Expert Systems With Applications 59 (2016) 217–225

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Sliding window-based support vector regression for predicting micrometeorological data

Yukimasa Kaneda[a,*], Hiroshi Mineno[b,c]

[a] Graduate School of Integrated Science and Technology, Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka 432-8011, Japan
[b] College of Informatics, Academic Institute, Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka 432-8011, Japan
[c] JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan

ARTICLE INFO

Article history:
Received 4 February 2016
Revised 29 March 2016
Accepted 13 April 2016
Available online 23 April 2016

Keywords:
Predicting micrometeorological data
Data extraction
Dynamic aggregation
Support vector regression
Ensemble learning

ABSTRACT

Sensor network technology is becoming more widespread and sophisticated, and devices with many sensors, such as smartphones and sensor nodes, have been used extensively. Since these devices have more easily accumulated various kinds of micrometeorological data, such as temperature, humidity, and wind speed, an enormous amount of micrometeorological data has been accumulated. In recent years, it has been expected that such an enormous amount of data, called big data, will produce novel knowledge and value. Accordingly, many current applications have used data mining technology or machine learning to exploit big data. However, micrometeorological data has a complicated correlation among different features, and its characteristics change variously with time. Therefore, it is difficult to predict micrometeorological data accurately with low computational complexity even if state-of-the-art machine learning algorithms are used. In this paper, we propose a new methodology for predicting micrometeorological data, sliding window-based support vector regression (SW-SVR) that involves a novel combination of support vector regression (SVR) and ensemble learning. To represent complicated micrometeorological data easily, SW-SVR builds several SVRs specialized for each representative data group in various natural environments, such as different seasons and climates, and changes weights to aggregate the SVRs dynamically depending on the characteristics of test data. In our experiment, we predicted the temperature after 1 h and 6 h by using large-scale micrometeorological data in Tokyo. As a result, regardless of testing periods, training periods, and prediction horizons, the prediction performance of SW-SVR was always greater than or equal to other general methods such as SVR, random forest, and gradient boosting. At the same time, SW-SVR reduced the building time remarkably compared with those of complicated models that have high prediction performance.

1. Introduction

Sensor network technology is becoming more widespread and sophisticated, and devices with many sensors have been used extensively. The devices can very easily obtain various kinds of micrometeorological data such as temperature, humidity, and wind speed. Micrometeorological data is affected strongly by the surface of the earth and is related to our lives and industrial activity. Accordingly, the data has been used by many applications such as environmental control systems for greenhouses (Othman & Shazali, 2012; Park & Park, 2011). Moreover, more advanced applications exploit the data to a greater extent by using machine learning and data mining technology. Furthermore, an enormous amount of micrometeorological data has been accumulated by many devices, and it has been expected that analyzing such an enormous amount of data, called big data, will produce novel knowledge and value.

To predict micrometeorological data effectively, a number of researchers have studied machine learning (Smith, Hoogenboom, & McClendon, 2009). These researchers described prediction methods for micrometeorological data; particularly, prediction performance and computational complexity were often mentioned. Meanwhile, micrometeorological data has a complex correlation among different features such as temperature and humidity. Moreover, its characteristics change variously with time. Therefore, even if big data is given as training data, it is not easy to predict micrometeorological data accurately. Furthermore, in many cases, so that models can have high prediction performance, they have to become complicated, and the computational complexity increases. Accordingly, some models probably cannot be built from big data in a

* Corresponding author.
E-mail address: kaneda@minelab.jp (Y. Kaneda).

http://dx.doi.org/10.1016/j.eswa.2016.04.012

practical amount of computing time. In other words, there is a trade-off relationship between high prediction performance and low computational complexity. However, compatibility is required in some practical use. As the prediction performance in applications becomes higher, the quality provided by the applications becomes better. For example, in the case of environmental control systems based on prediction (Kolokotsa, Pouliezos, Stavrakakis, & Lazos, 2009), the higher prediction performance enables the systems to provide precise control, precise management, and better environments. On the other hand, models that need a long time for training are worthless in practical use. In current situations where the amount of usable data has increased remarkably, this trade-off relationship has become a more critical issue.

Recently, one type of machine learning algorithm, support vector machines (SVMs), have been used successfully in various fields. The basic theory is a more efficient learning method based on probably approximately correct (PAC) learning. Moreover, SVMs can separate non-linear data with low computational complexity. Since most data observed in the real world is likely to have non-linear relationships, SVMs have also been applied to micrometeorological data prediction (Antonanzas, Urraca, Martinez-de-Pison, & Antonanzas-Torres, 2015; Mohammadi, Shamshirband, Anisi, Alam, & Petković, 2015; Urraca, Antonanzas, Martinez-de-Pison, & Antonanzas-Torres, 2015). Moreover, SVMs led to better prediction performance than other algorithms such as artificial neural networks (ANNs) and the autoregressive integrated moving average (ARIMA) model (Chevalier, Hoogenboom, McClendon, & Paz, 2011; Maity, Bhagwat, & Bhatnagar, 2010). However, when SVMs learn big data, the computational complexity is still a matter of concern. Another alternative learning method, ensemble learning, has also been used more widely for predicting micrometeorological data (Singh, Gupta, & Rai, 2013). The prediction performance of ensemble learning is greater than or equal to that of SVMs. The basic methodology is a combination of weak learners built from different kinds of training data. The combination yields a higher generalizing capability that a single model cannot represent. In particular, some researchers proposed improved methods that could be applied to micrometeorological data prediction (Wang & Japkowicz, 2009; Xie, Li, Ngai, & Ying, 2009). However, it is difficult to apply the methods to regression, and it is possible that the models will not be able to follow micrometeorological data whose characteristics always change with time.

In this paper, we propose a new methodology for predicting micrometeorological data, sliding window-based support vector regression (SW-SVR). SW-SVR involves a novel combination of support vector regression (SVR) and ensemble learning. To represent complicated micrometeorological data easily, SW-SVR builds several SVRs specialized for each representative data group in various natural environments, such as different seasons and climates. The specialized SVRs are built based on our previous proposed method, dynamic short-distance data collection (D-SDC) that extracts effective data for specific data prediction by taking account of movements: changes in data during prediction horizons. Each weak learner built from each extracted data specializes on specific data and predicts accurately the data similar to the specialized data. Then, SW-SVR aggregates all the predicted values based on weights decided by the similarity between test data and each data specialized by weak learners. This new ensemble learning methodology that changes weights dynamically enables following micrometeorological data whose characteristics hardly change with time. Our results demonstrated that the prediction performance of SW-SVR was always greater than or equal to that of other general methods such as SVR, random forest, and gradient boosting. At the same time, SW-SVR reduced the building time remarkably compared with that of complicated models that have high prediction performance.

