Day-Ahead Building Load Forecasting with a Small dataset [⋆]

Marco Lauricella, **Zhongtian Cai**, and **Lorenzo Fagiano**

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, Italy.
e-mail: {marco.lauricella | lorenzo.fagiano}@polimi.it.

Abstract: A new method is presented, to derive an algorithm that provides a forecast of one-day-ahead electricity consumption of a building. The approach aims to obtain high accuracy with a small dataset of 1-2 weeks, motivated by practical situations where the building is new or subject to relatively frequent changes, and/or limited local computation and memory are available. The method introduces a fictitious input signal that captures the prior information on the periodic behavior of building load time series. Moreover, the use of a linear model structure enables the derivation of guaranteed accuracy bounds on the forecast error, which can be used in day-ahead energy scheduling and optimization. Using an experimental dataset with measurements collected from an office building, it is found that the fictitious input can largely improve the prediction accuracy of the model, outperforming linear predictors and scoring a performance similar to that of nonlinear ARX models, such as recurrent neural networks, while retaining the capability to provide guaranteed accuracy bounds.

Keywords: Load Forecasting, Smart grid, Smart Buildings, Energy Prediction, System Identification

1. INTRODUCTION

Buildings are major energy consumers worldwide, accounting for 20%-40% of the total energy demand (EIA, 2015; Pérez-Lombard et al., 2008). Energy consumption forecasts allow system operators to plan the energy use over time, shift demand to off-peak periods, and make more favorable energy purchase plans. This motivates the increasing interest in building load forecasting (Bourdeau et al., 2019), at different time-scales. Approaches in the literature range from autoregressive models (AR, ARX, ARIMA, Seasonal ARIMA) (Espinoza et al., 2005; Fan and McDonald, 1994; Yun et al., 2012; Taylor and McSharry, 2007), to Support Vector Machines (SVM) (Chen et al., 2004), Artificial Neural Networks (ANN) (González and Zamarreño, 2005; Rahman et al., 2018; Cai et al., 2019) and Genetic Algorithms (GA). For a more comprehensive list see e.g. Ahmad et al. (2014); Deb et al. (2017), and the references therein. These black-box models are entirely developed from historical data, of which they require a relatively large amount (typically 1-2 years). Buildings also feature daily/weekly/monthly/seasonal trends in their energy consumption, in addition to an influence from weather (Massana et al., 2015). Forecasting methods were proposed to take into account such periodic trends, resorting for example to Fourier Series (Dhar et al., 1998), periodic AR models (Taylor and McSharry, 2007), or indexed ARX models (Yun et al., 2012). These approaches obtain comparable results to those of Neural Networks (NN) (Yun et al., 2012), with the advantage of having a simpler and more easily understandable model structure. Still, they require the availability of big datasets to identify seasonal trends only from data.

Such a need for large datasets represents a first potential problem of the mentioned approaches, making them unreliable when only recent energy consumption data are available, or when the memory for data storage is limited. In fact, even in the era of Big Data there is still an interest in the development of accurate forecasting models for industrial applications with limited memory and computational power, without resorting to cloud services and large databases. In addition, one can be interested in a ready-to-go forecasting algorithm to predict the energy consumption of a newly constructed or renovated building. Finally, it is of interest to derive an accurate forecasting model from a small, recent dataset also when the building may be subject to relatively frequent changes in purpose, number of occupants, or installed/used equipments. The development of a forecasting algorithm able to achieve state-of-the-art performance using a small dataset is thus a relevant open problem. A second problem of the existing approaches is the fact that prediction uncertainty is seldom considered in the algorithm derivation phase. Yet, algorithms that can provide both a nominal forecast and an associated error interval are extremely important when taking decisions on the basis of the load prediction. In particular, deriving tight error bounds for complex nonlinear models, such as ANNs, is currently an open problem, while recent results have been developed for linear models (Lauricella and Fagiano (2020)).

In this paper, we propose a novel method that aims to solve both problems. On the one hand, we inject the available prior information on the periodicity of energy consumption trends by constructing a fictitious input signal that is fed to the model, in addition to weather forecast. The fictitious input mimics a frequency decomposition of the periodic part of the load behavior, but does not require any data to be constructed. On the other hand, it captures well the nonlinearity in the load behavior, and allows us to resort to a linear ARX predictor which, identified with a short dataset (e.g., two weeks), achieves on our test-

[⋆] Marco Lauricella was financially supported by the ABB - Politecnico di Milano Joint Research Center through the PhD scholarship "Methodologies for the Estimation and Monitoring from Data of Electric Systems". Corresponding author: Marco Lauricella

ing data much better performance than nonlinear models trained with the same dataset, and comparable to those of nonlinear models trained with much larger datasets (e.g., two years). Furthermore, the model being linear, we apply the Set Membership (SM) approach proposed in Lauricella and Fagiano (2020) to compute tight guaranteed bounds on the forecasting error.

