Modeling functional outliers for high frequency time series forecasting with neural networks: an empirical evaluation for electricity load data

Nikolaos Kourentzes

Abstract— **This paper discusses and empirically evaluates alternative methodologies in modeling functional outliers for high frequency time series forecasting. In spite of several modeling and forecasting methodologies that have been proposed, there have been limited advancements in monitoring and automatically identifying outlying patterns and even less in modeling those for such times series. This is a significant gap considering the difficulty and the cost associated with manual exploration and treatment of such data, due to the vast number of observations. This study proposes and assesses the performance of different modeling methodologies focusing on two key aspects, the accuracy that the outliers are modeled and the impact of each methodology on modeling normal observations. The evaluated methodologies model functional outliers using binary, integer or trigonometric dummy variables, outlier profiles or isolate them into new time series and forecast them separately. Neural networks are employed to produce the forecasts, taking advantage of their flexible nature to accommodate the different methodologies and their superior performance in high frequency time series forecasting. Hourly electricity load data from the UK are used to empirically evaluate the performance of the different methodologies.**

Keywords: functional outliers, neural networks, multilayer perceptron, forecasting, electricity load.

I. Introduction

FORECASTS of electricity load data are required for a large variety of applications, such as trading electricity and scheduling production. In the forecasting literature such data are considered high frequency time series, where the data are collected and predicted in hourly or shorter time buckets. Although there is no strict definition of what constitutes a high frequency time series, in practice such time series are collected in daily or smaller time buckets and have vast amounts of data [1], introducing new issues in data handling, analysis and modeling. Use of conventional statistical modeling, designed for low frequency time series becomes problematic in these cases [2]. In the electricity load forecasting research several modeling methodologies for time series that exhibit these properties have been proposed [3], [4], [5], [6]; however, there have been limited advancements in both data monitoring and automatic outlier identification as well as modeling and automatic treatment of such outliers.

This is an important gap as time series models often require data cleaning, which involves modifying or removing outliers and obvious errors in the database [7], implicitly assuming that this information is a) available, which in fact requires costly manual collection and b) the analyst has a methodology to tackle outliers. Not cleaning the data can have substantial effects on model specification and parameters [7]. Outliers will introduce forecasting errors, as they do not follow the normal data generating process and they will also bias the model parameters, resulting in poor fit of the model to the data. In the literature, Taylor et al. acknowledges this issue and removes such days altogether [6], while Conejo et al. try to automatically correct outliers using conventional time series modeling approaches, but do not manage to improve the results [3], due to the high frequency nature of the data.

High frequency time series, and particularly electricity load data, typically exhibit periodic behavior. A new type of outlier appears in this family of time series. Whole periods may exhibit outlying behavior. For example electricity demand may be different throughout the day during a bank holiday in comparison to the normal demand profile. Such outliers can be analyzed as functional outliers [8]. In this case, we are interested in analyzing data providing information about curves, surfaces, etc as a whole varying over time. Such outliers can differ both in level, like normal outliers, but also in shape over the duration of a fixed period. To identify functional outliers there are different approaches, based on functional box- and bagplots [9], [10], time series clustering and classifications methodologies [11] belonging to a broader group of outlier detection research using unsupervised, supervised and semi-supervised learning algorithms, such as k-means, self-organizing maps, MLP networks, etc [12], [13], [14].

Once the outliers are known, one has to decide how to model them. In the case of electricity load forecasting, neural networks have shown good forecasting performance and is common to divide the time series into simpler ones and model those [15]. For instance, break the initial hourly time series into 24 new time series, one for each hour of the day. The functional outliers are now broken down to normal outliers, which can be modeled following conventional approaches [16]. However, [17] showed that this approach can lead to substantial loss of accuracy and increase the sensitivity to modeling decisions, concluding that it is preferable to forecast the complete time series, retaining all its dynamics. In this case one cannot avoid but model the functional outliers. In the context of time series forecasting

Nikolaos Kourentzes is with the Department of Management Science at Lancaster University Management School, Lancaster, LA1 4YX, United Kingdom. (email: n.kourentzes@lancaster.ac.uk).

there has been no focused research on the topic of modeling functional outliers for time series forecasting.

The contribution of this paper is to propose and evaluate a series of methodologies to model functional outliers on high frequency time series, specifically on hourly electricity load. These methodologies are based on both novel approaches and extensions of already existing conventional outlier modeling methods. In this paper the performance of each method is demonstrated and compared against a benchmark control model. This research concludes that using a trigonometric coding of the functional outliers results in the highest forecasting accuracy, while being robust to the stochasticity in the training of the neural networks.

The rest of this paper is organized as follows: Section II provides details of the proposed methodologies. In section III the experimental setup is described, while Section IV presents the empirical evaluation results. Section V concludes.

