

Article

Machine Learning-Based Short-Term Prediction of Air-Conditioning Load through Smart Meter Analytics

Manoj Manivannan , Behzad Najafi * and Fabio Rinaldi 

Dipartimento di Energia, Politecnico di Milano, Via Lambruschini 4, Milano 20156, Italy; manoj.manivannan@mail.polimi.it (M.M.); fabio.rinaldi@polimi.it (F.R.)

* Correspondence: behzad.najafi@polimi.it; Tel.: +39-02-2399-8518

Received: 2 October 2017; Accepted: 15 November 2017; Published: 19 November 2017

Abstract: The present paper is focused on short-term prediction of air-conditioning (AC) load of residential buildings using the data obtained from a conventional smart meter. The AC load, at each time step, is separated from smart meter's aggregate consumption through energy disaggregation methodology. The obtained air-conditioning load and the corresponding historical weather data are then employed as input features for the prediction procedure. In the prediction step, different machine learning algorithms, including Artificial Neural Networks, Support Vector Machines, and Random Forests, are used in order to conduct hour-ahead and day-ahead predictions. The predictions obtained using Random Forests have been demonstrated to be the most accurate ones leading to hour-ahead and day-ahead prediction with R^2 scores of 87.3% and 83.2%, respectively. The main advantage of the present methodology is separating the AC consumption from the consumptions of other residential appliances, which can then be predicted employing short-term weather forecasts. The other devices' consumptions are largely dependent upon the occupant's behaviour and are thus more difficult to predict. Therefore, the harsh alterations in the consumption of AC equipment, due to variations in the weather conditions, can be predicted with a higher accuracy; which in turn enhances the overall load prediction accuracy.

Keywords: residential buildings; air-conditioning; smart meter analytics; machine learning; short-term load prediction

1. Introduction

A significant portion of the global energy consumption is due to the consumption of buildings and thus the corresponding share of the building sector in the total energy consumption in India, Europe and USA is around 40% [1,2]. Previous studies have demonstrated that around half of the energy demand of buildings can be attributed to their heating, ventilation and air-conditioning (HVAC) systems [3]. Accordingly, the consumption of air-conditioning systems has a significant impact on the electrical grid and the precise prediction of its variations can provide the grid management with notable benefits such as competitiveness in the day-ahead market, dispatch management, demand-side management and control optimization. Apparently, a straightforward solution in order to simulate the behaviour of buildings, and thus predicting the variations in their HVAC consumption, is developing physical models employing their geometrical and construction characteristics [4], infiltration properties, required ventilation rate, occupancy profiles and other details. However, grid management firms and utilities do not commonly have access to such details about the characteristics of their consumers' buildings.

One potential solution to deal with the mentioned issue is the development of data-driven models, which correlate the consumed power of air conditioners in the individual buildings with the

corresponding ambient and temporal conditions and can thus estimate harsh variations in HVAC consumption due to changes in the ambient conditions. Nevertheless, the latter approach requires installing dedicated meters permanently connected to the HVAC equipment in each individual building and consequently results in additional costs for the user or the utility firm.

Another alternative approach, which is implemented in the present study, is to utilize the aggregate consumed power of individual buildings, which is obtained from smart meters. Once the aggregate consumed power at each time step is obtained, energy disaggregation methodologies are employed in order to determine the corresponding share of air-conditioning consumption. In order to train the disaggregation algorithm, a very short (one week) measurement period with additional meters for individual devices, is needed. However, the latter requirement can eventually be evaded once a database with abundant measured data corresponding to the operation of different combinations of various residential devices is available. The ubiquitous spread of smart meters throughout the world, and specifically in industrialized countries, facilitates utilizing the proposed approach.

The disaggregation of aggregate data by non-intrusive method was first proposed by Hart, Kern and Schweppe at Massachusetts Institute of Technology (MIT) in the 80s and was further developed by Hart and termed Non-Intrusive Appliance Load Monitoring (NALM), in the 90s [5]. Supervised NALM algorithms utilize a set of the labelled signatures, e.g., voltage and current waveforms, of electrical devices in order to identify them from the aggregate load. This requires a one-time intrusion intervention in the household of interest where each device is identified and labelled based on its unique signature. In unsupervised NALM, the algorithm undergoes training and the devices are clustered from the aggregate waveform by matching either ON/OFF signals, voltage or current spikes in the aggregate data. The clustering is a way of labelling the devices and performing NALM with total non-intrusiveness. For the actual disaggregation, appliance models such as ON/OFF, Finite State Machine (FSM) and Continuously Variable proposed by Hart [6] along with Zero-Loop-Sum-Constraint, are utilized owing to their corresponding simplicity and ease of developing disaggregation algorithms. Combinatorial Optimisation and Factorial Hidden Markov models, which are elaborated in Section 3, are two widely employed state-of-the-art energy disaggregation algorithms, which are accordingly employed in the present work.

The second step of the present work is focused on utilizing machine-learning algorithms to conduct short-term prediction of air-conditioning load employing the data obtained from the disaggregation step. Many previous studies had investigated the possibility of using machine-learning algorithms for predicting the energy demand of buildings. Anstett and Kreider [7] employed artificial neural networks to predict the daily energy consumption in a complex institutional building. Jain et al. [8] developed a similar approach using Support Vector Machine (SVM), based on an empirical dataset from a multi-family residential building, aimed at predicting the overall energy consumption. They also examined the effect of temporal granularity on the resulting prediction and their results demonstrated an overall coefficient of variation (CV) of 11.4, 0.54 and 0.08 on a daily, hourly and 10 min data granularity respectively. In the latter study, the CV metric was defined as the square root of squares of deviations of the predicted and the actual values divided by the number of samples multiplied by the mean. Chen et al. [9] performed a short-term prediction of electric demand in the building sector via hybrid SVM and compared it with pure SVM. A relative improvement in mean absolute error of 6% was achieved. Artificial neural networks were utilized by Karatasou et al. [10] to predict hour-ahead and day-ahead energy consumption in buildings employing two different datasets from the Energy Predictions Shootout I contest and an office building in Athens. The prediction results showed a CV of 2.39–5.59 and 2.57–24.35 for hour-ahead and day-ahead prediction applied on the first and the second datasets respectively. Edwards et al. [11] conducted a study focused on prediction of residential loads employing various algorithms including regression, Feed Forward Neural Network (FFNN), Support Vector Machine (SVM), Least-Square Support Vector Machine (LS-SVM), Hierarchical Mixture of Experts (HME) and Fuzzy C-Means (FCM) on the ASHRAE Great Energy Prediction Shootout with 15 min granularity and utilizing the Campbell Creek house database. Their results