After describing the method, we test and compare it with established ones using an experimental dataset with measurements collected in an office building hosting around 200 employees, equipped with electrical heating and cooling (Fig. 1).

Fig. 1. ABB office building providing the measurements of the adopted experimental dataset.

2. PROBLEM DESCRIPTION AND FORECASTING PROCEDURE

We assume to have a dataset containing the time series of electric power consumed by a building, averaged every quarter-hour, for a number N of subsequent days in the recent past. Specifically, the measured values $\tilde{y}_i(k)$ are available, $k = 1, \ldots, 96$, each one representing the average electric power consumed in the k-th quarter-hour of day i, $i = 1, \ldots, N$. Moreover, for the same days we assume that weather measurements are available as a sequence of vectors $u_{w,i}(1), \ldots, u_{w,i}(96)$, $u_{w,i}(k) \in \mathbb{R}^{nw}$. Components of the weather vector in each time interval typically include the external temperature, relative humidity, solar irradiation and wind speed, measured by a weather station close to the considered building.

The problem we address can be classified as a short-term load forecasting one. Using the available dataset covering N days, we want to derive a one-day-ahead forecasting algorithm. Namely, at the beginning of each day, we want the algorithm to predict the course of quarterly electricity consumption for the same day, together with prediction error bounds. The algorithm will entail the simulation, for one day, of a discrete-time autoregressive model of order n_y with suitable predicted input signals, and we allow the model to be initialized with the first n_y load measurements of the day. Thus, the derived forecasting algorithm will produce $96 - n_y$ predicted values of electricity consumption at time step n_y of each day. As an example, with $n_y = 3$ the algorithm will generate, at time $0:45$ of the considered day, the estimated electricity consumption from time $1:00$ every 15 minutes up to time $24:00$. We anticipate that this choice is arbitrary and without loss of generality, since in our approach the derived model is time-invariant and can be initialized at any instant to predict the load at any future time step.

Finally, in our setup it is intended that the forecasting algorithm is used for a limited number M of days, after which a new one is derived from a new dataset of N most recent past days and used for further M days, in a moving horizon fashion. For example, with $N = 14$ and $M = 7$, we obtain the following procedure.

1) At the beginning of each week, use the two most recent past weeks of data to derive the one-day-ahead forecasting algorithm valid for that week;
2) At the beginning of each day of the week, compute the one-day-ahead load forecast and uncertainty interval, using the forecasting algorithm obtained at step 1), initialized with first n_y load measurements of the considered day;
3) At the end of the week, update the dataset with the most recent measurements, adopting a moving window criterion with a fixed length of two weeks, and go to 1).

As anticipated in the introduction, the use of a small dataset is seldom considered in the literature and poses problems to established techniques, such as those based on neural networks and Fourier series, which need large amount of data in the training phase. On the other hand, the capability to produce accurate forecasts with a few, recent past data enables the load prediction function on industrial equipment, with limited memory and computational power, and allows one to employ the above-described moving horizon approach, which is able to adapt to changes in the building (e.g. renovation works, installation/replacement of equipment, change of purpose, etc.), and can be used on new buildings where no historical data is available.

The key point of the procedure is the method employed to derive the forecasting algorithm, introduced next.

3. LOAD FORECASTING METHOD

The proposed approach combines the use of: a) a fictitious input signal, aimed to capture the nonlinear and periodic trend of electricity consumption; b) the identification of a discrete time autoregressive model with exogenous input (ARX) using a simulation error method (SEM); and c) the use of Set Membership methods to estimate bounds on the simulation error of the model, which will provide the forecast error interval. Each one of these ingredients is described in the following sub-sections.

3.1 Fictitious input

Buildings have daily, weekly and seasonal patterns in their consumption profile (Massana et al., 2015). Several literature contributions employ this knowledge in different ways, some of which are reported in Section 1. All these approaches share the common need for a large-enough dataset, to be able to identify all the seasonal trends that are present in the analyzed time series. Here, we propose a method to exploit this prior knowledge even with short datasets. We introduce an artificial signal, named fictitious input, to excite the system model according to the expected periodic behavior of the energy consumption. The fictitious input is simply a linear combination of sine and cosine functions with unitary amplitude. The idea is similar to that of Fourier series, but the harmonics of the signal are not computed from the data: rather, they are a priori chosen as fractions of days and weeks based on an expert choice of the relevant frequency contributions.

