Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods

Suryanarayana, Gowri; Lago Garcia, Jesus; Geysen, Davy; Aleksiejuk, Piotr; Johansson, Christian

DOI
10.1016/j.energy.2018.05.111

Publication date
2018

Document Version
Final published version

Published in
Energy

Citation (APA)
Suryanarayana, G., Lago Garcia, J., Geysen, D., Aleksiejuk, P., & Johansson, C. (2018). Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods. *Energy*, *157*, 141-149. https://doi.org/10.1016/j.energy.2018.05.111

Energy 157 (2018) 141–149

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods

Gowri Suryanarayana [a, b, *], Jesus Lago [a, b, c], Davy Geysen [a, b], Piotr Aleksiejuk [e], Christian Johansson [d]

[a] EnergyVille, Thor Park 8310, 3600 Genk, Belgium
[b] VITO, Boeretang 200, 2400, Mol, Belgium
[c] Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, Delft, The Netherlands
[d] NODA, Biblioteksgatan 4, 374 35, Karlshamn, Sweden
[e] Institute of Heat Engineering of Warsaw University of Technology, Poland

ARTICLE INFO

Article history:
Received 22 November 2017
Received in revised form
9 May 2018
Accepted 16 May 2018
Available online 19 May 2018

Keywords:
District heating
Linear models
Regression
Deep learning
Machine learning
Day ahead forecasting

ABSTRACT

Recent research has seen several forecasting methods being applied for heat load forecasting of district heating networks. This paper presents two methods that gain significant improvements compared to the previous works. First, an automated way of handling non-linear dependencies in linear models is presented. In this context, the paper implements a new method for feature selection based on [1], resulting in computationally efficient models with higher accuracies. The three main models used here are linear, ridge, and lasso regression. In the second approach, a deep learning method is presented. Although computationally more intensive, the deep learning model provides higher accuracy than the linear models with automated feature selection. Finally, we compare and contrast the proposed methods with earlier work for day-ahead forecasting of heat load in two different district heating networks.

1. Introduction

The digitization effort in modern district heating systems facilitates the collection of large amounts of online data. This data can be used to implement a range of refined analysis methods in general and model generation techniques in specific. Such models can, for example, be used for forecasting thermal demand in a district heating system. The ability to forecast demand is a vital component of most optimization approaches for the operation of the network, and this especially applies to the more data-driven and automated approaches used in modern 4th generation networks. The primary difference between 3rd and 4th generation networks are lower system temperatures, but there is also a trend in transforming more clearly from reactive control to proactive control. Being proactive means planning ahead, and to plan ahead successfully a forecast of the system in question is vital. Another aspect of this is that 4th generation networks, at least in part, tend to decrease the operational quality of service margins compared to 3rd generation networks, simply due to the fact that lower system temperatures mean network temperatures closer to the delivered operational temperatures within customer systems. Lower margins of error make it more important for the control process to know what is going to happen in the near future. Finally, 4th generation district heating is associated with dynamic pricing schemes, for example based on marginal costs in production and distribution. This is yet another aspect that increases the need for accurate demand forecasts. In general, the more complex operational environment of a 4th generation network, possibly including distributed generation and prosumers, is a key driver in the development of more advanced forecasting technologies.

The research in this paper builds on top of recent work by authors [2,3], which proposed machine learning based approaches to solve operational day-ahead heat demand forecasting in district heating systems, and in Ref. [4], which shows that support vector regressor (SVR) is the best model for forecasting the heat load of district heating. In Ref. [2], a generic ensemble method using three different forecasting algorithms based on extra-trees and extreme learning machines was presented. For the proposed case study, the

[*] Corresponding author. EnergyVille, Thor Park 8310, 3600 Genk, Belgium.
E-mail address: gowri.suryanarayana@vito.be (G. Suryanarayana).

https://doi.org/10.1016/j.energy.2018.05.111
0360-5442/© 2018 Elsevier Ltd. All rights reserved.

best mean absolute percentage error (MAPE) was shown to be 11.7% for the winter months. In Ref. [3], an expert advice system was presented based on four different forecasters: linear regression, extremely randomized trees, feed-forward neural network and SVR. Here, with the same case study as that of [2], the best error was shown to be 11.5%, albeit for an extended test period including the autumn months. In more recent work [5], a customized recursive least square forecaster was used for forecasting the short term greenhouse heat load in a district heating system; the model was shown to be a simple, yet reliable forecaster. Polynomial regression models were shown to supplement artificial neural networks in Ref. [6]. In Ref. [4], SVR, regression trees, feed forwards neural network (FFNN) and multiple linear regression (MLR) were again used for forecasting the day-ahead heat load of smart district heating systems; the comparison showed SVR to be the best performing method.

While the research covered so far in literature has added tremendous value, there are some gaps in it that we address. Firstly, all the methods proposed above require lengthy, expensive and manual feature selection. Particularly, to obtain the optimal set of explanatory variables, the models need to be retrained multiple times, and human intervention is usually needed to analyze the results. Secondly, the rising popularity of deep learning and its success in several energy-related tasks [7–13], shows potential for its application even in forecasting the heat-load of district heat networks. With that as motivation, the goal of this paper is to investigate new techniques that consider the two afore mentioned gaps. We first investigate the usage of automatic feature selection techniques and their impact on the accuracy of heat load forecasting. We compare the results of the new methods with those in Refs. [2–4,6], and show that these methods not only eliminate the lengthy feature selection process, but also lead to better accuracies. Additionally, we investigate the usage of deep learning as a viable technique to forecast heat load of district heat networks. In particular, we consider a forward neural network with two hidden layers that uses state-of-the-art deep learning techniques, e.g., ReLU, dropout, training using stochastic gradient descent, and we compare its accuracy with a range of other models. Based on the obtained results, we show how the proposed deep learning technique is able to generalize better and obtain more accurate forecasts.