2. Related work

As mentioned in the introduction, to predict micrometeorological data effectively, SVMs and ensemble learning have generally been used. These algorithms have higher prediction performance for micrometeorological data than traditional methods because SVMs use not only a margin maximizing algorithm whose great performance was proved by PAC learning but also the kernel trick that enables non-linear separation. On the other hand, ensemble learning provides higher generalizing capability that a single model cannot represent. In this section, a brief summary of these algorithms and some improved algorithms are given. Moreover, so that SW-SVR can draw advantages from both SVMs and ensemble learning, several problems of these algorithms for practical use are discussed.

2.1. Support vector regression

SVMs, introduced by Vapnik,(1995), have been used successfully in various fields. In the simplest case, binary classification, SVMs obtain a separating hyperplane decided by maximizing the margin. The margin means the norms between different classes. PAC learning proved that maximizing the margin produces high generalization ability. Moreover, the kernel trick enables SVMs to separate data non-linearly with low computational complexity. Various kinds of data observed in the real world are likely to have non-linear relationships. Accordingly, SVMs are used in many applications such as micrometeorological data prediction (Kisi & Cimen, 2012; Maity et al., 2010). Meanwhile, SVMs for regression, support vector regression (SVR), uses the same methodology as SVMs that have the highest generalization ability. In this section, a brief summary of SVR is given as follows.

First, the linear function for regression is given as follows:

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b.$$

Then, as with SVMs, SVR also minimizes the norm of the weight vector $\mathbf{w}$; the L^2 norm $\|\mathbf{w}\|^2$ is often used, and minimizing $\|\mathbf{w}\|^2$ corresponds to maximizing the margin. Meanwhile, SVR tolerates prediction error ϵ. Therefore, the primal problem of SVR is shown as follows:

$$\text{minimize} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \begin{cases} y_i - \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) \le \epsilon \\ \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) - y_i \le \epsilon. \end{cases}$$

Moreover, to take some errors into account further, the same slack variables ξ as soft margin SVMs are introduced. The slack variables mean penalties and increase in proportion to errors between true values and predicted values. The problem that the slack variables are introduced into is shown as follows:

$$\text{minimize} \frac{\|\mathbf{w}\|^2}{2} + C\sum_i \left(\xi_i + \xi_i^*\right)$$

$$\text{subject to} \begin{cases} y_i - \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) \le \epsilon + \xi_i \\ \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) - y_i \le \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0. \end{cases}$$

where the constant C means the balance between the effect of maximizing the margin and penalties. To minimize the above formula, the slack variables in the formula must also be minimized. Accordingly, the slack variables depending on the errors are shown as follows:

$$\xi_i = \begin{cases} 0 & \left(y_i - \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) \le \epsilon\right) \\ y_i - \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) - \epsilon & \text{otherwise} \end{cases}$$

$$\xi_i^* = \begin{cases} 0 & \left(\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) - y_i \le \epsilon\right) \\ \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) - y_i - \epsilon & \text{otherwise.} \end{cases}$$

The above formulas mean that a penalty is not given when the error is lower than ϵ, but the error is regarded as a penalty that cannot be tolerated when the error is higher than ϵ. In other words, SVR tolerates errors less than ϵ, but errors over ϵ are solely taken into account as penalties. Finally, the dual problem is derived from the above primal problem by Lagrange multiplier and corresponds to a quadratic programming problem as with SVMs. As a result, since a unique global optimal solution is solved, SVR is superior to traditional algorithms that might fall into a local optimal solution, such as ANNs. The dual problem derived by Lagrange multiplier is shown as follows:

$$\text{maximize} -\frac{1}{2}\sum_{i,j}(\alpha_i + \alpha_i^*)\left(\alpha_j + \alpha_j^*\right)\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j - \epsilon\sum_i(\alpha_i + \alpha_i^*) + \sum_i y_i(\alpha_i - \alpha_i^*)$$

$$\text{subject to} \begin{cases} \sum_i(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C]. \end{cases}$$

Moreover, the above dual problem can easily involve nonlinear map φ to consider a higher dimension. To introduce nonlinear map φ in the above problem, kernel function $\mathrm{K}(x_i, x_j) = \varphi^t(x_i)\varphi(x_j)$ is defined and used instead of $x_i^{\mathrm{T}}x_j$. Then $\varphi^t(\mathrm{x}_i)\varphi(\mathrm{x}_j)$ is determined based on $\mathrm{K}(\mathrm{x}_i, \mathrm{x}_j)$ without calculation on a mapped higher dimension; this method is called the kernel trick. SVR based on maximizing the margin and the kernel trick yields high prediction performance.

Meanwhile, conventional quadratic programming solvers, such as the steepest descent method, have very high computational complexity; the computational complexity is approximately $O(N^3)$ where N is the number of training data. Accordingly, a quadratic programming solver for SVMs, sequential minimal optimization (SMO), has become de facto standard (Platt, 1998). SMO specialized for SVM reduce the computational complexity of SVM to approximately $O(N^2)$. Nevertheless, when an enormous amount of data is inputted as training data, the computational complexity increases substantially. To solve the problem, a theory that regards the quadratic programming problem as a computational geometry problem, core vector machine (CVM), was proposed (Tsang, Kwok, & Cheung, 2005). The prediction performance of CVM is comparable to that of SVMs, and the computational complexity decreases substantially. However, according to a paper (Loosli, 2007), prediction performance and computational complexity of CVM strongly depend on the values of parameters. Therefore, when essential parameter tuning for practical use is taken into account, CVM does not always satisfy both high prediction performance and low computational complexity.

SVR is one of the best algorithms in machine learning from the viewpoint of prediction performance. In particular, it has been expected that the kernel trick used in the dual problem is effective for predicting micrometeorological data that has a complex correlation among different features. However, the computational complexity to solve the dual problem is often still long for practical use. Thus, it is difficult to apply conventional SVR directly to micrometeorological data prediction.

2.2. Ensemble learning

Ensemble learning has been studied recently and used increasingly. The basic methodology of ensemble learning is a combination of weak learners built from different kinds of training data. The combination yields a higher generalizing capability that a single model cannot represent. As with SVMs, ensemble learning can represent non-linear relationships and has been used for predicting micrometeorological data. In particular, the two kinds of approaches, bagging and boosting, have often been used in ensemble learning. The approaches differ greatly on the method to build weak learners and aggregate them.

Algorithm 1
Bagging for regression.