Thus, at each time step $k = 1, \ldots, 96$ of each day $i \geq 1$, the fictitious input can be written as:

$$u_{f,i}(k) = \begin{bmatrix} \cos(\omega_1(k+96(i-1))) \\ \vdots \\ \cos(\omega_{n_\omega}(k+96(i-1))) \\ \sin(\omega_1(k+96(i-1))) \\ \vdots \\ \sin(\omega_{n_\omega}(k+96(i-1))) \end{bmatrix} \qquad (1)$$

where n_ω is the number of considered frequency contributions, and each $\omega_j = 2\pi f_j$ corresponds to one of the selected harmonics. Thus, we have $u_{f,i}(k) \in \mathbb{R}^{n_f}$ with $n_f = 2n_\omega$.

Here, considering that the sampling period is 15 minutes, we define $f_j = \frac{j}{4 \cdot 24 \cdot 7}$, so that, for example, with $n_\omega = 14$ one would consider sine and cosine waves with periods in the interval between 7 days ($j = 1$) and 12 hours ($j = 14$). As anticipated, the multivariable signal (1) is given as input to a linear ARX model to be identified. The subsequent model identification phase will then shape the amplitude of each component in order to match the resulting power consumption with the one measured and stored in the training dataset, as detailed next.

3.2 Model structure and parameter identification

Let us denote with $\hat{y}_i(k|n_y)$ the load predicted at time step $k + n_y$ of day i, given the information available at time n_y of the same day. The one-day-ahead load forecast is computed by simulating a dynamical model of the form:

$$\hat{y}_i(k+1|n_y) = \varphi_i(k|n_y)^T\theta, \quad k = 0, \ldots, (95 - n_y) \qquad (2)$$

where ^{T} denotes the matrix transpose operation, n_y is the model order, $\theta \in \mathbb{R}^{n_y+n_u\,(n_W+n_f)}$ is the vector of model parameters to be identified, and vector $\varphi_i(k|n_y) \in \mathbb{R}^{n_y+n_u\,(n_W+n_f)}$ ("regressor") is given by:

$$\varphi_i(k|n_y) = \left[Y_i^T(k|n_y)\; U_i^T(k)\right]^T. \qquad (3)$$

If $k \geq n_y$, vector $Y_i(k|n_y) \in \mathbb{R}^{n_y}$ in (3) is built as:

$$Y_i(k|n_y) = [\hat{y}_i(k|n_y),\, \hat{y}_i(k-1|n_y), \ldots, \hat{y}_i(k-n_y+1|n_y)]^T \qquad (4)$$

otherwise, for each $k = 0, \ldots, n_y - 1$, the estimated values $\hat{y}_i(k-j|n_y)$ in (4), $j = k, \ldots, n_y - 1$, are replaced by the measured values $\tilde{y}_i(k-j+n_y)$, which are available at time step n_y of day i (initial conditions).

Finally, vector $U_i(k) \in \mathbb{R}^{n_u(n_W+n_f)}$ in (3) contains the input signals as:

$$\begin{aligned} U_i(k) = \big[&u_{w,i}(k)^T,\, \ldots,\, u_{w,i}(k-n_u+1)^T, \\ &u_{f,i}(k)^T,\, \ldots,\, u_{f,i}(k-n_u+1)^T\big]^T, \end{aligned} \qquad (5)$$

where n_u is a value between 1 and n_y. We adopt the same value of n_u for both the weather inputs and the fictitious input for the sake of simplicity, without loss of generality.

Having defined the model structure, the parameter identification is carried out by solving the following unconstrained nonlinear program:

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} \left\|\tilde{\boldsymbol{Y}}_i - \hat{\boldsymbol{Y}}_i(\theta)\right\|_2^2 \qquad (6)$$

where N denotes the number of days in the considered simulation horizon (i.e. $N = 14$ for a two weeks long dataset), and

$$\begin{aligned} \tilde{\boldsymbol{Y}}_i &= [\tilde{y}_i(n_y+1),\, \ldots,\, \tilde{y}_i(96)]^T \\ \hat{\boldsymbol{Y}}_i(\theta) &= [\hat{y}_i(1|n_y)\, \ldots\, \hat{y}_i(96-n_y|n_y)]^T, \end{aligned}$$

with $\hat{y}_i(k|n_y)$ computed via recursion of (2)-(5), hence the dependence of vector $\hat{\boldsymbol{Y}}_i$ on θ. Notice that, to compute $\hat{y}_i(k|n_y)$, weather forecast should be used in place of future weather input measurements. Problem (6) shall be formulated and solved at step 1) of the procedure outlined in Section 2. Standard sequential quadratic programming algorithms, such as the Gauss-Newton one, are well-suited for this problem class and very efficient in terms of required computational time. Then, the obtained parameter vector $\hat{\theta}$ is used in step 2) of the procedure to compute the wanted load forecasts, by simply iterating the model (2)-(5) for the future days.