1.1. Contributions

In the first approach, we show how regressors based on linear models can provide improved accuracies when used with appropriately chosen features. In this context we explore a family of three linear models: linear, ridge, and lasso regression. All three models establish a linear dependence of the target variable on the explanatory variables. The main advantage of using these models is their simplicity: they are easy to formulate and the computational complexity of training these models is very low, which means that they can be retrained for drifts in parameters very swiftly. However, when used naively, linear regressors are fairly limiting as many dependencies that need to be captured can be non-linear. This has been demonstrated several times, where forecasters that capture complex non-linear dependencies have proven to be superior to linear regression, for e.g., see Refs. [3,14]. In this paper we propose an alternative over the naive approach. First, we encode the non-linearities explicitly as additional features in the training process. The difficulty here is that in most scenarios, the variables on which there are non linear dependencies or the nature of such non-linearities is not known. To circumvent this problem, we build a super set of features involving many degrees of non-linearities and the valid non-linearities are chosen through a special feature selection process. While for the simple linear regressor we propose an explicit automatic feature selection algorithm, the ridge and lasso regressors perform embedded feature selection through regularization. The parameter influencing the regularization term in the ridge and lasso regressors is chosen through hyperparameter optimization.

In the second approach, we use a forecaster based on deep learning. This choice was motivated by the many advances made in the field of neural nets (more recently referred to as deep learning). These advances started with the overcoming of challenges inherent to neural nets such as computation cost of training large models; see Ref. [15], where efficient training of *deep belief* networks was done with a greedy layer-wise *pre-training*. Subsequent improvements lead to efficient training of networks with multiple hidden layers, giving better results that were applicable to wider domains. Studying these new architectures and methodologies was then termed deep learning, where the term deep referred to the ability to train a neural network model whose depth was not limited to a single hidden layer [16]. Although deep learning models were originally developed for computer science applications such as image recognition [17], speech recognition [18], and machine translation [19], their success in energy market applications became widespread in the last two years [7–13]. Forecasting accuracies vastly improved, especially in wind power forecasting [10,12] and electricity markets [1,13]. In particular, within the context of electricity prices, [13] showed that deep learning models outperform a large benchmark of 98 prediction models. Motivated by these improvements, we extend the deep learning research to the field of heat load forecasting.

The paper is structured as follows. Section 2 presents the preliminary concepts used throughout the paper. In particular, Section 2.2 and Section 2.3, present the details of the novel techniques used for hyperparameter tuning and feature selection. In Section 3, the basic theory behind the forecasters used in this paper is presented. Section 4 gives an overview of the case studies considered in this work. Here, in addition to the heat network introduced in Refs. [2,3], the proposed methods are also tested on an additional heat network. Finally, in Section 5, the results of the forecasting models are presented for the two test cases.

2. Preliminaries

In this section, the theoretical concepts and algorithms that are used and adapted in the research are introduced.

2.1. Baseline forecasters

In order to evaluate the models that we propose in the following sections, we consider two of the most successful machine learning forecasting methods as baseline algorithms. We use the SVR model considered in Refs. [3,4] as the first baseline model. For the second baseline model, we use the extreme gradient boosting (XGBoost) algorithm [20], a forecaster which is similar to the extreme tree regressors model used in Ref. [3] and is also based on an ensemble of trees. However, in contrast with the latter, it uses boosting in place of bagging in order to build the ensemble. We use XGBoost instead of extreme tree regressors as it is known to obtain more accurate results in practice [21,22].

In addition, as introduced in a later section, we also consider the polynomial regression method of [6]. However, instead of using the polyfit function of Matlab, we proposed a modification to perform automatic feature selection.

2.2. Hyperparameter optimization

In general, any machine learning model has some hyperparameters (set before training the model) determining its performance. Needless to say, to obtain optimal results from a model, these hyperparameter have to be chosen appropriately. The most widely used techniques to perform hyperparameter optimization in the machine learning community are *Bayesian optimization* algorithms [23], a family of algorithms for optimizing black box functions. For hyperparameter optimization, the black box functions of interest are the performance indicators of forecasters expressed as functions of hyperparameters. Bayesian optimization algorithms require a much lower number of function evaluations than other alternatives such as evolutionary optimization techniques or grid search. With every sample of the black box function, these algorithms update the prior belief used for sampling the next value. This way, the number of samples drawn can be reduced leading to efficient evaluation of the optimum value.

One such technique Bayesian optimization is the *Tree-Structured Parzen Estimator (TPE)* [24], which is a *sequential model-based optimization* (SMBO) algorithm [25]. A SMBO method iteratively approximates the black box function (with every sample) and finds the local optimum of the resulting approximations. At the *i*th iteration, the black box function is first evaluated at a point $\boldsymbol{\theta}_i$ in the parameter space. Next an approximation $\mathscr{F}_i$ is obtained by fitting all the function evaluations from sample points so far. The next sampling point $\boldsymbol{\theta}_{i+1}$ is then obtained by optimizing for $\mathscr{F}_i$. This process is continued till the maximum number of iterations is reached, and the best sample so far is returned.

An illustration of the sequential model-based optimization method is given in Algorithm 1. In this paper, we extensively use this algorithm for tuning of hyperparameters of all the models.

Algorithm 1
Hyperparameter Optimization using sequential model-based optimization.

```
procedure hyperopt $(T,\boldsymbol{\theta}_0)$
  $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_0$
  $\mathscr{P} \leftarrow \varnothing$
  for $i = 1, \ldots, T$ do
    $p_i \leftarrow$ TrainModel$(\boldsymbol{\theta}_i)$
    $\mathscr{P} \leftarrow \mathscr{P} \cup \{(p_i, \boldsymbol{\theta}_i)\}$
    if $i < T$ then
      $\mathscr{F}_i(\boldsymbol{\theta}) \leftarrow$ EstimateModel$(\mathscr{P})$
      $\boldsymbol{\theta}_i \leftarrow \mathrm{argmax}_{\boldsymbol{\theta}}\, \mathscr{F}_i(\boldsymbol{\theta})$
  $\boldsymbol{\theta}^* \leftarrow$ BestHyperparameters$(\mathscr{P})$
  return $\boldsymbol{\theta}^*$
```

2.3. Feature selection

As mentioned earlier, feature selection plays an important role in model estimation. Feature selection algorithms can mainly be classified into three categories: *filter*, *wrapper*, and *embedded methods* [26]. Each of these families come with their advantages and drawbacks. While wrapper methods look for the best set of features among the sample space of all the features, embedded methods such as regularization methods perform implicit feature selection during the process of estimating the model. For the sake of completion we mention that filter methods use certain statistical measures to select important features. These methods do not estimate the models during feature selection, leading to fast computation times, but with feature selections that are not entirely reliable due to the absence of accuracy performance indicators. In comparison with embedded methods, wrapper methods are often computationally more intensive (though leading to more objective feature selection) as the sample space of the feature set is often very large. Due to their focus on the underlying model and complex dependence on explanatory variables, the methods we propose will be based on wrapper and embedded feature selections algorithms.

