

RESEARCH

Open Access



Adequacy of neural networks for wide-scale day-ahead load forecasts on buildings and distribution systems using smart meter data

Oleg Valgaev^{1*} , Friederich Kupzog¹ and Hartmut Schmeck²

*Correspondence:

oleg.valgaev@ait.ac.at

¹ Austrian Institute of Technology, Vienna, Austria

Full list of author information is available at the end of the article

Abstract

Power system operation increasingly relies on numerous day-ahead forecasts of local, disaggregated loads such as single buildings, microgrids and small distribution system areas. Various data-driven models can be effective predicting specific time series one-step-ahead. The aim of this work is to investigate the adequacy of neural network methodology for predicting the entire load curve day-ahead and evaluate its performance for a wide-scale application on local loads. To do so, we adopt networks from other short-term load forecasting problems for the multi-step prediction. We evaluate various feed-forward and recurrent neural network architectures drawing statistically relevant conclusions on a large sample of residential buildings. Our results suggest that neural network methodology might be ill-chosen when we predict numerous loads of different characteristics while manual setup is not possible. This article urges to consider other techniques that aim to substitute standardized load profiles using wide-scale smart meters data.

Keywords: Smart grid, Machine learning, Load forecasting, Day-ahead, Neural network, Local load, Buildings, Microgrids, Distribution system

Introduction

Following the ongoing transformation of the European power system, it will be necessary to locally balance the increasing share of decentralized renewable energy supply. Distribution system operation will require a versatile and reliable model to obtain numerous *day-ahead load forecasts (DALF)* on various levels of load aggregation. Wide-scale installation of smart meters allows to apply specialized data-driven techniques for local day-ahead forecasting instead of currently used *standardized load profiles (SLP)*. Among various machine learning approaches to load forecasting discussed in the literature, *artificial neural networks (ANN)* are the most intensively investigated methodology (Ahmad et al. 2018; Bourdeau et al. 2019).

There exist multiple models predicting a specific load time series one-step-ahead (e.g. hourly consumption of a university campus) (Ahmad et al. 2018). Nevertheless, only few proposals were made to use neural networks for day-ahead (multi-step) forecasts which will become more important for the distribution system operation. In particular, ANNs have been applied for predicting day-ahead loads of larger buildings (Bagnasco et al. 2015; Chitsaz et al. 2015a; Ryu et al. 2016), consumer aggregations (Marinescu et al. 2013) and microgrids (Amjady et al. 2010; Hernández et al. 2014). Further, there have been attempts to forecast a single household load using novel *deep learning techniques* (Amarasinghe et al. 2017; Mocanu et al. 2016).

Proposed models allow only limited conclusions about applying ANN for DALF on a *wide-scale*¹. The well-performing setup required detailed knowledge of the load and was often found and manually tuned through trial and error. Further, the networks require large amounts of historic data and are subject to overfitting (Amarasinghe et al. 2017; Chalal et al. 2016). The accuracy substantially varied depending on the load. While a proposed model was accurate for a given building, similar setup performed notably worse when applied on a different consumer (Fouquier et al. 2013).

When deciding on a general methodology to substitute SLPs for wide-scale distribution system operation, statistical analysis is required to avoid case-based reasoning. In this work, we provide a comprehensive evaluation of the ANN methodology for local day-ahead load forecasting. While many existing models forecast the time-series one-step-ahead, we discuss various strategies to adopt a network for predicting the entire daily load curve. From the literature, we select and evaluate four different ANN architectures that appear suitable for a wide-scale application. We provide evidence that the prominent ANN methodology, as currently implemented, might be ill-suited for predicting local consumption on a wide-scale, despite being accurate for particular loads.

Background

In this section, we provide theoretical background for our study. We start by describing the loads we will be predicting and continue with an overview of the ANN methodology as well as its state of the art applications for local load forecasting. The provided background allows unprepared readers to comprehend the selection and setup of ANN models that we describe in the subsequent section.

Local loads

Local loads range from single family homes to a small distribution system area supplied by an MV/LV substation including microgrids (Dang-Ha et al. 2017). Naturally, their consumption is very diverse and more volatile than the transmission system load.

A consumption pattern mainly depends on the aggregation size. At a small level, it highly depends on the consumer behavior which makes it harder to predict. Generally, load is autoregressive and underlies annual and weekly seasonalities but to different extent depending on the size (Sevlian and Rajagopal 2014) (Fig. 1).

Artificial neural networks

Neural network methodology is a set of methods used in many machine learning applications. It can model almost any nonlinear relation between multiple inputs and outputs.

¹Wide-scale application implies predicting separately numerous loads of different size and characteristics ranging from single buildings to distribution system areas.

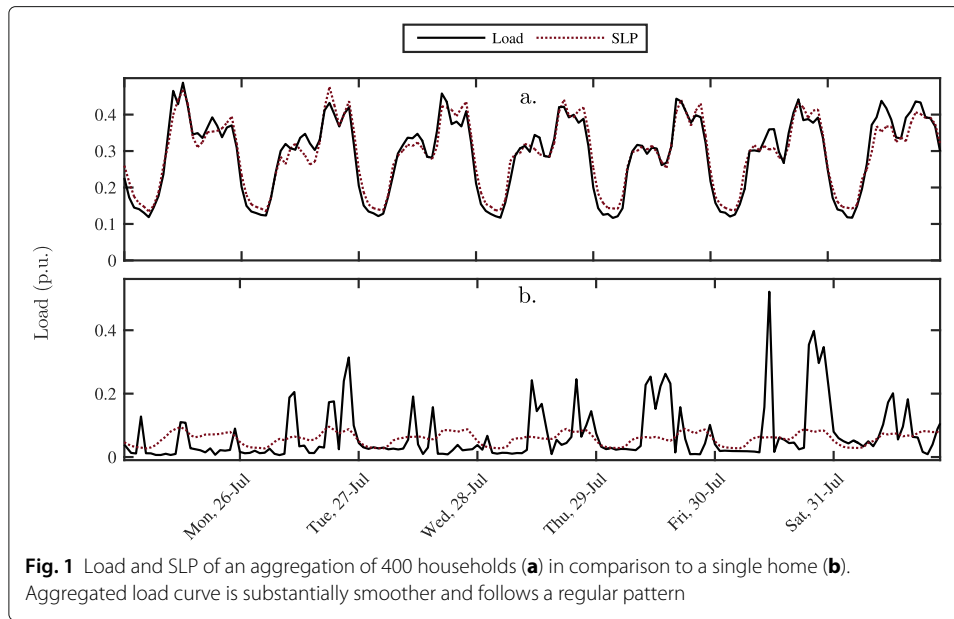


Fig. 1 Load and SLP of an aggregation of 400 households **(a)** in comparison to a single home **(b)**. Aggregated load curve is substantially smoother and follows a regular pattern

Assuming existence, the relation is self-learned from historic data by a specialized training algorithm that solves the *regression equation* (Friedman et al. 2008)

$$y = \mathbf{r}(X) + \epsilon \quad (1)$$

where *regression function* \mathbf{r} describes the systematic information about scalar variable y , and *residual error* ϵ represents all the uncaptured influences independent of vector input X .

