

## Article

# Day-Ahead Electric Load Forecasting for the Residential Building with a Small-Size Dataset Based on a Self-Organizing Map and a Stacking Ensemble Learning Method

Jaehyun Lee <sup>1</sup>, Jinho Kim <sup>1,\*</sup> and Woong Ko <sup>2</sup>

<sup>1</sup> School of Integrated Technology, Gwangju Institute of Science and Technology, 123 Cheomdangwagi-ro, Buk-gu, Gwangju 61005, Korea; 1992jhlee@gist.ac.kr

<sup>2</sup> Research Institute for Solar and Sustainable Energies, Gwangju Institute of Science and Technology, 123 Cheomdangwagi-ro, Buk-gu, Gwangju 61005, Korea; kwoong2001@gist.ac.kr

\* Correspondence: jeikim@gist.ac.kr; Tel./Fax: +82-62-715-5322

Received: 3 February 2019; Accepted: 19 March 2019; Published: 24 March 2019



**Abstract:** Electric load forecasting for buildings is important as it assists building managers or system operators to plan energy usage and strategize accordingly. Recent increases in the adoption of advanced metering infrastructure (AMI) have made building electrical consumption data available, and this has increased the feasibility of data-driven load forecasting. Self-organizing map (SOM) has been successfully utilized to cluster a dataset into subsets containing similar data points. These subsets are then used to train the forecasting models to improve forecasting accuracy. However, some buildings may have insufficient data since newly installed monitoring devices such as AMI have no choice but to collect a limited amount of data. Using a clustering technique on small datasets could lead to overfitting when using forecasting models following an SOM network to be trained with clusters. This results in a relatively high generalization error. In this study, we propose to address this problem by employing the stacking ensemble learning method (SELM) that is well-known for its generalization ability. An experimental study was conducted using the electricity consumption data of an actual institutional building and meteorological data. Our proposed model outperformed other baseline models, which means it successfully mitigates the effect of overfitting.

**Keywords:** building electric load forecasting; self-organizing map; stacking ensemble; small-size dataset; overfitting

## 1. Introduction

Forecasting electric load demand plays a crucial role in power systems and power economics. The use of forecasting enables system operators to manage and plan energy usage, and it allows utilities to predict electricity prices more accurately. In the past, research on load forecasting was mainly performed at the grid level, and the number of studies on small-scale systems such as buildings is relatively low [1,2].

As reported by the International Energy Agency (IEA), commercial buildings consume 32% of the final electricity produced in OECD countries [3]. Therefore, it is necessary to optimally manage building electricity consumption and reduce peak loads to achieve not only economic benefits from the perspective of a building owner by saving energy but also environmental benefits by reducing the CO<sub>2</sub> emission generated in the process of energy production. In addition, distributed energy resources such as photovoltaic (PV) generators, energy storage systems (ESSs), and electric vehicle (EV) charging stations are increasingly being integrated into buildings. Moreover, demand response (DR)

has become an efficient way to balance the power system to improve its stability, such that the economic efficiency of the entire power system increases [4]. With the background of the deregulated electricity market, building owners may be able to sell their surplus electricity by adequately scheduling or controlling their energy resources and electricity loads. Sellers of surplus energy, who also consume energy, are known as prosumers. Therefore, the more accurate the forecast, the greater the benefits the building owners reap. On the contrary, uncertainty in the forecast reduces the room for efficient energy management, which in turn results in unnecessary costs. Consequently, it is necessary to optimize energy consumption and generation, and the most important part of this is future electricity consumption forecasting.

Load forecasting tasks can be categorized by their forecasting horizons. Short-term load forecasting (STLF) is used to forecast an hour, a day, or a few weeks ahead. Mid-term load forecasting (MTLF) forecasts a few months ahead, and long-term load forecasting (LTLF) forecasts more than a year ahead. STLF is usually used for hour-ahead and day-ahead scheduling, or demand response applications. MTLF is typically used for unit commitment or energy trading, and LTLF is used for system planning and decision-making relating to energy policy [5]. In this study, we develop an STLF model, and conduct day-ahead hourly load forecasting of an institutional building, which would be applicable to perform next-day building energy scheduling.

One of the strategies for forecasting the electricity load of a building is to conduct physical modeling depending on the thermodynamics of the building. This can be achieved by software such as “EnergyPlus,” which can be used to simulate building energy consumption given detailed environmental parameters such as construction details, operation schedules, climate data, and heating, ventilation, and air-conditioning (HVAC) design [6]. The disadvantage of this approach is that it is difficult to obtain such information and to effectively capture the complex characteristics of a building [7].

However, the problem mentioned above can be overcome. Because buildings are increasingly being equipped with advanced metering infrastructure (AMI), the electricity usage patterns of the building can be monitored and the data recorded. The availability of these data has made it possible to forecast future electricity consumption when used in combination with data pertaining to factors that affect the load patterns of a building. Meteorological data, such as the temperature, humidity, or solar radiation, and temporal parameters, such as the day of the week, month, or season, are well known to be associated with building electricity consumption, and many researchers have utilized these factors for forecasting [8–12]. This prompted us to determine the factors that strongly influence the electricity consumption pattern of our target building and to decide which features to select.

Previously, various data-driven techniques have been used to achieve higher performance by lowering the error rates in the load forecasting domain. They include autoregressive models such as ARMA [13] or ARIMA [14], which analyze historical data points in times series that are highly correlated with the future value, and statistical regression models such as multiple linear regression (MLR) [15,16], which estimates the relationship between the input (i.e., features) and output variables. Machine learning models such as artificial neural networks (ANNs) [17], which emulate the structure of biological neural networks and can approximate complex and nonlinear functions, and support vector regression (SVR) [18], which finds the flattest function with the largest epsilon-deviation with regard to an entire training dataset and maps the input space to higher dimension using a kernel function to process nonlinear problems, are also included. Attempts to combine different techniques for improved performance were also reported. An enhanced version of empirical mode decomposition was employed to improve the Elman neural network on small-scale building load forecast. ARIMA was used to initially forecast the linear basic part of the load, after which SVR was used to correct the deviation of former forecasting [19]. Extensive and dedicated reviews are available and well described [3,7,20,21].

The clustering technique, self-organizing map (SOM), has also been successfully employed prior to the forecasting task to improve the performance [11,22–24]. In the first stage, SOM is used to cluster

the training set into separate subsets in an unsupervised scheme, such that the subsets have similar patterns and characteristics. Subsequently, the same number of forecasting models is trained by each subset to take advantage of all past information and similar dynamic properties [11]. However, in the case of buildings for which the accumulated load data are insufficient because equipment such as AMI was only installed recently, clustering techniques have limited use. Once clustering is complete, the number of samples in each cluster is apparently smaller than the total number of samples in the entire dataset. For instance, if a dataset that has 1000 data points were clustered into four clusters, each cluster could contain 300, 200, 100, and 400 data points, which is smaller than their original size. The reduced number of samples would negatively affect forecasting models in the second stage of the forecasting procedure. Clusters with a small size could result in the “overfitting” problem, which refers to the situation in which a model fits the training set too closely or even contains noise such that it fails to predict unseen data (i.e., the testing set) and loses its generalization ability. This problem is exacerbated as the number of dimensions of a feature or the complexity of the model increases. Moreover, when dealing with small-scale systems such as buildings that have more noise signals and fluctuations, the chances that overfitting occurs increases because of intrinsic randomness (a larger-scale system aggregates a number of small-scale systems and offsets their randomness).

To overcome this obstacle, we propose applying an ensemble technique to each cluster rather than employing a single model. Ensemble techniques, which use multiple forecasting models to improve the forecasting performance beyond that possible for any single model, have been successfully used in the time series forecasting and load forecasting domains [21]. They are well known to have excellent generalization ability, which means they are able to reduce overfitting [25]. Especially, among the ensemble techniques, we used the stacking ensemble method [26]. The basic concept of stacking is that “multiple heads are better than one.” The method uses several models to produce forecasts and combines the forecasted value of each to obtain a more accurate result. The combination could involve simply averaging the results, or it could entail introducing an additional aggregator model to make a final prediction based on the predictions of all the single models. The use of this approach would enable us to mitigate the effect of the overfitting problem that results from the reduction in the number of samples when clustering precedes the forecasting task.

The main contributions of this study are summarized as follows.

1. We utilized a stacking ensemble learning method to solve the overfitting problem led by the reduced number of each cluster derived by clustering techniques such as SOM in the context of having a small-size dataset and targeting small-scale systems (in our case, a building). To show the effectiveness of our proposed method, we used a small-size dataset (covering less than 2 years) of a real institutional building.
2. This is the first attempt to combine SOM and a stacking ensemble learning method to solve building-scale STLFI.

