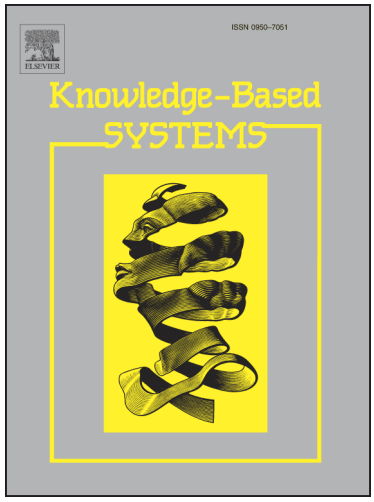
Accepted Manuscript



Correlation and instance based feature selection for electricity load forecasting

Irena Koprinska, Mashud Rana, Vassilios G. Agelidis

PII: S0950-7051(15)00071-4
DOI: http://dx.doi.org/10.1016/j.knosys.2015.02.017
Reference: KNOSYS 3078

To appear in: *Knowledge-Based Systems*

Received Date: 12 July 2014
Revised Date: 12 January 2015
Accepted Date: 21 February 2015

Please cite this article as: I. Koprinska, M. Rana, V.G. Agelidis, Correlation and instance based feature selection for electricity load forecasting, *Knowledge-Based Systems* (2015), doi: http://dx.doi.org/10.1016/j.knosys.2015.02.017

Correlation and Instance Based Feature Selection for Electricity Load Forecasting

Irena Koprinska[a], Mashud Rana[a], Vassilios G. Agelidis[b]

[a]School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia
email: {irena.koprinska, mashud.rana}@sydney.edu.au

[b]Australian Energy Research Institute, University of New South Wales, Sydney, NSW 2052, Australia
email: vassilios.agelidis@unsw.edu.au

Abstract

Appropriate feature (variable) selection is crucial for accurate forecasting. In this paper we consider the task of forecasting the future electricity load from a time series of previous electricity loads, recorded every 5 minutes. We propose a two-step approach that identifies a set of candidate features based on the data characteristics and then selects a subset of them using correlation and instance-based feature selection methods, applied in a systematic way. We evaluate the performance of four feature selection methods – one traditional (autocorrelation) and three advanced machine learning (mutual information, RReliefF and correlation-based), in conjunction with state-of-the-art prediction algorithms (neural networks, linear regression and model tree rules), using two years of Australian electricity load data. Our results show that all feature selection methods were able to identify small subsets of highly relevant features. The best two prediction models utilized instance and autocorrelation based feature selectors and an efficient neural network prediction algorithm. They were more accurate than advanced exponential smoothing prediction models, a typical industry model and other baselines used for comparison.

Keywords: electricity load forecasting, feature selection, autocorrelation, mutual information, linear regression, neural networks

1. Introduction

Forecasting the future electricity load is an important task in the management of modern energy systems. It is used to make decisions about the commitment of generators, setting reserve requirements for security and scheduling maintenance. Its goal is to ensure reliable electricity supply while minimising the operating cost.

Electricity load forecasting is classified into four types based on the forecasting horizon: long-term (years ahead), medium-term (months to a year ahead), short-term (1 day to weeks ahead) and very short-term (hours and minutes ahead). In this paper we consider Very Short-Term Load Forecasting (VSTLF), in particular 5 minutes ahead forecasting. VSTLF plays an important role in competitive energy markets such as the Australian national electricity market. It is used by the market operator to set the required demand and its price and by the market participants to prepare bids. The importance of VSTLF increases with the emergence of the smart grid technology as the demand response mechanism and the real time pricing require predictions at very short intervals [1].

Predicting the electricity load with high accuracy is a challenging task. The electricity load time series is complex and non-linear, with daily, weekly and annual cycles. It also contains random components due to fluctuations in the electricity usage of individual users, large industrial units with irregular hours of operation, special events and holidays and sudden weather changes.

Various approaches for VSTLF have been proposed; the most successful are based on Holt-Winters exponential smoothing and Autoregressive Integrated Moving Average (ARIMA) [2], Linear Regression (LR) and Neural Networks (NNs) trained with the backpropagation algorithm [3-7]. The problem of feature selection for VSTLF, however, has not received enough attention, and it is the focus of this paper.

Feature (variable) selection is the process of selecting a set of representative features (variables) that are relevant and sufficient for building a prediction model. It has been an active research area in machine learning [8-10]. Good feature selection improves the predictive accuracy, leads to faster training and smaller complexity of the prediction model. It is considered as one of the key factors for successful prediction.

Most of the existing approaches for VSTLF identify features in a non-systematic way or use standard autocorrelation analysis, which only captures linear dependencies between the predictor variables and the output variable that is predicted. The main goal of this paper is to show how advanced machine learning feature selection methods can be applied for electricity load forecasting, and more generally to energy time series forecasting. In particular, our contribution can be summarized as follows:

- We adapt and apply three advanced machine learning feature selection algorithms - Mutual Information (MI), RReliefF (RF) and Correlation-Based Selection (CFS) – to the task of load forecasting. We chose these methods as they are appropriate for the nature of the electricity load data - they can identify both linear and non-linear relationships (MI and RF) and capture both relevant and redundant features (CFS, RF), see Section 3. For comparison we also apply a method based on Autocorrelation (AC). We show how these feature selection methods can be applied in a systematic way to energy time series.
  - We propose a two-step approach for feature selection. In the first step we form a set of candidate features by applying a 1-week sliding window. A 1-week sliding window greatly reduces dimensionality while still capturing the main characteristics of data. In the second step we use a feature selection method to evaluate the quality of the candidate features and select a final subset of features.
  - We use the selected features with state-of-the-art prediction algorithms: NN, LR and Model Tree Rules (MTR). Hippert *et al.* [11] reviewed the application of NNs for electricity load forecasting and noted the need for systematic and fair comparison between NNs, standard linear statistical methods such as LR and other prediction algorithms.
- We conduct a comprehensive evaluation using two years of Australian electricity data. This includes a comparison with exponential smoothing (one of the most successful methods for load forecasting), a typical prediction model used by industry forecasters and several other benchmarks.
- We investigate additional aspects of the feature selection algorithms such as effect of the number of neighbors in AC and the number of features in MI and RF.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 analyses the data characteristics. Section 4 describes the proposed feature selection methods and how they were applied to our task. Section 5 presents the prediction algorithms we used and their parameters. Section 6 describes the methods used for comparison. Section 7 summarizes the experimental setup. Section 8 presents and discusses the results. Finally, Section 9 concludes the paper.

2. Previous Work

VSTLF is a relatively new area that has become important with the introduction of competitive electricity markets, and more recently, with the arrival of the smart grid. In contrast, short-term load forecasting has been widely studied, e.g. see [11-14].

There are two main groups of approaches for VSTLF: traditional statistical and computational intelligence. Prominent examples of the first group are exponential smoothing and ARIMA; these methods are linear and model-based. The most popular examples of the second group are NNs; for a survey on NNs for electricity load forecasting see [11]. NNs are attractive as they can model non-linear input/output relations and can learn them from a set of examples as opposed to the traditional statistical methods that fit a model and estimate its parameters.

One of the first studies on VSTLF was conducted by Liu *et al.* [3] who applied NN, fuzzy logic and autoregressive models to predict the load for every minute of a 30-minute forecasting horizon. They found that the NN and fuzzy rules were more accurate than the autoregressive models.

Charytoniuk and Chen [4] compared several NN-based methods for the prediction of 10-minute ahead electricity load using the load in the previous 20-90 minutes. They forecasted load differences instead of actual load. The best method achieved prediction error MAPE=0.4-1% and was implemented in a power utility in the United States, showing good accuracy and reliability.

Shamsollahi *et al.* [5] used a NN for 5-minute ahead electricity load forecasting. The data was processed by applying a logarithmic differencing of the consecutive loads; the NN architecture used 1 hidden layer and the stopping criterion was based on a validation set. They obtained an excellent MAPE=0.12% and the method was integrated into an energy market system for the region of New England in USA.

Chen and York [6] developed a complex hierarchical NN architecture for 15-minute ahead prediction. To predict the load for each day of the week, five NNs were used to cover different time intervals of the 24-hour period and their decisions were combined using another NN. They reported MAPE=0.28-0.87%.

Reis and Alves da Silva [15] predicted the load from 1 to 24 hours ahead using North American data. They first decompose the load series into several components using wavelet transform and then used NN-based approaches to make the prediction. The best approach achieved MAPE of 1.12% for 1-hour ahead prediction.

