energies

MDPI

Article

Enhancement of a Short-Term Forecasting Method Based on Clustering and kNN: Application to an Industrial Facility Powered by a Cogenerator

Giulio Vialetto * and Marco Noro

Department of Management and Engineering, University of Padova, 36100 Vicenza, Italy; marco.noro@unipd.it
* Correspondence: giulio@giuliovialetto.it

Received: 13 October 2019; Accepted: 18 November 2019; Published: 20 November 2019

Abstract: In recent years, collecting data is becoming easier and cheaper thanks to many improvements in information technology (IT). The connection of sensors to the internet is becoming cheaper and easier (for example, the internet of things, IOT), the cost of data storage and data processing is decreasing, meanwhile artificial intelligence and machine learning methods are under development and/or being introduced to create values using data. In this paper, a clustering approach for the short-term forecasting of energy demand in industrial facilities is presented. A model based on clustering and k-nearest neighbors (kNN) is proposed to analyze and forecast data, and the novelties on model parameters definition to improve its accuracy are presented. The model is then applied to an industrial facility (wood industry) with contemporaneous demand of electricity and heat. An analysis of the parameters and the results of the model is performed, showing a forecast of electricity demand with an error of 3%.

Keywords: data analytics; big data; forecasting; energy; polygeneration; clustering; kNN; pattern recognition

1. Introduction

Data management, machine learning, and artificial intelligence have been emerging themes in the energy sector during recent years, thanks to the increasing availability of data and the decreasing cost of sensors, storage, and data manipulation. Data analytics methods have already been used to analyze collected data to improve energy efficiency, for example in buildings [1,2], or combined with machine learning methods [3]. Different machine learning methods have been already defined [4], such as clustering, k-nearest neighbors (kNN), regression models, principal component analysis (PCA), artificial neural networks (ANNs), and support vector machines (SVMs). These methods are mainly used in the energy sector. In [5], ANNs are used to predict residential building energy consumption. In [6], SVMs and ANNs are applied to predict heat and cooling demand in the non-residential sector, whereas in [7] ANNs and clustering are used to predict photovoltaic power generation. PCA is considered to analyze and forecast photovoltaic data in [8] and [9], meanwhile, in [10] and [11], SVM is used. Data are also used to perform analytics on energy: In [12], open geospatial data are used to plan electrification, whereas in [13] social media data are proposed to better define energy-consuming activities. In another study, a methodology based on energy performance certification is defined to estimate building energy demand using machine learning (decision tree, SVM, random forest, and ANN) [14]. Ganhadeiro et al. evaluates the efficiency of the electric distribution companies using self-organizing maps [15]. Machine learning methods are implemented in different environments: MATLAB [16–18] and R [19,20] are the most famous ones for research. Recently, Fowdur et al. have provided an overview of the available platforms (mainly commercial, such as IBM solution,

Hewlett-Packard Enterprise Big Data Platform, SAP HANA Platform, Microsoft Azure and Oracle Big Data, but also open source software, such as H2O) on machine learning, and how it could be used for big data analytics [3]. Commercial environments have a more robust tool already developed and test with better documentation than open source software such as R. These environments would be preferred if the research aim is to develop commercial software. Open source software would be preferred for the aim of research, thanks to the possibility of full access to code and algorithms, in order to propose improvements and the free distribution of results.

In this paper, an enhancement of short-term forecasting based on clustering and kNN is proposed. In this context, "short-term" means few hours. When energy demand is sampled frequently (for example, every 15 min) and a dataset is available, data can be used to train a model to predict the energy request of the next few hours, with the scope of improving the operation strategy of the energy generation system and optimizing energy storage. Clustering is used to define the average curves, meanwhile kNN (the k-nearest neighbors algorithm) classifies each observation and forecasts the energy demand.

The clustering method has been already used to classify daily load curves [21,22] and to forecast energy demands [23–27]. Clustering and kNN are proposed in this study as forecasting methods and compared to other machine learning techniques, such as the previously cited ANN, SVM, or PCA. Such methods forecast data just by comparing the energy demand observed to the historical data. As a matter of fact, many industrial facilities collect energy demand data without considering other process variables that could be necessary to increase the accuracy of forecasts, for example weather conditions (air temperature, humidity, etc.). The complexity of the problem increases if the production process is a batch-type instead of continuous, as more variables are necessary (such as the properties of raw materials). As novelties compared to previous studies that proposed clustering and kNN for forecasting, an innovation on data normalization and an alternative criterion to define the most suitable number of clusters are suggested in this study in order to increase the accuracy of forecasts. Martinez deeply analyzed the use of clustering and kNN to forecast energy data using pattern similarity on historical data, silhouette criteria, and Dunn and Davies-Bouldin indices to define the optimum number of clusters (the clustering hyperparameter) [25]. Then, he improved forecasting accuracy with a weighted multivariate kNN algorithm in [27]. These papers present a similar algorithm to that proposed in the present article, even if improvements on the hyperparameter definitions are suggested here. The authors have already studied how to increase the efficiency by using innovative operation strategies and polygeneration systems, such as in [28–31]. Solid oxide fuel cells and heat pumps have been proposed to increase the efficiency on energy generation for the residential sector in different climates. Solid oxide fuel cells and electrolyzers have been suggested for energy generation in industrial facilities producing hydrogen [32]. In [33,34], some energy audits in industrial facilities have been performed to analyze the main inefficiencies, and to define which improvements are required. The main scope of this study is to define a method for improving the performances of short-term forecasting and, consequently, to use it in order to improve the operation strategy of the energy generation plant of an industrial facility. As a matter of fact, forecasting can help to optimize energy storage by giving suggestions on the energy demand of the next hours. The case study relates to a wood industry that requires low temperature heat to dry wood into steam-powered kilns by using a cogeneration plant. The industrial firm is organized with a batch process, and no data are available on the process (quantity of the wood into the kilns, properties and humidity of the raw wood, weather conditions, etc.). Energy demand data (both electricity and heat) are used to test the proposed improvements. The results show that the proposed methodology is able to predict the accuracy of the forecasting.

2. Method

In this section, the description of the method proposed in order to forecast the energy demand of an industrial facility is firstly reported. Successively, the method of the model training and the definition of its parameters are described.

2.1. Forecast Method Introduction

In this study, a forecast method based on clustering and kNN is proposed and applied to an industrial facility. Industry uses energy (thermal and electric) both for industrial processes and auxiliary purposes (lighting, compressed air, etc.). Generally speaking, energy uses related to the production processes are strictly connected to the variety and entity of the production output. If the production output remains constant in terms of the type and quantity of items, it is expected that the energy use does not vary significantly. Moreover, if the production output varies significatively (for example, because the industrial process is organized by batch), the complexity of the problem increases, and more variables are required. The aim of this study is to define a model based on a machine learning technique that allows the forecasting of energy demands for a short period (for example, the next hour) based on the demands observed and using a clustering approach without any other variables that could describe the process and/or the environmental conditions. In this study it is supposed that average profiles can be defined by using a dataset of at least one year of observation, in order to perform the forecast. Other machine learning methods, such as ANN or PCA are not proposed for this forecast problem due to the lack of variables which could describe the industrial process. Moreover, even if an ANN could be trained with only historical data to perform the forecast, the advantage of using clustering combined with kNN is the knowledge of the forecast process. If ANN would be used, a neural network is trained so a "grey box" model is defined, where the user knows the connection into the network, but it is unknown how the network works when varying the input variables. Instead, the methodology proposed here uses clustering to define similar patterns on historical data, where the user has a high control on the forecast process and may know each pattern proposed.