The remainder of this paper is organized as follows. Section 2 provides a detailed literature review of SOM and ensemble techniques. Section 3 describes the techniques and data used in our proposed method. Section 4 presents the experimental study and its results, and Section 5 discusses the meaning of results. Section 6 presents our conclusions and suggestions for further research.

## 2. Related Work

In this section, we provide a detailed literature review focused on SOM and ensemble techniques, which are relevant to our proposed method.

### 2.1. Self-Organizing Map (SOM)

SOM was first proposed by Kohonen in 1990 [27]. Since then, it has been applied to various fields including the load forecasting domain. Day-ahead hourly load forecasting was conducted using historical data corresponding to an area in central Spain from 1989 to 1999 [24]. The researchers first

classified the historical data for each day according to its load profile by means of SOM. ANNs were then trained using each class. Finally, they performed the prediction by using previously trained recurrent neural networks. Their experimental results showed that the forecasting performance of their method was superior to statistical techniques in terms of accuracy and robustness.

Other researchers carried out day-ahead hourly load forecasting using a two-stage hybrid network with SOM and a support vector machine (SVM) [11]. SOM was applied to cluster the input dataset into subsets and to separate the dataset into regular days and anomalous days. SVMs were then fitted using each cluster. They found this structure to be robust against different data types and the non-stationarity of load data. They used historical load data from a New York Independent System Operator as a case study, and compared the results of using a single SVM with their proposed method. Their method outperformed the single SVM model in terms of mean absolute error and mean absolute percentage error.

López et al. [28] presented MTLF to predict the daily peak load of the next month for which they proposed the SOM-SVR model. SOM was used to cluster the historical load data into two subsets in an unsupervised manner, and two epsilon SVRs corresponding to each subset were employed to fit the data and to make predictions. They used an electricity load dataset in the European Network on Intelligent Technologies competition, and benchmarked Malaysian and PJM electricity load datasets. Their practical application results demonstrated that their proposed method far outperformed previous methods in terms of accuracy.

Nagi et al. [29] presented interesting work. They used SOM not only for clustering, for which it was used most commonly, but also as a valid forecasting engine. The experimental results showed that their model is competitive compared to results obtained with more commonly used techniques such as ANNs and SVR. They also proposed that this structure of the model can be considered as an initial approach to standardize the load forecasting process. An input selection method that was proven to significantly reduce MAPE was discussed.

Hernández et al. [22] focused on STLF at the microgrid scale. Their proposed method was composed of three stages, of which the first was pattern recognition by SOM. After the first stage, input data are represented by their best matching units, and these units are clustered by means of a k-means clustering technique in the second stage. Finally, each cluster is fed into the corresponding ANNs. The case study was performed on data from the Spanish company Iberdola. The results produced lower errors compared to other simple models without clustering.

Panapakidis et al. [23] carried out STLF in small-size loads (i.e., the buses of transmission and distribution systems). They conducted day-ahead and hour-ahead load prediction, and proposed models based on ANNs and SOM. Four models were proposed. Model A was a simple ANN model for day-ahead STLF and was constructed to perform benchmarking. Model B was a combination of Model A and SOM clustering. Model C was a simple ANN model for hour-ahead STLF for benchmarking. Model D added SOM clustering to Model C. The experimental results showed that the proposed models (B and D) produced superior forecasting accuracy.

As can be seen in the above research, SOM has been successfully applied in the field of load forecasting. However, there has been no attempt to address the overfitting problem that may occur after clustering, especially in the context of having a small-size dataset and targeting small-scale systems. This motivated us to do this research.

## 2.2. Ensemble Learning Methods

Ensemble learning methods are widely used for many machine-learning problems. As a major branch of the ensemble methods, the stacking ensemble learning machine was first proposed by Wolpert in 1992 [26]. Owing to its great generalizing ability, it has been successfully used in the field of load forecasting. This section reviews the literature mainly relating to ensemble methods, especially the stacking learning ensemble method.

Burger and Moura [30] pointed out that studies in the literature have mostly focused on specific buildings, but that a method that is widely applicable to general buildings regardless of locations, seasons, and types is yet to be proposed. They attempted to address this problem by employing the stacking ensemble learning method with moving horizon training optimization to carry out STLTF. They trained the model weights in a real-time scheme using load data streams and a moving horizon training technique, and their case study of eight buildings on the campus of the University of California showed that the proposed method outperformed the use of a single model for each building.

Ahmad et al. [31] compared the forecasting performance of ANN and random forest (RF), a tree-based ensemble technique. Their real-world experiment involved data of HVAC energy consumption of a hotel in Madrid, Spain, and they also incorporated social parameters. The results showed that the performance of the ANN was slightly more accurate than that of the RF in terms of the root mean square error (RMSE), but RF was more advantageous in that it allowed multi-dimensional complex data to be adjusted and modeled more easily, which is a typical case when modeling buildings. Therefore, they concluded that both of these techniques were nearly equally useful for building energy consumption forecasting.

Khairalla et al. [32] pointed out that forecasting a time series with a complex pattern is challenging with a single conventional statistical method. They therefore proposed the stacking multi-learning ensemble (SMLE) model for time series forecasting with various horizons to improve the forecasting accuracy. They used SVR and an ANN as base learners of the first layer, and MLR as a meta learner of the second layer. Their empirical study was conducted on global oil consumption, and the results revealed that the proposed SMLE model surpassed all the other benchmark methods in terms of accuracy, time series similarity, and directional accuracy.

Divina et al. [33] proposed a strategy for STLTF based on a stacking ensemble learning scheme. They used three base learners: an ANN, RF, and regression trees based on evolutionary computation. In addition, as a top layer meta learner, they used a gradient boosting machine (GBM) to obtain the final prediction. Their experimental study was conducted on the energy consumption in Spain for a period of more than nine years. Superior results were obtained using their proposed method compared to existing state-of-the-art techniques applied to the same dataset.

These studies verify the effectiveness of the stacking ensemble learning method and its ability to generalize. We utilized this method to mitigate the effect of overfitting after clustering.

### 3. Materials and Methods

This section presents a description and exploratory analysis of the data used and details development of our proposed model for day-ahead hourly building electricity load forecasting. First, we describe the dataset and present the exploratory data analysis (EDA), which is an investigation of the main characteristics of the dataset with visualization. The EDA enables us to determine what the data would be able to reveal; thus, the EDA allows us to pre-check the apparent patterns and shapes that we can expect before modeling or preprocessing. Subsequently, we explain the method and techniques that we utilized in our proposed model.

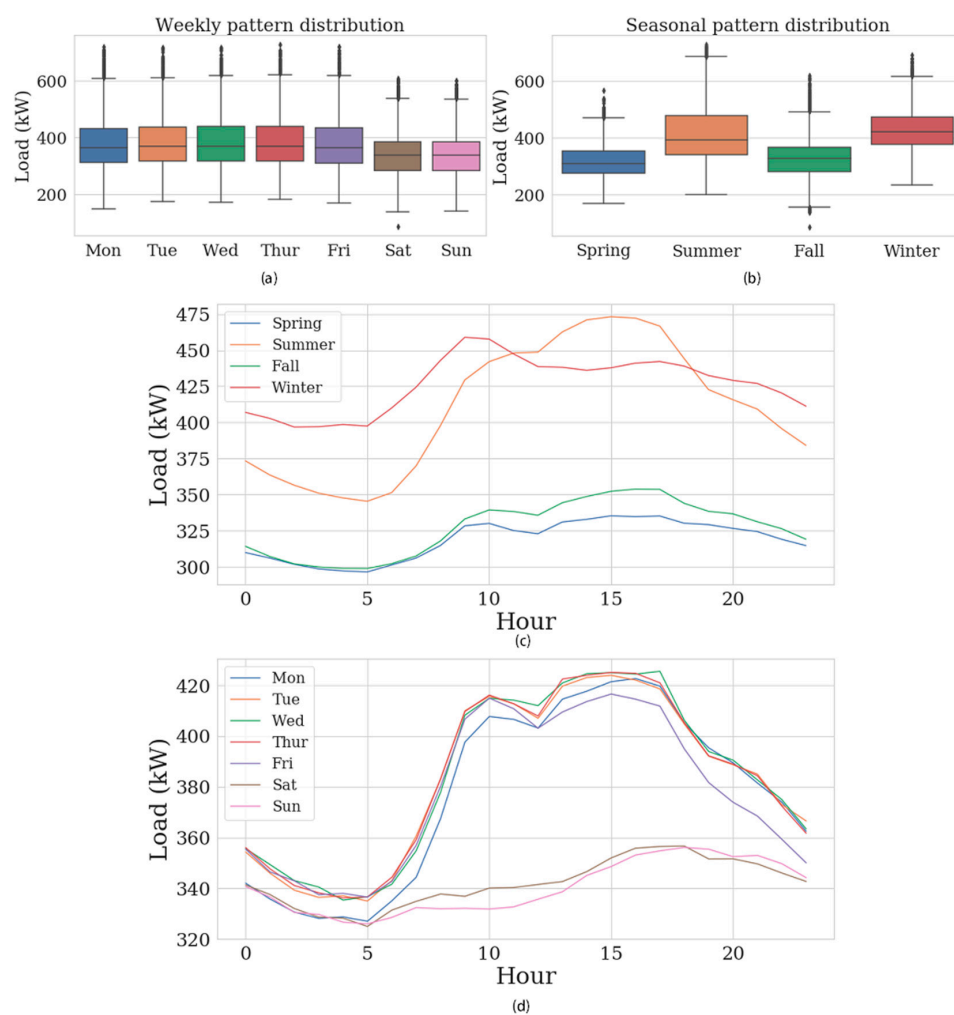
#### 3.1. Data

The load data we used in this work were obtained by recording the electricity consumption of a real institutional building every hour, from May 2016 to February 2018. The total numbers of data points and days in the dataset are 15,120 and 630, respectively. The meteorological data for the same period include the temperature, humidity, solar radiation, wind speed, and forecasted temperature. We examined the association between these variables and electricity load, and identified those that are influential. All datasets were subjected to preprocessing, which includes outlier and missing value replacement. The outliers were detected using a box plot, which means data points above and below the whiskers of the box plot were considered as outliers, and were then replaced by the values of the same hour of the neighboring days. Missing values were also inserted in the same way.