To reduce the computation time of wrapper methods but still benefit from their higher accuracy, [1] proposed a wrapper method that reduced the number of iterations required when performing the search across the feature space. In particular, instead of performing a regular grid search, it considered the TPE method, which was described in Section 2.2, to infer the relations between selected features and model performance, and then use these inferred relations to guide the search. In detail, the feature selection methods first model the features as two types of model hyperparameters:

1. Inclusion-exclusion features which can be modeled with a binary hyperparameter. These are the most common type of features and can be used to decide whether to use a specific input.
2. Features that represent some length. An example could be how many days of past grid load we need to consider in order to forecast the day-ahead grid load. This type of feature is modeled with an integer hyperparameter.

Then, after performing an optimization using the algorithm described in Section 2.2, the procedure fine tunes the feature selection using functional ANOVA [27].

It is important to note that this algorithm is really beneficial when the feature space is big. Particularly, as it infers the relations between performance and features to conduct the search for the best set of features, we can observe its advantages when performing a full search across the feature space is infeasible.

2.4. Performance metric

A performance metric is needed to evaluate and compare the accuracy of the forecasters. In this paper, we use the mean absolute percentage error (MAPE):

$$\mathrm{MAPE} = \frac{100}{N} \sum_{k=1}^{N} \frac{|y_k - \widehat{y_k}|}{|y_k|}, \tag{1}$$

where $[y_1, \ldots, y_N]^\top$ are the observed values and $[\widehat{y}_1, \ldots, \widehat{y}_N]^\top$ the forecasted values.

2.5. Diebold-Mariano test

While the MAPE is a good metric to provide a first assessment, we can not infer from it a proper comparison between forecasters. In particular, while based on MAPE a forecater might have a better accuracy, that result might be the product of the stochasticity of the data or the model estimation. Therefore, to assert whether a certain forecaster is statistically significantly better than others, we need to use statistical testing. In our application, we test the statistical significance of the difference in the accuracies obtained by two forecasters using the Diebold-Mariano (DM) Test, see Ref. [28].

Let $[y_1, \ldots, y_N]^\top$ be the time series vector to be forecasted, and $[\widehat{y}_1, \ldots, \widehat{y}_N]_{F_1}^\top$ and $[\widehat{y}_1, \ldots, \widehat{y}_N]_{F_2}^\top$ be the forecasted values from two models F_1 and F_2 respectively. We obtain the corresponding errors in forecasting $[\varepsilon_1, \ldots, \varepsilon_N]_{F_1}^\top$ and $[\varepsilon_1, \ldots, \varepsilon_N]_{F_2}^\top$ and define the following loss differential function:

$$d_k^{F_1,F_2} = L\left(\varepsilon_k^{F_1}\right) - L\left(\varepsilon_k^{F_2}\right), \tag{2}$$

where L is a loss function that has to be chosen so that $d_k^{F_1,F_2}$ is covariance stationary. We use a test called the one-sided test, very similar to the widely used *two-sided test*. Here, the null hypothesis H_0 is that the forecaster F_1 has the same accuracy as that of F_2 and the alternate hypothesis is that the accuracy of F_1 is better than that of F_2:

$$\begin{aligned} H_0 &: \mathbb{E}\left[d_k^{F_1,F_2}\right] \geq 0, \\ H_1 &: \mathbb{E}\left[d_k^{F_1,F_2}\right] < 0, \end{aligned} \tag{3}$$

where $\mathbb{E}$ denotes the expected value. For the test to be reliable, the loss differential needs to be covariance stationary, and a loss function of the following form is typically used to ensure that:

$$L\left(\varepsilon_k^{F_i}\right) = \left|\varepsilon_k^{F_i}\right|^p, \tag{4}$$

where $p \in \{1,2\}$.

3. Forecasters

This section presents the important concepts behind each of the forecasting models used: linear regression, lasso regression and ridge regression for the first approach and deep neural network for the second.

3.1. Polynomial linear regression

As the name suggests the method involves establishing a linear relationship between the target variable $\boldsymbol{y} \in \mathbb{R}^n$ and the explanatory variables, that are columned in a *feature matrix* $X \in \mathbb{R}^{m \times n}$. Here, n is the number of observations and m is the number of explanatory variables. The model looks as follows

$$\boldsymbol{y} = X\boldsymbol{\theta} + \varepsilon, \tag{5}$$

where $\boldsymbol{\theta}$, the parameters of the model are to be determined and ε denotes the error term (unobserved). In linear regression (ordinary least squares), the parameters are estimated by maximizing the logarithm of the likelihood $\mathscr{L}(\theta) = p(y|\theta)$ assuming that the errors ε are Gaussian, i.e.,:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_{LR} &= \arg\min_{\boldsymbol{\theta}} \|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 \\ &= \left(X^T X\right)^{-1} X^T \boldsymbol{y}, \end{aligned} \tag{6}$$

where $\|\bullet\|_2$ is used to denote the ℓ_2 norm. In polynomial regression, any non-linear dependence on explanatory variables can be encoded explicitly. For instance, a polynomial dependence of the form

$$\boldsymbol{y} = \theta_1 \boldsymbol{x} + \theta_2 \boldsymbol{x}^2 + \theta_4 \boldsymbol{x}^4$$

on some explanatory variable $\boldsymbol{x}$ can be modeled by having three different features in the feature matrix, one for each degree.

While this method is similar to the polynomial regression method proposed in Ref. [6], it has a key distinction: to handle the complexity of the large input feature space, the method is modified to consider an automatic feature selection method. This modification is briefly motivated in the paragraph below and explained in more detail in the next sections.

One of the main assumptions made in this method of finding the parameters is that the feature matrix has full rank, i.e., that the explanatory variables are linearly independent of one-another. However, if this linear independence does not hold due to correlated features, the parameters obtained by this model are prone to instabilities. When considering a large number of features, there might arise co-linearities between some of the variables, and this could lead to higher errors in the estimator [29]. Hence, we need a robust feature selection procedure to be in place. For the least-squares estimator, we use the method described in Section 2.3.