The first concept to introduce is the energy demand curve. It represents a temporal sequence of observations and forecasts of energy demand. Each curve can be split in two parts, namely, support and forecast. The former is the part of the data that will be provided to the model, constituted by the latest observations. The latter is the predicted data based on the support (Table 1). The length of the support (s) and of the forecast (f) is fixed by the user. In this model, it is proposed that $0 < f \leq s-2$. In the following discussion section, the performances of the model, varying f and s, for a real case study, will be described.

Table 1. Example of curves, definition of support and forecast (sample dataset).

Support								**Forecast**			
$i = 1$	$i = 2$			...			$i = s = 8$	$j = 1$		...	$j = f = 4$
10	11	10	13	12	14	16	12	11	12	18	13

To perform the forecast, the model features a workflow (Figure 1) based on the following steps:

1. Model training: A dataset of observations is used to train the model. Observations define the average demand curves and train the classification model;
2. Classification: Observations are used to classify which is the most similar average curve;
3. Forecast: Average curve forecast is used to define forecast of the observations.

The model proposed is based on two machine learning methods, clustering and kNN. Clustering is a method used only in the training process to define the average curve, while kNN is used to classify the observations and to relate them with the average curves.

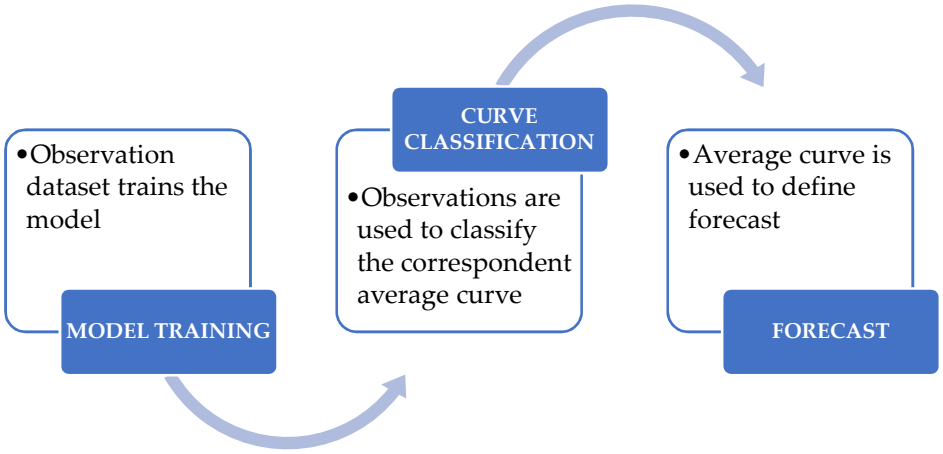

Figure 1. Workflow of the forecast method proposed.

2.2. Introduction to Clustering

Clustering is a data analytics method used to classify data and to perform data segmentation [4]. The samples are grouped into subsets or "clusters", where, in each cluster, objects are more likely related to one another than those assigned to different clusters. Clustering is strictly related to the concept of "degree of similarity" (or "degree of dissimilarity") between the objects of the same subset. A cluster method groups similar objects whereas similarity is defined, for example via a distance function.

K-means is a clustering method used when all the variables are quantitative, and the Euclidean distance between the objects is defined as a dissimilarity function, where the lower the distance, the greater the similarity ([4,35]). The Euclidean distance between each object x_a and x_b is measured by using the variable $i = 1...n$, which describes each object (Equation (1)):

$$d(x_a, x_b) = \sum_{i=1}^{n} \left(x_{a,i} - x_{b,i}\right)^2 \quad (1)$$

If a dataset with m objects is provided, K-means divides the dataset into N clusters, minimizing the Euclidean distance between each object of the cluster. The number of clusters, N, must be defined by the user as a hyperparameter. A hyperparameter is a value of a machine learning model that is defined before the training process. Silhouette [36], gap criterion [37] and other methods have been already developed and proposed to define the suitable number of clusters to divide a dataset. These methods try to define the minimum number of clusters to maximize the distance between the clusters themselves. For example, in [25], the performance of a forecasting method based on clustering and kNN with the silhouette, Dunn, and Davies-Bouldin methods, used to define the optimum number of cluster, is analyzed. In this paper, it the use of a criterion based on the clusters distance is not proposed, but instead to define the minimum number of clusters that minimizes the error of prediction under a threshold that is chosen by the user. In Section 2.7, discussing the hyperparameter definition, such a criterion will be described.

2.3. Introduction to kNN

kNN (k-Nearest Neighbors) is a machine learning method used mainly for classification and regression [4]. In the proposed forecast method, firstly, clustering training dataset is divided into N clusters, then an average curve for each cluster is defined. When a new observation occurs, it is necessary to classify which is its cluster. Here, kNN performs the classification task by analyzing how the k-neighbors nearest to the observation are classified, and the distances between them. In the model here proposed, kNN is used to define which is the cluster (and consequently, the average curve) defined with the training dataset closer to the new observation. kNN requires two hyperparameters,

the number of neighbors (k), and the distance function. Section 2.7, discussing the hyperparameter definition, describes how they are defined.

2.4. Model Training

The main task to define the forecast model is the training process. The training process requires at least one year of observations. The observations are ordered and then used to defined curves with support and forecast. These curves define a dataset. The workflow of the training can be divided in the following steps (Figure 2):

1. Define dataset: Firstly, it is necessary to define and to normalize the dataset. Successively, it is randomly divided into three subgroups, namely, the validation, training, and test datasets. These subgroups represent 25%, 50% and 25% of the total observations, respectively. The validation dataset is used to define the hyperparameters of the model, whereas the training dataset is used to train both the cluster and kNN models. Finally, the test dataset is useful to verify the performance of the trained model.
2. Define hyperparameters: As previously mentioned, the proposed model defines both the cluster and kNN models. Both methods require the definition at least the distance function and the number of clusters (cluster model), or the number of observations for classification (kNN model). The Euclidean distance function is proposed for the cluster model, meanwhile, the number of clusters and number of observations for classification are defined using the validation dataset.
3. Train cluster model: When all the hyperparameters are set, the training dataset is used to train the cluster model and to define the average forecast curves.
4. Train kNN model: When both the cluster model and the consequently average forecast curves are defined, kNN is defined. kNN is used to forecast the observations.
5. Test model: The test dataset is used to test the trained model and to check its performance by using the mean absolute percentage error (MAPE) and root mean square error (RMSE) criteria.

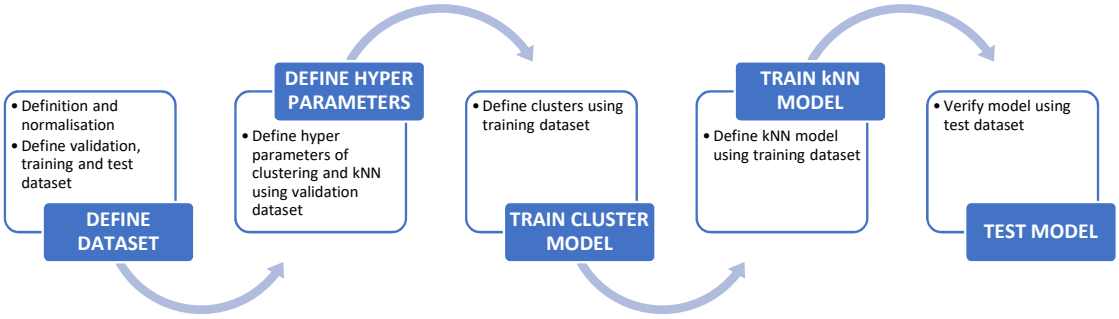

Figure 2. Workflow to train the model.