### 3.2. Exploratory Data Analysis (EDA) for Load Data

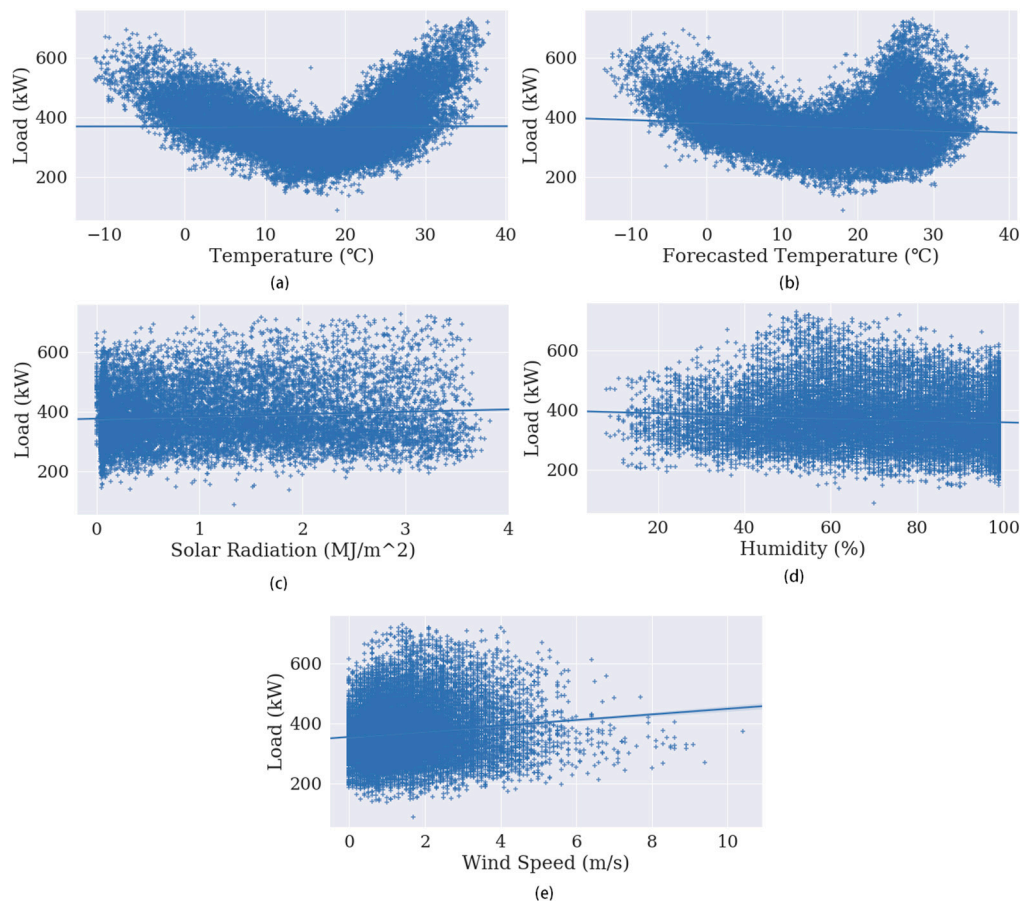
This section is devoted to investigating the distribution of the electricity load data by the season and the day of the week, using line plots and box plots. The patterns and distributions of the electricity load according to the seasons and the day of the week are shown in Figure 1a–d. We considered March to May as spring, June to August as summer, September to November as fall, and December to February as winter. As seen in the box plot describing the seasonal pattern, summer and winter display relatively larger values, and spring and fall display lower values. This is consistent with our assumptions that the cooling load is larger in summer due to the hot weather and that the heating load is larger in winter because of the cold weather. Moreover, the box plot describing the patterns according to the day of the week indicates that weekdays have almost similar values, whereas the values on weekends are relatively lower. This fact is also taken for granted because most people work on weekdays. Typically, these seasonal and weekly effects should be taken into account when selecting features and modeling. For instance, the season and day of the week are regarded as categorical variables; hence, they are usually transformed into integers [4] or one-hot encoded vectors [23]. However, in our work, we assume that SOM automatically filters out these features. Moreover, the standard SOM is only capable of processing numerical data and not categorical variables [34,35]. Hence, we decided to only consider the load patterns and meteorological data as features.



**Figure 1.** Box plot of load patterns according to the (a) season and (b) the day of the week. Line plot of load patterns according to the (c) season and (d) the day of the week.

### 3.3. Exploratory Data Analysis (EDA) for Meteorological Data

In this section, we examine the relationship between meteorological data and load data. A method that is commonly used for this purpose is Pearson's correlation coefficient (PCC), which measures the linear correlation between two variables. Because only the "linear" relationship is measured, it cannot capture the nonlinear relationship between variables. In other words, even though a clear relationship exists between the variables of interest, PCC would not reveal it. We therefore decided to simply investigate the relationship between the meteorological data and load data using the scatter plots. Figure 2 presents the scatter plots, which reveal clear relationships between the temperature and forecasted temperature with load (although these relationships are not linear). This is because the temperature and the load have an opposite relationship according to the season. In summer, the higher the temperature, the higher the consumption because of the cooling load. In contrast, in winter, the lower the temperature, the higher the consumption due to the heating load. Factors other than those plotted in Figure 2a,b do not seem to be relevant to the load; thus, we selected only temperature and forecasted temperature as features for the forecasting model.



**Figure 2.** Relationship between load data and meteorological data: (a) temperature; (b) forecasted temperature; (c) solar radiation; (d) humidity; (e) wind speed.

### 3.4. Methods

#### 3.4.1. Self-Organizing Map (SOM)

SOM is a type of ANN that expresses high-dimensional input data spaces as low-dimensional spaces (e.g., 1D, 2D, or even 3D). These lower-dimensional spaces can be recognized by humans through visualization [27].

At first glance, SOM appear to be typical ANN. However, the basic principles of SOM are completely different. SOM uses competitive learning techniques rather than error-based learning, such as gradient descent or backpropagation algorithms.

One of the most important features of SOM is that it has a topological structure. The input data are usually mapped to a 2D or 3D grid or to a hexagonal-shaped feature map composed of a certain number of neurons where the relative positions of the data points are preserved. That is, data points at similar locations are represented by neurons close to one another. Therefore, SOM can also be used as a tool for clustering high-dimensional data. The neurons in SOM are represented by the following 2D arrays:

$$M = \{m_1, m_2, \dots, m_k\} \quad (1)$$

$$m_i = [m_{i1}, m_{i2}, \dots, m_{iN}] \quad (2)$$

where  $k$  denotes the number of neurons,  $m_i$  is a vector that represents a neuron, and  $N$  denotes the dimension of each neuron (the same as that for a data point).