A network can be seen as an interconnection of single elements called *artificial neurons* with an individual output

$$y_{\text{neuron}} = \phi(u), \text{ where } u = \sum_{i=1}^{n_u} w_i v^{(i)}. \quad (2)$$

The n_u neuron inputs $v^{(i)} \in \mathbb{R}$ weighted with $w_i \in \mathbb{R}$ are processed by an *activation function* ϕ . If several of such neurons are interconnected into a network, theoretically, any non-linear relationship can be approximated. For this, neurons are organized in layers and the data traverses the network from input to output passing through some neurons in the hidden layers one or several times.

We formalize an ANN as follows. Let \mathcal{N} be a neural network with n_x inputs, n_y outputs, and n_w interconnections that exists between the neurons. It is fully determined by its architecture, and the set of weights for each neuron. The network provides a regression function estimate

$$Y = \mathbf{r}_{\mathcal{N}}(X, W) \quad (3)$$

which maps a multivariate input $X \in \mathbb{R}^{n_x}$ to the output $Y \in \mathbb{R}^{n_y}$ for a given set of weights $W = [w_1, \dots, w_{n_w}] \in \mathbb{R}^{n_w}$ which are determined during the model training. To forecast with an ANN model we proceed as follows:

Architectures

We select a network architecture, relevant inputs and set *hyper-parameters* that can not be learned directly from the data and are determined prior to training the network. A *network architecture* includes inputs, network type, number of neurons in the layers and their topology. While an extensive overview of different architectures contrasting the differences can be found in Raza and Khosravi (2015), there are two general types.

Feed-forward neural network (FFNN) is the type, where starting at the input, the data traverses the network without any cycles or loops. The most prominent architecture of this type is the *multi-layer perceptron (MLP)*. It includes at least one hidden layer where each hidden neuron has a nonlinear activation function.

Recurrent neural networks (RNN) have feedback loops that allow data to traverse the network both ways. The connections between the neurons form a directed graph, and unlike a FFNN, a recurrent network has a dynamic internal state.

Deep neural networks (DNN) are those that have at least two hidden layers allowing complex topology both of FFNN and RNN types. It was shown, that increasing network size and complexity can often improve the accuracy, but elevates the time and historic data needed to train the network (Goodfellow et al. 2016).

Each network has a set of hyper-parameters (number of neurons, activation function, training algorithm etc.) that significantly affect the prediction accuracy. The hyper-parameters are often manually selected and iteratively fine-tuned given an in-depth knowledge of the forecasting problem.

Network training

In the training phase, \mathbf{r} needs to be estimated finding W that defines the network \mathcal{N} so that $\mathbf{r}_{\mathcal{N}}$ yields the lowest error on the given training data and is expected to *generalize* well - i.e., produce the lowest error on the unseen data. The weights are initialized randomly and calculated with a *supervised training algorithm* using either a *back-propagation (BP)* of training error (Rumelhart et al. 1986) combined with different variants of gradient descent optimization (Kiefer and Wolfowitz 1952; Kingma and Ba 2017; Riedmiller and Braun 1993) or *Levenberg Marquardt (LM)* algorithm (Hagan and Menhaj 1994; Moré 1978).

Evaluation

At last, the forecast \hat{Y} is obtained with a previously trained network for a given input X^* as $\hat{Y} = \mathbf{r}_{\mathcal{N}}(X^*)$. The model is evaluated on test data using an appropriate error definition (Haben et al. 2014).

Setting hyper-parameters and training often requires a vast amount of historic data, and it is hard to interpret the weights of the resulting network. Nonetheless, for some applications where a considerable amount of data is available, computation time is not an issue and there is an extensive problem knowledge allowing manual fine-tuning, ANN models can be very accurate. Inspired by the advances in machine learning, there are several applications for local load forecasting that we describe next.

Neural network applications

Numerous researchers applied the methodology described above to predict the load in microgrids (Hernández et al. 2014), distribution system areas (Wang et al. 2018) and

buildings with a recent review citing over 90 publications (Runge and Zmeureanu 2019). The model performance depends fundamentally on the characteristics of time series. Thus, we focus on applications for short-term forecasting of local loads with (sub-)hourly resolution. This substantially reduces the number of related works to the ones summarized in Table 1.

The models have several aspects in common. They were developed for a specific load - a given building or an area of a distribution grid. The well performing architecture is setup manually, given explicit knowledge of the problem, researcher experience and intuition combined with trial and error. Model inputs are related to historical load, calendar features and, sometimes, daily weather. Further, they require large amounts of historic data. For instance, DNN model *convolutional neural network (CNN)* (Amarasinghe et al. 2017), *restricted Boltzmann machine (RBM)* (Mocanu et al. 2016; Ryu et al. 2016), *long short-term memory (LSTM)* (Marino et al. 2016; Kong et al. 2017) and *echo-state network (ESN)* (Shi et al. 2016) - all required years of data to setup and train the network.

Despite a common preconception, researchers are ambiguous about using weather related inputs such as *outside ambient temperature (OAT)* or solar irradiation. Some, do consider weather modeling electrical heating and PV at the level of MV/LV feeder (Hayes and Prodanovic 2016) or larger buildings (Pirjan et al. 2017). An in-depth sensitivity analysis can highlight an existing weather dependency (Llanos et al. 2012). Others, observe the models that do not use any weather data performing better for disaggregated loads (Hayes et al. 2015; Marinescu et al. 2013; Hernández et al. 2014). In Bagnasco et al. (2015), authors test two networks on the same dataset with and without such data. They observe no consistent advantage for either model. Alternatively, some researchers assume that the instantaneous weather changes do not affect the load significantly and consider only the month and the day arguing that the temperature does not change substantially from day to day (Hernández et al. 2014).

Hyper-parameters have a major impact on the performance (Mena et al. 2014; Pirjan et al. 2017). From the publications in Table 1 and more general review studies on ANN applications for load forecasting (Humeau et al. 2013; Runge and Zmeureanu 2019), we see that the models are usually setup and fine-tuned manually. Nevertheless, there have been attempts for automated trial and error (Amjady et al. 2010; Hernández et al. 2014).

Yet, despite the increasing interest, the development of fully automated models based on ANN is in a preliminary stage (Hutter et al. 2019). As of today, the load forecasting models rely on explicit a priori knowledge of the consumer and manual fine-tuning (Table 1). The well performing network architecture presented in the results is often found through a trial and error process and requires large amounts of historic data and computational resources

Methods

In this section, we formulate the day-ahead forecasting problem and describe the model architectures, their setup and simulation. We will present the simulation results afterwards.