II. Methods

A. Multilayer Perceptrons for Time Series Forecasting

In this study multilayer perceptrons (MLP) are employed, which represent the most widely employed NN architecture [18], [19]. These have been well researched and have proven abilities in time series prediction and universal approximation [20]. They are able to approximate and generalize any linear or nonlinear functional relationship to any degree of accuracy without any prior assumptions about the underlying data generating process, providing a powerful forecasting method for linear or non-linear, non-parametric, data driven modeling [21], [22]. MLPs can be used to model time series in a univariate forecasting framework, using as inputs only time lagged observations of the time series, to predict the future values modeling nonlinear autoregressive NAR(p)-processes. Additional intervention variables and covariates can be used to capture additional information, modeling NARX(p)-processes. Data are presented to the network as disjunct vectors of a sliding window over the time series history. MLPs are organized in layers; the input layer, any number of hidden layers and the output layer that provides the predicted values for the time series. In forecasting applications one hidden layer is found to be adequate in most cases [23], which is also used here. The neural network learns the underlying data generating process by adjusting its connection weights $\mathbf{w} = (\beta, \gamma)$, minimizing an objective function on the training data, typically a squared error loss. Let $Y = (y_t)$ be the time series that needs to be predicted, with $t = (1, \ldots, T)$ observations and $X = (x_{tk})$ an array of $k = (1, \ldots, K)$ input variables, which can be lagged observations of either the time series or external variables. The predicted value $\hat{y}$ of the time series, one step ahead from time t using single hidden layer MLPs is:

$$\hat{y}_{t+1} = \beta_0 + \sum_{h=1}^{H} \beta_h g \left(\gamma_{h0} + \sum_{k=1}^{K} \gamma_{hk} x_{tk} \right), \tag{1}$$

where $\beta = (\beta_1, \ldots, \beta_H)$, $\gamma = (\gamma_{11}, \ldots, \gamma_{HK})$ are the weights for the output and the hidden layer respectively. The β_0 and γ_{h0} are the biases of each neuron. The hidden nodes use a nonlinear transfer function $g(\cdot)$, which is usually either the sigmoid logistic or the hyperbolic tangent function. The modeler must choose the appropriate data pre-processing, the number of hidden nodes, the transfer function within nodes, the training algorithm and the cost function of the MLP. An adequate MLP architecture is routinely determined by running simulations on the time series; a set of candidate MLPs is trained using different architectural parameters and the architecture which shows the lowest in-sample error is selected. We provide further details in section III, where the experimental setup is discussed.

B. Methodologies for Modeling Functional Outliers

While conventional outliers are classified as additive or innovative outliers, requiring particular modeling in each case [7], functional outliers are not distinguished in separate classes [9]. In this study different alternative methodologies to model functional outliers are proposed. These are based on extensions of conventional outlier modeling or novel approaches, taking advantage of the unique nature of functional outliers in the context of time series forecasting. For all these methodologies it is assumed that the data generating process of normal observations is captured adequately and the outliers are already labeled as such.

1) Single Binary Dummy Variable: In conventional linear regression modeling persisting effects on the level of a time series or additive outliers can be captured by using a single indicator dummy variable I. Given a time series y_t following a process z_t such an event can be modeled as:

$$y_t = z_t + \omega I[t = \tau], \tag{2}$$

where ω is the size of the shift or the additive outlier and $I[t = \tau]$ is the dummy variable taking a value of 1 when $t = \tau$, i.e. there is an outlier, and a value of zero otherwise. The process z_t is uncontaminated by outliers, but unobserved [24]. Although linear regression, as it is apparent from (2) always shifts z_t by the same amount ω, MLPs were shown to be able to output for the same binary indicator variable several different values for y_t [25]. Based on this finding, a MLP should be able capture the shape of a functional outlier using a single binary dummy variable that is equal to 1 when it is occurring and zero otherwise.

2) Multiple Binary Dummy Variables: Given that a functional outlier lasts for several observations S, one can employ several binary dummy variables to code each of its observations $s = 1, \ldots, S$ with a different shift from the unobserved underlying process z_t equal to ω_s. In the linear regression context one would be required to use S different binary variables, where each I_s would be equal to one if the observation is the s^{th} value of a functional outlier and zero otherwise. If a constant term is assumed outside of the process z_t then $S{-}1$ binary dummies can be used instead. Assuming that the differences of the functional outlier and the normal observations are not significant for each s in S one could remove the corresponding indicator binary dummies for these periods, thus reducing the degrees of freedom of the model and simplifying its estimation. This process can be automated through the use of stepwise regression, where insignificant indicator variables are automatically dropped from the final model. In the context of neural networks the same principles can be applied directly, however there is higher incentive to remove insignificant indicators from the model, given that each variable increases the degrees of freedom of the model by H, i.e. the number of the hidden nodes. Also, due to the multiple bias terms $\gamma_{hk'}$ it is not straightforward to decide a-priori whether all S or $S-1$ binary dummies should be considered, where k' refers to the indicator of the binary dummy input variables.