The detailed learning process is as follows. **Step 1.** Initialize  $k$ -neurons randomly with the same dimensions as the input data, and place them in the data space. **Step 2.** Select a single data point from the input data. **Step 3.** Find the neuron closest to the selected data point. The distance between the data point and neuron is determined based on the Euclidean distance and is calculated as follows:

$$D = \sqrt{\sum_{i=1}^N (V_i - W_i)^2} \quad (3)$$

where  $V_i$  denotes a data point, and  $W_i$  denotes the weight of the neuron. The closest neuron is referred to as the best matching unit (BMU). **Step 4.** Move the BMU closer to the selected data point. The distance of the movement is determined by the learning rate, which decays exponentially over time. **Step 5.** Move the neighboring neurons near the BMU (i.e., within a certain radius that decreases over time) in inverse proportional to the distance from the BMU. The influence rates of neighboring neurons are determined by their Gaussian curve, such that neurons that are closer to the BMU are influenced more than the more distant ones. **Step 6.** Repeat Steps 1–6 for all data points. **Step 7.** Once the above process is complete, update the learning rate and the neighborhood radius. The learning rate  $L$ , neighborhood radius  $\sigma$ , and the influence rate of neighboring neurons  $\theta$  are dependent on time, that is, the number of iterations. The above variables are defined as follows:

$$L(n) = L_0 \exp\left(-\frac{n}{\alpha_L}\right) \quad (4)$$

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\alpha_\sigma}\right) \quad (5)$$

$$\theta(n) = \exp\left(-\frac{D^2}{2\sigma^2(n)}\right). \quad (6)$$

The weights of these variables are updated using the following equation:

$$W(n+1) = W(n) + \theta(n)L(n)(V(n) - W(n)). \quad (7)$$

At the end of this process, when the neurons are finally located in the data space, they are similar in distribution to the original data points. That is, the grid of neurons adapts to the topological shape of the data points. This result allows us to visualize a dataset using a U-matrix or frequency map—the distinct groups of neurons represent underlying clusters.

SOM has been successfully applied to the load forecasting problem to cluster data points that include load profiles within subsets containing similar ones. By doing this, a predictive model can be trained better than it can without clustering. Examples are discussed in Section 2. We will utilize this



method prior to a forecasting task, as in the literature, and we will also apply the stacking ensemble learning method in order to address the overfitting problem after clustering due to the reduced number of data samples in each cluster. This will be discussed in the next section.

### 3.4.2. The Stacking Ensemble Learning Method (SELM)

The ensemble method is one of the most successful approaches for performing time series forecasting tasks [36,37]. It refers to a method that combines multiple predictive learning algorithms to obtain results superior to those that could be obtained from a single algorithm. For instance, in a real-world application, it is not easy to generalize a given dataset because it has numerous underlying patterns and features. Although a particular model might be able to capture certain patterns or features, other models may not be able to achieve this. Likewise, different models capture the different patterns and features of the dataset. If multiple models learn patterns from the dataset and if their predictions are appropriately combined, it is possible to obtain more accurate results than a single model would be able to produce.

Mehta et al. [38] present a clear explanation as to why ensemble methods are effective. The first reason is statistical. When the training dataset is too small, a situation that arises after clustering, a learning algorithm can typically identify several models in the hypothesis space that show the same performance with respect to the training data. If the models are not correlated, averaging them reduces the risk of choosing the wrong hypothesis. The second reason is computational. Many machine learning algorithms, such as ANNs, present a risk of falling into local optima. Therefore, the algorithms are greatly affected by the initialization of the weights. Ensemble methods mitigate this problem by taking multiple models built from many different starting points resulting in an improved approximation of the true function. The last reason is representational. In most cases, because the training dataset has a finite size, the true function cannot be completely represented by any models in the hypothesis space. By combining several models, it may be possible to expand the representable function space to obtain a more effective model for the true function.

The most commonly used ensemble methods are the following:

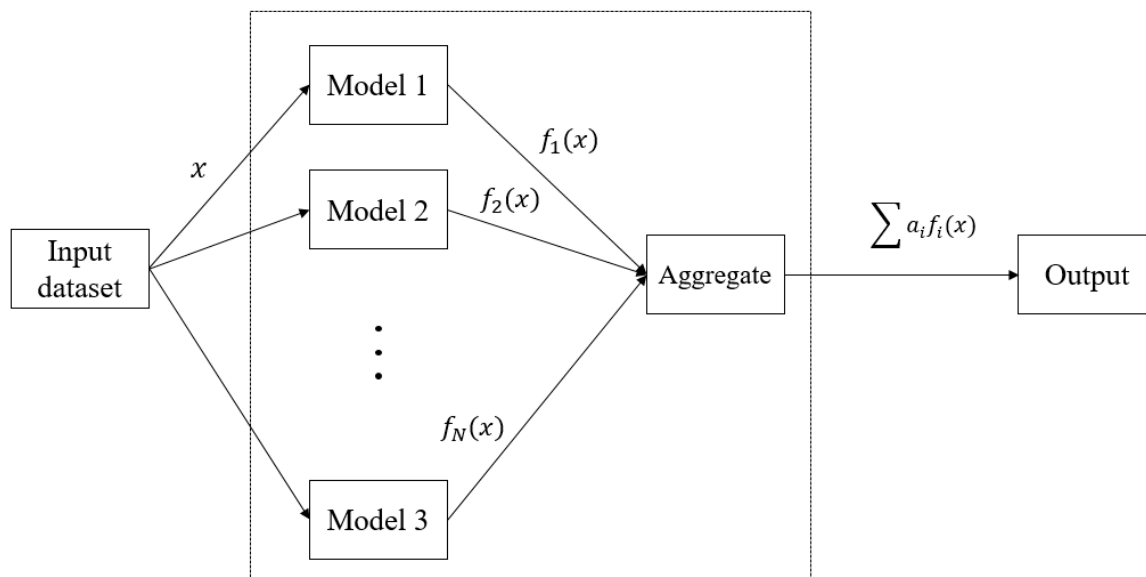
- bootstrap aggregating (bagging);
- boosting;
- stacking.

Bagging creates several training subsets. This is achieved by sampling same-sized subsets from the training dataset and then by training models with different subsets and aggregating the results, which have a parallel learning shape [39]. Popular bagging algorithms are bagging meta-estimators, which follow the general bagging rules that are briefly explained above, and RFs, which are extensions of bagging estimator algorithms. Decision trees are usually used as the base estimators, and, unlike bagging meta-estimators, RFs randomly choose a set of features that are used to determine the best split at each node of the decision tree [40].

Boosting is, unlike bagging, a sequential learning technique. The basic concept of boosting is that new models are improved by weighting the training points that the previous model was unable to predict correctly. That is, the first subset is created from the original dataset, and the base model is then trained by assigning the same weights to all training points. Next, errors are calculated, and weak areas that produce inaccurate predictions are identified. Subsequently, a new model is trained using the modified training set, to assign more weight to the weak areas. Boosting has been shown to improve the performance compared to bagging, but it can over fit the training data. The most common boosting algorithm is adaptive boosting (ADB) [41]. Other boosting-based methods include the gradient boosting machine (GBM) [42] and extreme gradient boosting (XGB) [43].

The basic concept of the SELM is that “multiple heads are better than one.” It is an ensemble technique that uses an aggregating model that learns the way in which the best combination of predictions can be made by multiple models or that simply averages them [30]. Figure 3 shows

a conceptual structure of the SELM. It is based on the fact that no model is perfect, which means models always produce errors. Additionally, because different models tend to capture the data from different points of views, the SELM takes advantage of leveraging the strength of each model to build a more robust predictive model. In this work, assuming the context of having a small-size dataset and targeting small-scale system, we use a simple averaging method for the aggregator rather than employing another trainable model because, when using a trainable model, we need a validation set or a cross-validation technique to train the model, which is inefficient for a small-size dataset.



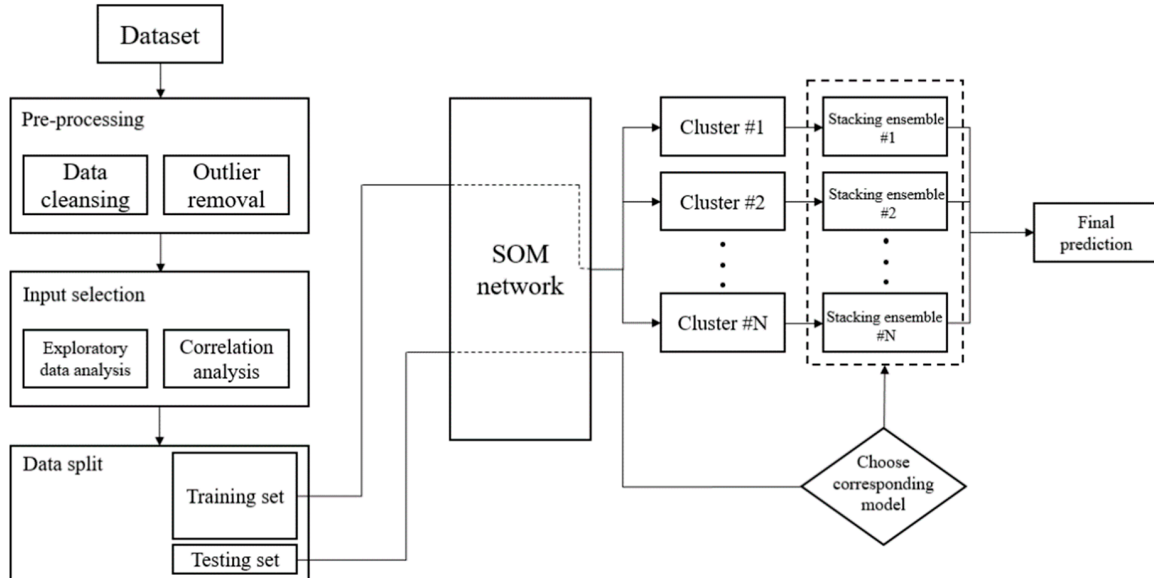
**Figure 3.** Conceptual diagram of the stacking ensemble learning method.