Day-ahead forecasting requires to predict the entire curve for the next day. The exact time at which we need the predicted curve depends on the particular application. We assume that the forecast has to be done shortly before midnight for the upcoming day, as it is done in other studies (Table 1). We formulate the DALF problem as follows.

Table 1 Neural networks in the literature predicting (sub-)hourly local loads. Description is provided in “Neural network applications” section

Reference	Network	Load	Dataset (months)	Horizon	Inputs	Setup	Hidden layers	Hidden neurons	Training algorithm
Bagnasco et al. (2015)	MLP	hospital	12	DALF	load, calendar, weather	manual	1	20	BP
Marinescu et al. (2013)	MLP	aggregation	17	DALF	load, weather	manual	1	15	BP
Hernández et al. (2014)	MLP	microgrid	54	DALF	load, calendar, weather	auto	1	21	var
Amjady et al. (2010)	MLP	microgrid	12	DALF	load, calendar, weather	auto	1	n/s	LM
Llanos et al. (2012)	MLP	microgrid	5	48 h	load	manual	1	8	LM
Chitsaz et al. (2015b)	RNN	edu. building	12	DALF	load, calendar, weather	manual	1	n/s	LM
Mena et al. (2014)	RNN	lab. building	18	1 h	load, calendar, weather	manual	1	10	LM
Prijan et al. (2017)	RNN	hypermarket	12	1 h	load, calendar, weather	manual	1	var	var
Hayes and Prodanovic (2016)	RNN	46 MV/LV	24	24 h	load, calendar, weather	manual	1	10	LM
Mocanu et al. (2016)	RBM	home	48	DALF	load	manual	1	10	LM
Marino et al. (2016)	LSTM	home	48	60 h	load, calendar	manual	2	10	BP
Amarasinghe et al. (2017)	CNN	home	48	DALF	load, calendar	manual	2	20	BP
Shi et al. (2016)	ESN	office building	48	1 h	load	auto	1	50	n/s
Kong et al. (2017)	LSTM	69 homes	48	6 h	load, calendar	manual	2	20	BP
Ryu et al. (2016)	RBM	40 enterprises	36	DALF	load, calendar, weather	manual	4	150	LM

Given a *training set* $\mathcal{T} := \{(X_1, y_1), \dots, (X_m, y_m)\}$ that consists of m observations of multivariate input $X = \{X_1, X_2, \dots\}$ and univariate output $y = \{y_1, y_2, \dots\}$ time series, predict the vector

$$Y_{d+1} = [y_1, \dots, y_n] \quad (4)$$

of n consecutive values representing the next-day load curve. With an input X_{d+1} , the prediction $\hat{Y}_{d+1}(X_{d+1})$ has to minimize the *root mean squared error (RMSE)* defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \epsilon_i^2}. \quad (5)$$

where $\epsilon_i = y_i - \hat{y}_i$ is the *residual error* between actual y_i and predicted value \hat{y}_i according to (1).

To find \hat{Y}_{d+1} we use different model architectures described next in “[Model architecture](#)” section. Each model is setup as denoted in “[Model setup](#)” and evaluated through the simulation described in “[Simulation](#)” sections.

Model architecture

We use network architectures that are the most common among the load forecasting applications (Table 1). Studying the literature, we have not seen any fundamental reason why any of the networks is superior for certain types of loads, given appropriate setup and vigor at manual fine-tuning (Bourdeau et al. 2019). However, for wide-scale usage, practicality becomes important. Hence, we do not consider DNN and other models that rely on abundant historic data or information from specific sensorial equipment (e.g. occupancy). We adopt two feed-forward and two recurrent networks for day-ahead prediction as described further in the text.

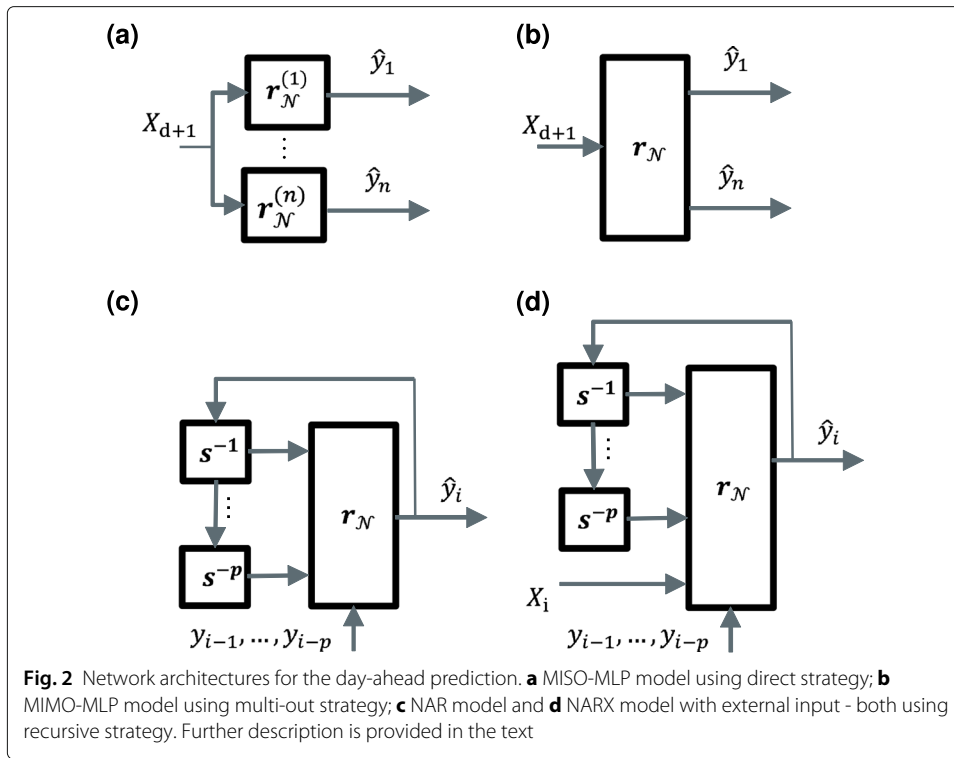
Day-ahead prediction

In time series forecasting, we often encounter situations where only *one-step-ahead prediction* is required or considered. For electrical load, such task eventually corresponds to an *intraday forecast*. Following the scope of this work, we focus on *day-ahead forecasts*, which are *multi-step predictions* and require forecasting n consecutive points. While n depends on the time series resolution, there are three general strategies (Ben Taieb et al. 2012) to adopt a one-step-ahead forecasting model for a multi-step prediction (Fig. 2).

Direct strategy is a straight forward approach that requires to setup and train n separate *multi-input single-output (MISO)* models with n_x inputs. However, such strategy disregards the dependencies between the predicted points of the curve. Moreover, it is computationally expensive, since we have to train n networks.

Multi-out strategy sets up and trains one complex *multiple-input multiple-output (MIMO)* network with n_x inputs and n outputs corresponding to the points of the forecast curve. It allows to consider the dependencies between the predicted time-steps and avoids the conditional independence assumption made by the direct strategy.