Taylor [2] used minute-by-minute British electricity load data to predict the load between 10 and 30 minutes ahead. He studied a number statistical methods based on ARIMA and exponential smoothing. Some of the methods ignored the seasonal patterns, others captured only the weekly cycle or both the daily and weekly cycles. The best forecasting method was an adaptation of the Holt-Winter's smoothing for double seasonality, achieving MAPE of about 0.4% for 30-minute ahead prediction; the best methods for 5-minute ahead prediction were double seasonal Holt-Winter's smoothing, restricted daily cycle smoothing and ARIMA, achieving MAPE of about 0.25%. In [16] Taylor, de Menezes and McSharry compared the performance of four methods for predicting the hourly demand for Rio de Janeiro from 1 to 24 hours ahead: ARIMA, double seasonal Holt-Winters exponential smoothing, NN and a regression method with principal component analysis. The simplest

method, exponential smoothing, was shown to be the most accurate.

In our previous work on 5-minute load forecasting [17] we applied autocorrelation analysis to extract and evaluate several nested feature sets of lag variables. The evaluation was limited to data for one month only. In [7] we used a larger dataset and constructed seasonal and yearly prediction models. We applied autocorrelation analysis to the whole training data, without a sliding window, and extracted 50 features. The most accurate prediction model was LR achieving MAPE=0.29%. We also found that there was no accuracy gain in building separate seasonal models in comparison to using a single model for the whole year. In this paper we extend our previous work by using a two-step feature selection process with a 1-week sliding window, applying and comprehensively evaluating the performance of a number of feature selection methods in addition to autocorrelation, and comparing the results with exponential smoothing and other baselines.

3. Data Analysis

We use electricity load data measured at 5-minute intervals for a period of two years: from 1st January 2006 until 31st December 2007. Each measurement represents the total electricity load for the state of New South Wales (NSW) in Australia. The data was provided by the Australian Electricity Market Operator (AEMO) [18].

In order to build accurate prediction models, it is important to understand the data characteristics and the external variables affecting the forecasting.

3.1 Data Characteristics

The electricity load data shows three main nested cycles: daily, weekly and yearly. These cycles are consistent with the human routine and the industrial and commercial activities.

Fig.1 plots the load for 2 consecutive weeks from our data. We can observe both the daily and weekly cycles.

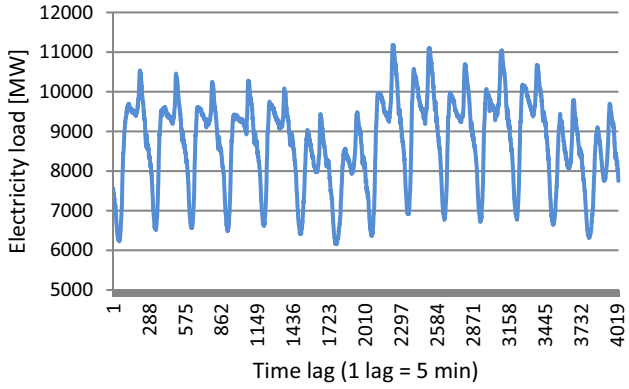


Fig. 1. Electricity load for two consecutive weeks (Monday, 1 May, to Sunday, 14 May, 2006)

The daily cycle is evident from the similarity of the load profiles of the individual days, e.g. the load profile of Monday is similar to the load profile of Tuesday, Wednesday and the other days. This is more clearly seen from Fig. 2 which shows the load for the different days of a single week.

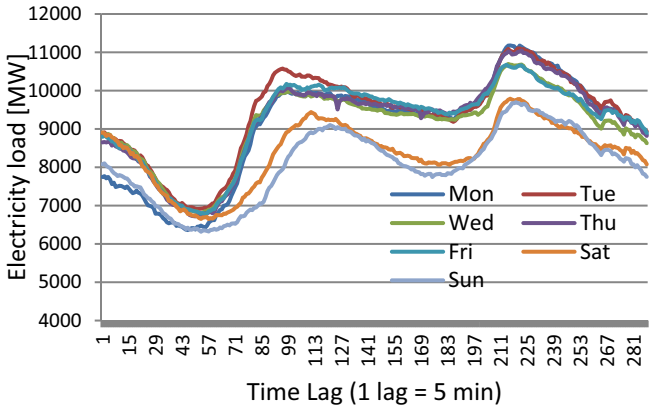


Fig. 2. Daily electricity load profiles for different days of a week

The weekly cycle is evident from the similarity of the load profiles of the same day of the week (e.g. the two Mondays in Fig.

1). We can also see that the load during the working days (Monday to Friday) is higher than the load during the weekend (Saturday and Sunday).

There is also a yearly pattern of the electricity load. For example, the load profile for 2006 is very similar to the load profile for 2007. This underpins the idea of building a yearly prediction model using the data for 2006 and then using this model to predict the load for 2007.

In this paper we capture the daily and weekly patterns with our feature selection methods and utilize the yearly pattern in the training of the prediction models.

3.2 Weather Variables

Weather variables such as temperature and humidity are relevant for electricity load forecasting in general but not for VSTLF. For small forecasting horizons such as 5-minute ahead, the weather changes are already captured in the load series. Prediction models using only previous load data have been shown to provide high accuracy and to outperform models that also use weather variables [2]. The use of weather variables was found to be beneficial for forecasting horizons beyond several hours [2, 19]. Hence, in this work, we do not consider weather variables; we only use previous load data.

3.3 Problem Statement

Given a time series of 5-minute electricity loads up to the time t, $X_1,...,X_t$, our goal is to predict the load at time $t+1$, X_{t+1}.

4. Feature Selection

Feature selection is the process of removing irrelevant and redundant features and selecting a small set of informative features that are necessary and sufficient for good prediction. Feature selection has been an active area of research in machine learning and statistics [8-10, 20]. Feature selection increases predictive accuracy by reducing overfiting and addressing the curse of dimensionality problem. It also affects the speed of the prediction algorithm – smaller feature set means faster training of the prediction model and faster forecasting of new data. Finally, it typically leads to a simpler prediction model (e.g. a regression function with smaller number of predictors or a decision tree with smaller size) and this in turn improves the understanding of the prediction model and results, and the user confidence in them. For this reasons, appropriate feature selection is one of the key factors for successful prediction.

A note on the terms “variable” and “feature” – in this paper we use them as synonyms. By definition, variables are the raw measurements while features are the inputs of the prediction model and a feature can be constructed using one or more variables. For example, in the industry feature set in Table 2 there are 11 variables (previous load values) and 10 features (each of them is a differences of two variables), as described in Sec. 6.2. In all other cases considered in the paper there is no difference between variables and features – they are electricity load values at a given time (lag variables).

4.1 Proposed Two-Step Feature Selection Approach

In order to select a set of relevant and informative lag variables, we propose a two-step approach: 1) forming an initial set of candidate variables by considering all variables (observation) from a 1 week previous data window and 2) selecting a subset of these variables using a feature selection method. Fig. 3 summarizes our approach.

For the first step we chose a 1-week window in order to capture the daily and weekly patterns in the electricity load data as discussed in Section I. As the data is recorded every 5 minutes, the candidate feature set contains 7*288=2016 lag variables.

In the second step we apply four feature selection methods: AC, MI, CFS and RF. There are important differences between these methods:

- AC is a traditional statistical method while the rest are advanced machine learning methods.
- AC, MI and CFS are correlation-based methods, while RF is an instance-based method.
- AC and CFS are able to detect only linear correlations while MI and RF can also detect non-linear correlations.
- CFS selects a subset of features explicitly while the other methods do not. Instead, they evaluate each feature individually and assign a score to it based on its importance. The user then decides which features should be included in the final feature subset, e.g. the N top ranked features based on the score.
- CFS and RF captures both the feature-to-feature and feature-to-output variable correlations, while AC and MI capture only the feature-to-output variable correlations. This means that CFS are RF are more suitable for identifying redundant variables – variables that are highly correlated or related to each other. All methods identify relevant variables – variables that are important to predict the output variable.

We now discuss each feature selection method in more details and how it was adapted and applied to our task.