3.2. Ridge regressor

Ridge regression, proposed in Ref. [29] gives a method to circumvent the problem faced by the ordinary least squares estimator in the presence of co-linear features. The model used here is the same as that given in (5), but in addition to the regular assumption on ε, the model also considers a prior Gaussian distribution $p(\theta)$ on the parameters θ and maximizes the logarithm of the likelihood $\mathscr{L}(\theta) = p(\theta|y)$ using Bayes rules, i.e.,:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_{RR} &= \arg\min_{\boldsymbol{\theta}} \left(\left\|X\boldsymbol{\theta} - \boldsymbol{y}_2^2\right\| + \alpha||\boldsymbol{\theta}||_2^2\right) \\ &= \left(X^T X + \alpha I\right)^{-1} X^T \boldsymbol{y}. \end{aligned} \tag{7}$$

This leads to a biased but stable estimator, even when the matrix $X^T X$ becomes ill-conditioned due to the presence of correlated features. The additional term in the minimization expression, which models the prior distribution $p(\theta)$, acts in practice as an ℓ_2 regularization term that penalizes the magnitude of the parameter estimators and that leads to an embedded feature selection. Note that the factor α plays an important role. While very low values of α could give results similar to that of linear regression, very high values, can suppress the parameters more than necessary. It thus needs to be chosen carefully, and will be subject to hyperparameter tuning.

3.3. Lasso regressor

This regressor is an alternative to the ridge regressor and also implicitly performs feature selection through the regularization of the parameters. However, in contrast to the ridge regressor, the prior distribution $p(\theta)$ is assumed to be Laplacian, which in turns leads to a regularization factor based on the ℓ_1-norm [30], i.e. the parameters are estimated via:

$$\widehat{\boldsymbol{\theta}}_{LLR} = \arg\min_{\boldsymbol{\theta}} \left(||X\boldsymbol{\theta} - \boldsymbol{y}||_2^2 + \alpha||\boldsymbol{\theta}||_1\right). \tag{8}$$

Note that the ℓ_1 norm can push the values of some parameters closer to zero compared to the ℓ_2 norm, leading to feature selection in a stricter sense. The value of α plays an important role here as well and needs to tuned appropriately.

3.4. Deep neural net (DNN)

Several architectures of DNN's such as the standard, convoluted or recurrent networks have been widely researched for forecasting applications; for this work, we consider a standard DNN, i.e., the extension of a multilayer perceptron to multiple hidden layers.

A general DNN with two hidden layers can be represented as in Fig. 1. In this representation, $\mathbf{X} = [x_1, \ldots, x_n]^\top \in \mathbb{R}^n$ is the input of the network, $\mathbf{Y} = [y_1, y_2, \ldots, y_m]^\top \in \mathbb{R}^m$ the output, n_k is the number of neurons of the kth hidden layer, and $\mathbf{z}_k = [z_{k1}, \ldots, z_{kn_k}]^\top$ is the state in the kth hidden layer,

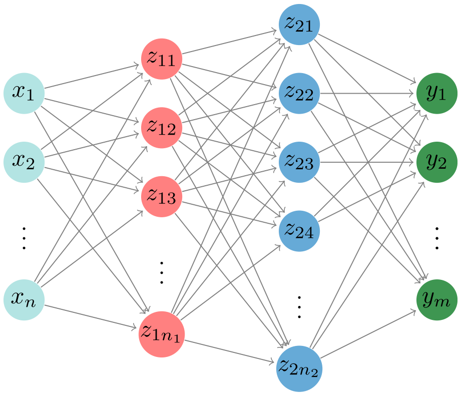

Fig. 1. Example of a DNN.

3.5. Model variants

For the linear models, two variants of each of the regressors are implemented. In the first version, we built a single model for all hours of the day (for day-ahead forecasting). In the second version, we considered a separate model for every hour. These models are denoted by linear-24, ridge-24 and lasso-24. The main advantage of building a model per hour is that the available historical data can be used more efficiently. The earlier hours can depend on more recent lags than the hours later in the forecasting horizon. The same holds even for temperature forecasts. The earlier hours can depend even on future forecasts, compared to the later hours. This has the potential of reducing the errors in forecasting for the earlier hours substantially, bringing the overall error down.

4. Two case studies

The two heat networks used as case studies for this work are explained in more detail in this section.

4.1. Rottne district heating network

The district heating network in Rottne, in the south of Sweden, was used as one of two use cases for this study. The district heating system (DHS) in Rottne is owned by Växjö Energi, and is operated as a stand-alone smaller district heating system. The network itself is a traditional 3*rd* generation DHS, with a central heat generation plant with a set of boilers and is connected to a surrounding piping network with a mixed demand consisting of residential buildings, single-family houses, schools and commercial buildings. The heat losses in the network vary roughly between 10 and 20% of the total demand, depending on seasonal variations. The total consumer base consists of about 200 buildings, with a majority of 150 of those being single-family dwellings. The rest are a combination of offices, schools and multi-family residential buildings.

The distribution network consists of roughly ten 300 m of piping with a total volume of about 64 m^3. The production site was built and made operational in 1998. Originally the production set-up consisted of a 1.5 MW dry wood burner in combination with a 3 MW fossil oil burner. Later, there was a need to use more moist wood chip fuels, and in 2004 the wood burner was refurbished to facilitate this. However, this lowered the heat capacity to 1.2 MW. In connection with this, a second wood chip burner with a capacity of 1.5 MW was installed. Furthermore, in 2012, the oil burner was retrofitted to use rapeseed oil based biodiesel instead of fossil oil. The combination of the two wood chip boilers satisfy most of the demand, and the biodiesel is primarily used to cover peak load and prolonged cold streaks.

Biodiesel produced from rapeseed oil is considerably more environmentally friendly than fossil oil. However, it is still much more expensive than the wood chips used for the base load, and should therefore be avoided if at all possible. The production site is controlled based on the relation between primary supply temperature and outdoor temperature. Basically, the colder it gets, the warmer the supply temperature needs to be. The pressure is maintained by automatically controlled pumps to ensure the desired differential pressure. The wood chip boilers will try to maintain the required supply temperature, but if the demand is high they will not be enough and this will cause a drop in supply temperature. This drop will automatically trigger the biodiesel boiler to start generating heat supply. The combined heat capacity of the two wood chip boilers is normally 2.7 MW (1.2 + 1.5), but since 2017 the larger boiler has been refitted to accept lower quality wood chips. This has lowered the heat capacity slightly, so now the total peak cap is about 2.5 MW.