demonstrated that an average CV values of 36.38, 31.83, 29.55, 27.62, 35.78, 28.35, 27.94 obtained by Regression, FFNN, Support Vector Regression (SVR), LS-SVM, HME-Regression, HME-FFNN and FCM-FFNN respectively. Basu et al. [12] developed a general model using a knowledge driven and data driven approach. The model was tested over IRISE of REMODECE datasets using different machine learning algorithms including Neural Networks, Nearest Neighbours and Decision Tree. Elevated prediction accuracies were obtained and were reported to be around 94.7%, 94.1% and 94.5% for lighting, washing machine and oven consumptions. Dong et al. [13] developed a hybrid model through data-driven techniques employing ANN, SVM, (LS-SVM), Gaussian process regression (GPR) and Gaussian mixture model (GMM) on four different residential data for hour ahead and day ahead forecast of AC load. The hybrid models performed slightly better than those in the works of Jain et al. [8] and Edwards et al. [11]. Dong et al. [14] discussed a similar approach using SVM and focused on optimizing the model's hyper parameters for predicting building energy consumption in tropical regions. However, in this work each model was built to yield maximum accuracy whilst predicting hour-ahead and day-ahead AC energy consumption. Kontokosta et al. [15] developed a predictive model using Linear Regression, SVM and Random Forest approaches to predict city-scale energy use in buildings in New York. It was shown that SVM performed the best among them. Owing to the availability of the building area and geo-location, the energy consumptions were correlated with the building area, occupants and floors. Li et al. [16] performed particle swarm optimization based LS-SVM for building cooling prediction. They found that the hyper-parameters of the model could be quickly optimized while attempting to conduct predictions for nonlinear and time series dataset like energy consumption. Fan et al. [17] developed a method for short-term cooling load prediction using supervised and unsupervised learning algorithm with deep neural network. Li et al. [18] used back propagation neural network, radial basis function neural network, general regression neural network and SVM for predicting the hourly cooling load in office buildings. Yao et al. [19] developed a combined forecast model based on analytic hierarchy process for day-ahead prediction. Analytic hierarchy process is a simple decision making procedure based on setting priorities. González et al. [20] modelled a feedback artificial neural network for hourly energy consumption in buildings. The model was not optimized based on the number of neurons, however they seemed to perform well on predicting the energy consumption. Ben-Nakhi et al. [21] developed a general regression neural network (GRNN) to predict cooling loads in buildings in Kuwait using a dataset from 1997–2001. The prediction model also used temperature forecast to aid in day-ahead predictions. Apart from machine-learning, genetic algorithms are also a promising method of analysing building energy performance. Castelli et al. [22] developed a model using genetic programming approach with geometric semantic genetic programming (GSGP). The model predicted both heating and cooling load of a set of residential buildings.

As previously pointed out, the present study involves two main steps. The first step is focused on extracting the air-conditioning consumption from the aggregate smart meter data of a building. The second step is dedicated to building a machine-learning model to predict hour-ahead and day-ahead consumption of air conditioner units from the obtained data. Yearly consumption data of a residential building, provided by Pecan Street Inc.'s Dataport™ [23] was utilized. Combinatorial Optimisation (CO) and Factorial Hidden Markov Model (FHMM) algorithms, which are implemented in the open source Non-Intrusive Load Monitoring Toolkit (NILMTK) [24], have been utilized to conduct the disaggregation step. The obtained air-conditioning load and the corresponding historical weather and time-related features are then employed as input features of the prediction procedure. The time-related data includes the hour, the day of the week, the weekday/weekend, and day/night, while the temperature and the irradiance constitute the employed weather data. The ambient temperature is provided within the dataset and, in order to include the effect of irradiance, the time-stamped generation of a nearby photovoltaic plant, is utilized. The use of Photovoltaics (PV) generation as an indication of irradiance will increase the general applicability of the proposed

method, as the grid managers have access to PV production at various locations, while the irradiance measurement devices are not ubiquitous.

Hour-ahead and day-ahead predictions are finally performed using several machine-learning algorithms such as Linear Regression, Random Forest Decision Tree, Support Vector Machines, and Multi-Layer Perceptron Neural Networks and their corresponding results are compared.

It is worth mentioning that the principal objective of the present work is performing short-term prediction of AC loads while only employing the aggregate data obtained from a conventional smart meter and in the absence of other detailed information about the building including the construction characteristics, occupancy, the ventilation rate, technical details of the air-conditioning unit, and other details. The latter situation is a problem that the utility companies and grid management units are commonly facing as they attempt to predict the aggregate power consumption of users (out of which a notable share is related to AC consumption) only employing the total consumed power communicated by the smart meter. However, they do not have access to any other information regarding the details of the building construction or the behaviour of the occupants. Hence, the main novelty of the present work, compared to previously conducted data-driven residential load prediction studies, is attempting to obtain increased accuracy while not having access to the mentioned detailed information about the building and its occupants.

2. Employed Dataset

The dataset used in the present work is the yearly consumption data of a residential building, located in Austin (TX, USA), which is measured in the year 2014, and is publicly accessible via Dataport™ (provided by Pecan Street Inc., Austin, TX, USA). This dataset includes the total consumed power of the house along with the power consumed by individual devices recorded with 1-min sampling rate. The devices include a split air conditioner, dish washer, washing machine, electric oven, water pumps, electric heater, fridge, fans, electric water heater, micro wave oven, toaster, television, miscellaneous electronic devices (laptops, tablets) and light bulbs of different types (fluorescent, incandescent, and light-emitting diodes (LED)).

The database also contains ambient temperature with 1-hour sampling rate which was added as a feature to the machine-learning model. As previously mentioned, in order to take into account, the effect of irradiance, time-stamped power generation of a photovoltaic unit installed on a nearby building has also been employed. Although detailed information regarding the model and the orientation of the PV panels, utilized in the unit, is not accessible, the corresponding power generation at any specific hour is proportional to the irradiance in that hour.

3. Energy Disaggregation Methodology

Energy disaggregation estimates appliance-by-appliance electricity consumption from a single meter that measures the total household's electrical consumption. First step in disaggregation is to establish appliance models, which describe the behaviour and electrical signature of appliance. There are several device models but the most simple and common models are the ON/OFF, Finite State Machines (FSM) and Continuously Variable [25]. The ON/OFF model considers that an appliance may be either ON or OFF at any given point in time. While it is ON there is no other state that the appliance may take (e.g., toaster, lights vacuum cleaners). FSM model considers appliances, which have several distinct switching states during ON mode. The appliance passes through different states every time the devices is used (e.g., washing machines, electric rice cooker, clothes dryer). The Continuously Variable model includes appliances like light dimmers, and variable-speed hand tools. These devices are very difficult to identify and disaggregate from the whole home energy data. It relies on high frequency harmonics to identify such devices. Appliance signatures like voltage, real and reactive power, current, root mean square (RMS) current, steady state harmonics and phase shifts are the electrical marks on the aggregate data from which appliances can be identified. The methods under the steady state make use of appliance signatures when the load is in steady state operation [26]. Appliances, whenever switched

ON, have a transient state momentarily before reaching a steady state which is caused by the sudden change in the circuit [27]. Transient behavior of most electrical appliances is unique, which makes it convenient for identification and disaggregation. The drawback, however, is the need for high sampling rate which may increase the cost of measurement and computation. Hybrid signatures are combination of steady-state signatures and transient signatures. H. H. Chang et al. [28,29] combined steady-state real power, reactive power and total transient energy to disaggregate different appliances with the same real and reactive power. Apart from steady-state and transient signatures, there are other methods of using features as signatures which need not be extracted from the measured appliance data. Hour of the day, frequency of appliance usage, usage duration and distribution over the day and the correlation between the usage of other appliances are some features which can be used to increase accuracy of identification and disaggregation [30]. Energy disaggregation, ultimately is to provide estimates [24], $y_i^{(n)}$, of the actual power demand, $y_t^{(n)}$, of each appliance n at time t , from the household's aggregate power readings, y_t . Generally, NALM algorithms are developed over the appliance models mentioned above. The NILMTK toolkit [24] uses $x_t^{(n)} \in Z_{>0}$ to represent the ground truth state, and $\hat{x}_t^{(n)}$ to represent the appliance state estimated by the disaggregation algorithm.