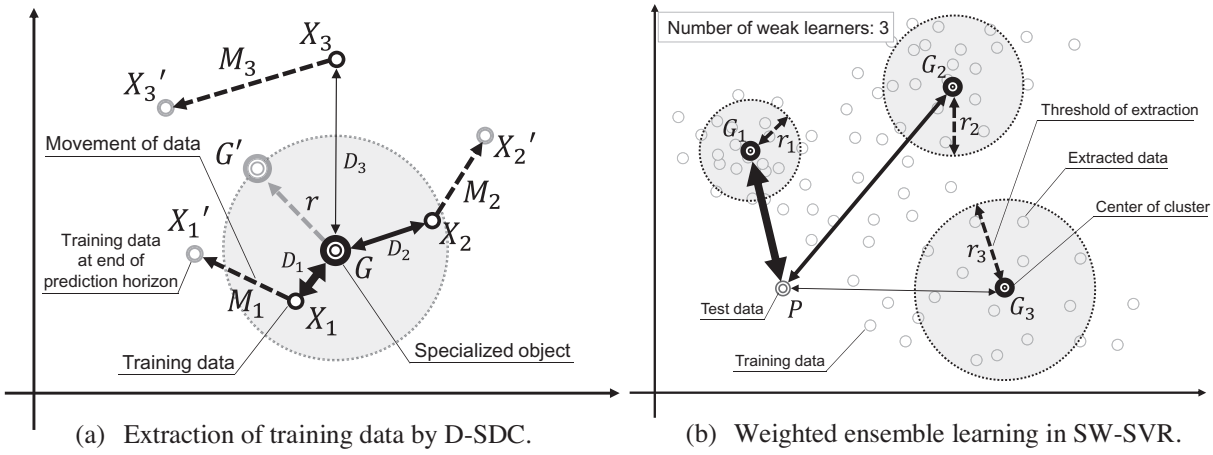
Input:
Training data: $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ where $\mathbf{x}_i \in X, y_i \in Y$
Number of weak learners: n
For $t = 1$ to n **do**
1. $D_t \leftarrow$ generate sample from D with replacement
2. $H_t(X) \leftarrow$ build a weak learner from D_t
Output:
$H(X) = \frac{1}{n}\sum_{t=1}^{n} H_t(X)$

Algorithm 2
Boosting for regression.

Input:
Training data: $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ where $\mathbf{x}_i \in X, y_i \in Y$
Number of weak learners: n
Weights: $\mathbf{w}_i = 1/N$
For $t = 1$ to n **do**
1. $H_t(X) \leftarrow$ build a weak learner from D by using weights $\mathbf{w}_t$
2. $\epsilon_t \leftarrow$ compute error rate of $H_t(X)$
3. $\alpha_t \leftarrow$ compute reliability of prediction result of $H_t(X)$ based on ϵ_t
4. $\mathbf{w}_{t+1} \leftarrow$ update weights $\mathbf{w}_t$ based on α_t
Output:
$H(X) = \sum_{t=1}^{n}(\alpha_t H_t(X)) / \sum_{t=1}^{n}\alpha_t$

Bagging uses several training data generated by bootstrap sampling. The algorithm of basic bagging for regression is shown in Algorithm. 1. In bagging, different kinds of training data are created by sampling inputted original training data with replacement. Then, weak learners are built from each sampled training data. Finally, each predicted value is aggregated by majority vote or arithmetic average. In particular, random forest, introduced by Breiman (Breiman, 2001), to which randomness in feature selection is also applied, often demonstrates better prediction performance than conventional models such as SVMs. Random forest is used in various applications and has been extended to other improved versions. For example, to predict imbalanced data observed frequently in the real world more accurately, improved balanced random forest (IBRF) has been proposed (Xie et al., 2009). IBRF involves an efficient sampling method for imbalanced data and cost-sensitive learning that penalizes misclassification of minority class more strongly. The authors showed that IBRF was more effective to predict imbalanced data than class-weighted SVMs and conventional improved random forest for imbalanced data prediction.

Boosting builds repeatedly weak learners by using weights based on the error rate. The algorithm of basic boosting for regression such as Adaboost (Freund & Schapire, 1997) is shown in Algorithm. 2. Unlike bagging, almost all boosting algorithms use the same training data, but the training data is weighted repeatedly. Boosting alternates between building weak learners by using weights and updating weights. Finally, each predicted value is aggregated by weighted average. Various kinds of algorithms in boosting have been studied and proposed; gradient boosting (Friedman, 2001) in particular has shown the best prediction performance in many competitions. Meanwhile, as with IBRF, the boosting algorithm for imbalanced data, boosting-SVM, has also been proposed (Wang & Japkowicz, 2009). The main characteristic of boosting-SVM is using asymmetric misclassification cost. The authors demonstrated that boosting-SVM enabled more accurate prediction of both the majority class and minority class.



(a) Extraction of training data by D-SDC. (b) Weighted ensemble learning in SW-SVR.

Fig. 1. Processing overview of SW-SVR.

When micrometeorological data including many unusual natural environments is regarded as imbalanced data, the above methods are likely to classify micrometeorological data more accurately. However, these approaches cannot be applied to regression. Moreover, according to our previous research (Suzuki, Kaneda, & Mineno, 2015), there is proper training data depending on test data. In other words, weights to aggregate weak learners built from different kinds of training data should depend on test data.

3. SW-SVR: Sliding window-based support vector regression

We propose a new methodology for predicting micrometeorological data, sliding window-based support vector regression, combining methodologies of SVR and ensemble learning. The basic theories are based on D-SDC, our previous proposed method to extract effective data for specific data prediction, and novel weighted ensemble learning as shown in Fig. 1. First, to represent complicated micrometeorological data easily, SW-SVR builds several SVRs specialized for each representative data group in various natural environments, such as different seasons and climates. The specialized SVRs are built based on D-SDC that extracts effective data for specific data prediction by taking account of movements: changes of data during prediction horizons (Fig. 1(a)). Each weak learner built from each extracted data specializes on specific data and accurately predicts the data similar to the specialized data. Afterward, each weak learner is aggregated with weights determined dynamically at the time of prediction so as to maintain the prediction performance of micrometeorological data whose characteristics always change with time (Fig. 1(b)). The weights are decided by the similarity between test data and each data specialized by weak learners. Even if the characteristics of micrometeorological data always change with time, SW-SVR always gives priority to weak learners that are more suitable for predicting test data. The details of the SW-SVR algorithm are shown in Algorithm. 3. The procedure for training consists of two kinds of preprocessing, iterated learning, and dynamic aggregation. The procedures of each part are shown as follows.

The below-mentioned algorithms in SW-SVR use the L^2 norm: the Euclid distance, and the performance is related to feature space. For example, if feature space includes noisy features or non-linear relationships between features, the performance will probably be reduced substantially. In particular, micrometeorological data has a complex correlation among different features such as temperature and humidity. Accordingly, feature space must be mapped into other feature space that takes into account the presence of noise and non-linear relationships. In our approach, we use kernel approximation (Rahimi & Recht, 2007) and partial least squares (PLS) regression (Tenenhaus, Vinzi, Chatelin, & Lauro, 2005) to map into new feature space. Kernel approximation generates new feature space and involves higher dimensions that represent non-linear data as linear data with a very low computational complexity. Actually, a combination of kernel approximation and linear SVMs led to faster prediction performance that is comparable to that of exact SVM (Cao, Naito, & Ninomiya, 2008). On the other hand, PLS regression is a supervised dimension reduction methodology. This method can reduce dimensions by extracting latent variables that have a strong relationship with a dependent variable. If feature space includes noisy features, the effect is reduced because of PLS regression. The combination of kernel approximation and PLS regression enables SW-SVR to use effective feature space for calculation of the L^2 norm in micrometeorological data.