After clustering, the number of samples in each cluster is smaller than the number of samples in the original dataset. This could be problematic, because the lack of training data in a forecasting task can lead to overfitting [44,45]. Several approaches have been used with small-size samples; the first of these approaches would be to reduce the complexity of the model. The second approach would be to reduce the dimensionality of the dataset using dimension reduction techniques such as principle component analysis (PCA), linear discriminant analysis (LDA), or SOM. The third approach would involve creating virtual samples by means of resampling techniques [46]. The last approach would entail the use of ensemble learning methods, which have been known to mitigate overfitting because of their ability to generalize. We decided to follow the last approach, especially the SELM, which was found to be effective for small-size datasets [47].

### 3.5. The Proposed Framework

The structure of the proposed model is shown in Figure 4. The dataset is first subjected to pre-processing, where data cleansing and outlier removal take place. The following stage is the input selection stage, where EDA and correlation analysis are employed to determine the data to be used as features. Next, the dataset is split into training and testing sets. Subsequently, the process is divided into a training phase and a test phase. In the training phase, the training dataset is divided into  $N$  clusters by SOM, and each cluster is used to train a corresponding forecasting model. Our proposed model uses the SELM for forecasting, but single models, such as MLR, ANN, and SVR, and ensemble models, such as GBM, ADB, and XGB, are also tested to demonstrate the superiority of our model. After the training phase ends, the forecasting performance of the model is evaluated in the test phase. First, we determine the clusters to which each data point in the testing dataset belongs by calculating its BMU, and we then forecast the future load using models trained by the corresponding clusters. The novelty of this method compared to the methods in the literature is that we use the SELM after

clustering in order to mitigate the effect of overfitting caused by the reduced number of samples in each cluster. We did not develop the techniques used in our proposed method. We devised the strategy of handling the overfitting problem, fully utilizing existing techniques. The experimental results demonstrating the effectiveness of our proposed method will be discussed in Section 4.



**Figure 4.** The proposed framework that describes the whole process from data pre-processing to final prediction produced by clustering using SOM and the stacking ensemble learning method.

### 3.6. Implementation Details

#### 3.6.1. Input Data

We chose load profiles of previous days and meteorological data as input data for the forecasting models. To build an accurate forecasting model, it is desirable to use factors that are closely related to the load. Obviously, the load profile of the targeted day is highly correlated with that of the previous day and the same day of the previous week, represented by  $L(d-1)$  and  $L(d-7)$ , respectively, where  $d$  denotes the targeted day. As discussed in Section 3.3, we decided to use the temperature and forecasted temperature as features among the various meteorological data, because they were ultimately the most closely correlated with the load through EDA. The entire dataset can be seen in Table 1. The dataset is divided into two parts—the training set and the testing set. The training set contains the first 441 days of the whole period, and the testing set contains the remaining. In addition, the dataset is min-max normalized using Equation (1) before entering SOM, where  $x$  denotes the data vector or the matrix, and  $z$  is the normalized vector or matrix.

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (8)$$

The input data that were selected are listed below:

- 24 h load profile of the previous day:  $L(d-1)$ ;
- 24 h load profile of the same day of the previous week:  $L(d-7)$ ;
- Average, maximum, and minimum temperature of the previous day:  $T_{\text{avg}}$ ,  $T_{\text{max}}$ ,  $T_{\text{min}}$ ;
- Forecasted average, maximum, and minimum temperature of the targeted day:  $T_{f, \text{avg}}$ ,  $T_{f, \text{max}}$ ,  $T_{f, \text{min}}$ .

**Table 1.** Input data matrix.

Column	1–24	25–48	49–51	52–54
Data	$L(d - 1)$	$L(d - 7)$	$T_{avg}, T_{max}, T_{min}$	$T_{f, avg}, T_{f, max}, T_{f, min}$

### 3.6.2. Configurations for Self-Organizing Map (SOM)

The number of neurons in the SOM grid was determined by Equation (2) [48].  $N$  denotes the number of neurons, and  $M$  is the number of data samples.

$$N \approx 5 * \sqrt{M}. \quad (9)$$

Our dataset contains 630 days; hence, the number of neurons was calculated as 11. We used a rectangular topological shape, and we set the learning rate as 0.5, sigma as 1.0, and the maximum number of iterative steps as 200. Mostly, the default values specified by the package designer were used, because it was effective with these values.

### 3.6.3. Configurations for Forecasting Models

The hyperparameters of the various predictive models that we used were mostly determined by trial and error with experiments. For ANNs, we established three hidden layers, each of which has 100, 200, and 200 hidden units. The learning rate was 0.001, the maximum number of iteration was 500, and a rectified linear unit (ReLU) [49] function was used as activation function. Further, for the SVR, the penalty parameter  $C$  was 0.25, the kernel coefficient was 0.1, epsilon, which identifies the epsilon-tube distance, was 0.005, and the kernel function was a radial basis function. For GBM, the subsample rate was set to 0.8, the number of sequential trees was 150, and the learning rate was 0.05. Other configurations for models such as RF, ADB, and XGB used the default values specified by the package designer because they worked fine with them.

### 3.6.4. Implementation Tools

The overall process performed with Python 3.5.3, and its package Pandas 0.23.4 [50], scikit-learn 0.20.0 [51], and R 3.5.2 and its package kohonen 3.0.8 [52].

## 4. Experimental Results

### 4.1. Experimental Setup

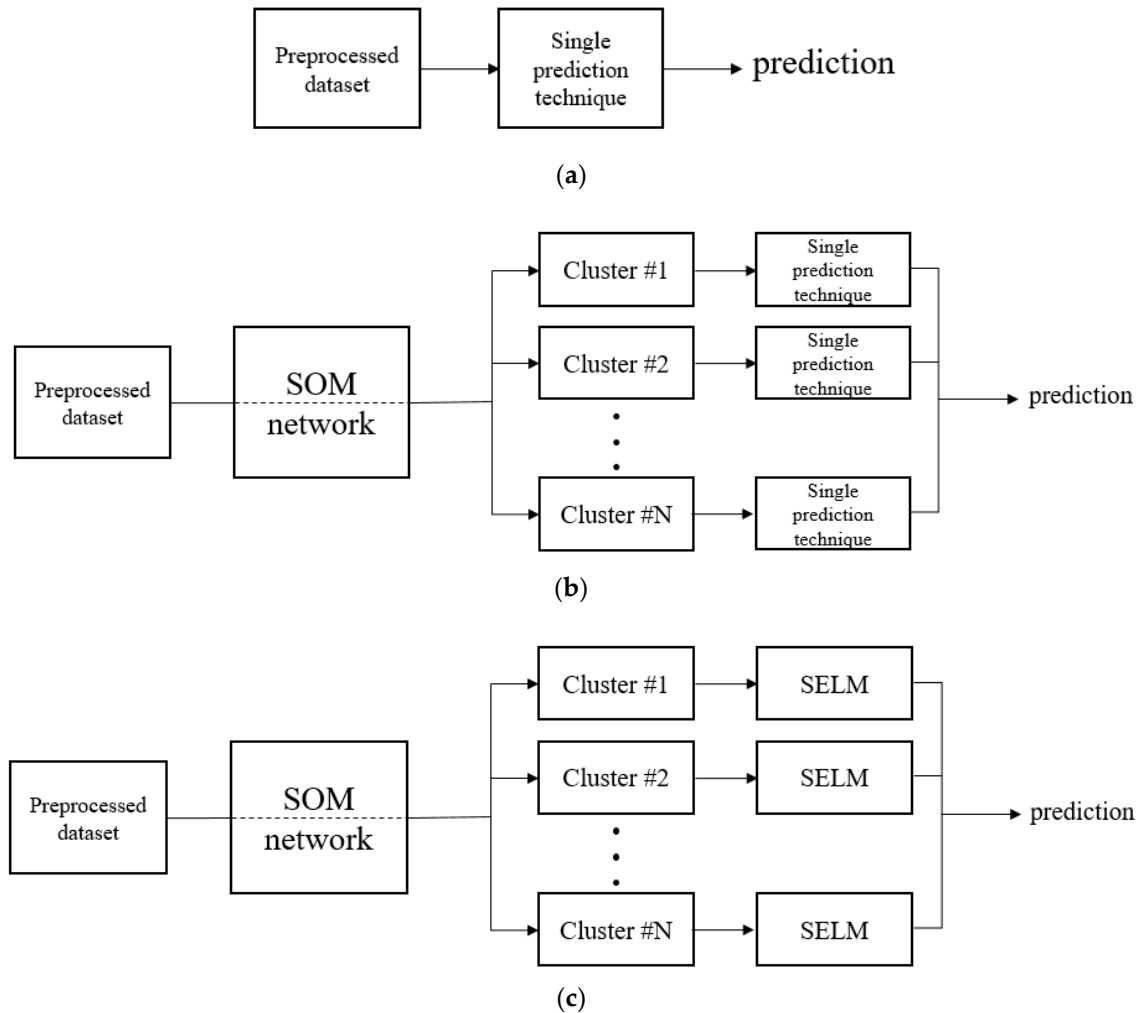
We evaluated the forecasting performance of various techniques and models using the mean absolute percentage error (MAPE) as a metric for calculating the error. MAPE is a measure indicating the extent to which the predicted value differs from the observed value. The formula is as follows:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - F_i}{A_i} \right| \quad (10)$$

where  $A_i$  is the actual value, and  $F_i$  is the forecasted value.

