used concurrently, the removal of noisy data and the use of a feature selection function are included. The optimal number of clusters was determined using parameter bracketing provided by OpenSubspace, and PROCLUS, which is a clustering-oriented approach, produced the best results. Nine hourly and seven daily power patterns were profiled as representative patterns of the wind turbines in the three regions of Korea. The time-interval features for wind speed, temperature, and wind direction were also used as feature vectors to accurately predict the profiled representative power patterns. The prediction model test applied the ANFIS, SMO, *k*-NN, and MLP algorithms; the precision, recall, and F_1-value as the accuracy evaluation criteria; and MAE and RMSE as the indexes to measure the error rate. As a result, ANFIS recorded the highest accuracy and lowest error rate for the hourly power pattern prediction, whereas SMO and MLP did the same for the daily power pattern prediction. The wind power generation prediction model suggested by this study can predict the wind power patterns of various time units, in theory. However, owing to the limitations of the dataset used, the discovery of various representative power patterns was difficult, and long-term forecasting such as discovering the monthly or yearly power patterns was impossible. Currently, Korea's wind power generation data continue to be accumulated and refined. Therefore, long-term forecasting of wind power generation patterns will be possible in future studies.

References

[1] M.R. Islam, S. Mekhilef, and R. Saidur, "Progress and Recent Trends of Wind Energy Technology," *Renewable Sustain. Energy Rev.*, vol. 21, May 2013, pp. 456–468.

[2] A. Kusiak, H. Zheng, and Z. Song, "Models for Monitoring Wind Farm Power," *Renewable Energy*, vol. 34, no. 3, Mar. 2009, pp. 583–590.

[3] L. Fugon, J. Juban, and G. Kariniotakis, "Data Mining for Wind Power Forecasting," *European Wind Energy Conf.*, Brussels, Belgium, Apr. 2008, pp. 1–6.

[4] A. Kusiak, H. Zheng, and Z. Song, "On-Line Monitoring of Power Curves," *Renewable Energy*, vol. 34, no. 6, June 2009, pp. 1487–1493.

[5] S. Li et al., "Comparative Analysis of Regression and Artificial Neural Network Models for Wind Turbine Power Curve Estimation," *J. Solar Energy Eng.*, vol. 123, no. 4, July 2001, pp. 327–331.

[6] T. Üstüntas and A.D. Sahin, "Wind Turbine Power Curve Estimation Based on Cluster Center Fuzzy Logic Modeling," *J. Wind Eng. Ind. Aerodynamics*, vol. 96, no. 5, 2008, pp. 611–620.

[7] M. Schlechtingen et al., "Using Data-Mining Approaches for Wind Turbine Power Curve Monitoring: A Comparative Study," *IEEE Trans. Sustain. Energy*, vol. 4, no. 3, 2013, pp. 671–679.

[8] R. Rahmani et al., "Hybrid Technique of Ant Colony and Particle Swarm Optimization for Short Term Wind Energy Forecasting," *J. Wind Eng. Ind. Aerodynamics*, vol. 123, Dec. 2013, pp. 163–170.

[9] H.B. Azad, S. Mekhilef, and V.G. Ganapathy, "Long-Term Wind Speed Forecasting and General Pattern Recognition Using Neural Networks," *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, 2014, pp. 546–553.

[10] L. Parsons, E. Haque, and H. Liu, "Subspace Clustering for High Dimensional Data: A Review," *ACM SIGKDD Exploration*, vol. 6, no. 1, June 2004, pp. 90–105.

[11] E. Müller et al., "OpenSubspace: An Open Source Framework for Evaluation and Exploration of Subspace Clustering Algorithms in WEKA," *Int. Open Source Data Mining Workshop*, Bangkok, Thailand, 2009, pp. 2–13.

[12] E. Müller et al., "Evaluating Clustering in Subspace Projections of High Dimensional Data," *Int. Conf. VLDB*, Lyon, France, vol. 2, no. 1, Aug. 24–28, 2009, pp. 1270–1281.

[13] R. Agrawal et al., "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," *ACM SIGMOD Int. Conf. Manag. Data*, Seattle, WA, USA, June 2–4, 1998, pp. 94–105.