The basic process of disaggregation can be divided into seven steps. Firstly, the whole house aggregate electricity data is collected through sensors or smart meters at the utility interface which measures the average power and the RMS voltage on the mains with a standard sampling interval (kHz, 1 s, 1 min, 15 min). Step 2 is to normalize the total load power or the measured signature with respect to the fluctuation in the mains. Supply voltage to consumers may have plodding or discrete changes due to factors like load-dependent voltage drops in transmission lines and transformers. This may lead to detecting step changes that may interfere with our appliance signature and ultimately with the disaggregation. The toolkit implements Hart's method based on linear model where admittance is preferred over power and current as a signature. The admittance $Y(t)$ is given by Equation (1), where $P(t)$ and $V(t)$ are the measured power and RMS voltage. The normalized power is then expressed as in Equation (2), which is admittance corrected by a constant value, resulting in a power normalized to 120 V. Step 3 involves passing the normalized power of the aggregate data through the edge detector, which evaluates time and size of all the step changes. It involves signal processing techniques like filtering, differentiating to detect peaks and to capture the step changes caused by appliance state changes. For an unsupervised learning algorithm, where the appliance labels are unavailable, more electrical signatures, such as reactive power, are considered when evaluating the step changes of unique devices. The detected step changes (e.g., ON/OFF) when mapped on the real-complex ΔP - ΔQ space, could be grouped into clusters based on equal and opposite components. Finally, each step changes are matched with the corresponding cluster in case of unsupervised learning or to appliances in case of supervised learning [24].

$$Y(t) = \frac{P(t)}{V^2(t)}, \quad (1)$$

$$P_{Norm}(t) = 120^2 Y(t) = \left(\frac{120}{V(t)} \right)^2 P(t). \quad (2)$$

3.1. Combinatorial Optimization

The total load depends on which appliance are switched on at any given moment, so a switching process, vector $a(t)$ is defined. The vector $a(t)$ is an n component Boolean vector defining the state of n appliances at time t :

$$a_i(t) = \begin{cases} 1, & \text{if appliance } i \text{ is ON at } t \\ 0, & \text{if appliance } i \text{ is OFF at } t' \end{cases} \quad (3)$$

For $i = 1, \dots, n$, the switch process modulates the power consumption of the individual appliances. A multiphase load with p phases can be modelled as a p -vector in which each component is the load on one phase. Then we model the measured power given by Equation (4). Where $P(t)$ is the p -vector as

seen at the utility at time t , and $e(t)$ is a small noise or error term. Equation (4) suggests a straightforward criterion for estimating the state of the individual appliances. If all n of the P_i are known and the measured power $P(t)$ is given, at each t choose the n vector $a(t)$ which minimizes $|e(t)|$, under the constraint that a is an n -dimensional Boolean vector [5]:

$$P(t) = \sum_{i=1}^n a_i(t)P_i + e(t), \tag{4}$$

$$\hat{a}(t) = \arg \min_a \left| P(t) - \sum_{i=1}^n a_i P_i \right|, \tag{5}$$

This is a familiar combinatorial optimization problem. Each time instant is a separate optimization problem and each time instant is independent. Combinatorial optimization is a subset sum problem and even with scalar P variables it's an NP complete "weighted set" problem. The computation becomes taxing as it is exponential with the number of appliances [5].

3.2. Factorial Hidden Markov Model

Hidden Markov Models (HMM) are temporal graphical models which are probabilistic methods. A simple representation of HMM is show in Figure 1 [31]. Several machine learning and artificial intelligence models implement Markov models. A well-known example is in the area of speech recognition [32] and word prediction. The HMM is sequence of discrete variables in which each variable emits a single continuous variable, which is dependent upon the value of the discrete variable. The discrete variables (sequence $z = z_1, \dots, z_T$) are not observed whereas the continuous variables (sequence $x = x_1, \dots, x_T$) are observed. T is the length of the sequence or the time step of each discrete variable. Each discrete variable z_T can correspond to one of K states, while each continuous variable can take on any real number. The three mains parameters describing a HMM are initial probability, transition probability and emission probability.

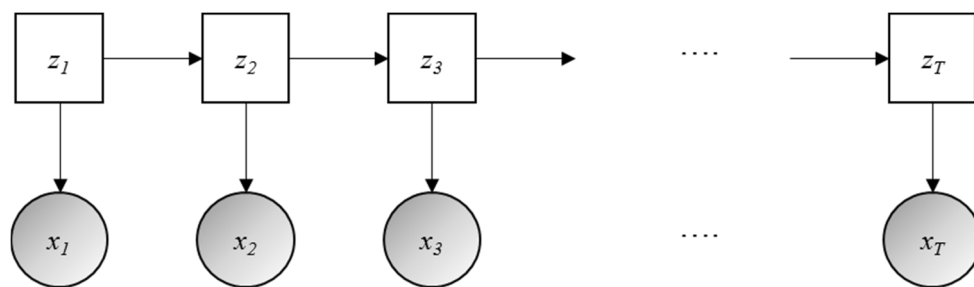


Figure 1. Hidden Markov Model [31].

Factorial Hidden Markov Models (FHMM) are a type of HMM wherein there are several independent Markov chains of hidden variables, $z_{(1)}, \dots, z_{(N)}$, in which N is the number of chains. Therefore, each continuous observed variable is dependent on multiple hidden variables [33]. The Figure 2 is a representation of a Factorial Hidden Markov model. Similar to a HMM, the joint likelihood of a FHMM is given by Equation (6), where $1: N$ represents a sequence of appliances $1 \dots, N$. The complexity of both learning and inference is greater for FHMMs than HMMs. The computational cost is exponential in the number of chains, N , the model will therefore become computationally intractable for large N [31]:

$$p(x^{(1:N)}, z | \theta) = \prod_{n=1}^N p(z_1^{(n)}) \pi \prod_{t=2}^T \prod_{n=1}^N p(z_t^{(n)} | z_{t-1}^{(n)}, A) \prod_{t=1}^T p(x_t | z_t^{(1:N)}, \phi), \tag{6}$$

In the FHMM, each of the n devices in a building is considered as a Hidden Markov Model. Each device has a discrete hidden state, denoted $x_t^{(i)} \in \{1, \dots, N_i\}$ for any given time t for device i , which corresponds approximately to the internal state of the device (ON/OFF or one of intermediate states if it were an FSM). At each time t , given the internal state, the i^{th} device produces a Gaussian distributed power, represented $x_t^{(i)}$, with state-specific mean and variance parameters.