In this section, we present the experimental results. To demonstrate the advantages of our proposed method that utilized SOM and the SELM, we compared forecasting performances of three different models: Model 1, Model 2, and Model 3. **Model 1** is a baseline model that uses a single prediction technique to produce an hourly load profile of the next day without using any other techniques such as clustering or ensemble. We used MLR, an ANN, SVR, RF, ADB, XGB, and a GBM for the single prediction technique. **Model 2** uses the same techniques as Model 1 for the forecasting tasks; however, before the forecasting task, the dataset is clustered into several clusters using SOM. The same number of forecasting models comprising homogeneous prediction techniques is then trained on the clusters. When it makes a prediction, a data point in the test dataset is assigned to one

of the clusters and the corresponding forecasting model predicts an hourly load profile of the next day. **Model 3** also includes the clustering process, but it applies the SELM in the forecasting task instead of single prediction techniques. Model 3 is our proposed method. The models are represented in the Figure 5.



**Figure 5.** (a) Model 1, (b) Model 2, and (c) Model 3. The “preprocess dataset” in the figure is the result of the original dataset having gone through the process of the front part of Figure 4.

#### 4.2. Experimental Results

The forecasted performance results of Model 1, Model 2, and Model 3 are presented in Table 2. The MAPE values of Model 2 and Model 3 in this table represent the best results obtained from the forecasting experiment according to the number of clusters. The overall results for this experiment are shown in Tables 3 and 4. In Table 3, the forecasted performance results of Model 2 using various prediction techniques are shown according to the number of clusters ranging from 2 to 6. Table 4 illustrates the forecasting performance results of Model 3. We measured the MAPE changing the number of clusters from 2 to 6 as we did with Model 2. In addition, we tested all possible combinations of prediction techniques used as components of the SELM and selected the best one. The best combination can be found in Table 4.



**Table 2.** Resulting MAPE of Model 1, Model 2, and Model 3 for various machine learning techniques.

Technique	Model 1	Model 2	Model 3
MLR	7.33	7.5	6.4 (By stacking MLR, GBM, and ANN)
ANN	6.86	6.77	
SVR	7.22	7.01	
RF	7.52	7.16	
ADB	7.64	7.43	
XGB	6.9	7.28	
GBM	6.85	7.27	

**Table 3.** Forecasting performance of Model 2 in terms of MAPE according to the number of clusters.

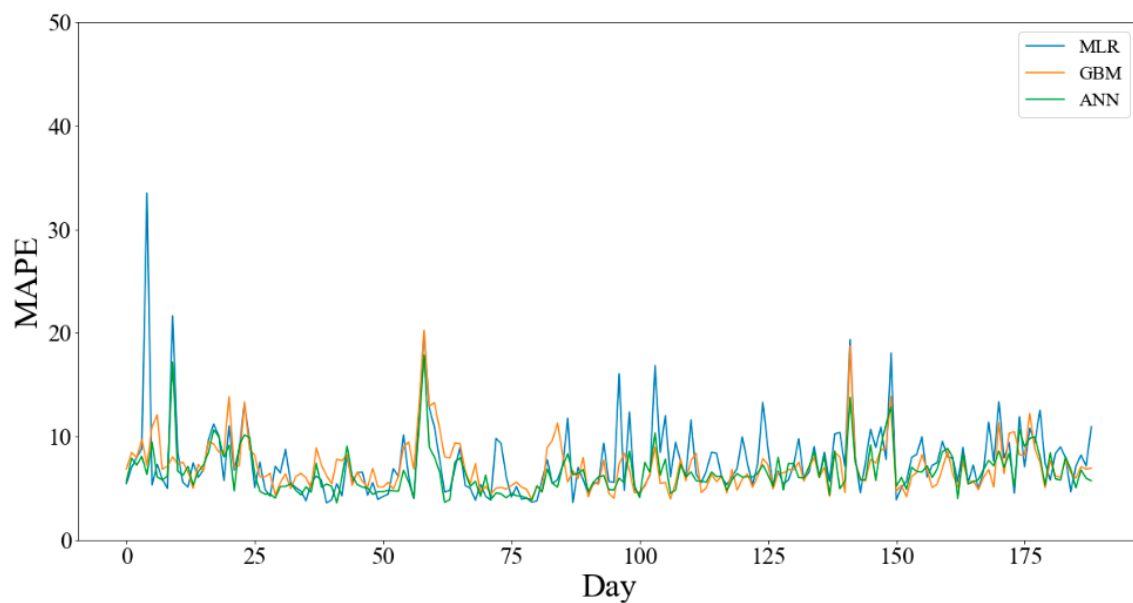
# of Clusters	MLR	ANN	SVR	RF	ADB	GBM	XGB
2	7.5	6.77	7.01	7.68	7.64	7.27	7.28
3	8.06	6.94	7.68	7.16	7.43	7.27	7.42
4	10.25	6.95	7.84	7.71	7.53	7.37	7.52
5	10.85	7.03	8.09	7.93	7.58	7.49	7.66
6	14.15	7.21	8.14	8.04	7.74	7.65	7.79

**Table 4.** Forecasting performance of Model 3 in terms of MAPE according to the number of clusters.

# of Clusters	Best Combination	Model 3
2	MLR/GBM/ANN	6.4
3	MLR/SVR/GBM/ANN/XGB	6.7
4	GBM/ANN	6.81
5	GBM/ANN	6.9
6	GBM/ANN	7.06

The forecasted performance results of Model 1 and Model 2 can be compared referring to the first and second columns of Table 2. The results indicate that, when SOM is used for clustering, performance is sometimes slightly more accurate, almost the same, or even less accurate than the results of Model 1. This contradicts the assumption that the forecasting performance improves when we take advantage of the reduced effect of overfitting that results from clustering the dataset into several clusters that contain data points with similar characteristics. The same number of forecasting models is trained on these clusters to produce predictions, which is the reason why SOM has been utilized in the literature. This occurs because clustering reduces the number of data samples in each model, thereby indicating that the chances of overfitting are very high, and the forecasting performance is rather poor for techniques that are sensitive to the number of data samples such as MLR.

Figure 6 shows the forecasting performance of the ANN, the GBM, and ADB in Model 2 on every data point in the testing set. As shown in Figure 6, the forecasting performance of each technique varies depending on the data points. This demonstrates that a certain technique does not produce more accurate predictions than others at all times. That is, for every data point in the testing dataset, the best performing technique always differs point by point. Even if the overall forecasting performance of a technique is worse than other models, at some points it may outperform the other techniques. Hence, trying to combine several techniques to produce improved results is reasonable. In this context, the SELM combines the predictions produced by individual techniques with a weighting coefficient to improve the final prediction. To determine the weighting, various methods ranging from simple averaging to regularized linear regression and even an ensemble method [33] have been used. Our study uses a simple averaging method because other methods require a validation set or cross-validation process to train the aggregating model and because we assumed a small-size dataset context.



**Figure 6.** MAPE over time for the ANN, the GBM, and ADB.

As can be seen from the third column in Table 2, the effect of overfitting successfully reduced by using the SELM, which enables us to select a group of models that more effectively complement each other's weaknesses when combined. In the experiments, MLR, the GBM, and the ANN were selected as the best group that produced the best performance, with the resulting MAPE being lower than that for all other cases. Considering that we performed building scale load forecasting, which exhibits more fluctuation and is noisier compared to larger-scale systems, and that we even used a small-size dataset (covering less than two years) with the assumption that smart metering equipment such as AMI was only recently installed in the building resulting in a lack of data, our results are fairly good given the conditions imposed.