After the training process, the model can be used to forecast new observations.

2.5. Data Normalization

One of the first step of data analytics is data normalization. As datasets have different values and scale effect may occur, classification methods such as clustering will not work properly if data are not normalized. Usually, normalization is performed using standard score or minimum-maximum scaling [4,38]. The standard score normalizes the dataset (X) by using the average (μ) and the standard deviation (σ), as described in Equation (2):

$$\frac{X-\mu}{\sigma} \tag{2}$$

In this model, the authors propose to differently normalize dataset. As the goal of the model is to forecast energy demand curves, the idea is that different curves may have different scales but similar

variation. The standard score would be normalized but the curves will still have a lower scale effect. Instead, in this study it is proposed to calculate the average of the observations for each curve, and then to calculate the variations between observations and average (Equation (3)):

$$n_{j,i} = \frac{o_{j,i}}{a_j} - 1 \tag{3}$$

where $o_{j,i}$ is the observation i of curves j, a_j is the average and $n_{j,i}$ the normalized observation. Figure 3 represents an example explaining the reason why this normalization is proposed. Curves 1 and 2 have different scales but similar variation. Firstly, the standard score is applied, then the average normalization follows. The average (avg) and standard deviation (std) for the standard score are calculated using all the support values. In the other case, the average of support of each curve is calculated and used for normalization. Forecast values are excluded because they only become known during the training process. As can be seen in Figure 3, curve 2 is 1.58 times larger than curve 1, and a noise is added. It is possible to appreciate that the proposed method (avg), that is based on the average of the curves, reduces the scale effect, but keeps the variation. As a matter of fact, the normalized curves 1 and 2 have similar values. Instead, the standard score method proposes normalized curves with different values because it normalizes not only the scale effect but the variation as well.

		SUPPORT							FORECAST					
STANDARD SCORE	Curve 1	10.0000	10.1000	10.0000	10.0000	10.1000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	avg	12.9369
	Curve 2	15.8000	15.8000	15.9580	15.8000	15.8000	15.8000	15.9580	15.8000	15.8000	15.8000	15.8000	std	3.0187
	norm, curve 1	-0.9729	-0.9398	-0.9729	-0.9729	-0.9398	-0.9729	-0.9729	-0.9729	-0.9729	-0.9729	-0.9729		
	norm, curve 2	0.9485	0.9485	1.0008	0.9485	0.9485	0.9485	1.0008	0.9485	0.9485	0.9485	0.9485		

		SUPPORT							FORECAST					
AVERAGE	Curve 1	10.0000	10.1000	10.0000	10.0000	10.1000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	avg, curve 1	10.0286
	Curve 2	15.8000	15.8000	15.9580	15.8000	15.8000	15.8000	15.9580	15.8000	15.8000	15.8000	15.8000	avg, curve 2	15.8451
	norm, curve 1	-0.0028	0.0071	-0.0028	-0.0028	0.0071	-0.0028	-0.0028	-0.0028	-0.0028	-0.0028	-0.0028		
	norm, curve 2	-0.0028	-0.0028	0.0071	-0.0028	-0.0028	-0.0028	0.0071	-0.0028	-0.0028	-0.0028	-0.0028		

Figure 3. Data normalization example.

2.6. Error Estimation

When a forecast method is proposed, it is necessary to estimate the error of the forecasting. As previously mentioned, error estimation is used also to define the hyperparameters. Here, MAPE- and RMSE-derived errors are suggested. *MAPE* is the acronym of mean absolute percentage error, and it is defined by Equation (4):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{l}\sum_{j=1}^{l}\left(\frac{p_{j,i}}{d_{j,i}} - 1\right)\right) \tag{4}$$

where n is the number of curves, l is the number of the forecasted values of each curve, $p_{j,i}$ is the model predicted value of the curve, and $d_{j,i}$ is the value observed. *RMSE* is the acronym of root mean square error. Here, it is proposed instead of mean square error (MSE) because it is possible to compare error using the same measurement unit of data. It is defined by Equation (5):

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{l}\sum_{j=1}^{l}\left(p_{j,i} - d_{j,i}\right)^2} \quad (5)$$

These errors are calculated on the entire forecast, meanwhile, the first forecasted value of each curve is the most important. $MAPE1$ and $RMSE1$ are calculated considering not all the forecasted values but only the first ($l = 1$).

2.7. Hyperparameters Definition

As previously mentioned, it is necessary to define the parameters for clustering and kNN. They are called hyperparameters. Clustering requires the "distance function" and the "number of clusters", while kNN requires the "number of the nearest neighbors" and the "distance function". Only the clustering distance function is defined a priori (Euclidean distance), whereas the other ones are defined using the validation dataset.

Firstly, the number of clusters is defined. As previously mentioned, different criteria have been already developed, and they usually try to minimize the number of clusters in order to maximize the distance between data. It is in the authors' opinion that a more suitable criterion for a forecasting method is to find the minimum number of clusters that minimize the forecasting error, for example under a threshold previously defined. The model proposed here clusters data to obtain average curves, and then it uses them to forecast the energy demand. It is proposed to vary the number of clusters (from 2 to N) and for each simulation to calculate $MAPE$ between the data and average curves of the clusters. The parameter is the minimum n that has a $MAPE$ lower than the average next three values:

$$\min(n) | \, MAPE(n) < \frac{MAPE(n+1) + MAPE(n+2) + MAPE(n+3)}{3} \quad (6)$$

Nevertheless, it is possible to define n as the minimum number of clusters associated with a $MAPE$ lower than a defined threshold:

$$\min(n) | \, MAPE(n) < MAPE_{limit} \quad (7)$$

This method can be seen as an early stopping method, because the number of clusters increases by as much as the accuracy of the system is increased. Figures 4 and 5 report how this method is applied to a validation dataset of electricity and heat demand, respectively. Each curve has 8 observations as support, and 4 observations as forecast (data refers to the case study defined in Section 3.1). It is possible to appreciate that the curves have a $MAPE$ decreasing rapidly between 2 and 10 clusters, whereas between 10 and 30 clusters they become more stable. With more than 30 clusters, the curves have very low gradient, and locally $MAPE$ increases, even if the number of clusters increases. In this case, if the criterion described by Equation (6) is applied, then 10 clusters for heat and 13 for electricity are suggested.

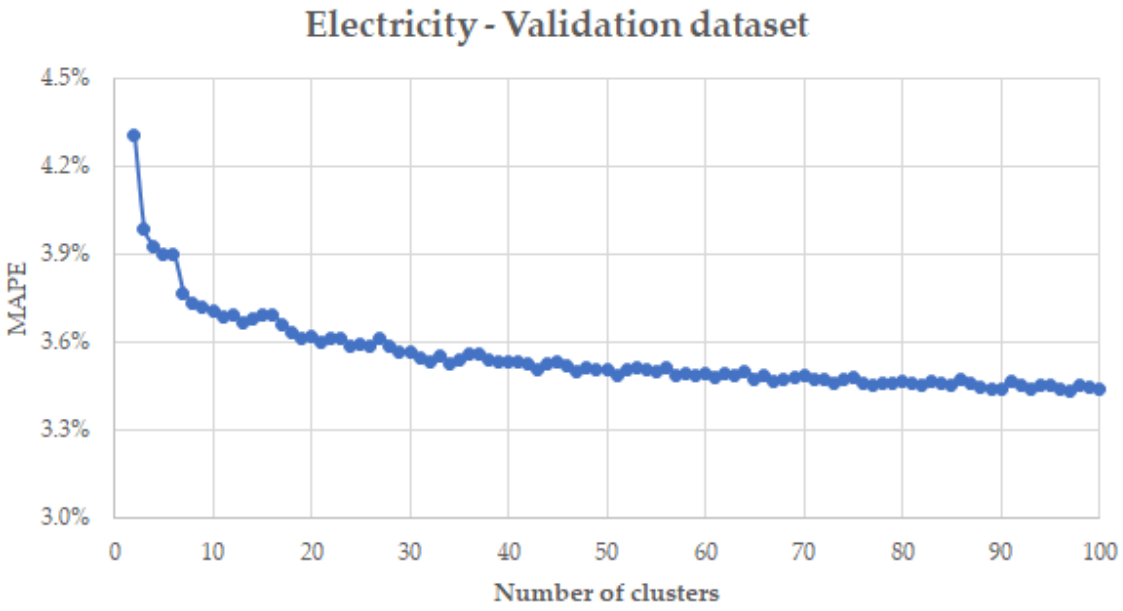

Figure 4. Electricity validation dataset mean absolute percentage error (*MAPE*) when varying the number of clusters from 2 to 100 for an 8-4 curve.