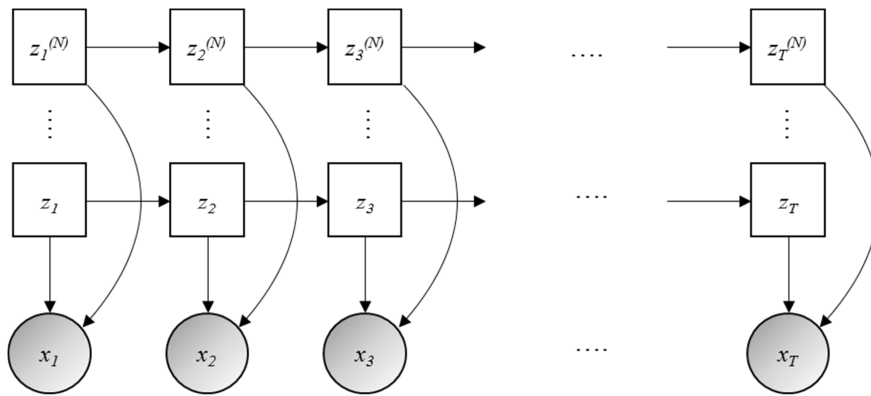


Figure 2. Factorial Hidden Markov Model [31].

Since, we only observe the sum of all the power outputs at each time as in Equation (7) [31]:

$$\bar{x}_t = \sum_{i=1}^n x_t^{(i)} \quad (7)$$

With a smart meter, in a practical scenario, the disaggregation task can then be framed as an inference problem. Given an observed sequence of aggregate energy $x_{\bar{1}}, \dots, x_{\bar{T}}$, we aim to compute the posterior probability of the individual device consumptions $x_t^{(i)}$ for $i = 1, \dots, n$ and $t = 1, \dots, T$ [34].

4. Short-Term Prediction Methodology

4.1. Machine Learning Models

In the present study, several machine-learning models including Linear Regression, Support Vector Machines, Random Forests Decision Trees, Neural Networks (Multi-Layer Perceptron) have been employed to conduct the predictions. Linear regression is a linear model that assumes a linear relationship between the input variables (x_1, x_2, \dots, x_n) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x_1, x_2, \dots, x_n). Therefore, a linear model can be represented as an equation that combines a specific set of input values (x) by a weight, the solution to which is the output variable (y). The weights are optimized by methods such as Ordinary Least Squares and Gradient Descent.

Support Vector Regression is a regression model from the Support Vector Machines, which is similar to the SVM classification. For a set of input variables (x) and single set of output variable (y), the goal is to find a function $f(x)$ that has the least deviation from the actual obtained targets (y). The problem can be solved as a convex optimization problem minimizing an error function.

Random Decision forests is an ensemble approach that is similar to a form of nearest neighbour predictor and can be applied for both classification and regression. A Random forest is a predictor consisting of a collection of randomized base regression trees by bootstrap aggregating or bagging. Random forest works by averaging multiple deep decision trees, trained on various parts of the same training set with the goal of reducing the variance [35].

Multi-Layer Perceptron (MLP) is a Feed Forward Artificial Neural Network, a universal function approximator which can learn complex mapping functions. MLP is made of several

interconnected layers and nodes in its hidden layer, apart from its input and output layer. Using error back-propagation as a gradient descent, an MLP estimates the weight of each node to map the set of inputs to the output. Back propagation keeps updating the weights by minimizing the mean squared error (MSE) with the addition of regularization term to decay the weights, which makes sure not to overfit the data that we are trying to map.

4.2. Accuracy Metrics

The model's performances are measured on the basis of two accuracy metrics, namely coefficient of determination (R^2) and root mean squared error (RMSE). R^2 is a general metric used in regression analysis, which is simply the square of sample correlation coefficient between the observed values to the predicted values. It gives a measure of how well the observed values are represented by the model, based on the proportion of total variation of values predicted by the model. If we consider the observed values to be y_1, \dots, y_n , and the corresponding predicted values to be f_1, \dots, f_n . The residuals are defined as $e = y_i - f_i$ and the mean of the observed data is given by Equation (8). The total sum of squares, which is proportional to the variance of the data, is given in Equation (9). The regression sum of squares and residual sum of squares are then defined as Equations (10) and (11), respectively. Finally, the definition of coefficient determination is given by Equation (12). This leads to a simple model interpretation where $R^2 = 1$ means that the model predicts exactly the observed values:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \quad (8)$$

$$SS_{tot} = \sum_i (y_i - \bar{y})^2 \quad (9)$$

$$SS_{reg} = \sum_i (f_i - \bar{y})^2 \quad (10)$$

$$SS_{res} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2 \quad (11)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (12)$$

The root mean squared error or root mean squared deviation is synonymous with the previous mentioned accuracy metric. It directly estimated the difference in predicted values to the corresponding observed values. The root mean squared deviation of predicted values for time t of a regression's dependent variable is computed for n different predictions as the square root of the mean of the squares of deviations, as expressed in Equation (13):

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}, \quad (13)$$

These two metrics are the most common metrics to evaluate a regression model.

5. Results

5.1. Disaggregation

Combinatorial Optimisation (CO) and Factorial Hidden Markov Model (FHMM) algorithms, which are conventional disaggregation methods implemented in the open source toolkit NILMTK [24], have been utilized to conduct the disaggregation of the considered building's aggregate load (measured in the year 2014). Similar to a machine learning approach, the disaggregation requires two datasets, one for training and another for testing. The training was conducted using a dataset of 28 days period from the chosen building of the year 2013 and the disaggregation using a dataset of 364 days

period from the same building of the year 2014. Furthermore, in order to evaluate the possibility of employing shorter training periods, disaggregation using a 7 day training dataset was also performed. The corresponding obtained R^2 score and RMSE of the algorithms along with the corresponding computational cost are provided in Table 1. The cost of computation is represented as training and disaggregation time, each of which is the time (in seconds) that the computer (equipped with an i5 Intel processor running at 2.2 GHz and 8 GB of memory) needed in order to perform the corresponding operation (training or disaggregating) and write the results to the hard drive.

Table 1. Comparison of CO and FHMM algorithms' accuracy and computational cost (for a PC equipped with an i5 Intel processor running at 2.2 GHz and 8 GB of memory).

Disaggregation Method	Training Period (Days)	Training Time (Seconds)	Disaggregation Period (Days)	Disaggregation Time (Seconds)	RMSE	R^2 Score (%)
Combinatorial Optimization	28	45	364	76	114.76	99.01
	7	14	364	77	121.80	98.67
FHMM	28	353	364	2800	284.49	92.07
	7	92	364	2780	295.67	91.82

As can be observed in this table, Combinatorial Optimization shows a notably superior performance compared to FHMM as it leads to higher accuracy while requiring even lower computational cost. It can also be concluded that utilizing a training period of 7 days instead of 28 days, does not result in a notable decrement in the disaggregation accuracy. Thus, a training period of 7 days will be enough in order to reach an acceptable accuracy. It is also noteworthy that, although one-week period of training with dedicated sensors is still needed in the methodology presented in this paper, this necessity can clearly be evaded once the utility or grid manager has access to a large database, which includes labelled electrical signature data of different combination of commercial domestic devices operating at the same time.

Figure 3 shows the results of disaggregation using Combinatorial Optimization (CO) and FHMM along with actual AC consumption of the considered building for a short interval (Aug 10th, 2014). As can be seen in this figure, an acceptable agreement between the AC consumption obtained through CO algorithm and the real data is observed.