As presented in Table 4, the ANN and the GBM were always included in the best group that performed the best, even though the GBM failed to outperform the other techniques alone. This means that these two techniques complement each other's weaknesses well, resulting in a more effective forecasting model. Therefore, when applying the SELM, it is reasonable to figure out which predictions of certain techniques are not highly correlated and select the least correlated ones. On the other hand, as shown in Figure 7, performance tends to decrease as the number of clusters increases. This occurs because, as the number of clusters increases, the number of data samples in each cluster decreases, which negatively affects model performance. This limitation can be overcome if a larger amount of data is available. Consequently, in this study, the number of clusters that produces the best forecasting performance is two. Examples of load forecasts for a certain week produced by the SELM model are presented in Figure 8.

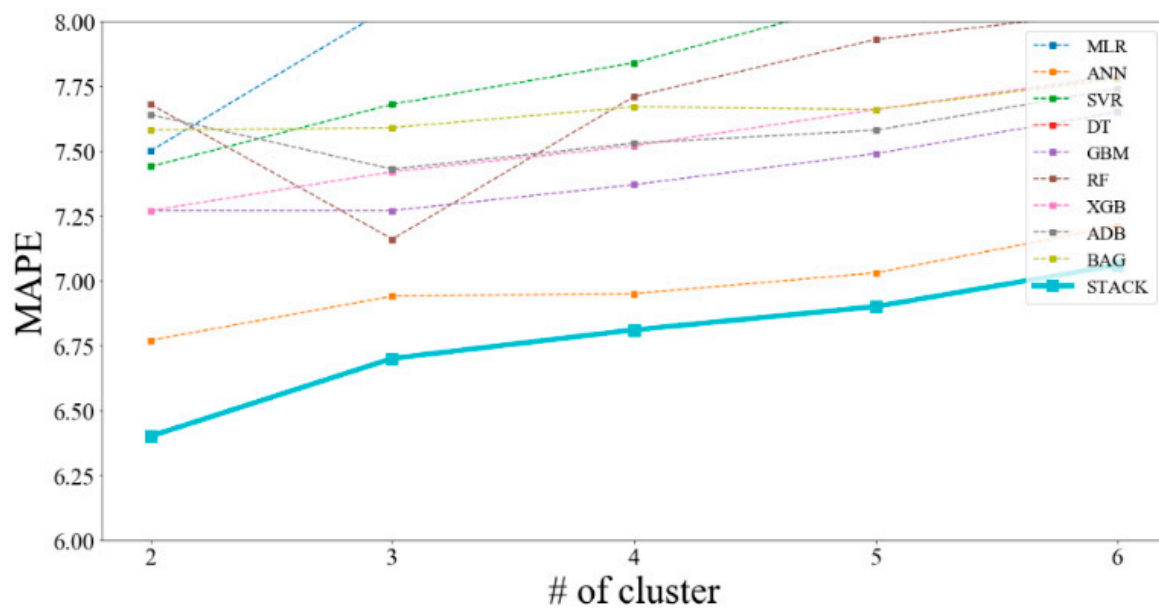


Figure 7. MAPE for each model according to the number of clusters.

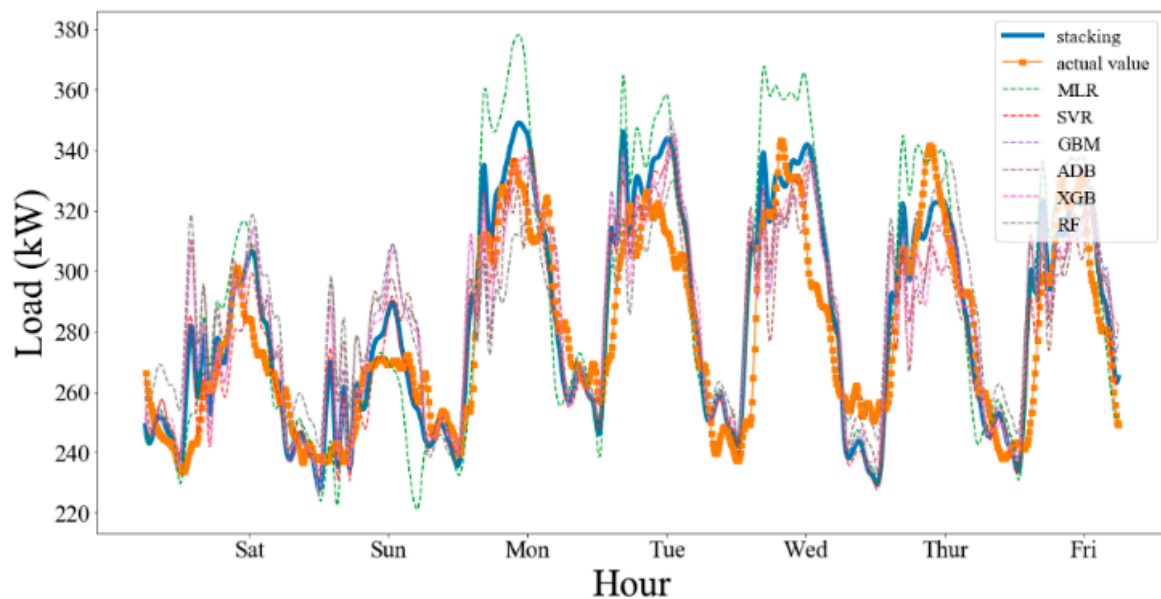


Figure 8. Examples of load forecasts for a certain week produced by the first layer models and the stacking ensemble learning model.

## 5. Discussion

In this study, we tried to address the overfitting problem caused by clustering in the load forecasting process, assuming the condition of having only a small-sized dataset and targeting a small-scale system such as a building. SOM has been successfully utilized in the load forecasting domain, and it helps to improve the forecasting performance of a model. However, in previous studies, it has not been validated in the context of a small-sized dataset (less than 2 years) and a small-scale system, which includes a building that employs a data collection system such as AMI. We argued that, in this situation, there would be an overfitting problem due to the number of data samples in each cluster. This can be observed in the first and second columns of Table 2, which present the forecasting accuracy of Models 1 and 2, and the difference between the two models regards whether clustering with SOM was applied. The results show that some of them slightly improved, but some techniques

vulnerable to overfitting were even worse as we expected. We overcome this problem with the SELM, which has been known to have an excellent generalization ability. The results of adopting the SELM (Model 3) can be found in the third column of Tables 2 and 4, which outperforms Models 1 and 2. This means the SELM worked as well as we expected.

## 6. Conclusions and Future Works

This paper presented a method to solve the day-ahead hourly building load forecasting problem. The method proposed in this paper combines self-organizing map (SOM) with a clustering technique and finally employs the stacking ensemble learning method (SELM). We attempted to address the overfitting problem due to the reduced number of samples after clustering. In particular, when the size of the original dataset is small and a small system such as a building is targeted, overfitting is more likely to occur. The effect of overfitting may result in a high generalization error. This can be seen in our experimental results. We employed the SELM to mitigate the effect of overfitting because it is known to have the ability to generalize by combining multiple models. The experimental results showed that the SELM achieved higher forecasting accuracy. Even with a small dataset (data covering less than two years, which is the smallest size reported in the literature to the best of our knowledge) and a small-scale system, which is noisier with more fluctuation than a larger system, our proposed model succeeded in producing a lower error compared to any individual model. The limitations of this study are as follows: hyperparameters were not found effectively (they were found manually), and the techniques used as components of the SELM were also selected by experiments. Therefore, a possible future research direction is to develop an effective hyperparameter tuning method that can be harmonized with clustering and the SELM. In addition, a method that automatically selects the best performing group would be more effective. More sophisticated feature selection and additional techniques to mitigate the effect of overfitting or to better improve forecasting accuracy will be studied.

**Author Contributions:** All the authors contributed to this work. J.L. designed the study, performed the literature review and the analysis, and wrote the paper. W.K. contributed to the conceptual approach and thoroughly revised the paper. J.K. led and supervised the research.

**Funding:** This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20181210301380) and by GTI research fund of GIST (GK08810).