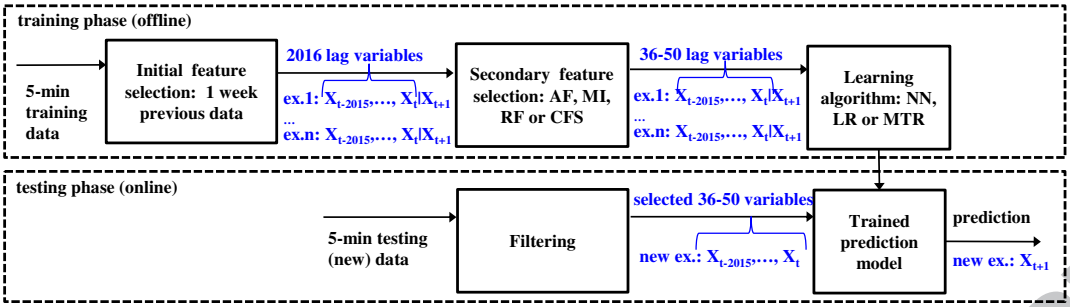


Fig. 3. Proposed approach

4.2 AC

Feature selection based on AC analysis is the most popular method for electricity load forecasting [7, 17, 21, 22]. The main idea is to compute the AC function and then select lag variables based on the strength of the identified correlations.

Let X_t be the value of a time series at time t and $\bar{X}$ be the mean value of all X in the given time series. The lag k autocorrelation coefficient r_k measures the linear correlation of the time series at times t and t-k:

$$r_k = r(X_t, X_{t-k}) = \frac{\sum_{t=k+1}^{n}(X_t - \bar{X})(X_{t-k} - \bar{X})}{\sum_{t=1}^{n}(X_t - \bar{X})^2}$$

Hence, r_1 measures the linear correlation, on average, between the X values that are 1 lag apart, r_2 - between the X values that are 2 lag apart, etc. The AC function contains all r_k values and it is used to identify cycles and patterns in a time series. Values close to 1 or -1 (i.e. spikes) indicate high positive or negative autocorrelation and values close to 0 indicate lack of autocorrelation.

A feature subset is formed by examining the AC function and selecting lag variables with high autocorrelation. For example, if there is a high correlation at lag 1, this means that X_t (the previous value) is a good predictor of X_{t+1}. More generally, if there is a high correlation at lags p and q, then X_{t-p+1} and X_{t-q+1}, the values p and q lags before $X_{t+1,}$ will be good predictors of X_{t+1}.

Fig. 4 shows the AC function for our training data (year 2006) for a 1-week data window, i.e. k = 2016 lags. We can see that there are several strong linear correlations. The strongest dependence is at lag 1 (i.e. values that are 1 lag apart), the second strongest dependence is at lag 2016 (i.e. values that are 1 week apart), the third strongest dependence is at lag 288 (i.e. values that are 1 day apart and so on. We can see that the AC graph reflects the weekly and daily patterns discussed in Section II and confirms their importance for feature selection for predicting the future electricity load. For example, to predict the future value X_{t+1}, the three strongest dependencies from the AC graph motivate the use of the previous value X_t, and the values at the same time 1 week ago (XD^7_{t+1}) and 1 day ago (XD^7_{t+1}), where XD^N_t means the load N days before the forecasting day at time t.

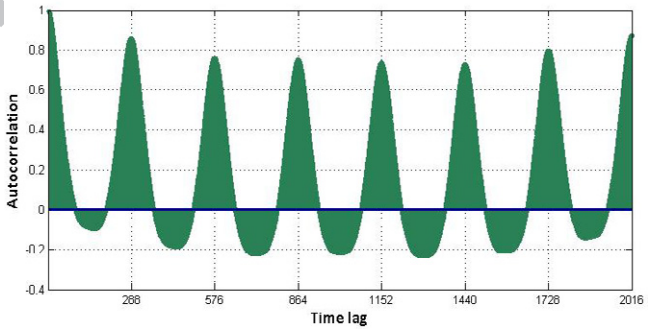


Fig. 4. Autocorrelation of training data

To form a feature subset, we extract lag variables from the seven highest peaks and their neighborhoods. The number of peaks and the size of the neighborhoods are selected empirically. The underlying principle for feature selection is that the higher the peak, the stronger the linear dependence is and, thus, the more informative the lag variable is. Hence, we extract more variables from the neighborhood of the higher peaks and smaller number of features from the lower peaks. Specifically, we extract the following 37 lag variables from the seven highest peaks:

- *from peak 1 (at lag 1; the highest peak):* the peak and the 10 lags before it (11 features); note that there are no lags after the highest peak;

- *from peak 2 (at lag 2016) and peak 3(at lag 288):* the peak and the three lags before and after it (7 features each);
- *from peaks 4 to 7 (at lag 1728, 576, 864, and 1152, respectively):* the peak and the surrounding 1 lag before and after it (3 features each).

This feature set is denoted by FS_{AC} and its features are listed in Table 1.

4.3 MI

MI is an information theoretic measure of the interdependence between two variables *X* and *Y*. If the two variables are independent, MI is zero; if they are dependent, MI has a positive value reflecting the strength of their dependency. It is a very suitable method for feature selection for electricity load data as it can capture both linear and non-linear correlations between the lag variables and the output variable.

In this paper, we apply a novel approach for estimating MI based on *k-nearest neighbor distances* [23]. It computes the MI between two variables without making any assumption about the underlying data distribution, and was shown to be efficient and more reliable than the traditional methods.

The MI between two random continuous variables *X* and *Y* with dimensionality *N* is estimated as:

$$MI(X,Y) = \psi(k) - \frac{1}{k} - \frac{1}{N}\sum_{i=1}^{N}[\ \psi(n_x(i)) + \psi\left(n_y(i)\right)]\ + \psi(N)$$

where $\psi(x)$ is the digamma function $\psi(x) = \Gamma(x)^{-1}d\Gamma(x)/dx$, k is the number of nearest neighbors (we used $k = 6$); $n_x(i)$ is the number of points x_j with a distance to x_i satisfying $\|x_i - x_j\| \le \epsilon_x(i)/2$ and $n_y(i)$ is the number of points y_j satisfying $\|y_i - y_j\| \le \epsilon_y(i)/2$, where $\epsilon_x(i)/2$ is the distance between x_i and its k-th neighbor in the *X* subspace and $\epsilon_y(i)/2$ is the distance between y_i and its k-th neighbor in the *Y* subspace.

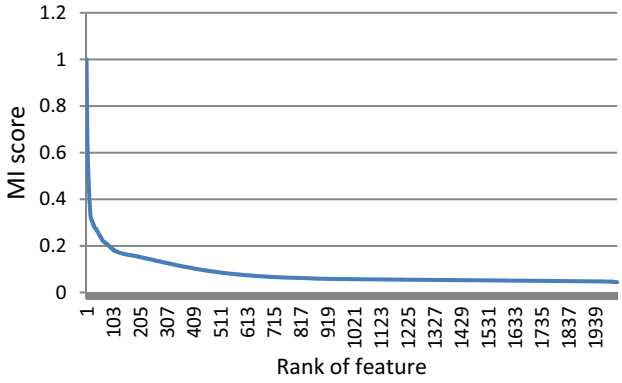


Fig. 5. MI score for each feature in ranked order

To conduct feature selection using MI, we first compute the MI score between each of the 2016 candidate lag variables and the output variable, and then rank the candidate lag variables in decreasing order based on their MI score. Fig. 5 plots the normalized MI scores for each variable in ranked order. We can make two important observations: 1) there is sharp drop of the MI score at about ranked variable 30 and 2) there is no significant improvement of the MI score after ranked variables 50 (the MI curve flattens gradually). Based on these observations, we can determine the cutoff point to be 50, and form the feature set FS_{MI} by selecting the top 50 ranked variables and disregarding the remaining variables. This results in about 97% feature reduction in comparison to the candidate feature set. The selected features are listed in Table 1.

Feature set	Number of features	Selected lag variables to predict X_{t+1}
FS_{AC}	37	X_{t-11} *to* X_t; XD^1_{t-3} *to* XD^1_{t+3}; XD^2_{t-1} *to* XD^2_{t+1}; XD^3_{t-1} *to* XD^3_{t+1}; XD^4_{t-1} *to* XD^4_{t+1}; XD^6_{t-1} *to* XD^6_{t+1}; XD^7_{t-3} *to* XD^7_{t+3}
FS_{MI}	50	X_{t-21} *to* X_t; XD^1_{t-5} *to* XD^1_{t+6}; XD^7_{t-9} *to* XD^7_{t+6}
FS_{CFS}	36	X_{t-16} *to* X_t; XD^1_{t-3} *to* XD^1_{t+3}; XD^2_{t+86}; XD^3_{t-29}; XD^5_{t-135}; XD^6_{t-35}; $XD^6_{t-20;}$ XD^7_{t-3} *to* XD^7_{t+3}
FS_{RF}	50	X_{t-6} *to* X_t; XD^1_{t-4} *to* XD^1_{t+4}; XD^2_{t-2} *to* XD^2_{t+1}; XD^3_{t-2} *to* XD^3_{t+2}; XD^4_{t-1} *to* XD^4_{t+1}; XD^5_{t-1} *to* XD^5_{t+1}; XD^6_{t-3} *to* XD^6_{t+3}; XD^7_{t-6} *to* XD^7_{t+5}
where: X_t – load on the forecasting day at time *t*; XD^N_t – load *N* days before the forecasting day at time *t*		

Table 1. Feature sets FS_{AC}, FS_{MI}, FS_{CFS} and FS_{RF}

4.4 CFS

CFS [24] is a state-of-the-art algorithm for feature subset selection. It explicitly produces a single subset of features, unlike MI and RF which rank all features individually and require a user input to form the final subset of features.