3.3 Accuracy bounds

Besides simplicity and the availability of a well-established theory, the advantage of using a linear ARX model like (2) is the possibility to apply the Set Membership (SM) approach presented in Lauricella and Fagiano (2020) to compute guaranteed accuracy bounds for the one-day-ahead load forecast over the considered horizon. Originally developed for linear time-invariant systems subject to output disturbances, whose input is perfectly known or measured without noise, this SM approach can be in fact applied in the framework considered here. By doing so, the computed bounds will account both for model mismatch and for the inaccuracy of the employed weather forecast. For the sake of completeness, the method is briefly recalled in the following. For a more complete description the interested reader is referred to Lauricella and Fagiano (2020).

We start by considering the one-step-ahead ARX predictor defined in (2), and all the multi-step predictors obtained by its recursion for all the considered time steps in the one-day-ahead forecasting horizon, which are denoted by:

$$\hat{y}_i(k|n_y) = \bar{\varphi}_{k,i}^T \bar{\theta}_k. \qquad (7)$$

Note that, in (7), the regressor $\bar{\varphi}_{k,i}$ is now a vector whose size increases with k, since it contains the initial conditions and all the subsequent input values applied to the system. Similarly, the parameter vector $\bar{\theta}_k$ increases in size with k, and its elements are polynomial functions of the elements of the model parameters θ in (2), obtained by the k-step model recursion. Then, for each k we can estimate the disturbance bound $\bar{\varepsilon}_k$ through the following linear program (LP):

$$\begin{gathered} \bar{\varepsilon}_k = \min_{\bar{\theta}_k,\, \varepsilon \in \mathbb{R}^+} \varepsilon \quad \text{subject to} \\ \left|\tilde{y}_i(k+n_y) - \bar{\varphi}_{k,i}^T \bar{\theta}_k\right| \leq \varepsilon, \;\; i = 1, \ldots, N, \end{gathered}$$

It is customary to employ a scaling factor $\alpha > 1$ on the obtained value, to account for the use of a finite dataset:

$$\hat{\bar{\varepsilon}}_k = \alpha \bar{\varepsilon}_k,\ \alpha > 1.$$

Then, we define the set of multi-step parameter values that are consistent with the measured data and with the estimated disturbance bound for each k-step predictor, called Feasible Parameter Set (FPS):

$$\Theta_k \doteq \left\{\bar{\theta}_k : \left|\tilde{y}_i(k) - \bar{\varphi}_{k,i}^T \bar{\theta}_k\right| \leq \hat{\bar{\varepsilon}}_k,\ i = 1, \ldots, N\right\}$$

Now, consider that the forecast algorithm obtained through the approach of Section 3.2 exploits the model parameters $\hat{\theta}$ (6). We indicate the corresponding multi-step parameters, obtained by iterating k times the one-step-ahead model (2), as $\hat{\bar{\theta}}_k$. Then, for a given regressor $\bar{\varphi}_{k,i}$, the local error bound for the computed forecast algorithm is estimated as:

$$\hat{\tau}_{k,i}(\bar{\varphi}_{k,i}) = \gamma\Big(\max_{\bar{\theta}_k \in \Theta_k} \left|\bar{\varphi}_{k,i}^T(\bar{\theta}_k - \hat{\bar{\theta}}_k)\right|\Big) + \hat{\bar{\varepsilon}}_k,\ \gamma > 1, \qquad (8)$$

where γ, like α, is a scaling parameter that accounts for the uncertainties due to the usage of a finite dataset. The computation of $\hat{\tau}_{k,i}$ entails the solution of 2 LPs. The obtained forecast accuracy bound, $\hat{\tau}_{k,i}$, holds for the $n_y + k$ time step of the considered day. Repeating the computation for $k = 1, \ldots, 96 - n_y$ yields the forecast error bounds along the whole day. In the next Section, we illustrate the performance of the method and the tightness of the computed bounds on an experimental dataset.