Recursive strategy forecasts one-step-ahead and uses the prediction \hat{y}_1 as a new observation with which forecast is reiterated for $i = 2$ and so on. The fundamental drawback is the sensitivity to prediction error, since it is accumulated while advancing the multi-step forecast. Applying a recursive strategy will only be accurate if we have a good representation of the underlying time series generating process.



Feed-forward models

A feed forward network can forecast day-ahead load curve adopting either direct or multi-output strategy. Given input

$$X = \{x^{(1)}, \dots, x^{(n_x)}\} \text{ and } X_{d+1} = [x_1, \dots, x_{n_x}], \quad (6)$$

we use the following multivariate forecasting models:

- 1 Multiple input single output multi-layer perceptron (MISO-MLP)
- 2 Multiple input multiple output multi-layer perceptron (MIMO-MLP)

In the first case, a separate MLP is trained to predict each point of Y_{d+1} (Fig. 2a). For the ease of exposition, we setup each network with same inputs and hyper-parameters described further in the text. The training output data is split into n separate series $y^{(1)}, \dots, y^{(n)}$, so that the n networks will have different weights after the training. The forecast output by the model consists of n separate predictions in

$$\hat{Y}_{d+1} = [\mathbf{r}_{\mathcal{N}}^{(1)}(X_{d+1}), \dots, \mathbf{r}_{\mathcal{N}}^{(n)}(X_{d+1})]. \quad (7)$$

In the second case, we train one multi-output feed-forward network with n_x inputs and n outputs. For a given X_{d+1} , the multivariate forecast is obtained as

$$\hat{Y}_{d+1} = \mathbf{r}_{\mathcal{N}}(X_{d+1}). \quad (8)$$

with one MLP $\mathbf{r}_{\mathcal{N}}$ to train (Fig. 2b).

Recurrent models

The concept of an RNN where past output values are fed back allows to create non-linear autoregressive time series models. With it, we apply a recursive strategy to the multi-step forecasting introducing the following models:

- 1 *Nonlinear autoregressive model (NAR)*
- 2 *Nonlinear autoregressive model with exogenous inputs (NARX)*

In the first case, we create a univariate autoregressive model with p lags defined as

$$\hat{y}_i = \mathbf{r}_N(y_{i-1}, \dots, y_{i-p}) \quad (9)$$

where prediction \hat{y}_i is a function of the p preceding values y_{i-1}, \dots, y_{i-p} of the time series y (Fig. 2c).

In the second case, NARX model presents an extension of the NAR model that can consider external inputs to create a multivariate time series model

$$\hat{y}_i = \mathbf{r}_N(y_{i-1}, \dots, y_{i-p}, X_i). \quad (10)$$

Here, a prediction \hat{y}_i is calculated as a function of its p lags y_{i-1}, \dots, y_{i-p} and an exogenous input X_i (Fig. 2d).

Model setup

In Table 2, we summarize the networks used in this study with corresponding number of inputs, outputs, training data and degrees of freedom (total number of weights). As in related works, we assume hourly time series resolution ($n = 24$). We now proceed explaining the setup and justifying the choice of hyper-parameters.

Inputs

We consider the dependency on past values, OAT and weekly seasonality. For the feed-forward models, we input the load curve of the previous day while NARX uses 24 lags ($p = 24$) recursively. Further, multivariate networks (MISO-MLP, MIMO-MLP, NARX) model the seasonality and annual temperature cycle explicitly. The weekly seasonality is considered with a *weekday number* and a (boolean) *public holiday* variable. The annual temperature cycle is modeled as a function of *month* and *day* number. In this study, we assume that short-term weather changes do not notably impact the load (Hernández et al. 2014). As we will see, the simulation results validate this assumption.

The univariate model (NAR) considers weekly patterns implicitly by using one week of lags ($p = 168$). The annual cycle is accounted for by using only two most recent months of data for training that is repeated monthly.

Table 2 Network setup overview

	MISO-MLP	MIMO-MLP	NAR	NARX
Type	FFNN	FFNN	RNN	RNN
Multi-step	direct	multi-output	recursive	recursive
Inputs	28	28	168	28
Outputs	1	24	1	1
Hidden layers	1	1	1	1
Hidden neurons	15	15	15	15
Total weights	79	424	2703	463
Training data	365	365	1440	8760

Network training is more efficient with normalized inputs (Dan Foresee and Hagan 1997). Following a standard practice, we apply *minmax-normalization* as follows. Each separate input $x^{(j)}$, with $j \in [1, \dots, n_x]$ is transformed to

$$\tilde{x}^{(j)} = \frac{2 \left(x^{(j)} - x_{\min}^{(j)} \right)}{x_{\max}^{(j)} - x_{\min}^{(j)}} - 1, \quad (11)$$

where the smallest input value $x_{\min}^{(j)}$ corresponds to $\tilde{x}^{(j)} = -1$ and the largest input $x_{\max}^{(j)}$ corresponds to $\tilde{x}^{(j)} = 1$. Herewith, every normalized training input lies in the range of $[-1; 1]$. We use the same normalization constants $x_{\min}^{(j)}, x_{\max}^{(j)}$ for the evaluation.

Outputs

Subsequently, network output must be transformed back into the range of original data. We provide the network with a pre-processing block that appears before the input layer and a post-processing block that appears after the output layer (Fig. 3)

Additionally, for recursive models (NAR, NARX) post-processing includes a *hold and release* unit. In their case, the output \hat{y} is a scalar prediction for a single time step. For those models, post-processing allows to output the entire forecast curve at once.

Training data

For wide-scale prediction of local loads it is important to rely on the smallest viable amount of training data. In addition to the availability issue, time series can constantly alter their regime with old data becoming irrelevant (inhabitants change, new equipment is installed). Multivariate networks (MIMO-MLP, MISO-MLP, NARX) require at least one year of data to learn the dependency on the month. Currently used SLPs also require total consumption over a year to scale the profile. In contrast, the NAR model is trained only on the two months of preceding data.