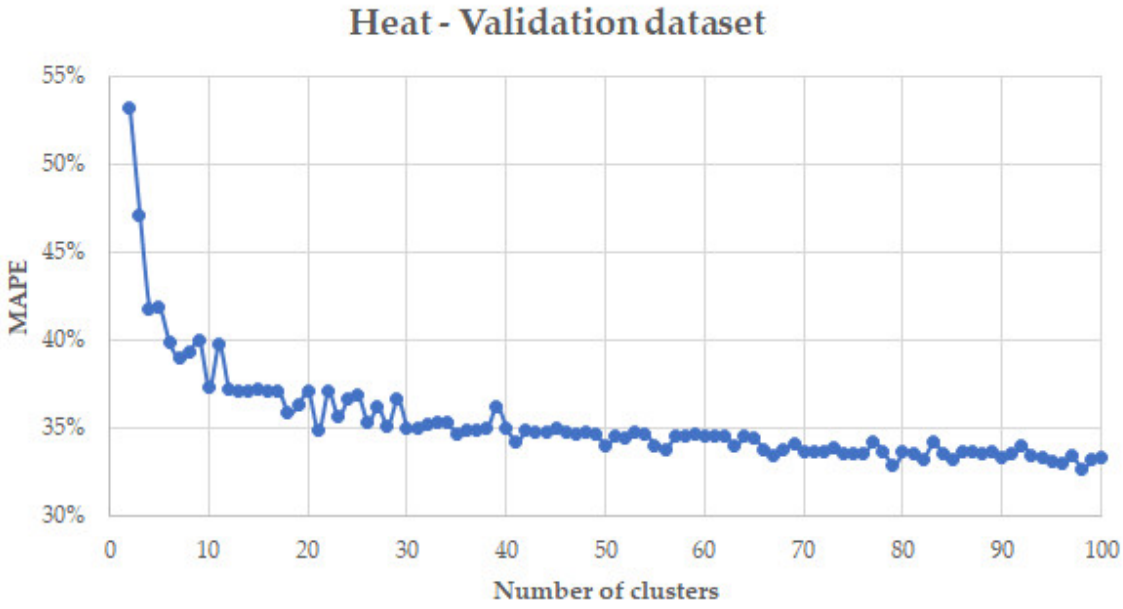

Figure 5. Heat validation dataset *MAPE* when varying the number of clusters from 2 to 100 for an 8-4 curve.

As previously mentioned, in other studies (such as [25]) where clustering and kNN are proposed for forecasting, the optimum number of clusters is defined by using a criterion such as silhouette or gap statistics. Here, the silhouette calculates the average distance between each member of a cluster from another cluster, and the minimum number of clusters that increases the distance is the optimum [36]. If the silhouette criterion was applied to the validation dataset (for both electricity and heat), the number of clusters suggested would be lower than the method proposed. In this regard, Figures 6 and 7 show that the number of clusters suggested is two in both cases. As a matter of fact, if this value was used, the *MAPE* would be the highest (Figures 4 and 5).

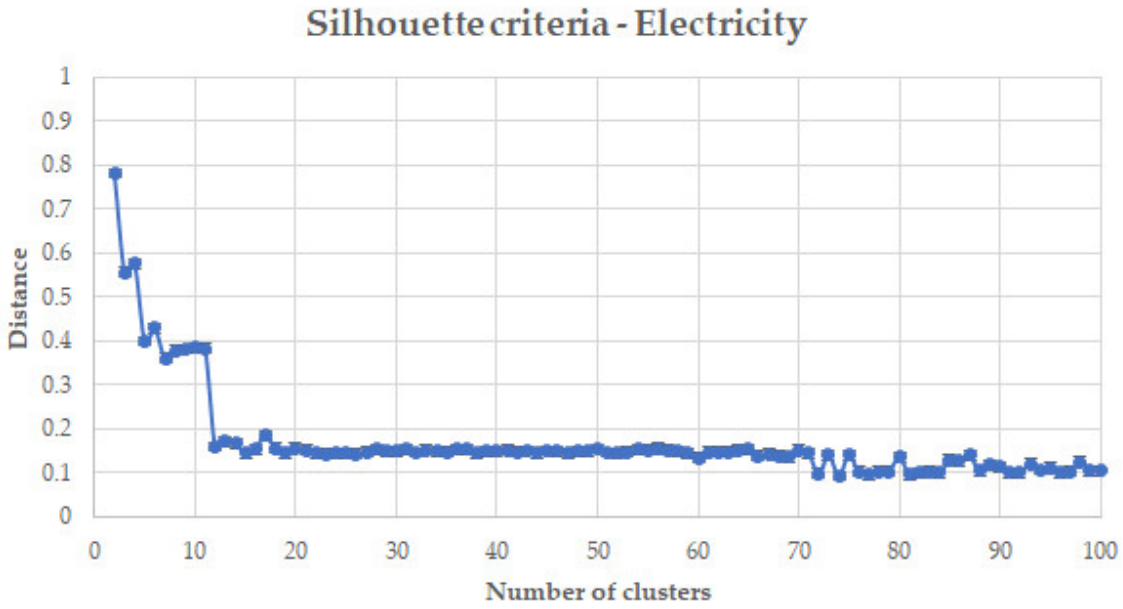

Figure 6. Silhouette applied to electricity validation dataset when varying the number of clusters from 2 to 100 for an 8-4 curve.

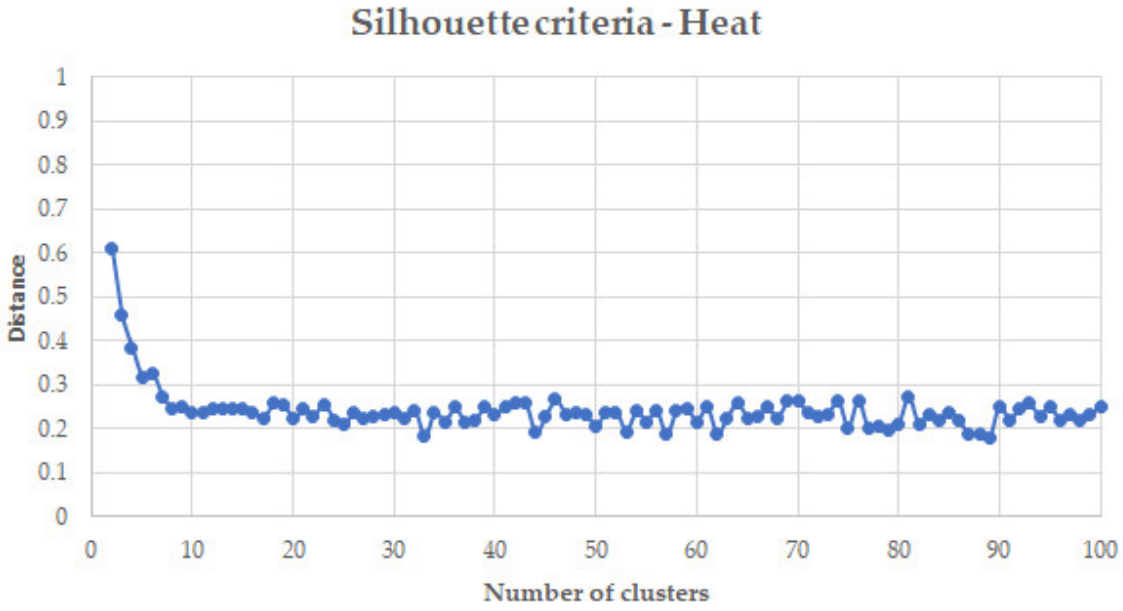

Figure 7. Silhouette applied to heat validation dataset when varying the number of clusters from 2 to 100 for an 8-4 curve.