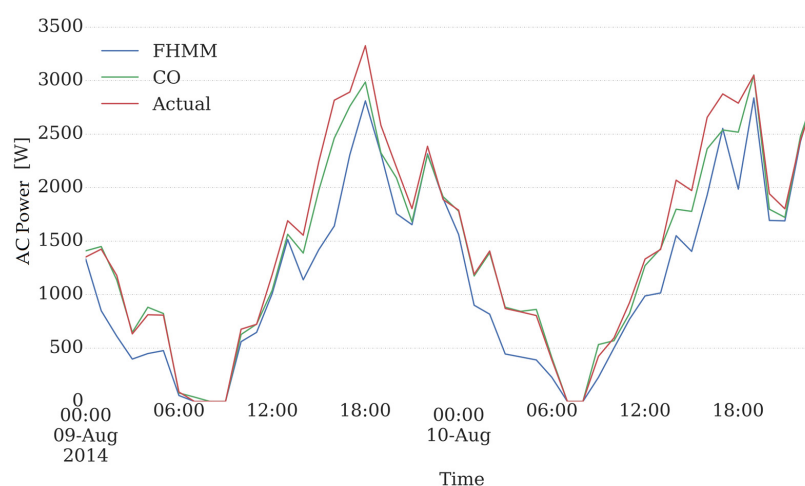


Figure 3. Comparison of CO and FHMM with actual consumption.

5.2. Correlation Study

Before building a machine-learning model, the correlation between the AC consumed power and the ambient conditions, including the temperature and the irradiance (presented as PV generation),

should be investigated. Figure 4 demonstrates the correlation between the power consumed by the air-conditioner and the ambient temperature. A lag between a rise in the ambient temperature and the resulting increment in the air conditioner's consumed power can be clearly noticed. Therefore, while attempting to predict the AC consumption of the next hour, apart from the predicted ambient temperature of the next hour, the values corresponding to the previous hours should also be provided as inputs. The correlation investigation demonstrated that, in order to take into account the mentioned delay, the ambient temperature value in the next hours and the last 5 h should be utilized as inputs.

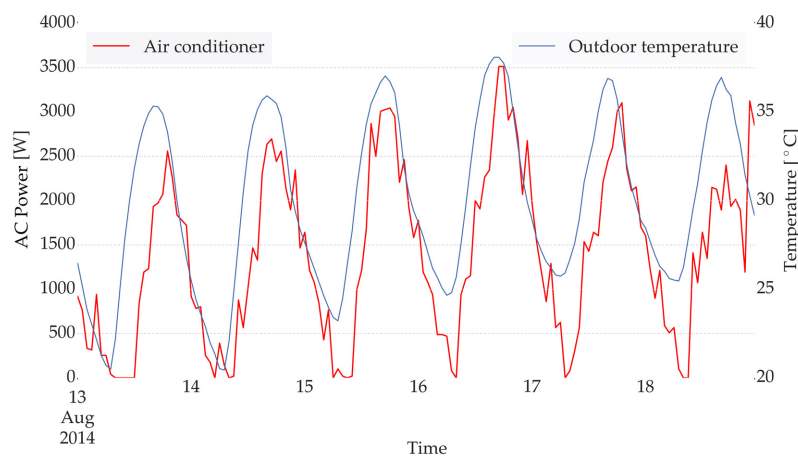


Figure 4. Correlation of temperature and AC power.

Figure 5 shows the correlation between the AC consumed power and the PV generation (which represent the solar irradiation in the considered location) in which a notably larger lag between the mentioned values can be observed. The conducted correlation tests demonstrated that, considering the mentioned remarkable lag, the PV generation values with the time lags of 5 and 6 h should be chosen as input features.

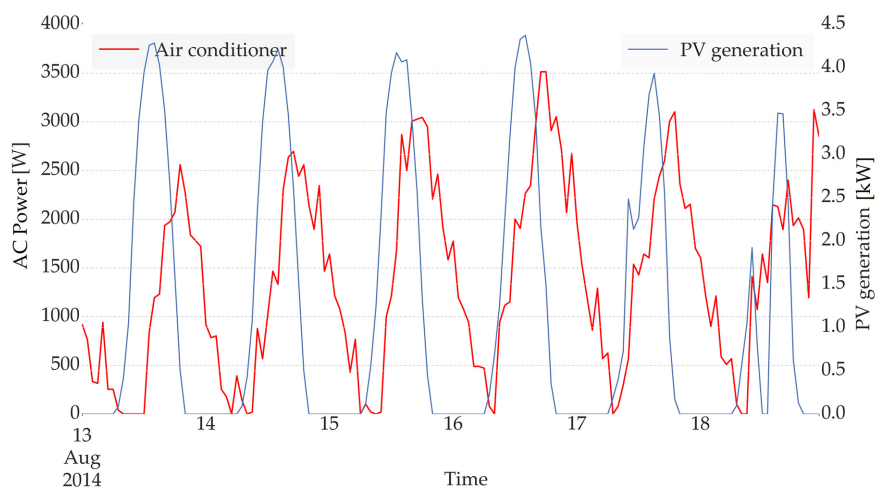


Figure 5. Correlation of irradiance and AC power.

It is noteworthy that an investigation was also conducted to evaluate the effect of relative humidity and a correlation was observed. However, it was demonstrated in the prediction step that including relative humidity as an input feature does not improve the accuracy of the developed prediction model. Hence, the relative humidity is not considered in the present work as one of the input features, which are provided to the machine-learning based models.

5.3. Considered Input Features and Employed Algorithms for Prediction

As previously mentioned, the power consumed by an AC unit at each time step is dependent on several variables which include the physical characteristics of the buildings, the occupancy profiles, the ventilation rate, and several other features. However, in the framework of the present work, the only accessible features are the AC consumption in the present and previous time steps, which are obtained through disaggregation of the smart meter's aggregate load, along with the ambient temperature and PV generation of the nearby plant (which represents solar irradiation). Considering the fact that the air-conditioner's consumed power profile is a time-series, in order to capture the consumption behaviour of the occupants, the corresponding values in the last hours should clearly be employed. Accordingly, the consumed power of AC in the last 24 h along with the corresponding value in the same hour (as the one, which is going to be predicted) in the previous week (lag of 168 h: 7 days \times 24 h) are considered as inputs. Furthermore, based on the results of the previous section, the ambient temperature in the next hour and the last 5 h along with the PV generation values (representing solar irradiation) with the time lags of 5 and 6 h are taken into account as weather related input features. Moreover, in an attempt to predict the occupancy profile in an indirect way, the parameters that represent the seasonality including the hour of day, day of the week, weekday/weekend are also utilized as inputs. Table 2 summarises the above-mentioned parameters which are considered as input features for the hour-ahead prediction. The given parameters with some minor differences (explained in Section 5.5) are also employed as input features for the day-ahead prediction.

Table 2. Features and target columns for the machine-learning model.

Input Description	N ^o of Inputs
AC consumed power in the last 24 h and 168 h (24 \times 7 days) before the hour which is predicted	25
Temperature in the next hour and last 5 h	6
PV production with time lags of 5th and 6th h	2
Hour of day	1
Weekend or Weekday (0 or 1)	1
Day or Night (0 or 1)	1

In order to carry out the prediction, various algorithms including Linear Regression, Support Vector Machine, Random Forest Decision Tree and MLP Neural Networks are employed. Hence, the implementation of these algorithms in Scikit-Learn [36], an open-source PythonTM-based dedicated machine learning package, are utilized.