CFS's main idea is that a good feature subset should contain features that are highly correlated with the output variable (the variable that is being predicted) but are not correlated with each other. Given a set of candidate features, it uses a search algorithm to find the best possible feature subset *S*, the one that maximizes the following heuristic:

$$Merit_s = \frac{k\,\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

where k is the number of features in S, $\overline{r_{cf}}$ is the average correlation between each feature f in S and the output variable c, and $\overline{r_{ff}}$ is the average feature to feature pair-wise correlation between the features in S.

As a search algorithm we used best-first search with forward selection. Employing an exhaustive search algorithm is not possible as the feature set is too big, e.g. there are 2^{2016} possible subsets for our set of 2016 lag variables. Best-first search is a heuristic search algorithm. Forward best-first search starts with an empty subset and adds one feature at a time. At each step it considers the current feature subset and all possible subsets formed by an addition of a single feature to it. It also considers all previous subsets that are not on the current path, i.e. it considers backtracking to a promising previous state if the current path is not promising any more. The heuristic values for all these subsets are computed, the best subset is selected and accepted if it is an improvement over the previous state. The search continues from there until there is no improvement or more features to add.

We used a modified version of the CFS algorithm. When we applied the original CFS algorithm to the 2016 candidate lag variables, it returned a feature set containing only four lag variables: lag 1 (X_t), lag 288 (XD^1_{t+1}), lag 1230 (XD^4_{t+78}) and lag 2016 (XD^7_{t+1}). This feature set makes sense as it includes variables that capture the daily and weekly cycles (lag 1, lag 288 and lag 2016), which were also the three highly ranked variables by AC. Nevertheless, it was too small and did not perform well.

We modified the CFS selection process by changing its starting point. Specifically, instead of starting with an empty subset, CFS starts with an initial set of 20 useful lag variables (called mandatory variables) based on the correlation analysis: lag 1 and the 5 lags before it (X_t to X_{t-6}), lag 288 and the 3 lags before and after it (XD^1_{t-3} to XD^1_{t+3}), and lag 2016 and the 3 lags before and after it (XD^7_{t-3} *to* XD^7_{t+3}). All of these variables are highly correlated with the output variable, but inevitable some of them, e.g. the ones extracted from the same peak, are also correlated with each other. The idea was that this initial feature set will be beneficial due to the high feature-to-output correlations, and that CFS will be able to refine it by adding features that reduce the feature-to-feature correlations, producing a final feature set that as a whole balances the two criteria. CFS was run with this initial subset and returned a final feature subset called FS_{CFS} that contained 36 features, see Table 1. Sec. 8.4 compares the results of the original and modified CFS algorithms, and shows that the modification was beneficial. The final feature set FS_{CFS} represents a feature reduction of 98% over the candidate feature set.

4.5 RF

RF [25] is an instance-based feature ranking method from the Relief family of methods for feature selection applicable to both classification and forecasting tasks. The main algorithm Relief [26] is used for two-class classification problems. Its main idea is that high quality features should have different values for instances from different classes and similar values for instances from the same class. RF is an extension for regression problems, such as our task, where the predicted value is numeric and the concept of two instances belonging to the same or different classes needs to be adapted. RF uses a probability value that the two instances are from the same or different classes, modeled as the relative distance between the predicted values of these instances.

Specifically, RF assigns a weight w_f to each feature f based on how well this feature distinguishes between instances from the same and different classes. It works by randomly selecting an instance R from the training data and finding its nearest neighbor from the same class (nearest hit H) and the opposite class (nearest miss M). It then updates the weight w_f of each feature using the following equation where *diff* is the difference between two instances (computed as sum of differences over all attributes), normalized to [0,1] values:

$$w_f = w_f - \left(\frac{diff_f(R,H)}{m} - \frac{diff_f(R,M)}{m}\right)$$

As a result, the weight w_f is increased if the feature f differentiates between instances from different classes and has the same value for instances from the same class. The process is repeated for m randomly selected instances R.

The use of m randomly selected examples from the training data may lead to variations in the ranking of the features for different runs of the algorithm. To reduce these variations, we used all training examples instead of m examples only. This makes the algorithm deterministic and also increases the reliability of the feature weights. It is also possible to use more than one nearest neighbor; in our experiments we used $k = 10$ nearest neighbors.

RF is appropriate for the electricity load data as it works well on noisy and correlated features and can detect higher order pairwise feature interactions. It has a linear time complexity, thus it is an efficient algorithm

To determine the final subset of features we follow the same procedure as with MI. Fig. 6 shows the RF score (weight) and ranking of the candidate features computed by RF. We can see that the RF graph is similar to the MI graph – the RF score drops rapidly at the beginning, and then flattens from about ranked feature 250 to ranked feature 1500, before further decreasing. As in MI, we decided to choose again the 50 highly ranked features; the resulting feature set is called FS_{RF} and shown in Table 1.

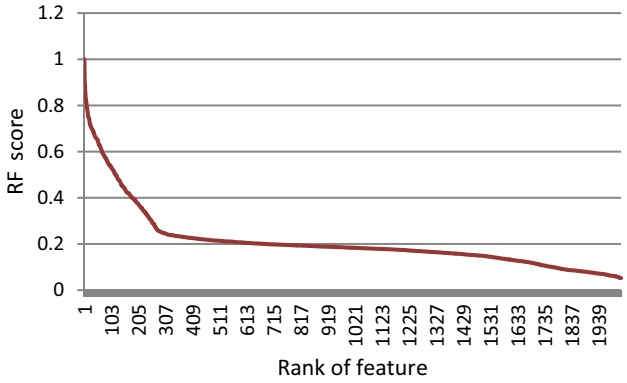


Fig. 6. RF score for each feature in ranked order

4.6 Comparison of Feature Sets

The features sets selected by the four algorithms are shown in Table 1. We can see that all sets capture the daily and weekly patterns as they include three main groups of features: 1) from the same day just before prediction time *t+1* (the previous 6-11 values), 2) from the previous day at *t+1* and the 3-6 surrounding values, and 3) from the previous week at *t+1* and the 3-6 surrounding values. In addition, AF and RF include features from the other previous days (2-6) at *t+1* and around this time. CFS also includes features from the other previous days but they are not from times around *t+1*, but from less expected times such as *t+86, t-29, t-135, t-20*. These features are added by the CFS algorithm to penalize the feature-to-feature correlation of the initial set of mandatory features and maximize the overall merit score.

5. Prediction Algorithms

We applied three state-of-the-art machine learning algorithms, representing different learning paradigms: NN, LR and MTR.

5.1 NN

NNs, in particular multi-layer perceptrons, are the most popular prediction algorithms for electricity load forecasting, used by both researchers and industry forecasters. They offer three main advantages in comparison to the traditional statistical models such as LR, ARIMA and exponential smoothing: 1) ability to model non-linear relationships between the predictor variables and the output variable, 2) ability to learn from examples and extract patterns, instead of making assumptions about the process that generated the data and fitting this model and 3) noise tolerance.

To develop a prediction model using NN, we used a multi-layer perceptron architecture with one hidden layer, trained with the Levenberg-Marquardt algorithm [27]. We chose the Levenberg-Marquardt algorithm over the standard steepest gradient descent backpropagation algorithm due to its faster convergence. The stopping criterion was: a maximum number of 2000 epochs is reached or there is no improvement in the error for 20 consecutive epochs. The number of hidden neurons was chosen using a validation set procedure as follows. We created *P* NNs, each with a different number of hidden neurons, from *1 to P*; these NNs were trained on the training set D_{train} and evaluated on the validation set $D_{valid,}$; the best NN architecture was selected and then evaluated on the testing set D_{test} and its performance is reported in the paper.