Algorithm 3
Sliding window-based support vector regression.

Input:
Training data set: $S=\{(\mathbf{x}_1,\ y_1,\ \mathbf{x}'_1),\ \ldots,(\mathbf{x}_N,\ y_N,\ \mathbf{x}'_N)\}$ where $\mathbf{x}_i \in X,\ y_i \in Y,\ \mathbf{x}'_i \in X'$
Test data: $\boldsymbol{P}$
Number of weak learners: n
Weight parameters: p, q
Preprocessing:
1. apply normalization to X and X'
2. fit kernel approximation and PLS regression to X and X'
3. $M_i = ||\mathbf{x}_i - \mathbf{x}'_i||,\ i = 1 \ldots N$
4. $\boldsymbol{G}_t \leftarrow$ each center of $kmenas(X),\ t = 1 \ldots n$
For $t = 1$ to n **do**
1. $D_{ti} = ||\boldsymbol{G}_t - \mathbf{x}_i||,\ i = 1 \ldots N$
2. $r_t = \sum_{i=1}^{N}(w_i M_i) / \sum_{i=1}^{N}(w_i)$ where $w_i = 1/D_{ti}^p$
3. $S_t = \{(\mathbf{x}_i,\ y_i) | D_{ti} < r_t\},\ i = 1 \ldots N$
4. $H_t(X) \leftarrow$ train $LinearSVR(S_t)$
Output:
$H(\boldsymbol{P}) = \sum_{t=1}^{n}(w_t H_t(\boldsymbol{P})) / \sum_{t=1}^{n}(w_t)$ where $w_t = 1/||\boldsymbol{G}_t - \boldsymbol{P}||^q$

According to our previous research, to accurately predict particular specific data in micrometeorological data, it is necessary to extract effective training data for specific data prediction (Suzuki et al., 2015). In SW-SVR, these several specific data is selected in advance, and weak learners are built from extracted effective training data for predicting each specific data. Meanwhile, micrometeorological data involves various natural environments such as different seasons and climates. Therefore, each selected specific data must represent more varied natural environments that probably will appear so as to represent micrometeorological data by several models. In SW-SVR, each specific data is selected by a clustering

algorithm, k-means (Macqueen, 1967). The k-means is one of the most famous non-hierarchical clustering algorithms and classifies data faster under several clusters than other clustering algorithms. In SW-SVR, the k-means classifies all training data into the same number of clusters as the number of weak learners given by users. Then, each center of clusters is used as specific data that represents various natural environments.

After selecting several specific data, SW-SVR iterates data extraction and building a model. First, SW-SVR extracts effective training data for predicting each specific data by D-SDC (Suzuki, Kaneda, & Mineno, 2014). The theory of D-SDC is similar to that of the k-nearest neighbor (k-NN) algorithm, and D-SDC also extracts some training data similar to a specialized object. However, in our D-SDC, the amount of extracted data depends on the movement of a specialized object with time. The movement r means the change of a specialized object during prediction horizons as shown in the following equation:

$$r_t = \|\boldsymbol{G}_t - \boldsymbol{G}'_t\|$$

where $\boldsymbol{G}$ is a specialized object, and $\boldsymbol{G}'$ is a specialized object after prediction horizons. D-SDC extracts training data whose norm from a specialized object is shorter than the movement r. Accordingly, extracted training data S by D-SDC is given as follows:

$$S_t = \left\{ (\mathbf{x}_i,\ y_i) |\ \|\boldsymbol{G}_t - \mathbf{x}_i\| < \|\boldsymbol{G}_t - \boldsymbol{G}'_t\| \right\}$$

where $\mathbf{x}$ is the feature of training data and y is the dependent variable of training data. D-SDC is based on the movement r because the movement r is strongly related to autocorrelation of data surrounding a specialized object. In micrometeorological data, movements in specific natural environments are mutually similar, and the autocorrelation becomes lower when these movements are bigger. For example, in Japan, the change of weather is drastic every spring, and the natural environments change various other natural environments with time. Meanwhile, when we predict time series data such as micrometeorological data, autocorrelation means correlation between features and a dependent variable, and more training data is required for highly accurate prediction when autocorrelation is lower. Since D-SDC extracts the amount of data surrounding a specialized object in proportion to the movement r, extraction that considers autocorrelation of data surrounding a specialized object is achieved. However, the movement r is unknown because $\boldsymbol{G}'$ is not observed. Meanwhile, as mentioned above, movements of data surrounding a specialized object are mutually similar. Therefore, D-SDC estimates the movement r based on movements of training data similar to a specialized object by weighted average, where the weights are reciprocals of norms between a specialized object and each training data. Movements of training data can be calculated by referring to the time when each training data is observed. The estimated movement r is given as follows:

$$r_t = \|\boldsymbol{G}_t - \boldsymbol{G}'_t\| \approx \frac{\sum_{i=1}^{N} w_i\ \|\mathbf{x}_i - \mathbf{x}'_i\|}{\sum_{i=1}^{N} w_i} \text{ where } w_i = \frac{1}{\|\boldsymbol{G}_t - \mathbf{x}_i\|^p},$$

N is the number of training data, and p is a weighted parameter. Afterward, SW-SVR builds several linear SVRs as weak learners based on the extracted data. As described above, a combination of linear SVR and kernel approximation is comparable to SVR using a kernel method. Moreover, linear SVR can be built much faster by using liblinear (Fan, Chang, Hsieh, Wang, & Lin, 2008), an optimized implementation for linear SVMs, instead of other general implementations of SVMs such as libSVM (Chang & Lin, 2011). Although a usable kernel in liblinear is restricted to the linear kernel, liblinear can build the model much faster by solving the primal problem instead of the dual problem. Furthermore, since all training data is divided into smaller amounts of extracted data, each model can be built faster, and it is easier to learn each extracted data by parallel processing.

The predicted values of SW-SVR take into account the change of natural environments with time. In general ensemble learning, prediction for regression depends on the weighted average, and the weights are determined at the time of training. However, SW-SVR determines weights dynamically at the time of prediction. The weights are determined by the norm between test data and each data specialized by weak learners. A final hypothesis of SW-SVR is shown as follows:

$$H(\boldsymbol{P}) = \frac{\sum_{t=1}^{n} w_t H_t(\boldsymbol{P})}{\sum_{t=1}^{n} w_t} \text{ where } w_t = \frac{1}{\|\boldsymbol{G}_t - \boldsymbol{P}\|^q},$$

$\boldsymbol{P}$ is the test data, n is the number of weak learners, $H(X)$ is a hypothesis, and q is a weighted parameter. In our approach, since the weights of ensemble learning are determined dynamically for every prediction, SW-SVR can follow micrometeorological data whose characteristics always change with time.