The kNN hyperparameters are defined, instead, using MATLAB optimization with the 'Fitchknn' function. The latter optimizes the kNN model by choosing the distance function and the number of neighbors to decrease the classification error [39].

3. Results

The proposed method was applied to a case study based on an industrial facility characterized by a simultaneous demand of electricity and heat. The production process is organized by batch, and no data such as environment conditions, raw material properties, etc., were available. Data are used to predict the two types of energy separately by also using energy demand data, and the length of support and forecast was varied in order to verify the dependency of error. The aim was to verify the forecasting

performances on energy demand (electricity and heat) of the proposed method. No improvements of the current energy generation system and/or industrial process are proposed.

3.1. Case Study Description

The energy consumption of an industrial facility selling wood (timber) laminated windows, plywood, engineered veneer, laminate, flooring, and white wood was analyzed. The industrial process requires heat to dry wood in kilns (working temperature of 70 °C), and to store it in warehouses. Electricity is used for the production equipment, offices, lighting in the warehouses, and to charge electric forklifts. Energy is generated by using two cogeneration systems (combined heat and power, CHP) based on internal combustion engines (ICE) to produce both electricity and heat. A natural gas fired boiler was present as an integration system for the kilns. Electricity is also exchanged with the grid when mismatching occurs between generation and demand. Figure 8 represents the energy fluxes and the interconnections between each component of the system.

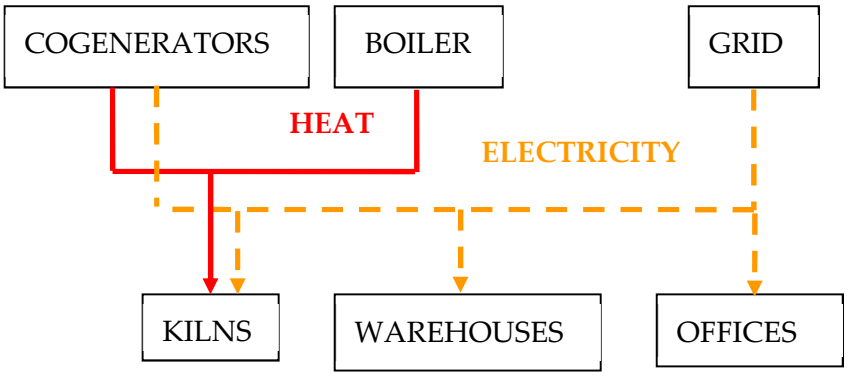

Figure 8. Electricity (yellow dot lines) and heat (red continuous lines) energy fluxes, connection between production (up boxes), and utilization (bottom boxes).

Energy use (both electricity and heat) was sampled every 15 min from 01/01/2015 to 25/09/2017. Electricity demand was available as mean power requested (kW). Heat demand, instead, was calculated by measuring the water flow rate (m^3/h) and inlet and outlet temperatures (°C) to heat the kilns. The data were stored in a structured SQL database. Here, we intended to use these data to define a curve with support and forecast, in order to train and to validate the forecast model. A dataset for heat demand, and another for electricity, has been defined.

As a matter of fact, these datasets can contain some sampling events with missing measurements or outliers. Missing measurements in a SQL database are managed with null values, so the events with at least one variable with a null value were not considered for the study, because the system was not able to sample the process, and the other variables could be affected by errors. Outliers could occur because the data were stored without any validation.

The data were plotted by a histogram (with a log scale on the x axis) and a probability plot of quartiles (QQ plot) to intercept outliers. The QQ plot was used to compare the dataset distribution with the normal distribution. The assumption here is that the data follow the latter, and if it does not, outliers are likely to be present. Figure 9 displays how the data were distributed. It is possible to appreciate that the outliers are present for both the electricity and heat demand. The electricity demand data were mainly between 100 and 1000 kW, while the maximum sampled value was higher than 106 kW. The same occurs for heat demand, where, in fact, the QQ plots show that the current dataset does not follow a standard distribution.

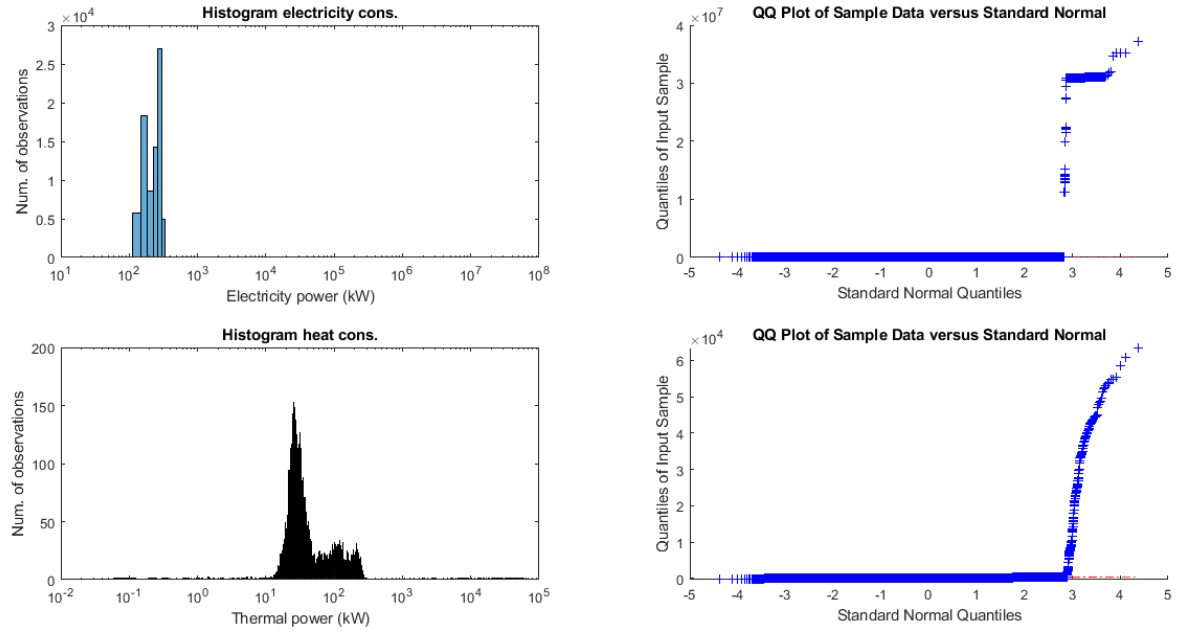

Figure 9. Representation of the dataset without filtering data, histogram, and probability plot of quartiles (QQ plot) of electricity (top) and thermal (bottom) power.