**Acknowledgments:** In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Papalexopoulos, A.D.; Hesterberg, T.C. A regression-based approach to short-term system load forecasting. *IEEE Trans. Power Syst.* **1990**, *5*, 1535–1547. [\[CrossRef\]](#)
2. Park, D.C.; El-Sharkawi, M.; Marks, R.; Atlas, L.; Damborg, M. Electric load forecasting using an artificial neural network. *IEEE Trans. Power Syst.* **1991**, *6*, 442–449. [\[CrossRef\]](#)
3. Yildiz, B.; Bilbao, J.; Sproul, A. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renew. Sustain. Energy Rev.* **2017**, *73*, 1104–1122. [\[CrossRef\]](#)
4. Lu, R.; Hong, S.H. Incentive-based demand response for smart grid with reinforcement learning and deep neural network. *Appl. Energy* **2019**, *236*, 937–949. [\[CrossRef\]](#)
5. Hong, T.; Fan, S. Probabilistic electric load forecasting: A tutorial review. *Int. J. Forecast.* **2016**, *32*, 914–938. [\[CrossRef\]](#)
6. Li, X.; Wen, J. Building energy consumption on-line forecasting using physics based system identification. *Energy Build.* **2014**, *82*, 1–12. [\[CrossRef\]](#)
7. Amasyali, K.; El-Gohary, N.M. A review of data-driven building energy consumption prediction studies. *Renew. Sustain. Energy Rev.* **2018**, *81*, 1192–1205. [\[CrossRef\]](#)

8. Alobaidi, M.H.; Chebana, F.; Meguid, M.A. Robust ensemble learning framework for day-ahead forecasting of household based energy consumption. *Appl. Energy* **2018**, *212*, 997–1012. [CrossRef]
9. Hong, T. Short Term Electric Load Forecasting. Available online: <https://repository.lib.ncsu.edu/bitstream/handle/1840.16/6457/etd.pdf> (accessed on 30 June 2018).
10. Chen, B.J.; Chang, M.W.; Lin, C.J. Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001. *IEEE Trans. Power Syst.* **2004**, *19*, 1821–1830. [CrossRef]
11. Fan, S.; Chen, L. Short-Term Load Forecasting Based on an Adaptive Hybrid Method. *IEEE Trans. Power Syst.* **2006**, *21*, 392–401. [CrossRef]
12. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Calavia, L.; Carro, B.; Sánchez-Esguevillas, A.; Cook, D.J.; Chinarro, D.; Gómez, J. A Study of the Relationship between Weather Variables and Electric Power Demand inside a Smart Grid/Smart World Framework. *Sensors* **2012**, *12*, 11571–11591. [CrossRef]
13. Pappas, S.S.; Ekonomou, L.; Karamousantas, D.C.; Chatzarakis, G.E.; Katsikas, S.K.; Liatsis, P. Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models. *Energy* **2008**, *33*, 1353–1360. [CrossRef]
14. Hor, C.-L.; Watson, S.J.; Majithia, S. Daily load forecasting and maximum demand estimation using ARIMA and GARCH. In Proceedings of the Probabilistic Methods Applied to Power Systems, Stockholm, Sweden, 11–15 June 2006.
15. Braun, M.R.; Altan, H.; Beck, S.B.M. Using regression analysis to predict the future energy consumption of a supermarket in the UK. *Appl. Energy* **2014**, *130*, 305–313. [CrossRef]
16. Hong, T.; Wang, P.; Willis, H.L. A naïve multiple linear regression benchmark for short term load forecasting. In Proceedings of the Power and Energy Society General Meeting, Detroit, MI, USA, 24–29 July 2011.
17. Kuo, P.-H.; Huang, C.-J. A High Precision Artificial Neural Networks Model for Short-Term Energy Load Forecasting. *Energies* **2018**, *11*, 213. [CrossRef]
18. Ceperic, E.; Ceperic, V.; Baric, A. A Strategy for Short-Term Load Forecasting by Support Vector Regression Machines. *IEEE Trans. Power Syst.* **2013**, *28*, 4356–4364. [CrossRef]
19. Nie, H.; Liu, G.; Liu, X.; Wang, Y. Hybrid of ARIMA and SVMs for Short-Term Load Forecasting. *Energy Procedia* **2012**, *16*, 1455–1460. [CrossRef]
20. Wei, Y.; Zhang, X.; Shi, Y.; Xia, L.; Pan, S.; Wu, J.; Han, M.; Zhao, X. A review of data-driven approaches for prediction and classification of building energy consumption. *Renew. Sustain. Energy Rev.* **2018**, *82*, 1027–1047. [CrossRef]
21. Wang, Z.; Srinivasan, R.S. A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. *Renew. Sustain. Energy Rev.* **2017**, *75*, 796–808. [CrossRef]
22. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A.; Lloret, J. Artificial neural networks for short-term load forecasting in microgrids environment. *Energy* **2014**, *75*, 252–264. [CrossRef]
23. Panapakidis, I.P. Clustering based day-ahead and hour-ahead bus load forecasting models. *Int. J. Electr. Power Energy Syst.* **2016**, *80*, 171–178. [CrossRef]
24. Marin, F.; Garcia-Lagos, F.; Joya, G.; Sandoval, F. Global model for short-term load forecasting using artificial neural networks. *IEE Proc.-Gener. Transm. Distrib.* **2002**, *149*, 121–125. [CrossRef]
25. Ren, Y.; Zhang, L.; Suganthan, P.N. Ensemble Classification and Regression-Recent Developments, Applications and Future Directions [Review Article]. *IEEE Comput. Intell. Mag.* **2016**, *11*, 41–53. [CrossRef]
26. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259. [CrossRef]
27. Kohonen, T. The self-organizing map. *Proc. IEEE* **1990**, *78*, 1464–1480. [CrossRef]
28. Nagi, J.; Yap, K.S.; Nagi, F.; Tiong, S.K.; Ahmed, S.K. A computational intelligence scheme for the prediction of the daily peak load. *Appl. Soft Comput.* **2011**, *11*, 4773–4788. [CrossRef]
29. López, M.; Valero, S.; Senabre, C.; Aparicio, J.; Gabaldon, A. Application of SOM neural networks to short-term load forecasting: The Spanish electricity market case study. *Electr. Power Syst. Res.* **2012**, *91*, 18–27. [CrossRef]
30. Burger, E.M.; Moura, S.J. Building Electricity Load Forecasting via Stacking Ensemble Learning Method with Moving Horizon Optimization. Available online: <https://escholarship.org/uc/item/6jc7377f#author> (accessed on 10 September 2018).
31. Ahmad, M.W.; Mourshed, M.; Rezgui, Y. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy Build.* **2017**, *147*, 77–89. [CrossRef]



32. Khairalla, M.A.; Ning, X.; Al-Jallad, N.T.; El-Faroug, M.O. Short-Term Forecasting for Energy Consumption through Stacking Heterogeneous Ensemble Learning Model. *Energies* **2018**, *11*, 1605. [CrossRef]
33. Divina, F.; Gilson, A.; Gómez-Vela, F.; García Torres, M.; Torres, J. Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting. *Energies* **2018**, *11*, 949. [CrossRef]
34. Hsu, C.C. Generalizing self-organizing map for categorical data. *IEEE Trans. Neural Netw* **2006**, *17*, 294–304. [CrossRef]
35. Chen, N.; Marques, N.C. An extension of self-organizing maps to categorical data. In Proceedings of the Portuguese Conference on Artificial Intelligence, Covilha, Portugal, 5–8 December 2005.
36. Allende, H.; Valle, C. Ensemble Methods for Time Series Forecasting. In *Claudio Moraga: A Passion for Multi-Valued Logic and Soft Computing*; Seising, R., Allende-Cid, H., Eds.; Springer International Publishing: Cham, Switzerland, 2017; Volume 349, pp. 217–232.
37. Qiu, X.; Zhang, L.; Ren, Y.; Suganthan, P.N.; Amaratunga, G. Ensemble deep learning for regression and time series forecasting. In Proceedings of the Computational Intelligence in Ensemble Learning (CIEL), Orlando, FL, USA, 9–12 December 2014.
38. Mehta, P.; Bukov, M.; Wang, C.-H.; Day, A.G.; Richardson, C.; Fisher, C.K.; Schwab, D.J. A high-bias, low-variance introduction to machine learning for physicists. *arXiv* **2018**; arXiv:1803.08823. [CrossRef]
39. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
40. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
41. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, Bari, Italy, 3–6 July 1996.
42. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
43. Chen, T.; He, T.; Benesty, M. Xgboost: Extreme gradient boosting. *R Package Version 0.4-2*. Available online: <http://cran.fhcrc.org/web/packages/xgboost/vignettes/xgboost.pdf> (accessed on 15 September 2018).
44. Liu, B.; Wei, Y.; Zhang, Y.; Yang, Q. Deep neural networks for high dimension, low sample size data. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017.
45. Zhao, W. Research on the deep learning of the small sample data based on transfer learning. *AIP Conf. Proc.* **2017**, *1864*. [CrossRef]
46. Bai, H.; Pan, W.; Wang, L.L.; Ritchey, P.N. Another Look at Resampling: Replenishing Small Samples with Virtual Data through S-SMART. *J. Mod. Appl. Stat. Methods* **2010**, *9*, 181–197. [CrossRef]
47. Ngo, K.T. Stacking Ensemble for auto\_ml. Virginia Tech. Available online: <https://vtechworks.lib.vt.edu/handle/10919/83547> (accessed on 12 September 2018).
48. Tian, J.; Azarian, M.H.; Pecht, M. Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm. In Proceedings of the European Conference of the Prognostics and Health Management Society, Nantes, France, 8–10 July 2014.
49. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
50. McKinney, W. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010.
51. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
52. Wehrens, R.; Buydens, L.M. Self-and super-organizing maps in R: The Kohonen package. *J. Stat. Softw.* **2007**, *21*, 1–19. [CrossRef]