4. EXPERIMENTAL RESULTS

In this paper, we use a dataset of measurements collected during two years in an office building of ABB SpA in Bergamo, Italy. The dataset consists of measurements of the overall building load, averaged every 15 minutes, and of weather variables measured every 15 minutes, specifically, ambient temperature, relative humidity, solar irradiation and wind speed. The considered building hosts around 200 employees, and it features an electric heating and cooling system, elevators, lighting system, a kitchen with fridges, computers and printers, a UPS, an e-vehicles charging station, and electronic laboratories. In Fig. 2, all the available building load measurements are illustrated through a box-plot, while Fig. 3 depicts the load measured during two weeks between August and September 2017.

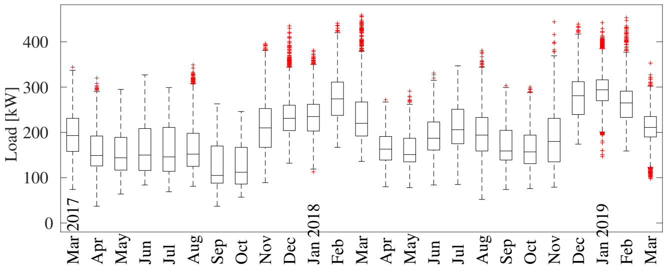

Fig. 2. Box-plot of the building load measured over the entire dataset.

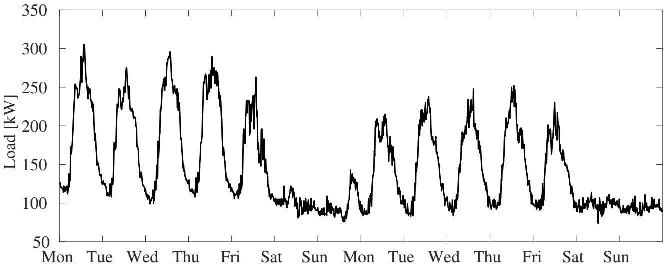

Fig. 3. Building load measurements over a two weeks interval (28 Aug 2017 to 10 Sep 2017).

The available measurements are then rearranged into two different sets, named Set A and Set B. Set A is a collection of nine randomly selected testing weeks, reported in Table 1, and of the corresponding training weeks, i.e. the two weeks preceding each considered testing week. During the random selection, all the weeks including a non-working day happening outside the weekend were excluded from the dataset. Set A is then used to train and test forecasting models with a limited amount of data. Set B is composed of two full years of training data (from 1 March 2017 to 28 February 2019), and of the four weeks of testing data listed in Table 2. The data used during the training and testing phases are previously normalized to zero mean and unitary variance, where the mean and the standard deviation are computed over the available training dataset. The obtained load forecast is then denormalized using the same mean and standard deviation used for the data normalization.

Table 1. Test sets for Group (A)

	2017			2018				2019	
Start	Jun19	Sep11	Nov20	Feb19	Jun18	Sep17	Dec10	Feb4	Mar18
End	Jun25	Sep17	Nov26	Feb25	Jun24	Sep23	Dec16	Feb10	Mar24

Table 2. Test sets for Group (B)

	2019			
Start	Mar4	Mar11	Mar18	Mar25
End	Mar10	Mar17	Mar24	Mar31

In this section, we compare the forecasting performance of the proposed prediction model, described in Section 3.2, with that of four different forecasting approaches. In particular, we use as benchmark an ARX predictor trained with the SEM approach, i.e., minimizing the simulation error across the considered prediction horizon, see e.g. Söderström and Stoica (1989), where the only inputs are the weather variables, which provides a baseline for standard linear regression models without fictitious input, and a nonlinear autoregressive network with exogenous input (NARX), which is the nonlinear counterpart of ARX predictors. A NARX architecture is a mixture of neural networks and autoregressive time series methods, where the output is computed based on its previous values, and on previous values of the exogenous variables. In the literature, NARX neural networks are often considered to be one of the best performing machine learning approaches for building load forecasting (Yildiz et al., 2017). Their training can be done both in open loop, if the load values used as input for the regression are the measured ones, or in closed loop, if the forecasted load is fed back to the network as input. The one-day-ahead forecast is always obtained in closed loop. The structure of the considered models, and the training procedures adopted for the different predictors are detailed in the following.

ARX models Model ① is the ARX predictor with fictitious input described by (2), while model ② is an ARX predictor without fictitious input, that corresponds to (2) where, in the regressor defined by (3), $U_i(k)$ contains only the weather variables $u_{w,i}(k)$. Both models are trained solving (6) using a two weeks long dataset, with an handmade optimizer coded on MatLab to solve unconstrained nonlinear programs (NLPs), employing a line search algorithm based on the Gauss-Newton method (Nocedal and Wright, 2006). The model order is then tuned to minimize cross-validation error and model complexity, trying different combinations of n_y and n_u on a randomly chosen subset of data. We found that $n_y = 3$ and $n_u = 1$ minimize the weekly average prediction error, while providing a sufficiently low standard deviation of the hourly forecasting error, with a limited model complexity.