To filter the outliers, it was proposed to define an upper limit for each of the variables, both for electricity and heat. The limit was set considering the maximum demand of electricity and heat of the system. Figure 10 represents the filtered data, where the QQ plots show that the filtered dataset was closer to a normal distribution and that the range of the dataset decreased.

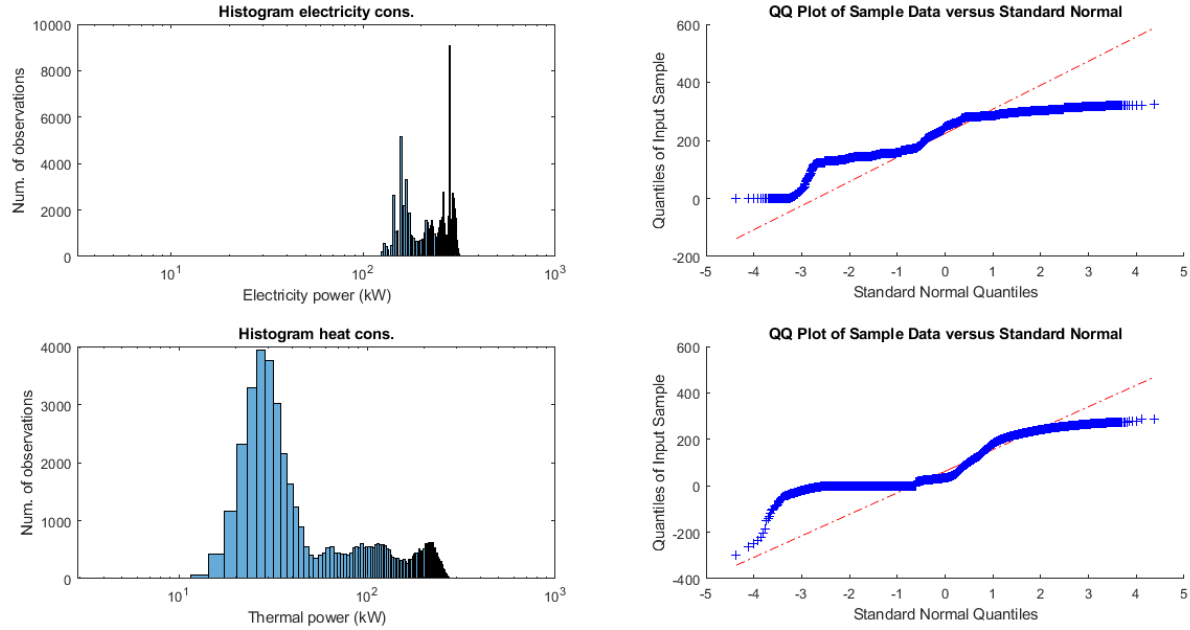

Figure 10. Representation of the dataset filtering data, histogram and QQ plot of electricity (top) and thermal (bottom) power.

3.2. Model Training and Test

Observations were used to define a dataset. The dataset was filtered by data related to null values or outliers. Here, it was randomly split into training, validation, and test datasets, representing 50%, 25%, and 25% of the entire dataset, respectively. The validation dataset was used to define the

hyperparameters of the model, whereas the training dataset was used to train the model, and the test dataset was used to check the accuracy of the model. Accuracy was defined by calculating the *MAPE* and *RMSE* between the forecasted value of the model and the observed value of the dataset. Curves of different lengths for the support and forecast were defined in order to discuss the influence of definition on the hyperparameters, in particular, the number of clusters. Table 2 shows some simulations of the model, considering energy demand curves of different length (for example, an 8-4 curve represents a curve with 8 observations as a support and 4 observations as a forecast). *MAPE* was calculated using the test dataset (error between forecasted values and observed values), once for the first forecasted value (here, the test dataset is *MAPE* 1) and once for the entire forecast (here the test dataset is the *MAPE*). The *MAPE* value calculated with the validation dataset was also added in order to define the hyperparameter number of clusters (Section 2.5). It is possible to appreciate that the *MAPE* calculated with the validation dataset is a good predictor of the *MAPE* of the test dataset. For example, with an 8-4 curve with electricity, the *MAPE* calculated with the validation dataset was 3.60%, whereas the *MAPE* calculated with the test dataset was 3.58%. The results also show a difference between the electricity and heat datasets, where an 8-4 curve has a *MAPE* of 3.58% and 34.11%, respectively. The difference can be explained with a higher variation of heat values.

Table 2. Simulation of the model with different curves length.

Curve	Type of Energy	Validation Dataset	Test Dataset			
		Mean Absolute Percentage Error (*MAPE*)	*MAPE*1	*MAPE*	*RMSE*1	Root Mean Square Error (*RMSE*)
8-4	Electricity	3.60%	2.75%	3.58%	5.15 kW	3.82 kW
8-4	Heat power	35.41%	32.95%	34.11%	93.43 kW	55.43 kW
10-4	Electricity	3.71%	2.74%	3.57%	5.15 kW	3.82 kW
10-4	Heat power	35.23%	32.70%	34.95%	93.20 kW	54.82 kW
10-8	Electricity	4.79%	2.90%	4.47%	5.47 kW	3.53 kW
10-8	Heat power	36.66%	35.30%	34.12%	90.03 kW	41.99 kW
12-8	Electricity	4.69%	2.80%	4.47%	5.31 kW	3.53 kW
12-8	Heat power	39.00 %	32.10%	37.21%	95.14 kW	43.05 kW

4. Discussion

In this section, the influence of the curve size and the type of normalization are both analyzed.

4.1. Influence of the Curve Size

Observations were used to define the curves in order to train and test the forecast model. Support is the part of the curve that is used to classify observation, and, consequently, it defines the forecasted value (forecast part). The length of the supports (s) and forecasts (f) may vary the hyperparameter number of clusters and, consequently, the error on forecasting. By increasing the forecast length (equally with support length), the forecast error is expected to increase, because the model needs to predict more observations. It is unknown what the effect of increasing the support length (with the same forecast length) could be, that is, increasing or decreasing the accuracy of the classification of the curve. Figures 11 and 12 represent the value of the *MAPE* criteria for the validation dataset, varying the support and the forecast for electricity and heat, respectively.

	Electricity	FORECAST						
		2	4	6	8	10	12	14
SUPPORT	4	2.9%						
	6	2.9%	3.6%					
	8	3.1%	3.6%	4.1%				
	10	3.1%	3.7%	4.1%	4.8%			
	12	3.2%	3.8%	4.3%	4.7%	5.1%		
	14	3.3%	3.9%	4.4%	4.9%	5.3%	5.8%	
	16	3.5%	4.0%	4.5%	5.1%	5.3%	5.8%	6.3%

Figure 11. Heatmap of *MAPE* of electricity validation dataset with curves with different support and forecast length.

	Heat Power	FORECAST						
		2	4	6	8	10	12	14
SUPPORT	4	37.7%						
	6	36.7%	35.2%					
	8	33.8%	35.4%	39.2%				
	10	36.8%	35.2%	37.3%	36.7%			
	12	40.2%	37.8%	39.0%	39.0%	36.9%		
	14	35.0%	36.9%	33.1%	36.1%	39.2%	37.4%	
	16	35.4%	39.0%	37.8%	39.0%	39.3%	37.9%	38.8%

Figure 12. Heatmap of *MAPE* of heat power validation dataset with curves with different support and forecast length.