Hyper-parameters

Activation function of the neurons must correspond to the applied normalization. With $x^{(j)} \in [-1; 1]$, each layer should have $\phi(u)$ with the same domain and range. Unfortunately, the majority of related works (Table 1) do not specify the activation function (Amjady et al. 2010; Hernández et al. 2014; Hayes and Prodanovic 2016; Marinescu et al. 2013; Mocanu et al. 2016; Pirjan et al. 2017; Marino et al. 2016). Among the rest, the most common functions are *linear* (Amarasinghe et al. 2017; Llanos et al. 2012; Mena et al. 2014; Ryu et al. 2016) and *tanh-sigmoid* (Bagnasco et al. 2015; Shi et al. 2016; Kong et al. 2017) defined as

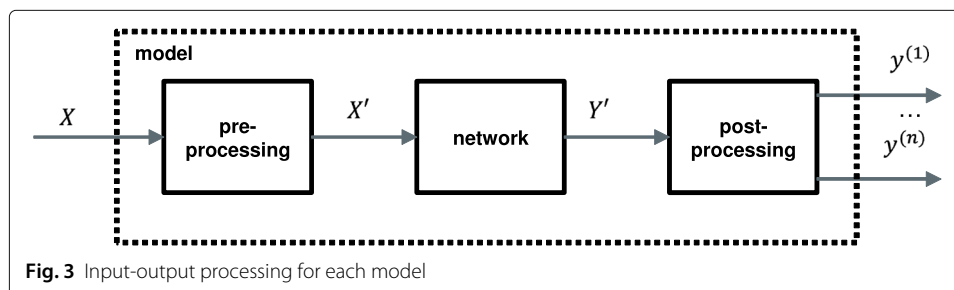


Fig. 3 Input-output processing for each model

$$\phi(u) = \frac{2}{1 + \exp(-2u)} - 1, \quad (12)$$

and which we use in this study since $\phi(u) \in [-1; 1], \forall u \in \mathbb{R}$.

Network size determines the predictive capacity of the model. For this work, we use networks with one hidden layer consisting of 15 neurons. The optimal number of hidden layers and neurons depends on the task and there is no theoretical methodology to determine it. Often, the size is chosen using experience in comparable problems. In related works, best performing networks also had one hidden layer with a similar number of neurons (Table 1).

Training algorithm for each network combines LM method with Bayesian regularization (Dan Foresee and Hagan 1997). The LM algorithm appears to be much faster than BP-based approaches for moderately-sized networks (Hagan and Menhaj 1994). Bayesian regularization does not keep some of the limited training data as a validation set in contrast to the commonly used early stopping technique (Friedman et al. 2008). At the same time, it is effective preventing overfitting and improving generalization of moderately over-sized networks (Dias et al. 2003).

Simulation

We conduct the following simulations predicting the load day by day and calculating RMSE for each predicted time series. The loads are taken from the publicly available smart meter data of *Irish Commission for Energy Regulation* (Commission for Energy Regulation (CER) 2012). Given a sample of over 900 homes² at the same location, we select the loads with no missing data and annual consumption within the interquartile range (IQR). The resulting data-set consists of 444 single buildings with 17 consecutive months of data. Each time series was re-sampled equidistantly with a 60 minute resolution and normalized by its maximum value to allow scale-free comparisons between the loads. We conduct the simulations using MATLAB-software. The hyper-parameters of the networks not mentioned explicitly in “[Model setup](#)” section were left to MATLAB defaults.

Simulation A We create aggregated loads of different size for which we predict the demand curves over one month (August). For each load, we train a network on preceding data 100 times to investigate the effect that random weights initialization has on the training solution. This gives us a sample of 100 forecast errors for each load and model.

Simulation B We predict the loads of single homes separately during another five months period (September – December). We retrain the networks every month using the preceding data for training on a rolling basis.

Results

We have observed substantial stochastic dispersion of the forecast error referable to random weight initialization (“[Forecasting loads of different size](#)” section) and variation among the households (“[Forecasting large sample of households](#)” section). Error distri-

²We consider only homes of the control group. Other homes in this dataset participated in a trial that affected their consumption.

Table 3 Mean RMSE of 100 networks predicting loads of different size (number of homes in the aggregation). Confidence intervals are denoted in Fig. 4

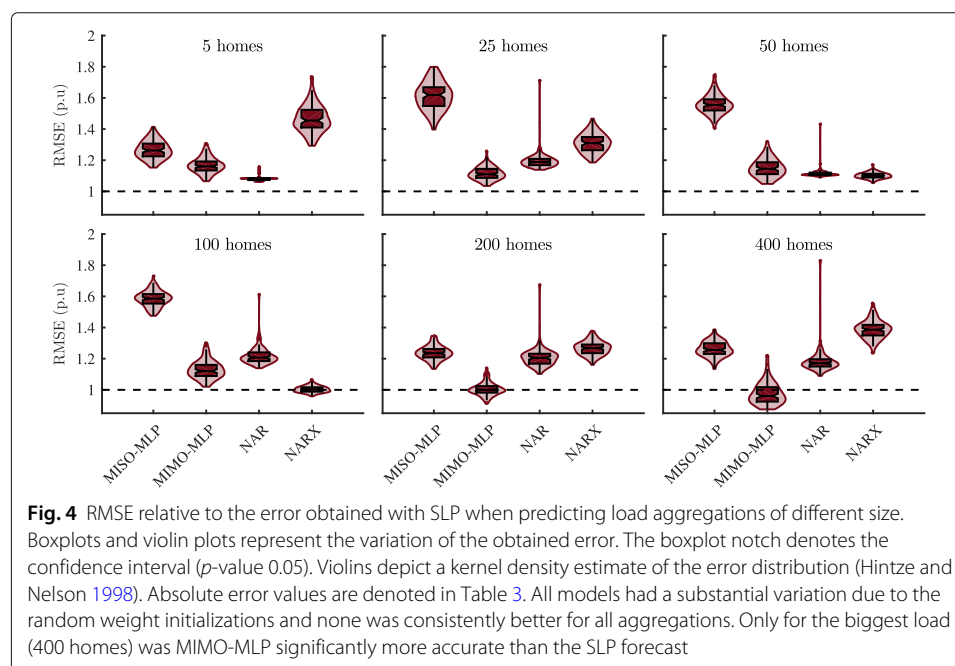
Load size (homes)	RMSE (p.u.)				
	SLP	MISO-MLP	MIMO-MLP	NAR	NARX
1	0,086	0,104	0,095	0,097	0,188
5	0,077	0,098	0,09	0,084	0,114
25	0,044	0,071	0,049	0,053	0,058
50	0,039	0,06	0,044	0,043	0,043
100	0,032	0,051	0,036	0,039	0,032
200	0,029	0,035	0,029	0,035	0,036
400	0,023	0,03	0,023	0,028	0,033