The price difference between wood chips and biodiesel is substantial. To optimize the operational behavior of the production units, and to avoid using biodiesel whenever possible, a demand side management system (DSM) is used in the DHS. The efficiency of such a system is heavily linked to the ability to forecast heat demand, which is why this study is relevant to the DHS of Rottne.

4.2. Karlshamn district heating network

The district heating network in Karlshamn is somewhat bigger than the Rottne network, although it is relatively geographically close. This makes it an interesting alternative as reference case. The Karlshamn DHS had about 70 GWh in annual delivery when it was made operational more than 25 years ago, and currently it has grown to about 200 GWh in annual heat delivery. About 95% of heat delivered is generated through excess heat from a nearby industry, and the rest is covered by a combination of bio-oil, fossil oil and natural gas. The network normally peaks at about 50–60 MW during a normal winter.

The distribution network is connected to a nearby paper mass factory, which is the source of the excess heat. This heat is then distributed to a nearby village as well as to the central city of Karlshamn. In Karlshamn the heat is further distributed to the city center as well as further away to two other smaller urban areas. The total distribution grid is about 250000 m and supplies about 1500 customers. Some 1000 of those are single family dwellings, while the rest are office buildings, schools, multi-residential buildings and public and commercial buildings of different kinds.

Similar to the Rottne DHS, the Karlshamn DHS also uses DSM's to avoid or reduce peak loads. In Karlshamn the DSM system covers roughly the hundred largest buildings in the DHS, and it has the capacity to reduce the heat demand of about 10–15% in total, and about 20–25% in certain parts of the network. As in Rottne, the DSM system is dependent on robust heat demand forecasting.

5. Implementation

All the code is implemented in Python. We implement all the models using the scikit package of python. For the GBT model, we employ the XGBoost [20] python library. For the DNN, we use the Keras [31] deep learning library together with the Theano [32] library for mathematical modeling. For hyperparameter tuning, we use the hyperopt [33] package.

5.1. Hyperparameter optimization

The hyperparameters that are considered and optimized for each of the models are listed in Table 1.

5.2. Feature selection

For the DNN, SVR and GBT, the following possible input features are considered: hour of the day, day of the week, last 7 days of heat load and temperature forecasts, and the next 24 h of temperature forecast. To select the optimal subset of features for each dataset, the method described in Section 2.3 is used.

For the linear model variants with the same model for all hours of the day, the following super set of features is used: hour of the day, day of the week, day of the year, and heat load and temperature forecast starting from the previous day up to one week in the past, and the next 24 h of temperature forecast. The reason for using the past information this way is that, while predicting the 24th hour from now, the model has past information only till 24 h in the past. This limitation is overcome in hourly linear models, where the models for earlier hours can depend on more recent heat load values, compared to the models of later hours. Polynomial dependence of up to 4th degree is added to each of the heat load and temperature forecast features. While for the linear regression model, the method in Section 2.3 is then used to find the optimal subset of features, the ridge and lasso regressors make use of embedded feature selection from regularization.

5.3. DM test for heat load forecasting

We now discuss the use of the DM tests to asses the statistical significance of the differences in forecasting accuracy. The following loss differential function is used:

$$d_k^{F_1,F_2} = \left|\varepsilon_k^{F_1}\right| - \left|\varepsilon_k^{F_2}\right|. \tag{9}$$

We consider a separate time series for each hour of the forecast horizon and perform the DM test independently for each of these series, as in Refs. [1,13,34,35]. In addition, we also perform a DM test considering the whole loss differential and serial correlation. There are several advantages of doing this. Firstly, the errors within a day are very likely to be correlated, as the same historical training information is used for all the hours. Secondly, hourly analysis helps us distinguish between the following three cases:

1. Forecaster F_1's accuracy is significantly better than that of F_2 for all the hours.
2. The overall accuracy of forecaster F_1 is significantly better than that of F_2, but there exist some hours where the latter has significantly better accuracies.
3. The accuracy of F_1 is not better that of F_2.

Table 1
Summary of the optimized hyperparameter for the models.

Model	Symbol	Definition
Ridge	α_r	Coefficient for ℓ_2 regularization
Lasso	α_l	Coefficient for ℓ_1 regularization
SVR	C	Penalty parameter of the error
	ε	Epsilon of the epsilon-SVR model
XGBoost	n_t	Number of trees
	d_{max}	Maximum tree depth
	lr	Learning rate
	γ	Minimum loss reduction needed to make a new partition on a leaf node
	α_x	Coefficient for ℓ_1 regularization
	λ_x	Coefficient for ℓ_2 regularization
	r_{sub}	Subsample ratio of the training set used for training a tree
	r_{col}	Subsample ratio of columns when training a tree
DNN	n_k	Number of neurons on the kth hidden layer, with $k = 1,2,3,4$.
	nonlin	Activation function on the hidden layers
	d	Dropout coefficient
	λ_{lr}	The initial learning rate used for the stochastic gradient descent method.
	BN	Binary hyperparameter to select if batch normalization is applied.

Finally, knowing the statistical significance of hourly accuracies, can help us build ensemble methods, where for each hour, the model proven to have significantly better accuracy can be used.

To make the distinction between the three cases described above, we do the following one sided DM tests:

1. For every hour h and pair of models F_1 and F_2, a DM test at a 95% confidence interval with the null hypothesis in the lines of (3):

$$\mathrm{DM}_h \begin{cases} H_0 : \mathbb{E}\left[d_{h,k}^{F_1,F_2}\right] \geq 0, \\ H_1 : \mathbb{E}\left[d_{h,k}^{F_1,F_2}\right] < 0, \end{cases} \quad \text{for } h = 1, \ldots 24, \tag{10}$$

where k is used to index the time series of the particular hour h.