5.4. Hour-Ahead Predictions

Employing the above-mentioned input features and utilizing the described machine-learning algorithms, the AC consumption in the next hour is estimated. The latter prediction is carried out for the months in which AC consumption exists which corresponds to May 1st 2014 until September 31st 2014 (except the first week in which one of the input features is missing) in the employed dataset.

The R^2 score and root mean squared errors (RMSE), which are achieved by utilizing the mentioned algorithms, are demonstrated in Table 3. The latter accuracy indices are determined by conducting cross validation on the whole dataset. It can be observed that Random Forest Decision Tree and MLP Neural Networks performed equally well on hour-ahead predictions leading to an R^2 score of 87.3%.

Table 3. Accuracy comparison of hour-ahead predictions conducted by various algorithms.

Algorithm	R^2 Score (%)	RMSE
Linear regression	84.7	196.963
Support Vector Machines	85.6	183.548
Random Forest Decision Tree	87.3	179.895
MLP Neural Network	87.3	178.447

Considering a practical scenario, the training and testing dataset's selection procedure can be different from that of cross-validation. The smart meter continuously collects consumption data in a real-time manner; thus, the available data for training is progressively increasing. Hence, in order to simulate such procedure, the training data can be considered to be updated month by month and the testing procedure can be conducted for the upcoming month. Table 4 shows the performance indices of Random Forest Decision Tree while conducting hour-ahead predictions using the mentioned method. It can be observed that, considering the notable changes in the ambient conditions in different months, more available data does not necessarily lead to an increment in the accuracy. Accordingly, the peak of accuracy (R^2 score of 90.1%) is achieved for the months of July for which the available training duration is 2 months. Such a training may also present real-time on-line prediction model using sliding window training or accumulative training as demonstrated by Yang et al. [37].

Table 4. Hour-ahead online training using a Random Forest Decision Tree.

Training Start	Training End	Testing Start	Testing End	Month	R^2 Score (%)
01-05-14	31-05-14	01-06-14	30-06-14	June	86.2
01-05-14	01-07-14	02-07-14	31-07-14	July	90.1
01-05-14	01-08-14	02-08-14	31-08-14	August	88.7
01-05-14	31-08-14	01-09-14	30-09-14	September	87.2

Figure 6 shows the comparison of hour-ahead predictions, using Random Forest Decision trees, and the real data for the interval between August 7th, 2014 and August 18th, 2014. The comparison shows an acceptable agreement apart from non-repeating peaks (due to the variations in the behaviour of the occupants during the weekends) which have not been predicted as accurately as other intervals.

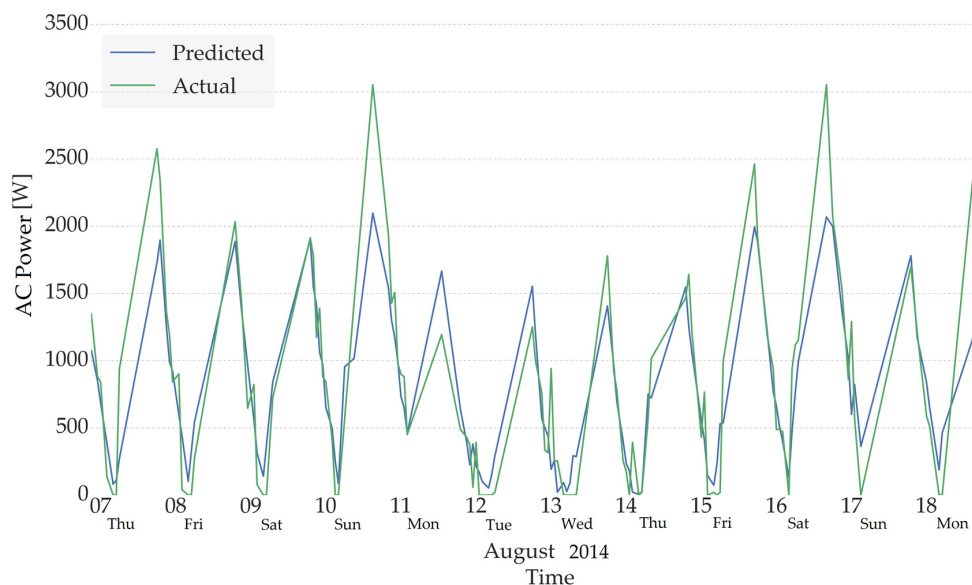


Figure 6. Hour-ahead predictions using Random Forest Decision Tree.

5.5. Day-Ahead Predictions

Since the consumption of air-conditioners, apart from weather conditions, is strongly dependent on the previous consumption patterns, the accuracy of the predictions models drop down for day-ahead predictions where the consumption of previous hours is not available. In the present study, it is assumed that accurate temperature and irradiance forecast data are available for the hours for which the prediction is performed. As shown in Table 2, the features for the prediction remain the same except for the lags in the AC consumption. The lagged consumption starts from 48 h to 24 h before

the current time step. The same machine learning models namely, Linear Regression, Support Vector Machine, Random Forest Decision Tree and MLP Neural Networks were employed to perform the day-ahead predictions. Table 5 compares the accuracy indices obtained using different algorithms while carrying out the day-ahead predictions. The Random Forest Decision Tree, with the R^2 score of 83.2%, shows a better accuracy compared to other algorithms. Apparently, the corresponding obtained accuracy is lower than the one of hour-ahead prediction as it latter lacks the information about the recent (last 24 h) consumption patterns.

Table 5. Performance metrics of day-ahead predictions.

Algorithm	R^2 Score (%)	RMSE
Linear Regression	74.4	255.011
Support Vector Machines	79.7	220.968
Random Forest Decision Tree	83.2	206.57
MLP Neural Network	80.6	220.282

Similar to hour-ahead predictions, the selection of training and testing datasets can also be performed in a sequential manner. Table 6 shows the performance of Random Forest Decision Tree on day-ahead predictions based on expanding training window for different months. A peak R^2 score of 85.7% is achieved for the month of July with available training data of 2 months.

Table 6. Day-ahead online training using Random Forest Decision Tree.

Training Start	Training End	Testing Start	Testing End	Month	R^2 Score (%)
01-05-14	31-05-14	01-06-14	30-06-14	June	82.2
01-05-14	01-07-14	02-07-14	31-07-14	July	85.7
01-05-14	01-08-14	02-08-14	31-08-14	August	84.8
01-05-14	31-08-14	01-09-14	30-09-14	September	83.1

A similar window from August 7, 2014 to August 18, 2014 for the day-ahead prediction is shown in Figure 7. As can be seen in this figure, the deviation of the predicted values from the real ones, for the intervals with non-repeating peaks (mainly taking place in the weekends), in the day-ahead predictions is larger than that of hour-ahead predictions.

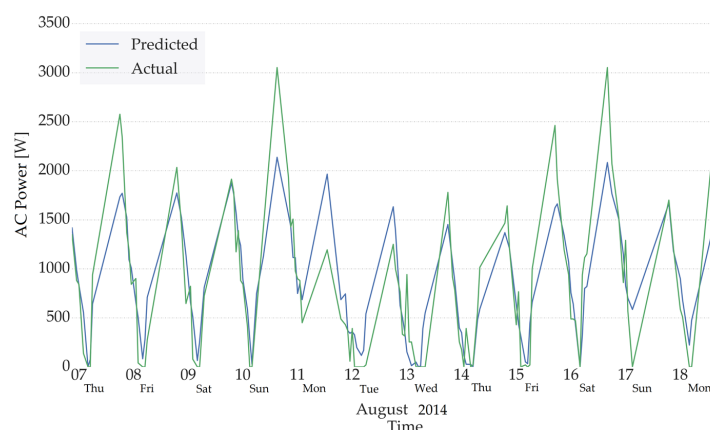


Figure 7. Day-ahead predictions using Random Forest Decision Tree.