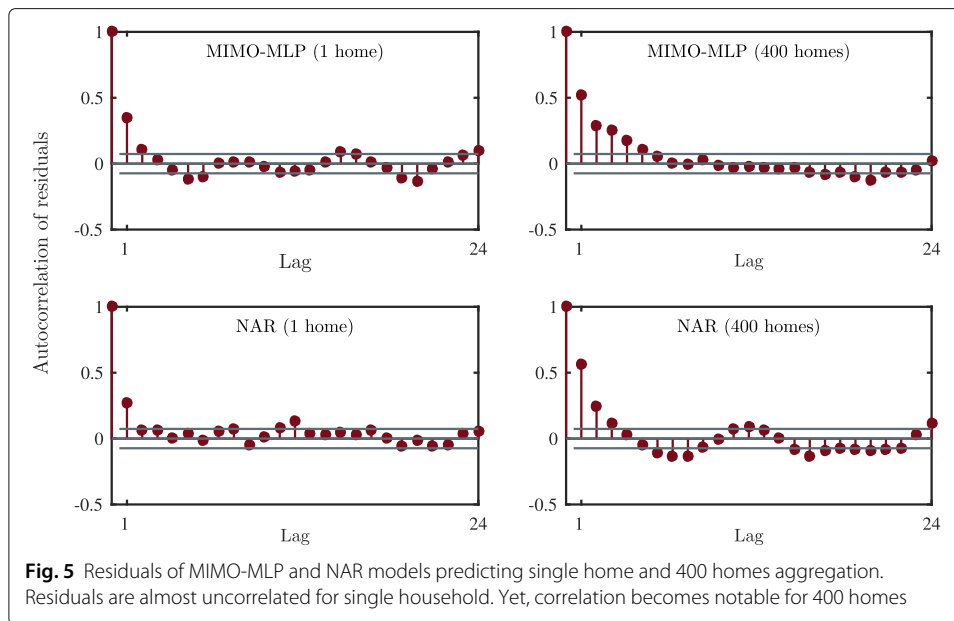
bution was represented with *violin-plots* while *box-plots* helped us to evaluate the range and statistical significance (p -value 0.05). Irish residential SLPs were used as a reference forecast (Irish standard load profiles 2014). Stochastic variation of errors had notable implications described in this section and discussed in the end of the article.

Forecasting loads of different size

We observed that mean RMSE changed drastically with load size and had notable dispersion. Average forecast of the biggest aggregation was three to four times as accurate as for a single home (Table 3). Further, forecast error had variation between 9% and 62% in terms of relative range (Fig. 4).

The ANN models were setup adequately. Residuals were mostly uncorrelated and according to Eq. 1 contained no systematic information (Fig. 5 left column). Uncaptured influence of a further variable would have been manifested in a substantial correlation which only started to appear for bigger loads (Fig. 5 right column). We attribute it to weather that began to have a notable effect on the overall consumption.





On the other side, the training algorithm failed to provide stable solutions. The dispersion could not be explained by the lack of historic data and was observed for each network. The most consistent training was achieved for the NAR model that is underdetermined with twice as many degrees of freedom as data (approx. 2:1). Yet, well-determined NARX (approx. 1:19) had wider IQR. In some cases, outliers indicating eventual overfitting were observed.

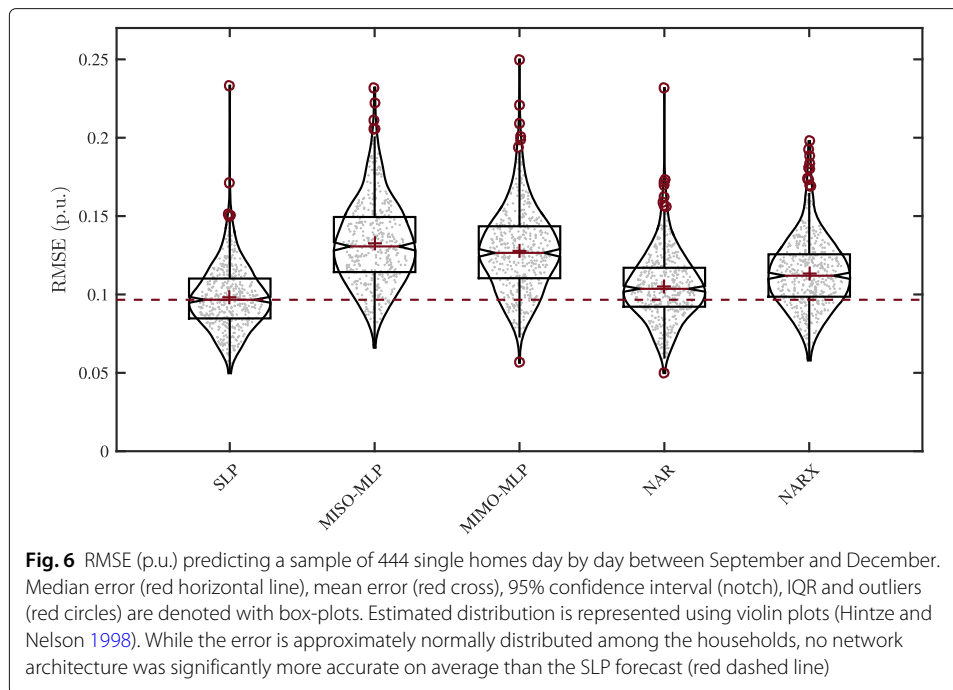
Only for the biggest load was an ANN forecast significantly more accurate than the reference (Fig. 4). Further, we observed the direct approach being the worst among multi-step strategies. Apart from this, we could not make any consistent comparisons between the architectures. While RNNs were accurate for smaller loads, MIMO-MLP delivered smaller error for bigger loads.

Due to the error dispersion, any meaningful conclusion can only be made using statistics on a relevantly large sample of trained networks. Our results indicated that the applied networks are only accurate for bigger aggregations (400 homes onwards) while for smaller loads they failed to infer the inherent patterns.

Forecasting large sample of households

We observed that the forecast error varies substantially depending on the given household (Fig. 6). The relative range spread between 162% (MISO-MLP) and 127% (NARX model) among the sample notwithstanding the homes had similar annual consumption and location.

On average, none of the networks reached the reference accuracy (SLP). As in the previous simulation, direct forecast (MISO-MLP) was observed to be least accurate among multi-step strategies while recursive prediction with RNN was the most successful (NARX, NAR). Nevertheless, while for some households NAR substantially improved the accuracy by up to 25% against the reference, such result was only valid for the specific load. Still, it had mean error that was 10% above the reference applied on the same sample of households.



Discussion

Our results suggest that ANN methodology might be inapt for wide-scale local load forecasting. Each of the four architectures produced dispersed forecast errors linked to the variation among training solutions and simulated buildings. As expected, mean error rapidly decreased for bigger loads. At the same time, no architecture was consistently better for each size. Even more surprisingly, no network was significantly more accurate for smaller loads than the forecast with an SLP (benchmark).

It is known that network performance depends on the characteristics of the time series linked to the load size and that ANNs do deliver accurate forecasts for some regular load curves (Wang et al. 2018). These models are often setup manually for one-step-ahead predictions using extensive training data (Runge and Zmeureanu 2019). To the best of our knowledge, ANNs were never applied in context of wide-scale day-ahead predictions on a statistically relevant sample of local loads where historic data is limited, manual adjustment is not possible and the loads can be highly volatile.

We have observed that no network architecture evaluated in this work was decisively superior for all loads (Fig. 4). Moreover, only for the biggest aggregation, there was a network significantly better than the benchmark. While this contradicts many specific cases where similar ANN models were shown to perform well (Table 1), a comparable conclusion has been reported earlier for small loads where even a naive approach outperformed an MLP (Hayes et al. 2015).