Firstly, it is possible to appreciate that the electricity validation dataset has a regular variation of *MAPE* in comparison to the heat validation dataset. When the electricity dataset is used, the *MAPE* increases when increasing support and/or forecast lengths. Here, it is supposed that the electricity demand varies differently from the heat demand. As expected, the electricity dataset shows that when increasing the forecast length of the curve the *MAPE* increases. Here, the *MAPE* increases from 3.5% for a 16-2 curve (4 support length, 2 forecast length) to 6.3% for a 16-4 curve. This shows that the error increases when the forecast period becomes longer. On the other hand, the increase of the support length is also related to the increase of *MAPE*, where it changes from 2.9% for a 4-2 curve to 3.5% for a 16-2 curve. Even if more observations are available to classify each curve, the error does not decrease.

4.2. Influence of the Normalization

As mentioned in Section 2.5, in this model, it is proposed to not use a normalization based on the standard score but instead on the percentage norm. Here, the aim is to reduce the scale effect of the curves but to maintain their variation. A representation of the *MAPE*, varying the number of clusters in the electricity validation dataset with a curve of 8 observations for support and 4 for forecast (Figure 13) and 10 for support and 4 for forecast (Figure 14) is reported. In both cases, it is possible to appreciate that the dataset normalized with the standard score has a higher *MAPE* with respect to the normalization with the proposed percentage norm.

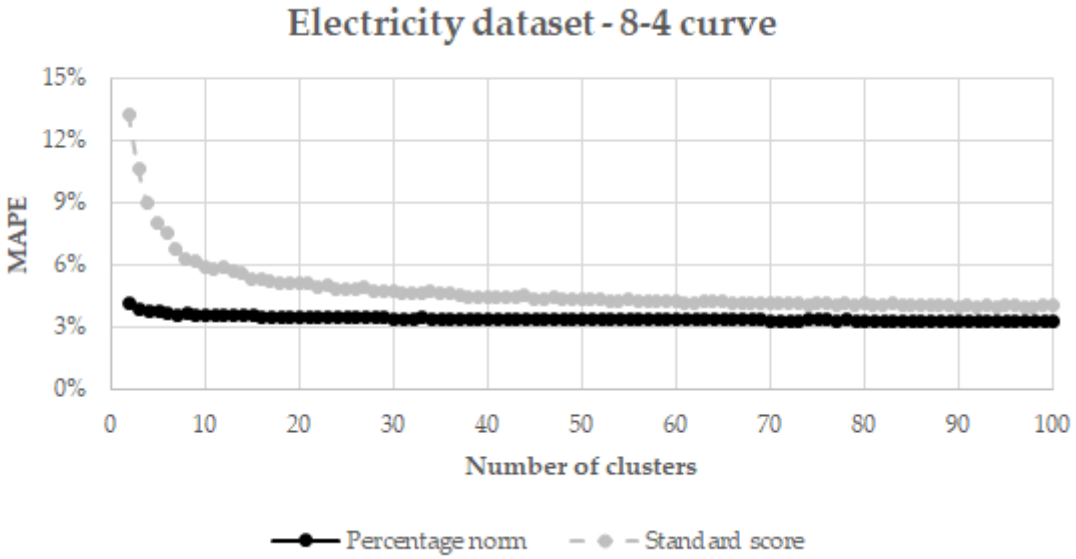

Figure 13. Comparison on *MAPE* with the electricity validation dataset, curve with 8 observation and 4 forecast values, normalization between percentage norm and standard score.

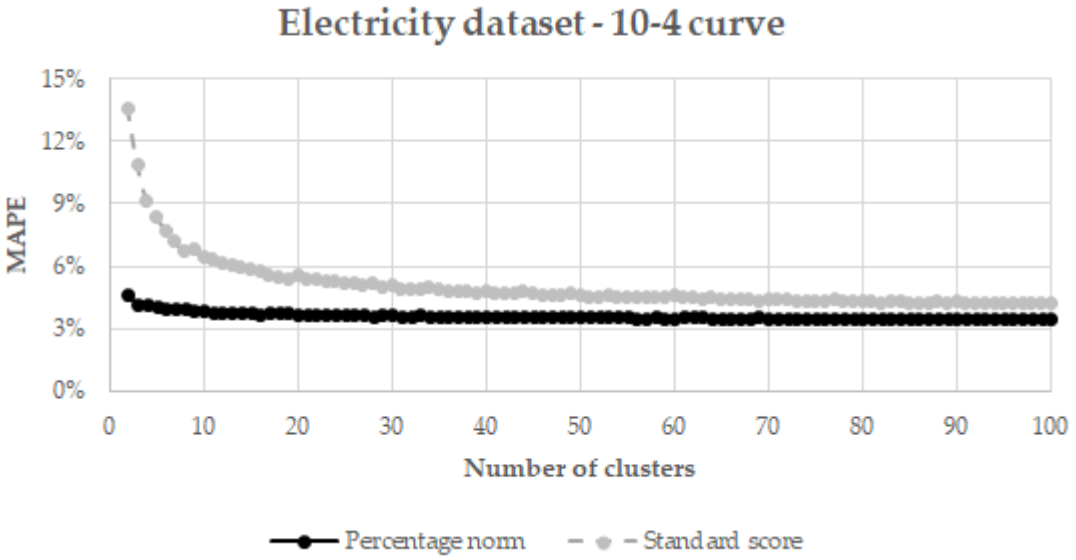

Figure 14. Comparison on *MAPE* with the electricity validation dataset, curve with 10 observation and 4 forecast values, normalization between percentage norm and standard score.

5. Conclusions

In this paper, the enhancements of a short-term forecasting method based on clustering and kNN machine learning techniques have been proposed and tested. A novel definition of hyperparameters (number of clusters) and data normalization compared to the state-of-art methods are presented here, in order to increase the accuracy on the forecast and to minimize errors. A dataset of observations is required to define the hyperparameters, in order to train the model and to test it. A case study based on an industrial facility with simultaneous electricity and heat demands was presented in order to apply the proposed energy forecast method. An analysis reported on how the length of the energy demand curves (numbers of observations and forecast) impacted the model performance. The industrial firm works with a batch process and only energy demand data were sampled and stored, as no other data on the process were available. The results show that the improvements suggested here, in terms of the definition of hyperparameters, decrease the error of forecasting compared to other criteria in the literature. An analysis of the effect of the length of the curves (both on support and forecast) on the

error was performed as well. For the dataset used here, the longer the length (both on support and/or forecast), the higher the error. The validation dataset was not only used to define the hyperparameters, as it could be used to predict the error of the forecast as well. It is in the authors' opinion that further improvements on the methodology could be achieved by studying the most suitable distance function for the dataset and/or by weighting observations. Moreover, an investigation on how this forecast method could improve energy production and efficiency could be of interest, for example reducing the production of unnecessary heat and/or improving suitable operation strategy to decrease the cost of energy generation.

Author Contributions: Conceptualization, G.V.; Data curation, G.V.; Validation, M.N.; Writing—original draft, G.V. and M.N.

Funding: This research received no external funding.

Acknowledgments: Authors thank Corà Domenico and Figli S.p.A., who provided the dataset for the case study to test the proposed methodology.