NARX models Model ③ is a NARX network trained in open loop over a two-weeks-long dataset, with inputs given by weather variables and the fictitious input signal. Model ④ is equivalent to model ③, but its training is performed in closed loop. In both cases, the NARX network is characterized by a single hidden layer with one neuron having an hyperbolic tangent (sigmoid) activation function, with autoregressive order equal to 3, where the output layer is given by one neuron with linear activation function.

The training is performed using the Bayesian Regulation Backpropagation algorithm provided with the MatLab 9.5 Deep Learning Toolbox (ver. 12.0), that minimizes a combination of mean squared forecasting error and weights. More details can be found in Foresee and Hagan (1997). The chosen network structure is the one providing the best performance in cross-validation among several trials.
Model ⑤ is a 'state-of-the-art' NARX network, whose structure emulates the best performing forecasting model among the ones reviewed in Yildiz et al. (2017). This network is then trained with Bayesian Regulation Backpropagation using two weeks of training data (Set A), and it is used as a term of comparison for the proposed forecasting approach when the amount of data available for model training is limited. The same model is also trained using the full length dataset (Set B) in order to obtain the best forecasting performance, as commonly done in the literature. Here, as in Yildiz et al. (2017), the inputs are given by weather variables, time indexes (week of the year, day of the week, hour of the day), holiday/business-day binary indicator, previous day peak and minimum load, previous week peak and minimum load, and the autoregressive order is 3. The optimal network structure, resulting from a preliminary phase where the cross-validation performance of several different network structures were compared, is given by 2 hidden layers with respectively 4 and 6 neurons when using Set A, and by two hidden layers, having respectively 10 and 4 neurons, when using Set B, with hyperbolic tangent (sigmoid) activation function. The output layer consists of one neuron which is activated by a linear regression function.
For the training of the NARX models, the training dataset was randomly split into a training part (85%), and a validation part (15%), and for each testing week, every model is trained 50 times with a different initialization, and the one providing the best performance in cross-validation was selected for the forecasting of the testing week.

Here, we employ R^2 and Mean Absolute Percentage Error (MAPE) as performance metrics of the forecasting models, since they are commonly adopted in the related literature. The R^2 is a metric that measures the fitting performance of a model on the training data, and it is defined as

$$R^2 = 1 - \frac{\sum_{j=1}^{N_t} \left(\tilde{y}(j) - \hat{y}(j)\right)^2}{\sum_{j=1}^{N_t} \left(\tilde{y}(j) - \bar{y}\right)^2}, \text{ with } \bar{y} = \frac{1}{N_t} \sum_{j=1}^{N_v} y(j),$$

where $\tilde{y}$ represents the measured building energy consumption, $\hat{y}$ is its forecast, and N_t is the number of data samples used for training. The MAPE measures the forecasting performance on the testing data, and it is defined as:

$$\text{MAPE} = \frac{100}{N_v} \sum_{t=1}^{N_v} \left| \frac{\tilde{y}(t) - \hat{y}(t)}{\tilde{y}(t)} \right|$$

where N_v is the number of data samples used for testing.

Table 3. Fitting and forecasting performance for the training and testing weeks of data Set A.

Model	①	②	③	④	⑤	①	②	③	④	⑤
Week			R^2					MAPE		
1	0.92	0.54	0.93	0.93	**0.97**	8.86	19.47	**8.84**	8.60	26.73
2	0.95	0.61	0.95	0.96	**0.98**	12.91	30.81	**9.97**	11.71	18.29
3	0.84	0.58	0.85	0.86	**0.97**	**7.49**	10.35	7.62	7.77	12.52
4	0.85	0.48	0.86	0.87	**0.95**	**7.19**	10.96	7.48	7.38	10.66
5	0.84	0.44	0.83	0.84	**0.90**	**8.45**	13.09	9.53	9.13	24.08
6	0.91	0.58	0.92	0.93	**0.95**	6.87	12.34	**6.07**	6.31	19.09
7	0.80	0.46	0.81	0.83	**0.94**	8.03	11.03	**7.66**	7.97	14.01
8	0.78	0.46	0.78	0.77	**0.91**	7.94	15.48	**7.28**	10.35	14.23
9	0.89	0.58	0.88	0.89	**0.96**	6.91	10.51	**6.02**	6.20	21.28
Avg.	0.86	0.53	0.87	0.88	**0.95**	8.29	14.89	**7.83**	8.38	17.88

Table 4. Fitting and forecasting performance for the training and testing weeks of data Set B.