2. For every hour h and pair of models F_1 and F_2, a DM test with a null hypothesis complementary to that in (10), i.e., that the forecaster F_2 has the same accuracy as that of F_1:

$$\widehat{\mathrm{DM}}_h \begin{cases} H_0 : \mathbb{E}\left[-d_{h,k}^{F_1,F_2}\right] \geq 0, \\ H_1 : \mathbb{E}\left[-d_{h,k}^{F_1,F_2}\right] < 0, \end{cases} \quad \text{for } h = 1, \ldots 24. \tag{11}$$

3. In case F_1 and F_2 each have significantly better accuracies for at least 1 h, a regular DM test considering serial correlation and the full loss differential, i.e., d^{F_1,F_2}, is considered:

$$\mathrm{DM}_{sc} \begin{cases} H_0 : \mathbb{E}\left[d^{F_1,F_2}\right] \geq 0, \\ H_1 : \mathbb{E}\left[d^{F_1,F_2}\right] < 0. \end{cases} \tag{12}$$

Following the procedure in Ref. [1], we make the following statements about the prediction accuracies of F_1 and F_2:

1. The predictive accuracy of F_1 is significantly better than that of F_2 if the following two conditions are met:
   (a) The null hypothesis is rejected for at least one of the hours for the regular DM_h, i.e. F_1 has accuracy significantly better than that of F_2 for at least 1 h.
   (b) The null hypothesis of none of the complementary $\widehat{\mathrm{DM}}_h$ tests is rejected, i.e. there is no hour where the predictive accuracy of F_2 is better than that of F_1.
2. If both F_1 and F_2 have at least 1 h in which they have significantly better accuracies, we consider the result of the DM test DM_{sc} that considers the whole differential loss with serial correlation. Then, if the null hypothesis of DM_{sc} is rejected, we conclude that the overall accuracy of F_1 is better than that of F_2, although there are some hours at which F_2's accuracy is significantly better.

6. Results

This section presents the results and comparison between the different models. For both data sets, roughly 27 months of data from November 2014 to February 2017 were available. For the first data set, we used the same test period as in Ref. [3], i.e., months from August 2016 to February 2017. For the second dataset, we chose the test period closer to a heating season - end of October 2016 (28-10-2016) to end of February (27-02-2017). For the linear models the rest of the data was used for training. The DNN, SVR and GBT models used the first three months of the data for validation and the rest for training. The reason for selecting these three months is twofold; firstly, due to the lack of sufficient data we could only use three months for validation. Secondly, as head load forecasting is crucial during the winter months, our test and validation sets focus on these months.

To evaluate the error in the test set, the models are retrained at every day so that testing is done as in real life conditions, i.e. using the most recent data to recalibrate the models. It is important to note that only the model is retrained; the hyperparameters are kept fixed and equal to the best configuration obtained during the hyperparameter optimization. We would again like to stress that the simplicity of the linear models used here allows for quick retraining, and that thus retraining every day in real-time is a feasible option.

6.1. Results MAPE

Table 2 gives the MAPE values for the different models used. Note that the deep learning model has the best MAPE in both cases, 8.08% and 4.15% respectively. The best linear models in both cases come second with 8.77% and 4.44% respectively. In both cases the hourly ridge regression gives errors very close to that of the deep learning model. Both deep learning and the best linear model outperform the SVR and GBT forecasters with respect to MAPE. We also note that for case study 1, while SVR gives results similar to that in Refs. [2,3], the models proposed in this paper give significantly higher accuracies. It is also important to remark that we have made improvements in the baseline models that were used in Refs. [2,3]. The GBT especially gives an improved MAPE compared to the extreme tree regressor that were used earlier.

Table 2
Model comparison in terms of MAPE for the two case studies.

	Case study 1	Case study 2
Linear regression	9.08	4.77
Ridge regression	9.06	4.76
Lasso regression	9.09	6.75
Linear regression-24	9.88	5.03
Ridge regression-24	8.77	4.44
Lasso regression-24	9.44	4.46
GBT	9.05	4.60
SVR	11.75	4.78
DNN	8.08	4.15

6.2. Results DM test

Tables 3 and 4 summarize the DM test comparison results for case studies 1 and 2 respectively. It is important to note that the table's entries are not fully anti-symmetric, i.e., F_1 not being significantly better than F_2 has no implication on whether or not F_2 is significantly better than F_1. Three scenarios arise:

1. The prediction accuracy of the model F_1 is significantly better than that of F_2, with the alternative hypothesis being accepted with 95% confidence (represented with ✓in the tables).
2. Although F_2 may be significantly better in at least one of the 24 h of the forecast horizon, the overall accuracy of F_1 for the full loss differential is still statistically significantly better (represented by $\checkmark_s$ in the tables).
3. The prediction accuracy of F_1 is not significantly better than F_2. (represented by blank entries in the tables)

Fig. 2 shows the DM test results of the DNN with respect to all other forecasters used for both case studies. By following each of the curves over the full day, one can determine whether or not the DNN is significantly better.

Based on Tables 3 and 4 and Fig. 2, we observe the following:

Table 3
DM test comparison results for case study 1.

F_1	F_2								
	Lasso-24	SVR	GBT	Linear	Linear-24	Lasso	Ridge	Ridge-24	DNN
Lasso-24									
SVR									
GBT									
Linear	$\checkmark_s$								
Linear-24	$\checkmark_s$								
Lasso	$\checkmark_s$								
Ridge	$\checkmark_s$			✓					
Ridge-24	$\checkmark_s$	$\checkmark_s$	$\checkmark_s$	✓	✓	$\checkmark_s$	$\checkmark_s$		
DNN	✓	$\checkmark_s$	✓	✓	✓	✓	✓	✓	

Table 4
DM test comparison results for case study 2.