[14] K. Sequeira and M. Zaki, "SCHISM: A New Approach for Interesting Subspace Mining," *IEEE Int. Conf. Data Mining*, Brighton, UK, Nov. 1–4, 2004, pp. 186–193.

[15] M. Ester et al., "Algorithms for Characterization and Trend Detection in Spatial Databases," *Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 27–31, 1998, pp. 44–50.

[16] H.-P. Kriegel et al., "A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data," *Int. Conf. Data Mining*, Houston, TX, USA, Nov. 27–30, 2005, pp. 250–257.

[17] K. Kailing, H.-P. Kriegel, and P. Kroger, "Density-Connected Subspace Clustering for High-Dimensional Data," *SIAM Conf.*

Data Mining, Lake Buena Vista, FL, USA, Apr. 22–24, 2004, pp. 246–257.

[18] C. Aggarwal et al., “Fast Algorithms for Projected Clustering,” *ACM SIGMOD Int. Conf. Manag. Data*, Philadelphia, PA, USA, 1999, pp. 61–72.

[19] G. Moise and J. Sander, “Finding Non-redundant, Statistically Significant Regions in High Dimensional Data: A Novel Approach to Projected and Subspace Clustering,” *Int. Conf. KDD*, Las Vegas, NV, USA, Aug. 24–27, 2008, pp. 533–541.

[20] M. Hayati, M. Seifi, and A. Rezaei, “Double Gate MOSFET Modeling Based on Adaptive Neuro-Fuzzy Inference System for Nanoscale Circuit Simulation,” *ETRI J.*, vol. 32, no. 4, Aug. 2010, pp. 530–539.

[21] S.K. Shevade et al., “Improvements to the SMO Algorithm for SVM Regression,” *IEEE Trans. Neural Netw.*, vol. 11, no. 5, 2000, pp. 1188–1193.

[22] L.M. Zouhal and T. Denoeux, “An Evidence-Theoretic *k*-NN Rule with Parameter Optimization,” *IEEE Trans. Sys. Man, Cybernetics*, vol. 28, no. 2, 1998, pp. 263–271.

[23] B. Durrant et al., *Weka 3: Data Mining Software in Java*, Machine Learning Group at the University of Waikato, 2014. Accessed Aug. 1, 2014. http://www.cs.waikato.ac.nz/ml/weka/

[24] E. Müller et al., *OpenSubspace: Weka Subspace-Clustering Integration*, 2013. Accessed July 16, 2014. http://dme.rwth-aachen.de/OpenSubspace/

[25] H. Lee, J.-H. Yoo, and D. Park, “Data Clustering Method Using a Modified Gaussian Kernel Metric and Kernel PCA,” *ETRI J.*, vol. 36, no. 3, June 2014, pp. 333–342.

Heon Gyu Lee received his BS degree in computer science from Kyonggi University, Suwon, Rep. of Korea, in 2002 and his MS and PhD degrees in computer science from Chungbuk National University, Cheongju, Rep. of Korea, in 2005 and 2009, respectively. Since 2009, he has been working for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is now a senior member of the researching staff. His main research interests are data mining, databases, bioinformatics, pattern recognition, and knowledge-based information retrieval.

Minghao Piao received his BS degree in science and technology from Yanbian University, Yanji, China, in 2007 and his MS and PhD degrees from Chungbuk National University, Cheongju, Rep. of Korea, in bioinformatics, in 2009 and computer science, in 2014, respectively. His main research interests include electrical customer classification, data mining application to energy data, discovery of emerging patterns in clustering analysis, and reduction of information redundancy for building classifiers.

Yong Ho Shin received his BS degree in industrial engineering from Seoul National University, Rep. of Korea, in 1993 and his MS and PhD degrees in industrial and system engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 1995 and 2003, respectively. From 2007 to 2009, he worked for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. Since 2009, he has been with the School of Business, Yeungnam University, Gyeongsan, Rep. of Korea, where he is now an associate professor. His main research interests are operations management, logistics management, data mining, and knowledge management systems.

http://dx.doi.org/10.4218/etrij.15.2314.0070