Model	①	②	③	④	⑤	①	②	③	④	⑤
Week			R^2					MAPE		
1	0.85	0.49	0.83	0.85	**0.98**	12.14	15.97	8.84	8.55	**7.61**
2	0.88	0.54	0.87	0.87	**0.98**	**8.81**	12.89	11.46	10.40	8.86
3	0.89	0.58	0.88	0.89	**0.98**	6.28	10.51	**6.02**	6.20	7.73
4	0.92	0.66	0.91	0.92	**0.98**	10.15	12.93	9.26	**9.18**	9.24
Avg.	0.88	0.57	0.88	0.88	**0.98**	9.35	13.08	8.89	8.59	**8.36**

Table 3 reports a comparison of the forecasting performance of models ①-⑤, all trained with a two-weeks-long dataset, for each of the nine testing weeks of Set A. It is possible to notice that the ARX model fed with the proposed fictitious input signal (model ①) has an average MAPE close to that of the two NARX network models fed with the same fictitious input (models ③ and ④), and it is able to achieve a comparable, and sometimes better, forecasting performance across the nine weeks, as can be seen from the MAPE values. It is important to observe that the predictors based on the fictitious input signal perform significantly better than the other models (② and ⑤), when the amount of data available for the training phase is limited. Fig. 4 depicts the forecasted load for the testing week 6 of data Set A obtained with model ①, with the associated bounds $\hat{\tau}_{k,i}$ computed solving (8) for every quarter-hour of all the days of the considered week, with $\alpha = 1.01$, $\gamma = 1.05$. Fig. 5 reports a comparison between the forecasting error of model ①, computed as absolute value of the difference between measured and forecasted load, and the corresponding accuracy bounds, detailed for a couple of days of testing week 6 of Set A.

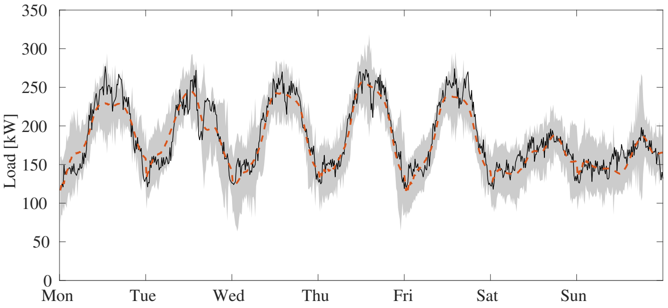

Fig. 4. Measured and forecasted load for testing week 6 of data Set A. Solid black line: measured load; red dashed line: model ① forecast; gray area: accuracy interval given by bounds $\hat{\tau}_{k,i}$.

Table 4 presents another performance comparison of the prediction models obtained using Set B, where model ⑤ uses the full two years long dataset for the training phase, while the other models only use the two weeks preceding every testing week for their training. Clearly, model ⑤ achieves the best forecasting performance, thanks to its ability to model nonlinear dynamics, and to the usage of a large amount of data, but the proposed ARX predictor (model ①) obtains a close enough prediction error, as measured by the MAPE, with a much simpler model structure, and using less data for the training. Finally, Fig. 6 shows the load measured during the testing week 3 of Set B, and the corresponding predictions obtained with models ①, ③ and ⑤.

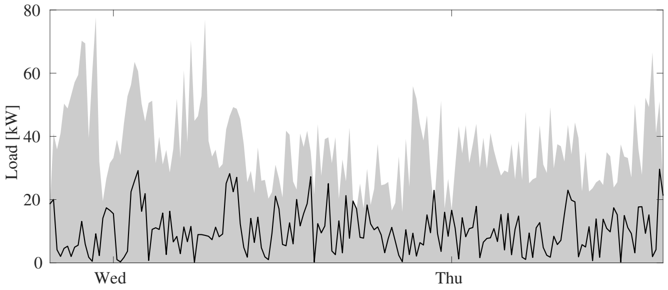

Fig. 5. Detail of forecasting error and accuracy bounds for testing week 6 of data Set A. Solid black line: forecasting error; gray area: accuracy interval bounds.

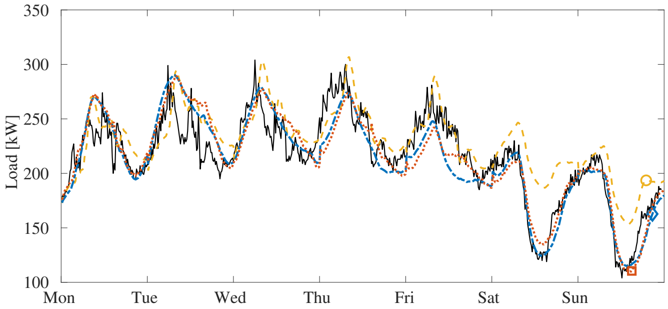

Fig. 6. Measured and forecasted load for testing week 3 of data Set B. Solid black line: measured load; red dotted line with □: model ①; blue dash-dotted line with ⋄: model ③; yellow dashed line with ∘: model ⑤.