5.6. Comparison of Predicting the Aggregate Load with and without Utilizing the Disaggregation Procedure

The results of the proposed methodology can also be employed in order to perform hour-ahead and day-ahead prediction of the overall aggregate load of the building. Accordingly, the same input

features, which were employed for predicting the air-conditioning (AC) consumptions, can also be utilized for predicting the remaining non-AC (aggregate loads subtracted by disaggregated AC) loads. The obtained predicted values can then be summed up with the predicted values of the AC consumption, thus predicting the overall aggregate load.

Clearly, a conventional alternative method to estimate the aggregate load is to directly predict the whole load using the same input features. The accuracy indices of predicting the aggregate load using these two methods are compared in Table 7 for hour-ahead prediction. It can be observed the former approach, which involves a disaggregation step, results in a higher accuracy. Thus, the methodology, which is proposed in the present study, can also enhance the prediction accuracy of the aggregate load. Table 8 shows a similar comparison for day-ahead predictions.

Table 7. Comparison of hour-ahead aggregate load prediction through the proposed 2-step method (which includes disaggregation) and prediction only using the aggregate load.

Model	R ² Score (%)		RMSE	
	Proposed 2-Step Method	Prediction Using Aggregate Load	Proposed 2-Step Method	Prediction Using Aggregate Load
Linear Regression	86.12	85.23	236.38	243.85
Support Vector Machines	85.24	84.48	243.78	244.837
Random Forest Decision Tree	88.67	87.57	213.58	223.649
MLP Neural Network	87.02	86.36	228.539	233.645

Table 8. Comparison of day-ahead aggregate load prediction through the proposed 2-step method (which includes disaggregation) method and prediction only using the aggregate load.

Model	R ² Score (%)		RMSE	
	Proposed 2-Step Method	Prediction Using Aggregate Load	Proposed 2-Step Method	Prediction Using Aggregate Load
Linear Regression	75.98	73.98	310.93	323.65
Support Vector Machines	78.35	78.00	295.24	288.235
Random Forest Decision Tree	82.42	81.4	266.02	269.64
MLP Neural Network	81.60	79.34	272.134	288.37

6. Discussion

The results of the disaggregation step demonstrated that only employing a short training period (one week) and using combinatorial optimization, which is a simple disaggregation algorithm with a low computational cost, an elevated yearly disaggregation accuracy (98.67%) can be achieved. Furthermore, a complete database including electrical signatures obtained from different combinations of domestic appliances can be employed as a training dataset; thus, the training period, with dedicated sensors for different devices, can accordingly be avoided. Therefore, it can be concluded the disaggregation step is not a practical impediment to the proposed method and does neither introduce a notable error in the prediction procedure, as the disaggregation accuracies are notably high.

The investigation on the correlation between the AC consumption and the ambient condition, demonstrated the expected lag between an increment in the ambient temperature and the resulting rise

in the AC consumption. The observed lag is due to the thermal storage of the walls, which results in a delay in the conversion of temperature increment to an increase in the AC load. Furthermore, an even larger time lag was observed while investigating the effect of solar irradiation (through generation profile of a nearby PV plant). The latter lag is due to the delay in the conversion of the radiation heat transfer into a convective one, as the incident solar irradiation will first heat up the walls and the objects inside the building (through windows) and they will in turn warm up the internal air through convection. Accordingly, the predicted ambient temperature of the next hour along with the corresponding values in the last 5 h were chosen as the inputs while the PV generations with the lag of 5 and 6 h were taken into account to represent solar irradiation. The latter demonstrates the importance of having the knowledge and keeping in mind the physical behaviour of buildings even while employing a purely data-driven approach.

The results of hour-ahead and day-ahead prediction demonstrates the fact that using the proposed approach elevated accuracies can be obtained while only the aggregate load data and historical weather data is available and no information regarding the building characteristics or occupants' behaviour was given. The latter is owing to the fact that providing the AC consumption for several time steps and the corresponding ambient temperature and PV generation (representing solar irradiation), while applying the appropriate time lags, facilitates simulating the physical behaviour of the building in an implicit way. Furthermore, considering the seasonality related parameters (hour, day of the week, weekday/weekend), and the consumption profiles in the previous hours, provides information regarding the occupancy profile in an indirect manner. The fact that outside weather condition are included can even result in a better prediction of occupancy as the occupants will be more willing to stay at home and use the air-conditioner at the hours with high ambient temperature and elevated solar irradiation. The level of the latter willingness is apparently learnt from similar conditions taking place in the previous measured periods.

However, the latter information is not enough in order to predict the non-repeating alterations (peaks) in the behaviour of the occupants. These alterations commonly happen in the weekends and the prediction results demonstrated that the developed models could not anticipate such sudden and non-repeating variations.

7. Conclusions

A two-step methodology was proposed and implemented in order to perform hour-ahead and day-ahead prediction of air-conditioning (AC) load in residential buildings. The first step was separating the AC load from the overall consumption and it was demonstrated that using Combinatorial Optimization method and with training data of seven days, a disaggregation R^2 score 98.67% can be obtained. The second step was focused on performing predictions using various machine learning algorithms. It was demonstrated that Random Forests Decision Trees provides the most accurate prediction and leads to R^2 scores of 87.3% and 83.2% for hour-ahead and day-ahead predictions respectively. Hence, it was shown that even without any available information regarding physical characteristics of the building, using the proposed methodology, elevated prediction accuracies can be obtained. It was also demonstrated that the same prediction procedure can be utilized in order to predict the remaining non-AC loads (aggregate load subtracted by disaggregated AC load) and the latter can then be summed up with the predicted AC consumption in order to estimate the overall aggregate load. It was demonstrated that, both for hour-ahead and day-ahead predictions, using the mentioned two-step methodology results in a higher accuracy compared to directly predicting the aggregate load. Furthermore, the proposed methodology provides the utility with some additional benefits including the possibility of providing itemized bills, tailor-made energy saving recommendations for customers and diagnosis of damaged or faulty air-conditioners.

Acknowledgments: The authors would like to acknowledge Politecnico di Milano's Department of Energy for providing financial support for this project.

Author Contributions: Manoj Manivannan carried out the disaggregation step; Behzad Najafi and Manoj Manivannan conducted the correlation study and built the machine-learning models; Behzad Najafi and Fabio Rinaldi provided domain knowledge in the area of building physics, which was employed in the feature selection step. Behzad Najafi and Fabio Rinaldi directed the project and prepared the paper.