Finally, we describe the computational complexity of SW-SVR. To represent complicated micrometeorological data easily, SW-SVR uses the various conventional methods besides D-SDC we proposed: kernel approximation, PLS regression, k-means, and linear SVR. The computational complexity of these methods in general increases linearly; in other words, the computational complexity is approximately equal to $O(N)$ where the number of training data N is even bigger than the number of the dimensions and each parameter of these methods. Moreover, the computational complexity of D-SDC corresponds to $O(nN)$ because D-SDC just iterates N times of distance calculation $n+1$ times where n is the number of weak learners in SW-SVR. Therefore, if N is even bigger than n, the computational complexity of D-SDC also increases linearly. The total computational complexity of SW-SVR is approximately equal to $O(N)$ that is even less than that of SVR.

4. Evaluation

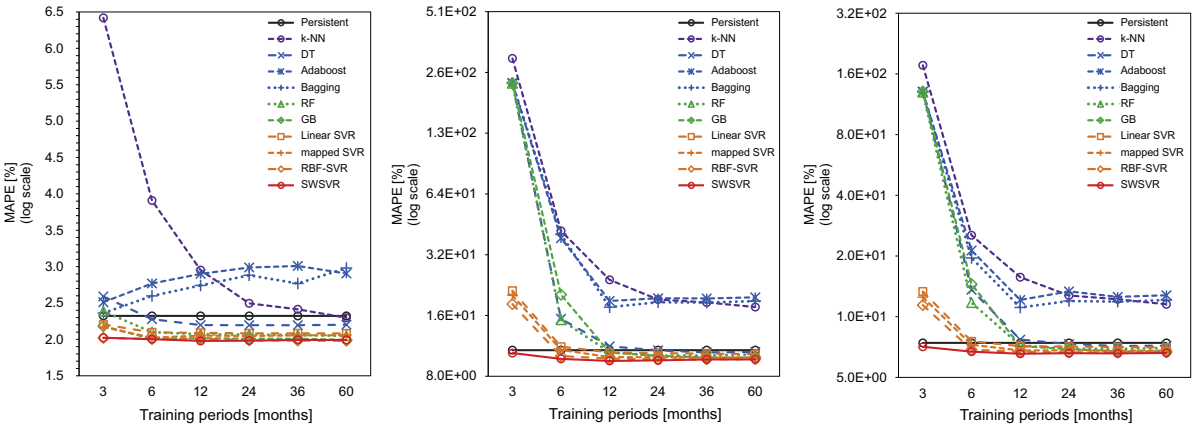
4.1. Experiment

We compared the performance of SW-SVR with other standard methods for regression: k-NN, decision tree (DT), Adaboost, bagging, random forest (RF), gradient boosting (GB), linear SVR, and SVR using a radial basis function (RBF) kernel that shows higher performance in various fields (RBF-SVR). Note that the kernel of kernel approximation in SW-SVR is also the RBF kernel, and the base learner in Adaboost and bagging is the decision tree that has been used generally. Moreover, to evaluate SW-SVR in more detail, we evaluated the performance of linear SVR with mapping: standard linear SVR to which the same mapping as SW-SVR is applied ("mapped SVR"). Mapped SVR clarifies each performance of mapping feature space and ensemble learning based on D-SDC. All parameters of the used models were adjusted by the grid search method. Baseline for this evaluation was the performance of the naivest persistent model as shown in the following formula:

$$\hat{y}_{i+\Delta t} = y_i$$

where $\hat{y}$ is the predicted value, y is the true value, and Δt is the prediction horizons.

We evaluated the performance by two ways: hold-out validation and 10-fold cross-validation. We predicted the temperature after 1 h and 6 h by using large-scale micrometeorological data in Tokyo (Japan Meteorological Agency, n.d.). The data consists of atmospheric pressure, temperature, relative humidity, wind speed, and irradiance. In hold-out validation, training periods are limited to the earlier periods than testing periods so as to assume practical use; test data is always predicted based on past training data in practical use. The training periods were from 3 months to 5



(a) Testing periods: 1 month. (b) Testing periods: 6 months. (c) Testing periods: 12 months.

Fig. 2. MAPE for prediction after 1 h for each algorithm. Note that (b) and (c) are shown with log scale.

years before September 1, 2014, and testing periods were from 1 month to 1 year later the same day. By varying the training periods and the testing periods, the performance under the various usage scenarios is evaluated. On the other hand, the periods for 10-fold cross-validation were 6 years from September 1, 2009 to September 1, 2015. Note that the amount of data per month was approximately 4000 because the data was accumulated every 10 minutes. In this evaluation, we used the mean absolute percentage error (MAPE) as the index of prediction error and building time calculated based on the CPU clock time as the index of computational complexity. MAPE is shown as follows:

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

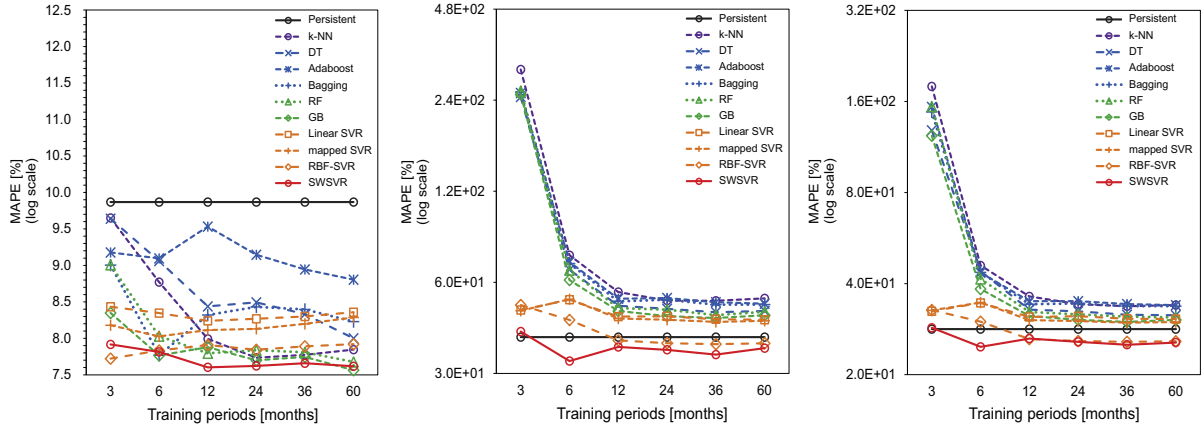
where N is the number of test data, y is the true value, and $\hat{y}$ is the predicted value. Moreover, we evaluated the average of each extraction rate by D-SDC in each experimental condition so as to analyze the performance of SW-SVR and D-SDC further. All implementations for this evaluation are in Python, and implementations in scikit-learn (Pedregosa et al., 2012) were used for all methods except SW-SVR. This evaluation was performed on a single core of a machine with an Intel Core i5-2500 K Processor and 12GB of RAM; even though several methods, such random forest and SW-SVR, can be performed on parallel processing, the methods were performed on a single core so as to evaluate the building time of all methods fairly.