5.2 LR

LR is a classical statistical method for forecasting. LR assumes linear relationship between the predictor variables and the variable that is predicted, and uses the least square method to find the regression coefficients. In this work we applied Weka's implementation of linear regression [28]. It has an inbuilt mechanism for feature selecting based on the M5 method. This method firstly builds a regression model using all features and then removes the features, one by one, in decreasing order of their standardized coefficient until no improvement is observed in the prediction error given by the Akaike Information Criterion (AIC).

5.3 MTR

MTR [29] is a representative of a different prediction paradigm, tree and rule-based models. This type of models hasn't received enough attention in the area of electricity load forecasting. MTR generates a small set of rules that can be easily interpreted by people. This is an important advantage over the NN models, whose decisions are difficult to explain and trace

back.

MTR generates rules from model trees. Model trees [30] are similar to the traditional decision trees for predicting nominal values except that: 1) their leaves correspond to linear regression functions instead of discrete class labels and 2) the attribute tests at each of their node are selected to minimize the intra-subset variation in the numeric class values of the instances that go down each branch instead of minimizing an entropy-based measure.

MTR operates by iteratively generating model trees at each step and converting the best leaf into a rule, i.e. it uses a separate-and-conquer strategy. A model tree is induced from the training data and the best leaf is converted into a rule; the training examples covered by the rule are removed and the procedure is applied recursively to the remaining examples until all of them are covered. The resulting rules are comparable in terms of accuracy with the model tree but are smaller and hence easier to interpret and understand.

Decision trees and model trees can be seen as having an inbuilt mechanism for feature selection – only the attributes that appear in the tree are selected. However, the feature selection is local – at each step one attribute is selected, the best one for the current subset of examples, which may not result in an optimal feature set in terms of overall predictive accuracy. It has been shown that the predictive performance of decision trees improves with appropriate feature selection before building the tree, and that this also typically reduces the number of nodes and makes the tree more compact and easy to understand [31, 32].

6. Prediction Methods Used for Comparison

We compare the performance of our approach with four baselines, a typical industry model and three different versions of the exponential smoothing method. Exponential smoothing is one of the most popular and successful econometric methods used for electricity forecasting.

6.1 Baselines

We used the following naïve forecasting methods as baselines:

1) B_{mean}: *mean load value in the training data*. The prediction for X_{t+1} is given by the mean value of X_{t+1} in the training data.

2) B_{plag}: *load from the previous lag (i.e. 5 minutes before).* The prediction for X_{t+1} is given by X_t.

3) B_{pday}: *load from the previous day at the same time.* The prediction for X_{t+1} is given by XD^1_{t+1}.

4) B_{pweek}: *load from the previous week at the same time.* The prediction for X_{t+1} is given by XD^7_{t+1}.

6.2 Industry Model

A typical feature set used by industry practitioners is shown in Table 2; we will refer to this feature set as FS_{IND}. It utilises electricity loads from the same day and the previous week, which is consistent with the daily and weekly patterns we discussed in Section II. More specifically, to predict the load X_{t+1}, it uses the load from the previous 5 lags on the same day ($X_t, \ldots, X_{t-4}$) and also the load at the same time, one week ago (XD^7_{t+1}) and the previous 5 lags ($XD^7_t,\ldots,XD^7_{t-4}$). A natural logarithmic difference is applied to the successive loads similarly to [5] in order to improve data stationarity. The variable that is predicted is $ln(X_{t+1}/X_t)$ and needs to be transformed back to X_{t+1}. In contrast, all the other prediction approaches discussed in this paper, use untransformed previous loads and predict directly X_{t+1}.

The industry feature set is employed in conjunction with a multi-layer perceptron NN; we will call this combination the *industry model* and use it for comparison.

Table 2. Industry feature set FS_{IND}

Predict X_{t+1} by predicting $ln(X_{t+1}/X_t)$ and then transforming it to X_{t+1}
Features: $lnD(X_t),\ldots,lnD(X_{t-4})$ *(4 features)* $lnD\ (XD^7_{t+1}), \ldots, lnD\ (XD^7_{t-4})$ *(5 features)*
where: $lnD(X_t)=ln(X_t)-ln(X_{t-1})=ln(X_t/X_{t-1})$

6.3 Exponential Smoothing Methods

We also implemented three different versions of the exponential smoothing method. Exponential smoothing is a very popular econometric method for electricity forecasting that has been shown to be very successful [2, 33]. The predicted value is a weighed combination of the previous values, where the more recent values are weighed higher than the older. The parameters of the model are estimated using an optimization procedure which minimizes the mean squared error. The Holt-Winters exponential smoothing is an extension of the standard exponential smoothing for dealing with cyclic (seasonal) data as the electricity load data. It decomposes the data into trend and seasonal components. The standard version can only include one seasonality but extensions for more than one seasonality have been proposed in [2, 33]. Following Taylor [33], we applied three Holt-Winters exponential smoothing methods:

- HW_{daily} – single seasonality: within-day. This is a standard Holt-Winters method that uses a 24-period seasonal cycle.
- HW_{weekly} – single seasonality: within-week. This is a standard Holt-Winters method that uses a 168-period seasonal cycle.

•HW_{double} – double seasonality: within-day and within-week. This is a Holt-Winter method proposed by Taylor that uses both a 24-period cycle for the within-day seasonality and a 168 period cycle for the within-week seasonality.

The parameters of these methods were derived from the training data using a standard nonlinear optimization method that minimizes the sum of the squared 1-step-ahead forecasting errors. The derived parameters are shown in Table 3.

Table 3. Holt-Winters parameters (α – smoothing, γ – trend, δ - daily seasonality, ω - weekly seasonality, ψ - error adjustment)

	α	**γ**	**δ**	**ω**	**ψ**
HW_{daily}	0.001	0.0	0.13	-	0.996
HW_{weekly}	0.001	0.0	-	0.11	0.999
HW_{double}	0.001	0.0	0.13	0.10	0.998

7. Experimental Setup

7.1 Data

The available data is a time series of 5-minute electricity loads for two years, 2006 and 2007. The total number of samples is 210,240 (= 2 years x 365 days x 24 hours x 12 measurements). There were 272 missing data points (0.1% of all data) that were replaced with the average of the previous 3 load values. The data has been normalized between -1 and 1. For our prediction task, one example is a 2016-dimensional feature vector after the initial feature selection and a 35-50-dimensional vector after the secondary feature selection, depending on the feature selection algorithm used, see Fig. 3.

The data is divided into 3 non-overlapping subsets: *training set (D_{train}), validation set (D_{valid}) and testing set (D_{test})*. D_{train} contains the first 70% of the data for 2006, D_{valid} contains the remaining 30% of the data for 2006 and D_{test} contains the data for 2007. All prediction models used $D_{train} + D_{valid}$ for feature selection and D_{test} for evaluating the predictive accuracy. LR and MTR use $D_{train} + D_{valid}$ for building of the prediction model, and NN uses D_{train} for building of the prediction model and D_{valid} for selecting the best NN architecture.

7.2 De- seasonalization

Our data is seasonal as there are periodic recurring patterns, e.g. daily and weekly cycles. Some standard statistical forecasting methods apply de-seasoning as a preprocessing step. This includes estimating and removing the seasonal component by differencing or division. This adjustment is based on the assumption that the seasonal fluctuations may dominate the other variations in the time series and make accurate prediction more difficult [34]. There is no consensus about the usefulness of de-seasoning, e.g. [35] recommends to use unadjusted data. Machine learning methods such as NNs are considered to be able to detect seasonality directly and even detect changing seasonal patterns, without the need for prior de-seasonalization [36]. However, [34] found that NNs also can benefit from prior de-seasonalization and de-trending. In this work we didn't apply de-seasoning or de-trending prior to the forecasting but this is an interesting direction for further investigation.

7.2 Evaluation Measures

To evaluate the accuracy of the prediction models, we use two standard performance measures: Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|X_i - \hat{X}_i|$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - \hat{X}_i}{X_i}\right| 100\%$$

where X_i and $\hat{X}_i$ are the actual and predicted load at lag i, respectively, and n is the total number of predicted loads.

MAPE can be seen as a normalized version of MAE where the normalization is achieved by a simple division by X_i. It is the preferred measure used by industry practitioners.

8. Results and Discussion

Table 4 shows the performance of the four proposed feature sets with NN, LR and MTR. Table 5 shows the performance of the baselines and the methods used for comparison.

Table 4. Performance of the proposed feature sets. "Rank" is the ranking of the feature set, for each prediction model, and is based on MAPE and statistical testing.