The accuracy of an ANN-model for wide range of loads can be improved by selecting the best architecture among a set of candidates for each given consumer. Such idea of an *ensemble model* is currently discussed (Ahmad et al. 2018; Bourdeau et al. 2019). However, we have observed notable dispersion of the forecast error obtained by the same network due to the random weight initialization which is a fundamental part of network training. This dispersion makes model selection based on trial and error (Hernández et al. 2014;

Amjady et al. 2010) and comparison in general, more difficult requiring to consider a statistically relevant sample of training solutions for each candidate.

Our results suggest further practical conclusions for using ANN methodology to forecast local loads day-ahead. Firstly, one-step-ahead models, which are more common, should not be adopted directly. Predicting each point of the load curve separately was significantly worse than any other strategy described in this work (Fig. 2).

Secondly, a more complex network architecture with additional input and training data (e.g., DNN) is unlikely to be substantially more accurate. The residual analysis shows that our simple setups had sufficient modeling capacity (Fig. 5). It also shows that, for smaller loads, no further inputs are required and it is enough to consider only annual temperature cycle instead of daily weather changes. Notwithstanding some special cases (e.g. substantial PV share), this counterintuitive observation is consistent with other studies (Hayes et al. 2015; Marinescu et al. 2013; Hernández et al. 2014) and contrasts to the transmission system level where weather explains up to 70% of load variation (Dang-Ha et al. 2017).

Most importantly, stochastic nature of the results becomes apparent when predicting numerous local loads. The variation among the loads and training solutions urges to consider statistical significance of any result. We have demonstrated substantial relative range of forecast error which undermines any conclusion based uniquely on mean error and such like. Related studies rarely consider confidence intervals of errors when evaluating an ANN model applied for local load forecasting (Table 1). Our statistical analysis shows that, even when aptly setup, ANNs may fail to reach the accuracy of currently used SLPs (Fig. 6).

We explain the weak performance of applied architectures by non-stationarity of local loads. Following the ANN methodology, a trained network forecasts unseen data, assuming that the statistical properties of the process generating the data remain constant. Once the regression function is estimated, a network does not adapt to the change in data characteristics which often arises with local loads. In such case, historic data quickly becomes irrelevant undermining the network training. The fact that the most accurate model used the least amount of training data supports this hypothesis (NAR model using only 2 months of data).

Stationarity assumption is central for any neural network architecture. Hence, it is unclear how ANN methodology, in general, can become effective predicting local loads on a wide-scale. Neural networks are known for their black-box character and it is hard to formulate a theory about their limitations for the given problem. In this situation, empirical evidence obtained through statistical analysis, as in this work, becomes important when deciding on a general approach for replacing SLPs. As currently applied, ANN methodology might be ill-chosen for wide-scale forecasting of local loads. Future research should focus on *adaptive models* (Ditzler et al. 2015; Kuznetsov and Mohri 2015) which may require an entirely different approach, despite ANNs being successful for other machine learning problems.

Conclusion

This work investigated neural network methodology for wide-scale day-ahead forecasting of local loads such as single homes and small aggregations encountered in microgrids or distribution system areas in general. Currently, grid operation relies on standardized profiles for such forecasts which fail to reflect the volatility and diversity of the loads

as they become of interest for local energy management. From numerous existing network architectures, we identify and apply several setups that are practical for wide-scale day-ahead prediction. As exist at present, ANNs do not yield any statistically significant improvement. Herewith, we provide empirical evidence that a prominent neural network methodology might be inadequate for wide-scale day-ahead forecasts of local loads.

Acknowledgements

Not applicable.

Authors' contributions

Oleg Valgaev conducted the literature review, simulations and drafted the manuscript. Hartmut Schmeck and Friederich Kupzog edited, read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

Smart meter data set with measured load curves of numerous single consumers is publicly available from Irish Commission for Energy Regulation (Commission for Energy Regulation (CER) 2012).

Competing interests

The authors declare that they have no competing interests.

Author details

¹Austrian Institute of Technology, Vienna, Austria. ²Karlsruhe Institute of Technology, Karlsruhe, Germany.

Received: 11 June 2020 Accepted: 23 October 2020

Published online: 13 November 2020

References

- Ahmad T, Chen H, Guo Y, Wang J (2018) A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. *Energy Build* 165:301–320. <https://doi.org/10.1016/j.enbuild.2018.01.017>
- Amarasinghe K, Marino DL, Manic M (2017) Deep neural networks for energy load forecasting. *IEEE*. pp 1483–1488. <https://doi.org/10.1109/ISIE.2017.8001465>
- Amjady N, Keynia F, Zareipour H (2010) Short-term load forecast of microgrids by a new bilevel prediction strategy. *IEEE Trans Smart Grid* 1(3):286–294. <https://doi.org/10.1109/TSG.2010.2078842>
- Bagnasco A, Fresi F, Saviozzi M, Silvestro F, Vinci A (2015) Electrical consumption forecasting in hospital facilities: An application case. *Energy Build* 103:261–270. <https://doi.org/10.1016/j.enbuild.2015.05.056>
- Ben Taieb S, Bontempi G, Atiya AF, Sorjamaa A (2012) A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Syst Appl* 39(8):7067–7083. <https://doi.org/10.1016/j.eswa.2012.01.039>
- Bourdeau M, Zhai XQ, Nefzaoui E, Guo X, Chatellier P (2019) Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustain Cities Soc* 48:101533. <https://doi.org/10.1016/j.scs.2019.101533>
- Chalal ML, Benachir M, White M, Shrahily R (2016) Energy planning and forecasting approaches for supporting physical improvement strategies in the building sector: A review. *Renew Sust Energy Rev* 64:761–776. <https://doi.org/10.1016/j.rser.2016.06.040>
- Chitsaz H, Shaker H, Zareipour H, Wood D, Amjady N (2015) Short-term electricity load forecasting of buildings in microgrids. *Energy Build* 99:50–60. <https://doi.org/10.1016/j.enbuild.2015.04.011>
- Chitsaz H, Shaker H, Zareipour H, Wood D, Amjady N (2015) Short-term electricity load forecasting of buildings in microgrids. *Energy Build* 99:50–60. <https://doi.org/10.1016/j.enbuild.2015.04.011>
- Commission for Energy Regulation (CER) (2012) CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010 [dataset]. 1st Edition. Irish Social Science Data Archive. SN: 0012-00. <http://www.ucd.ie/issda/data/commissionforenergyregulationcer/>. Accessed 1 June 2020
- Dan Foresee F, Hagan MT (1997) Gauss-Newton approximation to Bayesian learning. In: *Proceedings of International Conference on Neural Networks (ICNN'97)*, vol. 3. IEEE, Houston, TX, USA. pp 1930–1935. <https://doi.org/10.1109/ICNN.1997.614194>
- Dang-Ha TH, Bianchi FM, Olsson R (2017) Local short term electricity load forecasting: automatic approaches. *arXiv preprint arXiv:1702.08025*
- Dias FM, Antunes A, Mota AM (2003) Regularization versus early stopping: A case study with a real system. In: *2nd IFAC Conference Control Systems Design, Bratislava, República Eslovaca*. <http://cee.uma.pt/morgado/Down/CSD03.PDF>
- Ditzler G, Roveri M, Alippi C, Polikar R (2015) Learning in nonstationary environments: a Survey. *IEEE Comput Intell Mag* 10(4):12–25. <https://doi.org/10.1109/MCI.2015.2471196>
- Foucquier A, Robert S, Suard F, Stéphan L, Jay A (2013) State of the art in building modelling and energy performances prediction: A review. *Renew Sust Energy Rev* 23:272–288. <https://doi.org/10.1016/j.rser.2013.03.004>
- Friedman J, Hastie T, Tibshirani R (2008) *The elements of statistical learning*. vol. 1. Springer series in statistics Springer, Berlin
- Goodfellow I, Bengio Y, Courville A (2016) *Deep learning*. MIT Press, Cambridge. <http://www.deeplearningbook.org>