F_1	F_2								
	Lasso-24	SVR	GBT	Linear	Linear-24	Lasso	Ridge	Ridge-24	DNN
Lasso-24		$\checkmark_s$		$\checkmark_s$	✓	✓			
SVR					✓	✓			
GBT					✓	✓			
Linear						✓			
Linear-24						✓			
Lasso									
Ridge				✓		✓			
Ridge-24		✓	✓		✓	✓			
DNN		$\checkmark_s$	✓	$\checkmark_s$	✓	✓	$\checkmark_s$		

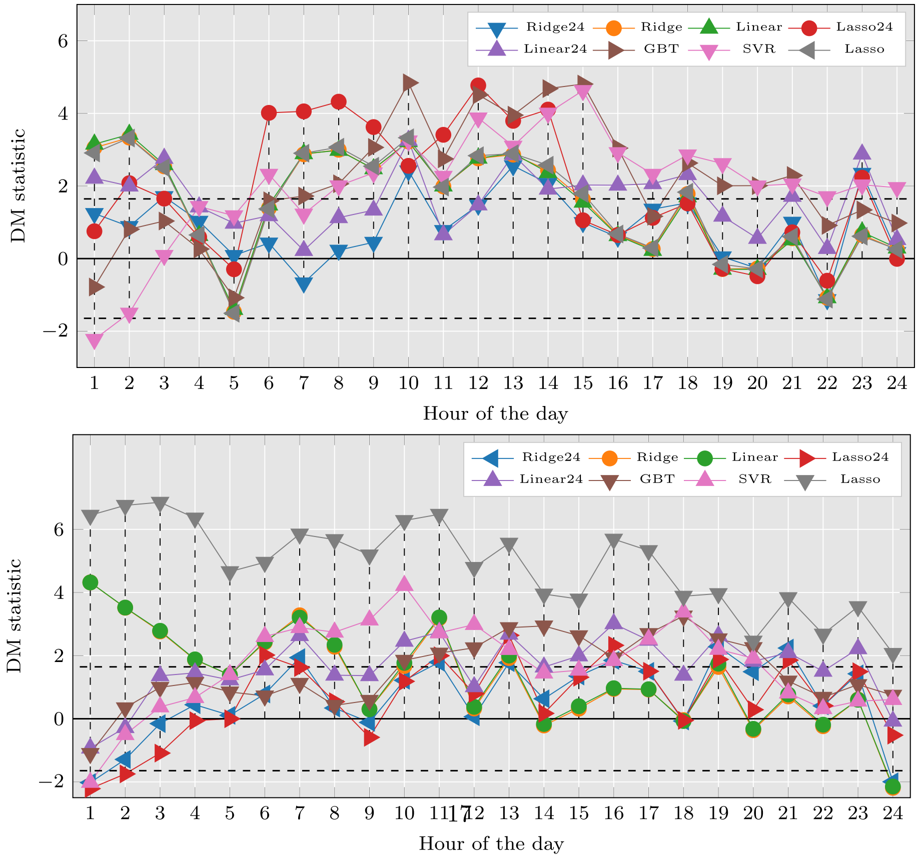

Fig. 2. DM results for the DNN model. **Top**: test results for case study 1. **Bottom**: test results for case study 2. Values above the top dashed line represent cases where, with a 95% confidence level, the DNN is significantly better. Similarly, values below the lower dashed line accept at a 95% confidence level that the DNN is significantly worse.

1. The DNN does significantly better than all the linear models as well as the baseline models in case study 1.
2. The hourly ridge regression model is significantly better than both the baseline models in both case studies, and also better than the rest of the linear models in case study 1.
3. In case study 2, the DNN is significantly better than both the baseline models, and four of the linear models. However, nothing can be concluded about its performance compared to the hourly lasso and hourly ridge regression models.
4. The baseline models are significantly better than none of the proposed models in case study 1, and better only compared to the linear-24 and lasso models in case study 2.
5. In case study 1, only the DNN is better than the ridge 24, and in case study 2, none of the models are better than ridge-24 and lasso-24.
6. In both case studies none of the models are significantly better than the DNN.

Overall, in case study 1 the DNN proves to be the strongest model and the hourly ridge regression model is the next. For case-study 2, although it cannot be concluded on which the best model is, the DNN, the hourly ridge and hourly lasso regressors perform the best. Particularly, we have demonstrated that the DNN and the ridge-24 proposed in this paper are significantly better than the state-of-the-art models.

7. Conclusion and future work

We have shown that simple linear models can be very powerful in forecasting heat loads in district heating networks when non-linearities can be accounted for and automatic feature selection is done. In particular, they can outperform many of the advanced machine learning forecasting tools and the state-of-the-art methods proposed in literature such as SVR and GBT. The ridge regressor with hourly models proved especially powerful in both the use cases, giving a MAPE as low as 8.77 in the first case and 4.44 in the second. This model even proved better even with respect to the DM test and performed nearly as well as the deep learning model. We also showed that deep learning models provide the best accuracies overall in terms of both MAPE (8.08 and 4.15) and the DM test, provided enough computation time is available.

The data used for this study was collected from two district heating systems in Sweden. The Karlshamn network, operated by Karlshamn Energi, uses industrial excess heat from an external source to cover more than 90% of the yearly energy demand. Since they are not fully in control of this heat supply, they are dependent on forecasting the demand as accurately as possible. The other grid is the Rottne district heating system, operated by Växjö Energi. The Rottne grid has been used as a demonstrator in the Horizon 2020 project STORM the last few years, in which an advanced grid controller has been deployed. Such grid controllers are dependent on accurate load forecasting and the results presented in this paper will contribute to an even higher level of efficiency in the grid. For both of these cases, the techniques presented in this study will facilitate increased operational efficiency. In general, it is expected that the results will contribute to the further development of modern grid controllers for 4th generation district heating and cooling.

We also foresee these forecasters to be used in other project contexts, especially to predict the electricity load of a cluster of buildings; particularly, as the heat and electricity consumptions of households follow similar patterns, i.e., trend, seasonality, the proposed methods could potentially be easily extended to the latter. This is an important area of research in the field of demand response. In the next step we also want to include these improved forecasters in an expert advice system. Additionally, having statistical significance results for each hour, we can even consider an ensemble of forecasters, where for each hour we can choose the best model for that hour.

Acknowledgment

This research has received funding from the European Union's Horizon 2020 research and innovation program under the grant agreement No 649743 (STORM) and the Marie Skodowska-Curie grant agreement No 675318 (INCITE). The work related to this research was done at EnergyVille/VITO, Noda, Växjö Energi and Karlshamn Energi.

References

[1] Lago J, De Ridder F, Vrancx P, De Schutter B. Forecasting day-ahead electricity prices in Europe: the importance of considering market integration. Appl Energy 2018;211:890–903. https://doi.org/10.1016/j.apenergy.2017.11.098.

[2] Johansson C, Bergkvist M, Geysen D, Somer OD, Lavesson N, Vanhoudt D. Operational demand forecasting in district heating systems using ensembles of online machine learning algorithms. Energy Procedia 2017;116:208–16. https://doi.org/10.1016/j.egypro.2017.05.068.