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

References

1. Noussan, M.; Nastasi, B. Data Analysis of Heating Systems for Buildings—A Tool for Energy Planning, Policies and Systems Simulation. *Energies* **2018**, *11*, 233. [CrossRef]
2. Tronchin, L.; Manfren, M.; Nastasi, B. Energy analytics for supporting built environment decarbonisation. *Energy Procedia* **2019**, *157*, 1486–1493. [CrossRef]
3. Fowdur, T.P.; Beeharry, Y.; Hurbungs, V.; Bassoo, V.; Ramnarain-Seetohul, V. Big Data Analytics with Machine Learning Tools. In *Internet of Things and Big Data Analytics toward Next-Generation Intelligence*; Springer: Berlin, Germany, 2018; pp. 49–97.
4. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer: New York, NY, USA, 2009.
5. Biswas, M.A.R.; Robinson, M.D.; Fumo, N. Prediction of residential building energy consumption: A neural network approach. *Energy* **2016**, *117*, 84–92. [CrossRef]
6. Koschwitz, D.; Frisch, J.; van Treeck, C. Data-driven heating and cooling load predictions for non-residential buildings based on support vector machine regression and NARX Recurrent Neural Network: A comparative study on district scale. *Energy* **2018**, *165*, 134–142. [CrossRef]
7. Cheng, K.; Guo, L.M.; Wang, Y.K.; Zafar, M.T. Application of clustering analysis in the prediction of photovoltaic power generation based on neural network. *IOP Conf. Ser. Earth Environ. Sci.* **2017**, *93*, 012024. [CrossRef]
8. Yang, D.; Dong, Z.; Lim, L.H.I.; Liu, L. Analyzing big time series data in solar engineering using features and PCA. *Sol. Energy* **2017**, *153*, 317–328. [CrossRef]
9. Malvoni, M.; De Giorgi, M.G.; Congedo, P.M. Photovoltaic forecast based on hybrid PCA–LSSVM using dimensionality reducted data. *Neurocomputing* **2016**, *211*, 72–83. [CrossRef]
10. Malvoni, M.; De Giorgi, M.G.; Congedo, P.M. Data on Support Vector Machines (SVM) model to forecast photovoltaic power. *Data Br.* **2019**, *9*, 13–16. [CrossRef]
11. Qijun, S.; Fen, L.; Jialin, Q.; Jinbin, Z.; Zhenghong, C. Photovoltaic power prediction based on principal component analysis and Support Vector Machine. In Proceedings of the 2016 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia), Melbourne, VIC, Australia, 28 November–1 December 2016; pp. 815–820.
12. Korkovelos, A.; Khavari, B.; Sahlberg, A.; Howells, M.; Arderne, C. The Role of Open Access Data in Geospatial Electrification Planning and the Achievement of SDG7. An OnSSET-Based Case Study for Malawi. *Energies* **2019**, *12*, 1395. [CrossRef]
13. de Kok, R.; Mauri, A.; Bozzon, A. Automatic Processing of User-Generated Content for the Description of Energy-Consuming Activities at Individual and Group Level. *Energies* **2018**, *12*, 15. [CrossRef]
14. Attanasio, A.; Piscitelli, M.; Chiusano, S.; Capozzoli, A.; Cerquitelli, T. Towards an Automated, Fast and Interpretable Estimation Model of Heating Energy Demand: A Data-Driven Approach Exploiting Building Energy Certificates. *Energies* **2019**, *12*, 1273. [CrossRef]
15. Ganhadeiro, F.; Christo, E.; Meza, L.; Costa, K.; Souza, D. Evaluation of Energy Distribution Using Network Data Envelopment Analysis and Kohonen Self Organizing Maps. *Energies* **2018**, *11*, 2677. [CrossRef]

16. MATLAB. Deep Learning Toolbox. Available online: https://www.mathworks.com/products/deep-learning.html (accessed on 14 July 2019).
17. Paluszek, M.; Thomas, S. *MATLAB Machine Learning*; Apress: Berkeley, CA, USA, 2017.
18. Kim, P. *MATLAB Deep Learning*; Apress: Berkeley, CA, USA, 2017.
19. Ghatak, A. *Machine Learning with R*; Springer: Singapore, 2017.
20. Ramasubramanian, K.; Singh, A. *Machine Learning Using R*; Apress: Berkeley, CA, USA, 2019.
21. Amri, Y.; Fadhilah, A.L.; Setiani, N.; Rani, S. Analysis Clustering of Electricity Usage Profile Using K-Means Algorithm. *IOP Conf. Ser. Mater. Sci. Eng.* **2016**, *105*, 012020. [CrossRef]
22. Müller, H. Classification of daily load curves by cluster analysis. In Proceedings of the Eighth Power Systems Computation Conference, Helsinki, Finland, 19–24 August 1984; pp. 381–388.
23. Wahid, F.; Kim, D. A Prediction Approach for Demand Analysis of Energy Consumption Using K-Nearest Neighbor in Residential Buildings. *Int. J. Smart Home* **2016**, *10*, 97–108. [CrossRef]
24. Wu, J.; Kong, D.; Li, W. A Novel Hybrid Model Based on Extreme Learning Machine, k-Nearest Neighbor Regression and Wavelet Denoising Applied to Short-Term Electric Load Forecasting. *Energies* **2017**, *10*, 694.
25. Martinez Alvarez, F.; Troncoso, A.; Riquelme, J.C.; Aguilar Ruiz, J.S. Energy Time Series Forecasting Based on Pattern Sequence Similarity. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1230–1243. [CrossRef]
26. Talavera-Llames, R.; Pérez-Chacón, R.; Troncoso, A.; Martínez-Álvarez, F. Big data time series forecasting based on nearest neighbours distributed computing with Spark. *Knowl. Based Syst.* **2018**, *161*, 12–25. [CrossRef]
27. Talavera-Llames, R.; Pérez-Chacón, R.; Troncoso, A.; Martínez-Álvarez, F. MV-kWNN: A novel multivariate and multi-output weighted nearest neighbours algorithm for big data time series forecasting. *Neurocomputing* **2019**, *353*, 56–73. [CrossRef]
28. Vialetto, G.; Rokni, M. Innovative household systems based on solid oxide fuel cells for a northern European climate. *Renew. Energy* **2015**, *78*, 146–156. [CrossRef]
29. Vialetto, G.; Noro, M.; Rokni, M. Innovative household systems based on solid oxide fuel cells for the Mediterranean climate. *Int. J. Hydrog. Energy* **2015**, *40*, 14378–14391. [CrossRef]
30. Vialetto, G.; Noro, M.; Rokni, M. Thermodynamic investigation of a shared cogeneration system with electrical cars for northern Europe climate. *J. Sustain. Dev. Energy Water Environ. Syst.* **2017**, *5*, 590–607. [CrossRef]
31. Vialetto, G.; Noro, M.; Rokni, M. Combined micro-cogeneration and electric vehicle system for household application: An energy and economic analysis in a Northern European climate. *Int. J. Hydrogen Energy* **2017**, *42*, 10285–10297. [CrossRef]
32. Vialetto, G.; Noro, M.; Colbertaldo, P.; Rokni, M. Enhancement of energy generation efficiency in industrial facilities by SOFC—SOEC systems with additional hydrogen production. *Int. J. Hydrogen Energy* **2019**, *44*, 9608–9620. [CrossRef]
33. Lazzarin, R.M.; Noro, M. Energy efficiency opportunities in the production process of cast iron foundries: An experience in Italy. *Appl. Therm. Eng.* **2015**, *90*, 509–520. [CrossRef]
34. Noro, M.; Lazzarin, R.M. Energy audit experiences in foundries. *Int. J. Energy Environ. Eng.* **2016**, *7*, 409–423. [CrossRef]
35. MATLAB. K-Mean Function-MATLAB. Available online: https://it.mathworks.com/help/stats/kmeans.html (accessed on 19 January 2019).
36. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
37. Tibshirani, R.; Walther, G.; Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc.* **2011**, *63*, 411–423. [CrossRef]
38. Lantz, B. *Machine Learning with R*; Packt Publishing: Birmingham, UK, 2013.
39. MATLAB. FitchkNN Function-MATLAB. Available online: https://it.mathworks.com/help/stats/fitcknn.html (accessed on 23 June 2019).