5. CONCLUSION

We presented a new approach to forecast one-day-ahead electricity consumption of non-residential buildings, based on the usage of linear models, that is able to obtain high accuracy with a small set of data. To do so, we introduced a fictitious input signal able to capture the nonlinear and periodic trend of building load time series from prior information. Moreover, the use of a linear prediction model lets us derive guaranteed accuracy bounds on the forecasting error resorting to the Set Membership framework. Experimental results obtained from a dataset with measurements collected from an office building illustrate the validity of the proposed approach, showing that the use of the fictitious input signal significantly improves the forecasting accuracy of linear models. The presented ARX predictor achieves performance comparable to that of NARX neural networks, while retaining a simpler model structure and the capability to provide guaranteed accuracy bounds on its load forecast.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Dr. Enrico Ragaini and to ABB Italia SpA, for providing the dataset with weather and load measurements, and for the support during the development of the forecasting approach.

REFERENCES

Ahmad, A., Hassan, M., Abdullah, M., Rahman, H., Hussin, F., Abdullah, H., and Saidur, R. (2014). A review on applications of ANN and SVM for building electrical energy consumption forecasting. *Renewable and Sustainable Energy Reviews*, 33, 102–109.

Bourdeau, M., qiang Zhai, X., Nefzaoui, E., Guo, X., and Chatellier, P. (2019). Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*, 48, 101533.

Cai, M., Pipattanasomporn, M., and Rahman, S. (2019). Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Applied Energy*, 236, 1078–1088.

Chen, B.J., Chang, M.W., and Lin, C.J. (2004). Load forecasting using support vector machines: A study on EUNITE competition 2001. *IEEE Transactions on Power Systems*, 19(4), 1821–1830.

Deb, C., Zhang, F., Yang, J., Lee, S.E., and Shah, K.W. (2017). A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74, 902–924.

Dhar, A., Reddy, T.A., and Claridge, D.E. (1998). Modeling hourly energy use in commercial buildings with fourier series functional forms. *Journal of Solar Energy Engineering*, 120(3), 217–223.

EIA (2015). Electric power monthly: with data for January 2015. Technical report, US Energy Inf Adm.

Espinoza, M., Joye, C., Belmans, R., and Moor, B.D. (2005). Short-term load forecasting, profile identification, and customer segmentation: a methodology based on periodic time series. *IEEE Transactions on Power Systems*, 20(3), 1622–1630.

Fan, J.Y. and McDonald, J.D. (1994). A real-time implementation of short-term load forecasting for distribution power systems. *IEEE Transactions on Power Systems*, 9(2), 988–994.

Foresee, F.D. and Hagan, M. (1997). Gauss-Newton approximation to Bayesian learning. In *Proceedings of International Conference on Neural Networks (ICNN'97)*. IEEE.

González, P.A. and Zamarreño, J.M. (2005). Prediction of hourly energy consumption in buildings based on a feedback artificial neural network. *Energy and Buildings*, 37(6), 595–601.

Lauricella, M. and Fagiano, L. (2020). Set Membership identification of linear systems with guaranteed simulation accuracy. *IEEE Transactions on Automatic Control*.

Massana, J., Pous, C., Burgas, L., Melendez, J., and Colomer, J. (2015). Short-term load forecasting in a non-residential building contrasting models and attributes. *Energy and Buildings*, 92, 322–330.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer.

Pérez-Lombard, L., Ortiz, J., and Pout, C. (2008). A review on buildings energy consumption information. *Energy and Buildings*, 40(3), 394–398.

Rahman, A., Srikumar, V., and Smith, A.D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy*, 212, 372–385.

Söderström, T. and Stoica, P. (1989). *System identification*. Prentice-Hall.

Taylor, J.W. and McSharry, P.E. (2007). Short-term load forecasting methods: An evaluation based on european data. *IEEE Transactions on Power Systems*, 22(4), 2213–2219.

Yildiz, B., Bilbao, J., and Sproul, A. (2017). A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews*, 73, 1104–1122.

Yun, K., Luck, R., Mago, P.J., and Cho, H. (2012). Building hourly thermal load prediction using an indexed ARX model. *Energy and Buildings*, 54, 225–233.