[3] Geysen D, Somer OD, Johansson C, Brage J, Vanhoudt D. Operational thermal load forecasting in district heating networks using machine learning and expert advice. arXiv eprint. 2017. arXiv:1710.06134.

[4] Idowu S, Saguna S, Hlund C, Scheln O. Applied machine learning: forecasting heat load in district heating system. Energy Build 2016;133:478–88. https://doi.org/10.1016/j.enbuild.2016.09.068.

[5] VoglerFinck P, Bacher P, Madsen H. Online short-term forecast of greenhouse heat load using a weather forecast service. Appl Energy 2017;205(Supplement C):1298–310. https://doi.org/10.1016/j.apenergy.2017.08.013.

[6] Petrichenko R, Baltputnis K, Sauhats A, Sobolevsky D. District heating demand short-term forecasting. In: 2017 IEEE international conference on environment and electrical engineering and 2017 IEEE industrial and commercial power systems Europe (EEEIC/I CPS Europe); 2017. p. 1–5. https://doi.org/10.1109/EEEIC.2017.7977633.

[7] Wang H, Wang G, Li G, Peng J, Liu Y. Deep belief network based deterministic and probabilistic wind speed forecasting approach. Appl Energy 2016;182: 80–93. https://doi.org/10.1016/j.apenergy.2016.08.108.

[8] Coelho I, Coelho V, Luz E, Ochi L, Guimares F, Rios E. A GPU deep learning metaheuristic based model for time series forecasting. Appl Energy 2017;201: 412–8. https://doi.org/10.1016/j.apenergy.2017.01.003.

[9] Fan C, Xiao F, Zhao Y. A short-term building cooling load prediction method using deep learning algorithms. Appl Energy 2017;195:222–33. https://doi.org/10.1016/j.apenergy.2017.03.064.

[10] Wang H-Z, Li G-Q, Wang G-B, Peng J-C, Jiang H, Liu Y-T. Deep learning based ensemble approach for probabilistic wind power forecasting. Appl Energy 2017;188:56–70. https://doi.org/10.1016/j.apenergy.2016.11.111.

[11] Kong X, Xu X, Yan Z, Chen S, Yang H, Han D. Deep learning hybrid method for islanding detection in distributed generation. Appl Energy 2018. https://doi.org/10.1016/j.apenergy.2017.08.014.

[12] Feng C, Cui M, Hodge B-M, Zhang J. A data-driven multi-model methodology with deep feature selection for short-term wind forecasting. Appl Energy 2017;190:1245–57. https://doi.org/10.1016/j.apenergy.2017.01.043.

[13] Lago J, De Ridder F, De Schutter B. Forecasting spot electricity prices: deep learning approaches and empirical comparison of traditional algorithms. Appl Energy 2018;221:386–405. https://doi.org/10.1016/j.apenergy.2018.02.069.

[14] Kawashima M, Dorgan C, Mitchell J. Hourly thermal load prediction for the next 24 hours by arima, ewma, lr and an artificial neural network. ASHRAE Trans 1995;101(1):186–200.

[15] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput 2006;18(7):1527–54. https://doi.org/10.1162/neco.2006.18.7.1527.

[16] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016. http://www.deeplearningbook.org/.

[17] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th international conference on neural information processing systems, NIPS'12. USA: Curran Associates Inc.; 2012. p. 1097–105. https://doi.org/10.1145/3065386.

[18] Hinton G, Deng L, Yu D, Dahl GE, Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. Signal Process Mag 2012;29(6):82–97. https://doi.org/10.1109/MSP.2012.2205597.

[19] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv eprint. 2014. arXiv:1409.0473.

[20] Chen T, Guestrin C. Xgboost: a scalable tree boosting system. In: Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. p. 785–94. https://doi.org/10.1145/2939672.2939785.

[21] Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms using different performance metrics. In: International conference on machine learning (ICML); 2005. p. 161–8. https://doi.org/10.1145/1143844.1143865.

[22] Caruana R, Karampatziakis N, Yessenalina A. An empirical evaluation of supervised learning in high dimensions. In: International conference on machine learning (ICML); 2008. p. 96–103. https://doi.org/10.1145/1390156.1390169.

[23] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Global Optim 1998;13(4):455–92. https://doi.org/10.1023/A:1008306431147.

[24] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. In: Advances in neural information processing systems; 2011. p. 2546–54. http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.

[25] Hutter F, Hoos HH, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. In: International conference on learning and intelligent optimization. Springer; 2011. p. 507–23. https://doi.org/10.1007/978-3-642-25566-3_40.

[26] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003;3:1157–82.

[27] Hutter F, Hoos H, Leyton-Brown K. An efficient approach for assessing hyperparameter importance. In: Proceedings of the 31st international conference on international conference on machine learning. Vol. 32 of ICML'14; 2014. p. 754–62. http://proceedings.mlr.press/v32/hutter14.pdf.

[28] Diebold FX, Mariano RS. Comparing predictive accuracy. J Bus Econ Stat 1995;13(3):253–63. https://doi.org/10.1080/07350015.1995.10524599.

[29] Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. Technometrics 1970;12(1):55–67. https://doi.org/10.2307/1267351.

[30] Tibshirani R. Regression shrinkage and selection via the lasso. J Roy Stat Soc B 1996;58(1):267–88.

[31] Chollet F. Keras. 2015. https://github.com/fchollet/keras.

[32] Theano Development Team. Theano: a Python framework for fast computation of mathematical expressions. arXiv eprint. 2016. arXiv:1605.02688.

[33] Bergstra J, Yamins D, Cox DD. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proceedings of the 30th international conference on machine learning; 2013. p. 115–23. http://proceedings.mlr.press/v28/bergstra13.pdf.

[34] Ziel F, Steinert R, Husmann S. Forecasting day ahead electricity spot prices: the impact of the EXAA to other European electricity markets. Energy Econ 2015;51:430–44. https://doi.org/10.1016/j.eneco.2015.08.005.

[35] Nowotarski J, Raviv E, Truck S, Weron R. An empirical comparison of alternative schemes for combining electricity spot price forecasts. Energy Econ 2014;46:395–412. https://doi.org/10.1016/j.eneco.2014.07.014.