- Haben S, Ward J, Vukadinovic Greetham D, Singleton C, Grindrod P (2014) A new error measure for forecasts of household-level, high resolution electrical energy consumption. *Int J Forecast* 30(2):246–256. <https://doi.org/10.1016/j.ijforecast.2013.08.002>
- Hagan MT, Menhaj MB (1994) Training feedforward networks with the Marquardt algorithm. *IEEE Trans Neural Netw* 5(6):989–993. <https://doi.org/10.1109/72.329697>
- Hayes B, Gruber J, Prodanovic M (2015) Short-term load forecasting at the local level using smart meter data. *IEEE*. <https://doi.org/10.1109/PTC.2015.7232358>
- Hayes BP, Prodanovic M (2016) State forecasting and operational planning for distribution network energy management systems. *IEEE Trans Smart Grid* 7(2):1002–1011. <https://doi.org/10.1109/TSG.2015.2489700>
- Hernández L, Baladrón C, Aguiar J, Calavia L, Carro B, Sánchez-Esguevillas A, Pérez F, Fernández Á, Lloret J (2014) Artificial neural network for short-term load forecasting in distribution systems. *Energies* 7(3):1576–1598. <https://doi.org/10.3390/en7031576>
- Hintze JL, Nelson RD (1998) Violin plots: a box plotx-density trace synergism. *Am Stat* 52(2):181. <https://doi.org/10.2307/2685478>
- Humeau S, Wijaya TK, Vasirani M, Aberer K (2013) Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households. In: 2013 Sustainable internet and ICT for sustainability (SustainIT). IEEE, Palermo, Italy. pp 1–6. <https://doi.org/10.1109/SustainIT.2013.6685208>
- Hutter F, Kotthoff L, Vanschoren J (2019) Automated Machine Learning: Methods, Systems, Challenges. In: The Springer Series on Challenges in Machine Learning. Springer International Publishing, Cham. <https://doi.org/10.1007/978-3-030-05318-5>
- Irish standard load profiles (2014). <https://rmdservice.com/standard-load-profiles/>. Accessed 1 June 2020
- Kiefer J, Wolfowitz J (1952) Stochastic estimation of the maximum of a regression function. *Ann Math Statist* 23(3):462–466. <https://doi.org/10.1214/aoms/1177729392>
- Kingma DP, Ba J (2017) Adam: a method for stochastic optimization. arXiv:1412.6980 [cs]. <http://arxiv.org/abs/1412.6980>. Accessed 1 June 2020
- Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, Zhang Y (2017) Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans Smart Grid*:1–1. <https://doi.org/10.1109/TSG.2017.2753802>
- Kuznetsov V, Mohri M (2015) Learning theory and algorithms for forecasting non-stationary time series. In: Advances in neural information processing systems, Montreal. pp 541–549
- Llanos J, Saez D, Palma-Behnke R, Nunez A, Jimenez-Estevez G (2012) Load profile generator and load forecasting for a renewable based microgrid using self organizing maps and neural networks. *IEEE*. pp 1–8. <https://doi.org/10.1109/IJCNN.2012.6252648>
- Marinescu A, Harris C, Dusparic I, Clarke S, Cahill V (2013) Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods. *IEEE*. pp 25–32. <https://doi.org/10.1109/SE4SG.2013.6596108>
- Marino DL, Amarasinghe K, Manic M (2016) Building energy load forecasting using deep neural networks. *IEEE*. pp 7046–7051. <https://doi.org/10.1109/IECON.2016.7793413>
- Mena R, Rodríguez F, Castilla M, Arahal MR (2014) A prediction model based on neural networks for the energy consumption of a bioclimatic building. *Energy Build* 82:142–155. <https://doi.org/10.1016/j.enbuild.2014.06.052>
- Mocanu E, Nguyen PH, Gibescu M, Kling WL (2016) Deep learning for estimating building energy consumption. *Sust Energ Grids Netw* 6:91–99. <https://doi.org/10.1016/j.segan.2016.02.005>
- Moré JJ (1978) The Levenberg-Marquardt algorithm: implementation and theory. In: Watson GA (ed). Numerical analysis, vol. 630. Springer Berlin Heidelberg, Berlin, Heidelberg. pp 105–116. <https://doi.org/10.1007/BFb0067700>
- Pirjan A, Oprea S-V, Cărușanu G, Petroșanu DM, Băra A, Coculescu C (2017) Devising hourly forecasting solutions regarding electricity consumption in the case of commercial center type consumers. *Energies* 10(11):1727. <https://doi.org/10.3390/en10111727>
- Raza MQ, Khosravi A (2015) A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renew Sust Energ Rev* 50:1352–1372. <https://doi.org/10.1016/j.rser.2015.04.065>
- Riedmiller M, Braun H (1993) A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: IEEE International Conference on Neural Networks. IEEE, San Francisco, CA, USA. pp 586–591. <https://doi.org/10.1109/ICNN.1993.298623>
- Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323(6088):533–536. <https://doi.org/10.1038/323533a0>
- Runge J, Zmeureanu R (2019) Forecasting energy use in buildings using artificial neural networks: a review. *Energies* 12(17):3254. <https://doi.org/10.3390/en12173254>
- Ryu S, Noh J, Kim H (2016) Deep neural network based demand side short term load forecasting. *Energies* 10(1):3. <https://doi.org/10.3390/en10010003>
- Sevlian RA, Rajagopal R (2014) A model for the effect of aggregation on short term load forecasting. In: 2014 IEEE PES General Meeting Conference & Exposition. IEEE, Washington DC. pp 1–5
- Shi G, Liu D, Wei Q (2016) Energy consumption prediction of office buildings based on echo state networks. *Neurocomputing* 216:478–488. <https://doi.org/10.1016/j.neucom.2016.08.004>
- Wang Y, Chen Q, Hong T, Kang C (2018) Review of smart meter data analytics: applications, methodologies, and challenges. *IEEE Trans Smart Grid*:1–1. <https://doi.org/10.1109/TSG.2018.2818167>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.