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

References

1. Ferreira, P.M.; Ruano, A.E.; Silva, S.M.; Conceição, E. Neural networks based predictive control for thermal comfort and energy savings in public buildings, energy and buildings. *Energy Build.* **2012**, *55*, 238–251. [[CrossRef](#)]
2. U.S. Energy Information Administration. Electric power monthly December 2016. Available online: <https://www.eia.gov/electricity/monthly> (accessed on 15 March 2017).
3. Álvarez, J.; Redondo, J.; Camponogara, E.; Normey-Rico, J.; Berenguel, M.; Ortigosa, P. Optimizing building comfort temperature regulation via model predictive control. *Energy Build.* **2013**, *57*, 361–372. [[CrossRef](#)]
4. Paterson, G.; Mumovic, D.; Das, P.; Kimpian, J. Energy use predictions with machine learning during architectural concept design. *Sci. Technol. Built Environ.* **2017**, *23*, 1036–1048. [[CrossRef](#)]
5. Hart, G.W. Nonintrusive appliance load monitoring. *Proc. IEEE* **1992**, *80*, 1870–1891. [[CrossRef](#)]
6. Hart, G.W. *Nonintrusive Appliance Load Data Acquisition Method*; Progress Report; MIT Energy Laboratory: Concord, MA, USA, 1984.
7. Anstett, M.; Kreider, J.F. Application of neural networking models to predict energy use. *ASHRAE Trans.* **1993**, *99*, 505–517.
8. Jain, R.K.; Smith, K.M.; Culligan, P.J.; Taylor, J.E. Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy. *Appl. Energy* **2014**, *123*, 168–178. [[CrossRef](#)]
9. Chen, Y.; Tan, H. Short-term prediction of electric demand in building sector via hybrid support vector regression. *Appl. Energy* **2017**, *204*, 1363–1374. [[CrossRef](#)]
10. Karatasou, S.; Santamouris, M.; Geros, V. Modeling and predicting building's energy use with artificial neural networks: Methods and results. *Energy Build.* **2006**, *38*, 949–958. [[CrossRef](#)]
11. Edwards, R.E.; New, J.; Parker, L.E. Predicting future hourly residential electrical consumption: A machine learning case study. *Energy Build.* **2012**, *49*, 591–603. [[CrossRef](#)]
12. Basu, K.; Hawarah, L.; Arghira, N.; Joumaa, H.; Ploix, S. A prediction system for home appliance usage. *Energy Build.* **2013**, *67*, 668–679. [[CrossRef](#)]
13. Dong, B.; Li, Z.; Rahman, S.M.M.; Vega, R. A hybrid model approach for forecasting future residential electricity consumption. *Energy Build.* **2016**, *117*, 341–351. [[CrossRef](#)]
14. Dong, B.; Cao, C.; Lee, S.E. Applying support vector machines to predict building energy consumption in tropical region. *Energy Build.* **2005**, *37*, 545–553. [[CrossRef](#)]
15. Kontokosta, C.E.; Tull, C. A data-driven predictive model of city-scale energy use in buildings. *Appl. Energy* **2017**, *197*, 303–317. [[CrossRef](#)]
16. Li, X.M.; Shao, M.; Ding, L.X.; Xu, G.; Li, J.B. Particle swarm optimization-based ls-svm for building cooling load prediction. *J. Comput.* **2010**, *5*, 614–621. [[CrossRef](#)]
17. Fan, C.; Xiao, F.; Zhao, Y. A short-term building cooling load prediction method using deep learning algorithms. *Appl. Energy* **2017**, *195*, 222–233. [[CrossRef](#)]
18. Li, Q.; Meng, Q.; Cai, J.; Yoshino, H.; Mochida, A. Predicting hourly cooling load in the building: A comparison of support vector machine and different artificial neural networks. *Energy Convers. Manag.* **2009**, *50*, 90–96. [[CrossRef](#)]
19. Yao, Y.; Lian, Z.; Liu, S.; Hou, Z. Hourly cooling load prediction by a combined forecasting model based on analytic hierarchy process. *Int. J. Therm. Sci.* **2004**, *43*, 1107–1118. [[CrossRef](#)]
20. González, P.A.; Zamarreño, J.M. Prediction of hourly energy consumption in buildings based on a feedback artificial neural network. *Energy Build.* **2005**, *37*, 595–601. [[CrossRef](#)]
21. Ben-Nakhi, A.E.; Mahmoud, M.A. Cooling load prediction for buildings using general regression neural networks. *Energy Convers. Manag.* **2004**, *45*, 2127–2141. [[CrossRef](#)]

22. Castelli, M.; Trujillo, L.; Vanneschi, L.; Popovič, A. Prediction of energy performance of residential buildings: A genetic programming approach. *Energy Build.* **2015**, *102*, 67–74. [[CrossRef](#)]
23. Pecan Street Inc. DataPort. Available online: <https://dataport.cloud/> (accessed on 15 January 2017).
24. Batra, N.; Kelly, J.; Parson, O.; Dutta, H.; Knottenbelt, W.; Rogers, A.; Singh, A.; Srivastava, M. Nilmtk: An open source toolkit for non-intrusive load monitoring. In Proceedings of the 5th international conference on Future energy systems, Cambridge, UK, 11–13 June 2014; pp. 265–276.
25. Zeifman, M.; Roth, K. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Trans. Consum. Electron.* **2011**, *57*, 76–84. [[CrossRef](#)]
26. Jiang, L.; Luo, S.; Li, J. Automatic power load event detection and appliance classification based on power harmonic features in nonintrusive appliance load monitoring. In Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), Melbourne, VIC, Australia, 19–21 June 2013; pp. 1083–1088.
27. Banerjee, R. Development of pc based transient current analysis system using microcontroller and hall effect sensor. *Int. J. Eng. Res. Gen. Sci.* **2015**, *3*, 321–326.
28. Chang, H.H.; Lin, C.L.; Lee, J.K. Load identification in nonintrusive load monitoring using steady-state and turn-on transient energy algorithms. In Proceedings of the 2010 14th International Conference on Computer Supported Cooperative Work in Design, Shanghai, China, 14–16 April 2010; pp. 27–32.
29. Hsueh-Hsien, C.; Ching-Lung, L.; Hong-Tzer, Y. Load recognition for different loads with the same real power and reactive power in a non-intrusive load-monitoring system. In Proceedings of the 2008 12th International Conference on Computer Supported Cooperative Work in Design, Xi'an, China, 16–18 April 2008; pp. 1122–1127.
30. Zoha, A.; Gluhak, A.; Imran, M.A.; Rajasegarar, S. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors* **2012**, *12*, 16838–16866. [[CrossRef](#)] [[PubMed](#)]
31. Parson, O. Unsupervised Training Methods for Non-Intrusive Appliance Load Monitoring from Smart Meter Data. Ph.D. Thesis, University of Southampton, Southampton, UK, 2014.
32. Virtanen, T. Speech Recognition Using Factorial Hidden Markov Models for Separation in the Feature Space. In Proceedings of the 9th International Conference on Spoken Language Processing, Pittsburgh, PA, USA, 17–21 September 2006.
33. Ghahramani, Z.; Jordan, M.I. Factorial hidden markov models. *Mach. Learn.* **1997**, *29*, 245–273. [[CrossRef](#)]
34. Kolter, J.Z.; Johnson, M.J. *Redd: A Public data Set for Energy Disaggregation Research*; SustKDD: San Diego, CA, USA, 2011.
35. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning—Data Mining, Inference and Prediction*; Springer-Verlag New York: New York, NY, USA, 2009.
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Vanderplas, J. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
37. Yang, J.; Rivard, H.; Zmeureanu, R. On-line building energy prediction using adaptive artificial neural networks. *Energy Build.* **2005**, *37*, 1250–1259. [[CrossRef](#)]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).