Feature Set	Metrics	Prediction Model		
		NN	LR	MTR
FS_{AC}	MAE [MW]	24.50	25.25	30.64
	MAPE [%]	0.279	0.288	0.349
	Time [s]	77	3	50
	Rank:	1	1	1
FS_{MI}	MAE [MW]	24.78	25.70	30.50
	MAPE [%]	0.282	0.293	0.347
	Time [s]	87	6	58
	Rank:	2	2	1
FS_{RF}	MAE [MW]	24.43	25.12	30.70
	MAPE [%]	0.278	0.287	0.350
	Time [s]	91	7	60
	Rank:	1	1	1
FS_{CFS}	MAE [MW]	24.71	25.94	30.46
	MAPE [%]	0.281	0.296	0.347
	Time [s]	70	3	47
	Rank:	2	3	1

8.1 Comparison of the Proposed Feature Sets

The main results of the comparison between FS_{AC}, FS_{MI}, FS_{RF} and FS_{CFS} can be summarized as follows:

- All four feature sets achieved good predictive accuracy (Table 4) – their prediction errors MAPE are between 0.282% and 0.350%, considerably lower than the MAPE of the three baselines (Table 5).
- Overall, the best performing feature sets are FS_{RF} and FS_{AC}, followed by FS_{CFS} and finally FS_{MI}. This conclusion is drawn after considering the statistical significance of the differences in accuracy between the four feature sets, for a given algorithm, see below, and calculating an overall ranking for each feature set.
- A comparison of the feature sets, for the same algorithm, shows that FS_{AC}, FS_{MI}, FS_{RF} and FS_{CFS} obtained similar predictive accuracy, e.g. the MAPEs are: 0.278-0.282% for NN, 0.287-0.296% for LR and 0.347-0.350% for MTR. To further investigate this, we conducted a statistical testing using the Wilcoxon rank-sum test. The results showed that: (i) for NN there were only two statistically significant differences in accuracy: FS_{RF} vs FS_{MI} and FS_{RF} vs FS_{CFS}, (ii) for LR all differences were statistically significant except FS_{RF} vs FS_{AC} and (iii) for MTR all differences were not statistically significant. Based on this, we ranked the features sets for each algorithm based on MAPE as shown in Table 4 under "rank". For example, for NN, FS_{RF} and FS_{AC} are ranked equally first and FS_{MI} and FS_{AC} are ranked equally second. Then we calculate the rank for each feature set, the total ranking scores and the final ranking of the feature sets: FS_{RF} and FS_{AC} (total ranking score = 3), FS_{MI} (total ranking score = 5) and FS_{CFS} (total ranking score = 6).
- The time required to extract the features was as follows: about a few seconds for AC, a minute for MI, seven minutes for CFS and 48 hours for RRelifF. Thus, for applications that require frequent re-running of the feature selection algorithm and retraining of the prediction model to adapt to highly variable load characteristics, RRelifF may not be a good choice. In our case, the feature selection is done once every year and all feature selection algorithms are suitable.

Table 5. Performance of baselines and methods used for comparison

Prediction Method	MAE [MW]	MAPE [%]
HW_{daily}	26.23	0.301
HW_{weekly}	28.40	0.324
HW_{double}	25.82	0.295
Industry model	27.58	0.301
B_{mean}	1159.42	13.484
B_{plag}	41.24	0.473
B_{pday}	453.89	5.046
B_{pweek}	451.03	4.940

8.2 Comparison with the Industry Feature Set

Table 6 shows the results of the industry feature set FS_{IND} when used with NN, LR and MTR. We can observe that in terms of MAPE the industry feature set is most accurate in conjunction with NN, MAPE=0.301%. Recall that the combination

FS_{IND}+NN is called the industry model as it is employed by industry forecasters. Hence, our results confirm that NN is a suitable prediction algorithm for FS_{IND}. However, the performance of FS_{IND}+NN is considerably lower than the best performing prediction model FS_{RF}+NN, MAPE=0.278%, and in fact considerably lower than any other feature set when used with NN or LR. Hence, the industry feature set FS_{IND} is outperformed by the proposed feature sets FS_{AC}, FS_{MI}, FS_{RF} and FS_{CFS}.

A possible reason for the relatively poor performance of FS_{IND} is its smaller number of features - 9 in comparison to 36-50 for the other feature sets. These features only partially capture the weekly and daily patterns of the electricity load; we can see this by examining the AC graph in Fig. 4 and by comparing FS_{IND} with FS_{AC}.

Table 6. Performance of FS_{IND}

Feature Set	Metrics	Prediction Model		
		NN	LR	MTR
FS_{IND}	MAE [MW]	27.58	27.87	27.08
	MAPE [%]	0.301	0.318	0.309
	Time [s]	59	0.5	42

8.3 Comparison of Prediction Models

Fig. 7 shows the MAPE values of all prediction models and the benchmark methods, for visual comparison. The main results can be summarized as follows:

- The most accurate prediction models are FS_{RF}+NN and FS_{AC}+NN, achieving MAPE of 0.278-0.279%. They considerably outperformed all methods used for comparison - the three exponential smoothing methods, the industry model and the four baselines.
- For all four proposed feature sets, NN is the best algorithm, followed by LR and then by MTR. All differences between prediction models that use the same feature set and a different algorithm (e.g. FS_{RF}+NN, FS_{RF}+LR and FS_{RF}+MTR, etc.) are statistically significant at $p<0.05$ (Wilcoxon rank-sum test for statistical significance).
- While the predictive accuracy of NN and LR is similar ($MAPE_{NN}$=0.278-0.282% and $MAPE_{LR}$=0.287-0.296%), the accuracy of MTR is considerably lower ($MAPE_{MTR}$=0.347-0.350%), and lower than some of the baselines (the exponential smoothing and industry model). However, as expected, MTR generated a small number of compact rules. In fact it produced only one rule in all four cases, involving between 7 and 16 variables - 11 for FS_{AC}, 16 for FS_{MI}, 7 for FS_{RF} and 13 FS_{CFS}. Hence, although MTR did not perform accurately in our case, it is a good option for generating compact prediction models that are easy to interpret by practitioners.
- Among the methods used for comparison the best one was the double seasonal exponential smoothing HW_{double}, followed by HW_{daily} and the industry model (the same accuracy), and finally HW_{weekly}. The exponential smoothing results are also consistent with Taylor [33] who found that the double seasonality outperformed the weekly and daily seasonality models.
- The four baselines B_{mean}, B_{pday}, B_{pweek} and B_{plag} were the least accurate methods. Their accuracy was significantly lower than the accuracy of the other methods. For example, in terms of MAPE, the best prediction models FS_{RF}+NN and FS_{AC}+NN outperformed B_{mean} with a factor of 49, B_{pday} and B_{pweek} with a factor of 18 and B_{plag} with a factor of 2.
- The time required to train the prediction models is shown in Table 4 and was 77-91 seconds for NN, 3-7 seconds for LR and 47-60 seconds for MTR. Thus, all prediction models are fast to train and suitable for practical applications, even if there is a need for frequent retraining.

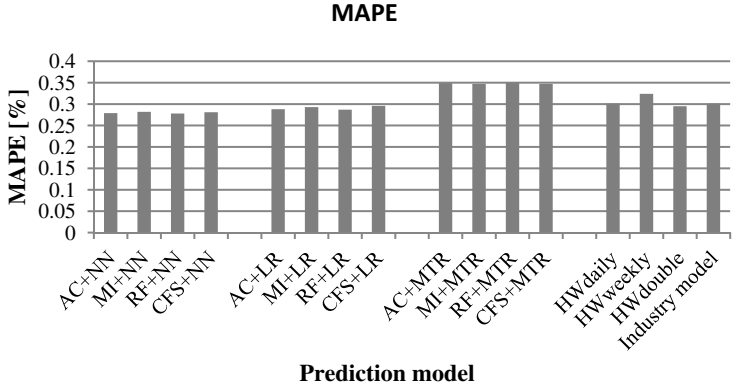


Fig. 7. Comparison of prediction models – MAPE
(The baselines B_{mean}, B_{plag}, B_{pday} and B_{pweek} are not shown as their MAPEs are disproportionaly higher)

8.4 Additional Investigation of Feature Selection Methods

1) Standard vs Modified CFS method

Fig. 8 compares the performance of the standard and modified CFS methods. Recall from Sec. 4.4 that the standard CFS selects only 4 features while the modified CFS starts with an initial set of 20 mandatory features that is further expanded to a final set of 36 features. We can see that the modified CFS considerably outperforms the standard CFS for all prediction algorithms.