4.2. Results and discussion

Fig. 2 and 3 show the prediction error in the prediction horizons of 1 h and 6 h, respectively. Note that a log scale is used in Figs. 2(b), (c), 3(b), and (c). The results indicate that SW-SVR produced the best average performance in all models during the whole testing periods, training periods, and prediction horizons. In particular, the effect occurs noticeably when testing periods are longer than training periods. On the other hand, in this situation, almost all methods except SW-SVR have often lower performance than the naivest persistent model as baseline. The results demonstrate that the conventional superior methods do not always display the great performance for micrometeorological data prediction depending on difficulty of the prediction caused by training periods and testing periods and prediction horizons. Moreover, in algorithms based on SVR, the prediction performance of SW-SVR is almost the best, followed in order by those of RBF-SVR, mapped SVR, and linear SVR. The difference between mapped SVR and linear SVR is due to the effect of mapping feature space. On the other hand, the difference between SW-SVR and mapped SVR is due to the effect of ensemble learning based on D-SDC. These comparisons demonstrated that both mapping feature space and ensemble learning based on D-SDC are effective for improving prediction performance. Meanwhile, mapped SVR also tended to have lower prediction performance than that of SW-SVR when the testing periods are longer than the training periods. Accordingly, under this condition, ensemble learning based on D-SDC is particularly effective. When the testing periods are longer than the training periods, the effective training data for predicting the test data is reduced. We considered that a little training data that D-SDC extracted for building models corresponded to the effective training data for predicting the test data. Actually, Fig. 4 indicates the average of each extraction rate by D-SDC and demonstrates that weak learners of SW-SVR are always built from a very small proportion of the whole training data. SW-SVR that always predicts micrometeorological data accurately regardless of the amount of training data is very practical and useful.

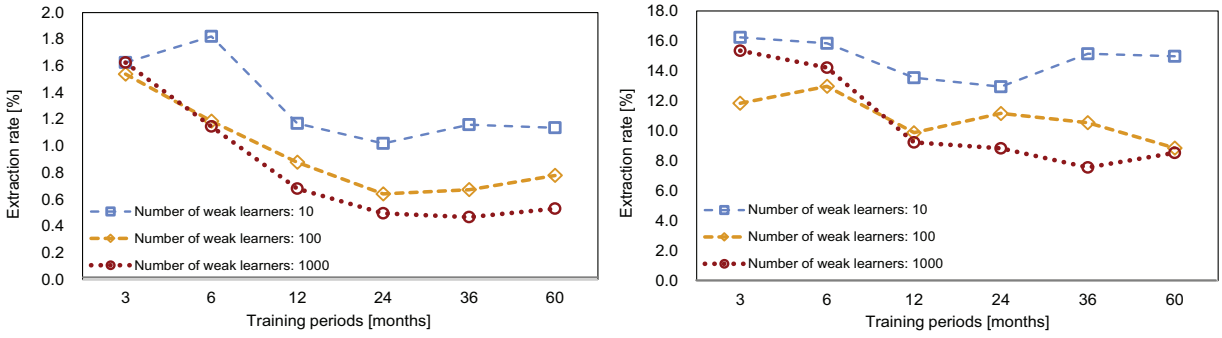
Table 1 shows the results of 10-fold cross-validation in the prediction horizons of 1 h and 6 h. SW-SVR was often superior to all methods including RBF-SVM in hold-out validation. However, in 10-fold cross-validation, although SW-SVR had higher the prediction performance than that of all methods except RBF-SVR, RBF-SVR was superior to SW-SVR slightly. The results demonstrate that the prediction performance of SW-SVR is affected by temporal order between training data and test data, and SW-SVR is particularly suited to be used for practical use in which test data is always predicted based on past training data. Meanwhile, even in 10-fold cross-validation, the magnitude relation of the prediction error between mapped SVR and linear SVR and SW-SVR was same as the case of hold-out validation. Therefore, both mapping feature space and ensemble learning based on D-SDC are effective for improving prediction performance in cross-validation.

Fig. 5 and 6 show the building time in the prediction horizons of 1 h and 6 h, respectively. Figs. 5(a) and 6(a) show the building time of models that have high prediction performance as shown in Figs. 2 and 3, RF, GB, RBF-SVR, and SW-SVR, when training periods were varied. Note that the number of weak learners was 1000 in the ensemble learning series, cost parameter was 1 in the SVR series, and σ of SW-SVR was 0.00001; σ of SW-SVR was a parameter of the RBF kernel in kernel approximation. These results demonstrated that the building time of ensemble learning, such as SW-SVR, increases more gently than that of SVR. In particular, the



(a) Testing periods: 1 month. (b) Testing periods: 6 months. (c) Testing periods: 12 months.

Fig. 3. MAPE for prediction after 6 h for each algorithm. Note that (b) and (c) are shown with log scale.

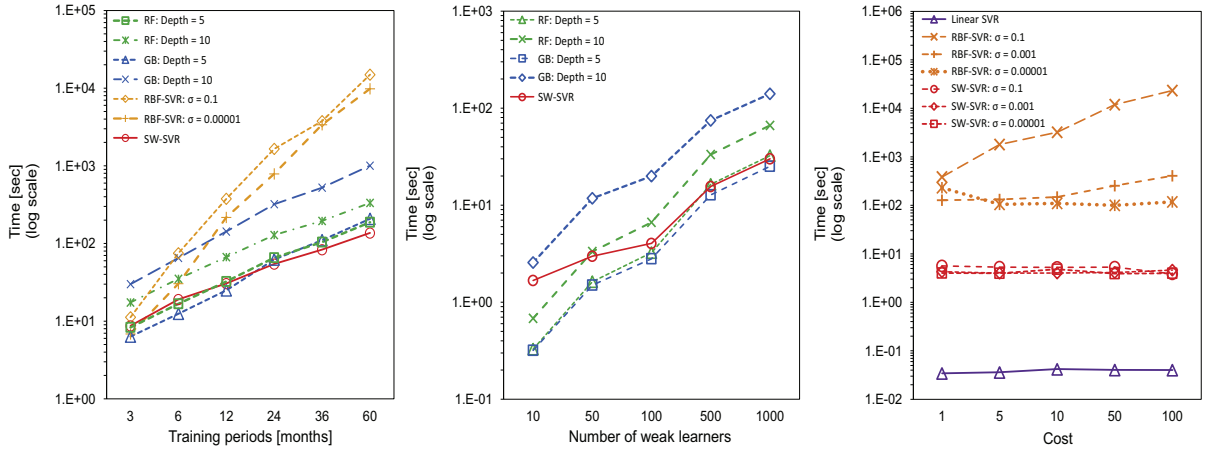


(a) Prediction horizons: 1 hour. (b) Prediction horizons: 6 hours.