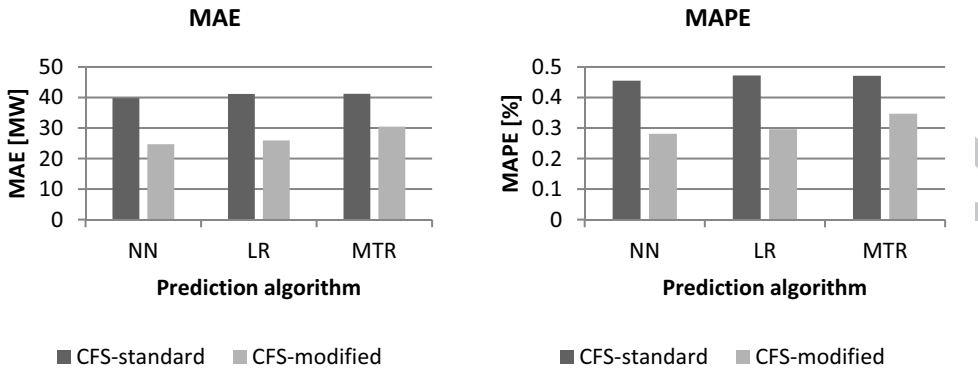


Fig. 8. Comparison of the standard and modified CFS feature sets – MAE and MAPE

2) AC – Effect of the Neighbors

In contrast to CFS, AF considers only feature-to-output variable correlations and ignores the feature-to-feature correlations. As discussed in Sec. 4.2, to form the feature set we extract features from the seven highest autocorrelation peaks and also from their neighbors. The neighbors are not only highly correlated with the output value but also with their corresponding peak. This raises an interesting question – are the neighbors redundant or contributing additional information? To answer this, we formed four nested feature sets, as described in Table 7, and evaluated their performance. Set 1 contains the lag variables corresponding to the 7 highest autocorrelation peaks only, without any neighbors. Sets 2, 3 and 4 contain the variables from Set 1 and also an increasing number of lag variables corresponding to the neighbors. Set 4 is the feature set FS_{AC} presented in Sec. 4.2.

Table 7. AC feature sets with increasing number of features extracted from the neighbors

Sets	Features extracted from:			Total number of features
	peak 1	peaks 2-3 each	peaks 4-7 each	
Set 1	1	1	1	7
Set 2	5	3	1	15
Set 3	9	5	3	31
Set 4	11	7	3	37

Fig. 9 shows the MAE and MAPE results for these four feature sets and all prediction algorithms. We can see that as the number of features increases, the accuracy improves. This improvement is biggest between Set 1 and 2, and smaller between the other sets. This means that the neighbors contribute additional information, although they are highly correlated with their corresponding peak. A possible reason for this is that all variables together capture temporal patterns that are useful for prediction. Thus, all variables together, provide significant performance improvement, with the prediction algorithms we consider.

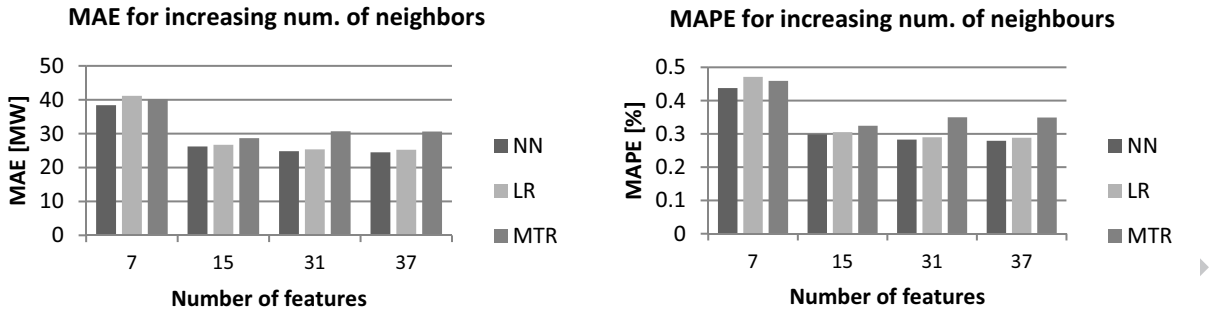


Fig. 9. Comparison of AC feature sets with increasing number of neighbors – MAE and MAPE

3) MI and RF – Effect of the Number of Features

The MI and RF feature selection methods do not explicitly select a feature subset. Instead, they evaluate each feature separately and assign a score to it, then all features are sorted based on the score and the ones that fall below a user specified threshold are discarded. To determine the threshold, a standard approach is to plot the graph showing the score values in decreasing order (as in Fig. 5 and 6) and select the *n* features with highest scores, before the graph flattens. In our case after visual inspection of the MI and RF graphs we selected the top 50 features. This approach doesn't guarantee finding the optimal subset but is fast and produces reasonable results.

To further investigate the effect of the number of features, we formed six feature subsets by selecting the top ranked 10, 20, 30, 40, 50 and 60 features. Fig. 10 and 11 show the results for MI and RF, respectively. For NN and LR, we can see that the accuracy improves as the number of features increases, but after 40 features this improvement is very small.

For MTR with MI, the accuracy almost doesn't change as the number of features increases. A further analysis showed that while MTR with the smallest feature set (10) struggled to produce a compact set of rules (it generated many rules using all features), MTR with the remaining five feature sets (20-60) generated only one rule with 13 features, which was the same or very similar for all cases. For MTR with RF, the highest accuracy is achieved with the smallest number of features (10). For all six sets, one different rule was generated involving 5-8 features, which is a very compact representation with a considerable further feature reduction. Although LR also has an inbuilt feature selection mechanism, its effect is very small – it further removed only 3-10% of all features, compared to 25-86% for MTR. Hence, MTR is less sensitive to the feature selection threshold compared to LR and NN.

In summary, the performance is sensitive to the feature selection threshold and different algorithms are affected in different ways. In our case, a threshold of 50 features based on visual inspection of the MI and RF graphs provided good overall results. The performance can be further optimized for a given algorithm by considering the trade-off between the number of features and accuracy and using the visual inspection threshold as a starting point to search for better subsets.

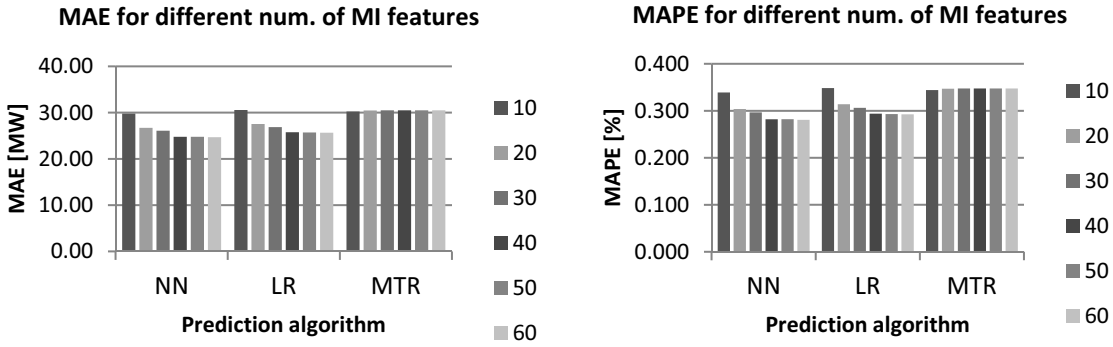


Fig. 10. Comparison of MI feature sets with 10-60 features – MAE and MAPE

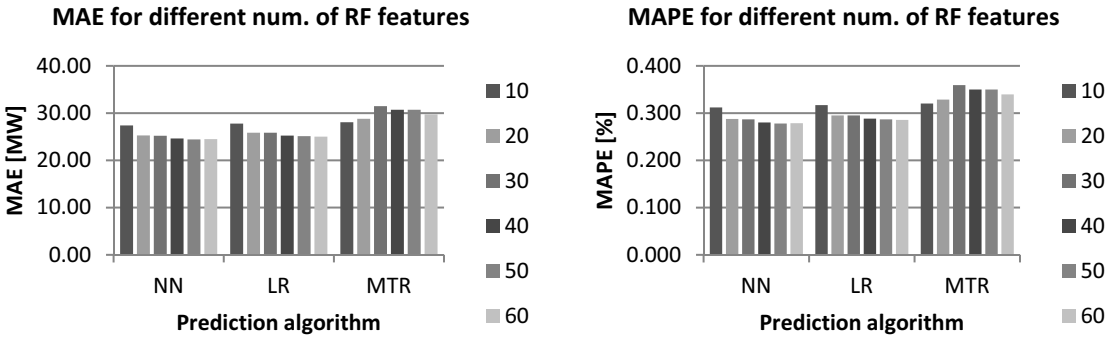


Fig. 11. Comparison of RF feature sets with 10-60 features – MAE and MAPE

9 Conclusions

We considered the task of predicting the electricity load one step ahead from a time series of previous electricity loads measured every 5 minutes. We evaluated the performance of four feature selection methods – three advanced machine learning (MI, RF and CFS) and one traditional statistical method (AC). These methods differ in the type of relationships they detect (both linear and non-linear), ability to capture relationships between features and the generation of the feature subset (explicit or ranking-based). The feature selection methods were used as a part of an efficient two-step process: identifying candidate features using a 1-week sliding window and then selecting a subset of them by applying the feature selection methods. The selected features were evaluated in conjunction with state-of-the-art prediction algorithms – NN, LR and MTR, using two years of Australian electricity data.

Our results showed that all feature selection methods were able to identify subsets of highly relevant features. The number of selected features was 36-50, which is about 2% of all features in the 1-week sliding window. The best prediction models were RF+NN and AC+NN, achieving a low prediction error MAPE of 0.278-0.279%. They were considerably more accurate than the three Holt-Winters exponential smoothing methods, the industry model and the four baselines used for comparison. The other two feature sets, CFS and MI, also performed very well with NN, and are a viable alternative. The best performing algorithm was NN, followed by LR and MTR. While LR performed similarly to NN, MTR was considerably less accurate although it produced very compact rules.

The proposed feature selection approach is generic and can be applied to load forecasting with different forecasting horizons and other energy time series tasks, e.g. forecasting solar and wind power, electricity prices and smart meter data. An interesting area for future work is selecting the best feature set for each test example as suggested in [37]. For VSTLF tasks this will require a fast and efficient feature selection method, and may further improve the accuracy. Another avenue for future work is investigating the performance of Radial-Basis Function (RBF) neural networks as an alternative to backpropagation neural networks. RBF networks are typically as accurate and noise tolerant as backpropagation neural networks but are faster to train [38]; they have been recently successfully applied for solar power forecasting [39] and multivariate time series forecasting [40].

References

[1] S. C. Chan, K. M. Tsui, H. C. Wu, Y. Hou, Y.-C. Wu, and F. F. Wu, Load/price forecasting and managing demand response for smart grids, IEEE Signal Processing Magazine (2012) 68-85.
[2] J. W. Taylor, An evaluation of methods for very short-term load forecasting using minite-by-minute British data," International Journal of Forecasting 24 (2008) 645-658.
[3] K. Liu, S. Subbarayan, R. R. Shoults, M. T. Manry, C. Kwan, F. L. Lewis, and J. Naccarino, Comparison of very short-term load forecasting techniques, IEEE Trans. Power Systems 11 (1996) 877-882.
[4] W. Charytoniuk and M.-S. Chen, "Very short-term load forecasting using artificial neutal networks, IEEE Trans. Power Systems 15 (2000) 263-268.
[5] P. Shamsollahi, K. W. Cheung, Q. Chen, and E. H. Germain, A neural network based very short term load forecaster for the interim ISO New England electricity market system,in: Proceedings of the 22nd IEEE PES International Conference on Power Industry Computer Applications (PICA), 2001, pp. 217-222.
[6] D. Chen and M. York, Neural network based very short term load prediction, in: Proceedings of the IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, Pittsburg, 2008.
[7] I. Koprinska, M. Rana, and V. G. Agelidis, Yearly and Seasonal Models for Electricity Load Forecasting, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2011, pp. 1474-1481.
[8] R. Kohavi and G. H. John, Wrappers for feature selection, Artificial Intelligence 97 273-324.
[9] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157-1182.
[10] L. Yu and H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5 (2004) 1205-1224.
[11] H. S. Hippert, C. E. Pedreira, and R. C. Souza, Neural Networks for short-term load forecasting: a review and evaluation, IEEE Transactions on Power Systems 16 (2001) 44-55.
[12] E. A. Feinberg and D. Genethliou, Load forecasting, in: Applied Mathematics for Restructured Electric Power Systems: Optimization, Control and Computational Intelligence, Springer, 2005, pp. 269-285.

[13] S. Fan and R. J. Hyndman, Short-term load forecasting based on a semi-parametric additive model, IEEE Transactions on Power Systems 27 (2012) 134-141.
[14] F. Martínez-Álvarez, A. Troncoso, J. C. Riquelme, and J. S. Aguilar-Ruiz, Energy time series forecasting based on pattern sequence similarity, IEEE Transactions on Knowledge and Data Engineering 23 (2011) 1230-1243.
[15] A. J. R. Reis and A. P. A. d. Silva, Feature extraction via multiresolution analysis for short-term load forecasting, IEEE Transactions on Power Systems 20, 189-198.
[16] J. W. Taylor, L. M. De Menezes, and P. E. McSharry, A comparison of univariate methods for forecasting electricity demand up to a day ahead, International Journal of Forecasting 22 1-16.
[17] I. Koprinska, R. Sood, and V. G. Agelidis, Variable selection for five-minute ahead electricity load forecasting, in: Proceedings of the 20th International Conference on Pattern Recognition (ICPR), 2010.
[18] AEMO. Australian Energy Market Operator. Available: www.aemo.com.au
[19] J. W. Taylor, Short-term load forecasting with exponentially weighted methods, IEEE Transactions on Power Systems 27 (2012) 458-464.
[20] A. Miller, Subset Selection in Regression, 2nd ed.: Chapman & Hall/CRC, 2002.
[21] S. Crone and N. Kourentzes, Input-variable specification for neural networks - an analysis of forecasting low and high time series frequency, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2009.
[22] G. A. Darbellay and M. Slama, "Forecasting the short-term demand for electricity - Do neural networks stand a better chance?, International Journal of Forecasting 16 (2000) 71-83.
[23] A. Kraskov, H. Stögbauer, and P. Grassberger, Estimating Mutual Information, Physical Review E 69 (2004) 1-16.
[24] M. A. Hall, Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning, in: Proceedings of the International Conference on Machine Learning (ICML), 2000, pp. 359-366.
[25] M. Robnik-Sikonja and I. Kononenko, Theoretical and Empirical Analysis of ReliefF and RReliefF,Machine Learning 53 (2003) 23-69.
[26] K. Kira and L. Rendell, A practical approach to feature selection, in: Proceedings of the 9th International Conference on Machine Learning (ICML), 1992.
[27] H. Yu and B. M. Wilamowski, Levenberg–Marquardt Training, in: Industrial Electronics Handbook. vol. 5 – Intelligent Systems, ed: CRC Press, 2011, pp. 12-1--12-15.
[28] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed.: Morgan Kaufmann, 2011.
[29] G. Holmes, M. Hall, and E. Frank, Generating rule sets from model trees, in: Proceedings of the Australian Joint Conference on Artificial Intelligence, 1999.
[30] Y. Wang and I. Witten, Inducing model trees for continuous classes, in: Proceedings of the European Conference on Machine Learning (ECML), Prague, Czech Republic, 1997.
[31] P. Perner, Improving the accuracy of decision tree induction by feature preselection, Applied Artificial Intelligence 15 (2011), 747-760.
[32] L. Rokach and O. Maimon, Data Mining with Decision Trees: Theory and Applications, 2nd ed.: World Scientific Publishing, 2005.
[33] J. W. Taylor, Short-term electricity demand forecasting using double seasonal exponential smoothing, Journal of Operational Research Society 54 (2003) 799-805.
[34] G. P. Zhang and M. Qi, Neural network forecasting for seasonal and trend time series, European Journal of Operational Research 160 (2005) 501-514.
[35] J. G. D. Gooijer and P. H. Franses, Forecasting and seasonality, International Journal of Forecasting 13 (1997) 303-305.
[36] P. H. Franses and G. Draisma, Recognising changing seasonal patterns using artificial neural networks, Journal of Econometrics 81 (1997) 273-280.
[37] G. Dudek, Variable selection in the kernel regression based short-term load forecasting model, in: Proceedings of the International Conference on Artificial Intelligence and Soft Computing (ICAISC), 2012.
[38] F. Feger and I. Koprinska, Co-training using RBF nets and different feature splits," in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2006.
[39] C. Chen, S. Duan, T. Cai, and B. Liu, Online 24-h solar power forecasting based on weather type classification using artificial neural networks, Solar Energy 85 (2011) 2856-2870.
[40] D. Chen and W. Han, Prediction of multivariate chaotic time series via radial basis function neural network, Complexity 18 (2013) 55-66.