Fig. 4. Average of each extraction rate by D-SDC in SW-SVR.

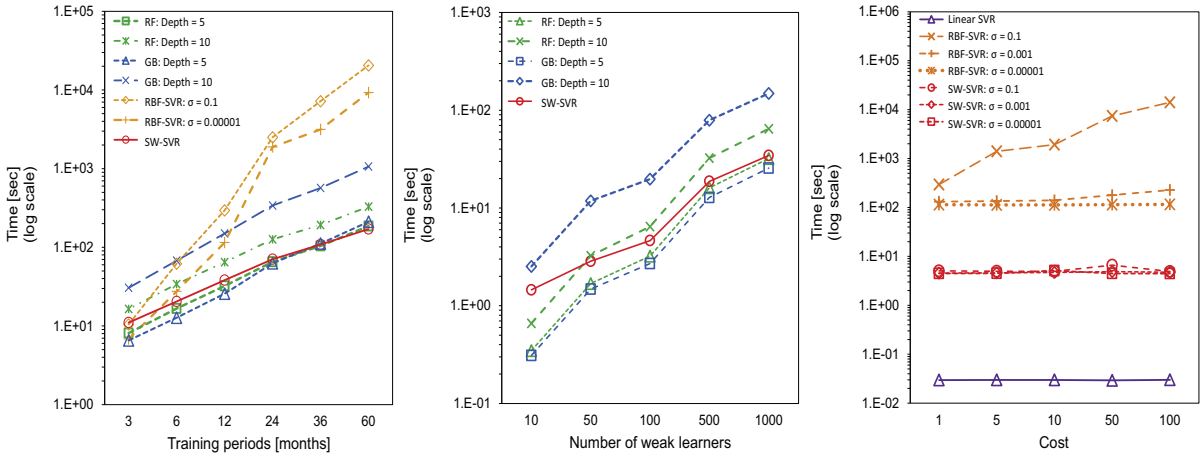
Table 1
MAPE of 10-fold cross-validation for each algorithm.

Prediction horizons	Methods										
	SW-SVR	k-NN	DT	Adaboost	Bagging	RF	GB	Linear SVR	mapped SVR	RBF-SVR	Persistent
1h	5.18608	8.59929	5.81042	11.10375	10.24014	5.57213	5.27190	5.43892	5.25274	5.16985	5.96816
6h	23.49826	26.52433	25.99290	29.93160	29.58125	25.55044	24.14987	24.68383	24.26108	20.94132	24.86800



(a) Different training periods. (b) Ensemble learning series. (c) SVR series.

Fig. 5. Building time for prediction after 1 h for each model. Note that all figures are shown with log scale.



(a) Different training periods. (b) Ensemble learning series. (c) SVR series.

Fig. 6. Building time for prediction after 6 h for each model. Note that all figures are shown with log scale.

building time of SW-SVR is shortest when the training periods become longer. In other words, the rate of building time increase of SW-SVR is the gentlest in all the methods when training data increases. These results indicate that, as mentioned above, the computational complexity of SW-SVR is less than that of conventional methods including random forest and gradient boosting. SW-SVR is effective for training of an enormous amount of data in terms of building time.

Next, Figs. 5(b) and 6(b) show the building time of the models with better performance in ensemble learning, RF, GB, and SW-SVR, when the number of weak learners was varied. Note that the cost parameter of SW-SVR was 1, σ of SW-SVR was 0.00001, and training periods were 12 months. SW-SVR needs a longer building time than RF and GB using shallow DT when the number of weak learners is lower. However, when the depth of DT becomes deeper or the number of weak learners becomes higher, SW-SVR can build the model faster than or at the same speed as RF and GB. Moreover, SW-SVR, as with RF, can be run easily in parallel environments, and it is expected that the building time of SW-SVR will become even shorter.

Finally, Figs. 5(c) and 6(c) show the building time of the models based on SVR when the parameters of SVR were varied. Note that the number of weak learners was 100, and the training periods were 12 months. These results indicate that the building time of SW-SVR is significantly shorter than that of RBF-SVR but longer than linear SVR. Meanwhile, Fig. 4 demonstrates that weak learners of SW-SVR are always built from a very small proportion of the whole training data. In particular, when prediction horizons were 1 h, the average of each extraction rate was 0.47 percent at best and 1.82 percent at worst. On the other hand, when prediction horizons were 6 h, the average of each extraction rate was 7.57 percent at best and 16.25 percent at worst. Nevertheless, the reason the computational complexity of SW-SVR is larger than linear SVR is that the increase of computational complexity due to building several models is larger. However, since the amount of training data of each weak learner reduces substantially, the computational complexity to build one model in SW-SVR reduces also. Accordingly, when the number of models one CPU builds reduces by using parallel processing, the computational complexity of the overall SW-SVR is lower than or equal to that of linear SVR. Meanwhile, as with linear SVR, SW-SVR never depends on the change of parameters related to SVR, and the building time is always a constant. As mentioned in the above discussion, the building time of SW-SVR solely depends on the number of weak learners and training periods. Therefore, SW-SVR can avoid an unexpected long building time in parameter tuning that changes each parameter variously.

These results demonstrate that SW-SVR predicts complicated micrometeorological data with the best prediction performance and the lowest computational complexity compared with standard algorithms. In particular, we found that dynamic aggregation of models built from very little extracted data by D-SDC is effective for compatibility of high prediction performance and low computational complexity. However, there are problems to be solved in SW-SVR. Firstly, the prediction performance of SW-SVR sometimes deteriorates despite an increase of training data. In particular, this problem occurred under the conditions that prediction horizons are 6 h as shown in Fig. 3. This is because data extracted by D-SDC involves unnecessary training data for highly accurate prediction. If D-SDC extracts the same data as the extracted data when training periods are shorter, the prediction performance of SW-SVR never deteriorates due to an increase of training data. Therefore, we must review both feature mapping and algorithms of D-SDC so as to avoid extracting unnecessary training data. Meanwhile, SW-SVR is based on a combination of several algorithms: kernel approximation, PLS regression, k-means, D-SDC, and linear SVR. Moreover, each algorithm has several parameters. Therefore, SW-SVR has more varied parameters, and it takes more time to tune the parameters. In this experiment, we used a grid search roughly so as to decide the parameters in a certain time. However, there is still room for improvement in the prediction performance by using other approaches such as a genetic algorithm instead of a grid search (Huang & Wang, 2006).

5. Conclusion and future work

In this paper, we proposed a new methodology for predicting micrometeorological data, SW-SVR that involves a novel combination of SVR and ensemble learning. To take the advantages of SVR and ensemble learning, SW-SVR builds several SVRs specialized for each representative data group in various natural environments by using D-SDC that extracts effective training data for specific data prediction. Moreover, to follow micrometeorological data whose characteristics always change with time, prediction of SW-SVR is based on dynamically weighted ensemble learning depending on the similarity between test data and each data specialized by weak learners. As a result of evaluation experiments using large-scale micrometeorological data, the prediction performance of SW-SVR is greater than or equal to other general methods such as SVR, RF,

and GB. Moreover, SW-SVR reduces the building time substantially compared with complicated models that have high prediction performance. We anticipate that dynamic aggregation of models built from various kinds of extracted data by D-SDC can contribute to more sophisticated studies of micrometeorological data prediction.

In future work, we should evaluate SW-SVR in more varied situations to show that SW-SVR works effectively. In particular, we will use more complicated data that consists of many features. Furthermore, when SW-SVR is applied to applications such as environmental control systems, the performance of overall applications should be evaluated. Currently, we have developed an agricultural support system using SW-SVR, which controls environments in greenhouses depending on the activity of the plants. The evaluation of the applications will describe the superiority of SW-SVR in practical use.

Acknowledgements

This study was partially supported by JST, PRESTO, and JSPS KAKENHI (26660198), Japan.

References

Antonanzas, J., Urraca, R., Martinez-de-Pison, F. J., & Antonanzas-Torres, F. (2015). Solar irradiation mapping with exogenous data from support vector regression machines estimations. *Energy Conversion and Management, 100*, 380–390.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32 http://doi.org/10.1023/A:1010933404324.

Cao, H., Naito, T., & Ninomiya, Y. (2008). Approximate RBF kernel SVM and its applications in pedestrian classification. *The 1st International Workshop on Machine Learning for Visionbased Motion Analysis - MLVMA'08*, 1–9 http://hal.archives-ouvertes.fr/inria-00325810/.

Chang, C., & Lin, C. (2011). LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*, 1–39 http://doi.org/10.1145/1961189.1961199.

Chevalier, R. F., Hoogenboom, G., McClendon, R. W., & Paz, J. A. (2011). Support vector regression with reduced training sets for air temperature prediction: A comparison with artificial neural networks. *Neural Computing & Applications, 20*(1), 151–159 Retrieved from <Go to ISI>://WOS:000286674800015.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning, 9*(2008), 1871–1874 http://doi.org/10.1038/oby.2011.351.

Freund, Y., & Schapire, R. (1997). A desicion-theoretic generalization of on-line learning and an application to boosting. *Computational Learning Theory, 55*(1), 119–139 http://doi.org/10.1006/jcss.1997.1504.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232.

Huang, C. L., & Wang, C. J. (2006). A GA-based feature selection and parameters optimizationfor support vector machines. *Expert Systems with Applications, 31*(2), 231–240 http://doi.org/10.1016/j.eswa.2005.09.024.

Japan Meteorological Agency. (n.d.).. *Japan meteorological agency* http://www.jma.go.jp/jma/indexe.html.

Kisi, O., & Cimen, M. (2012). Precipitation forecasting by using wavelet-support vector machine conjunction model. *Engineering Applications of Artificial Intelligence, 25*(4), 783–792 http://doi.org/10.1016/j.engappai.2011.11.003.

Kolokotsa, D., Pouliezos, A., Stavrakakis, G., & Lazos, C. (2009). Predictive control techniques for energy and indoor environmental quality management in buildings. *Building and Environment, 44*(9), 1850–1863 http://doi.org/10.1016/j.buildenv.2008.12.007.

Loosli, G. (2007). Comments on the core vector machines : fast SVM training on very large data sets. *The Journal of Machine Learning Research, 8*, 291–301.

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability: 1* (pp. 281–297). http://doi.org/citeulike-article-id:6083430.

Maity, R., Bhagwat, P., & Bhatnagar, A. (2010). Potential of support vector regression for prediction of monthly streamflow using endogenous property. *Hydrological Processes, 24*(7), 917–923.

Mohammadi, K., Shamshirband, S., Anisi, M. H., Alam, K. A., & Petković, D. (2015). Support vector regression based prediction of global solar radiation on a horizontal surface. *Energy Conversion and Management, 91*, 433–441.

Othman, M. F., & Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system. In *Procedia Engineering: 41* (pp. 1204–1210). http://doi.org/10.1016/j.proeng.2012.07.302.

Park, D. H., & Park, J. W. (2011). Wireless sensor network-based greenhouse environment monitoring and automatic control system for dew condensation prevention. *Sensors, 11*(4), 3640–3651 http://doi.org/10.3390/s110403640.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, É. (2011). Scikit-learn: machine learning in python. *The Journal of Machine Learning Research* 12, 2825–2830. http://doi.org/10.1007/s13398-014-0173-7.2

Platt, J. C. (1998). Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods*, 185–208 http://doi.org/10.1109/ISKE.2008.4731075.

Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems, 20*, 1177–1184 http://doi.org/10.1.1.145.8736.

Singh, K. P., Gupta, S., & Rai, P. (2013). Identifying pollution sources and predicting urban air quality using ensemble learning methods. *Atmospheric Environment, 80*, 426–437 http://doi.org/10.1016/j.atmosenv.2013.08.023.

Smith, B. A., Hoogenboom, G., & McClendon, R. W. (2009). Artificial neural networks for automated year-round temperature prediction. *Computers and Electronics in Agriculture, 68*(1), 52–61 http://doi.org/10.1016/j.compag.2009.04.003.

Suzuki, Y., Kaneda, Y., & Mineno, H. (2014). *SW-SVR improved by short-distance data collection method* (pp. 1–8) *IPSJ SIG Technical Report, 2014-MBL-73(9).*

Suzuki, Y., Kaneda, Y., & Mineno, H. (2015). Analysis of support vector regression model for micrometeorological data prediction. *Computer Science and Information Technology, 3*(2), 37–48 http://doi.org/10.13189/csit.2015.030202.

Tenenhaus, M., Vinzi, V. E., Chatelin, Y. M., & Lauro, C. (2005). PLS path modeling. *Computational Statistics and Data Analysis, 48*(1), 159–205 http://doi.org/10.1016/j.csda.2004.03.005.

Tsang, I. W., Kwok, J. T., & Cheung, P.-M. (2005). Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research, 6*, 363–392 http://doi.org/10.1111/j.1442-9993.2007.01810.x.

Urraca, R., Antonanzas, J., Martinez-de-Pison, F. J., & Antonanzas-Torres, F. (2015). Estimation of solar global irradiation in remote areas. *Journal of Renewable and Sustainable Energy, 7*(2), 023136.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*: Vol. 8. Springer http://doi.org/10.1109/TNN.1997.641482.

Wang, B. X., & Japkowicz, N. (2009). Boosting support vector machines for imbalanced data sets. *Knowledge and Information Systems, 25*(1), 1–20 http://doi.org/10.1007/s10115-009-0198-y.

Xie, Y., Li, X., Ngai, E. W. T., & Ying, W. (2009). Customer churn prediction using improved balanced random forests. *Expert Systems with Applications, 36*(3 PART 1), 5445–5449 http://doi.org/10.1016/j.eswa.2008.06.121.