3) Single Integer Dummy Variable: Capitalizing on the nonlinear mapping capabilities of neural networks one can use a single integer dummy variable to code the functional outliers. Such variable increases monotonically from 1 to S when there is a functional outlier and is zero otherwise. This coding resembles a sawtooth waveform. A similar technique has been employed to model deterministic seasonality with MLPs [19], [26]. The authors show that following this approach neural networks are able to capture deterministic seasonality in time series, although this approach underperforms in comparison to other methodologies. However, a key advantage of this approach is the minimum increase in the model's degrees of freedom, making it easier to train the MLPs.

4) Profile Dummy Variable: Instead of letting the neural network identify the nonlinear mapping between the integer dummy discussed above and the deviations of the functional outlier from z_t one could assist the network by providing an archetypal pattern that the functional outliers follow. Thus, by estimating the average profile of the functional outliers a dummy variable is constructed that is equal to the profile when there is an outlier and zero otherwise.

5) Trigonometric Dummy Variables: In modeling repeating patterns instead of using multiple indicator variables one can equivalently use trigonometric variables. This has been widely used in modeling seasonal time series with regression models, particularly for the case of deterministic seasonality [27]. When using MLPs it has been shown that due to their approximation capabilities only a pair of trigonometric variables (a single sine and a single cosine) can be used with minimal loss of fitting and predictive accuracy [19], [26]; yet reducing the additional degrees of freedom to only two for any longer seasonal periodicity. A analogous approach can be used to code functional outliers. The network is given a pair of sine and cosine with wavelength S, i.e. the duration of the functional outlier, when one is occurring and zero in other cases.

6) Model Separately: Instead of providing additional information and indicator variables to the MLP regarding the functional outliers, one can separate all the outliers in a new time series and replace those in the original time series with normal observations, based on the approximated process z_t. Although the modeling of the original time series is greatly simplified as there as no outliers, two separate time series have to be predicted. The newly constructed time series, containing all the functional outliers, can be relatively short in comparison to the original time series, leading to estimation issues. Once both time series are forecasted, the predicted functional outliers are replaced on the predicted series of z_t, resulting in a single series that both outlying and normal observations are predicted.

III. Experimental Design

A. Data

To assess the performance of the aforementioned functional outlier modeling methodologies electricity load time series data from the UK are used. These are sampled at hourly intervals, from the 1^{st} of April 2001 01:00 until the 1^{st} of November 2008 01:00, amounting to 66505 hourly observations or 2771 days. Figure 1 provides a plot of the first 3000 hourly observations. The time series exhibits triple seasonality, a daily, a weekly and an annual pattern.

The time series contains two leap years, 2004 and 2008, distorting the annual periodicity. The load profile of each day exhibits three distinct patterns, associated to winter, summer and transitional consumption profiles. This is demonstrated in figure 2, where data only for Thursday's are plotted. To avoid cluttering the figure, transitional profiles are plotted together with summer days. Finally, UK uses daylight saving, translated into one moving date day in either March or April having 23 hours and another moving date day in October having 25 hours every year.

Using the methodology outlined in [14] 63 functional outliers (or 1512 hourly observations) are identified, reflecting unusual electricity load profiles. These are illustrated in figure 3. Using these outliers the different methodologies discussed in section II will be applied and evaluated. A large number of these outliers can be explained by calendar events, such as bank holidays and are not connected to exogenous variables such as temperature, which is not used in this study.

The time series is split into three subsets for training the MLPs and evaluating their performance. The test set is years

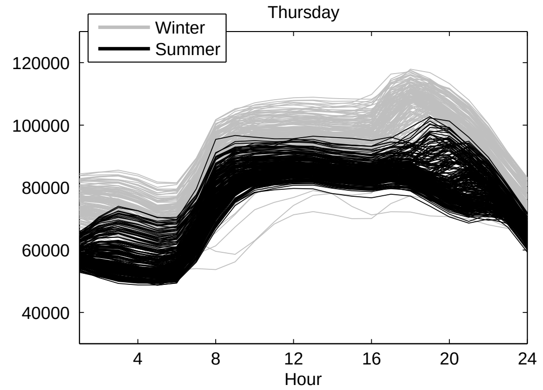

Fig. 2. Summer and winter consumption profiles for Thursday.

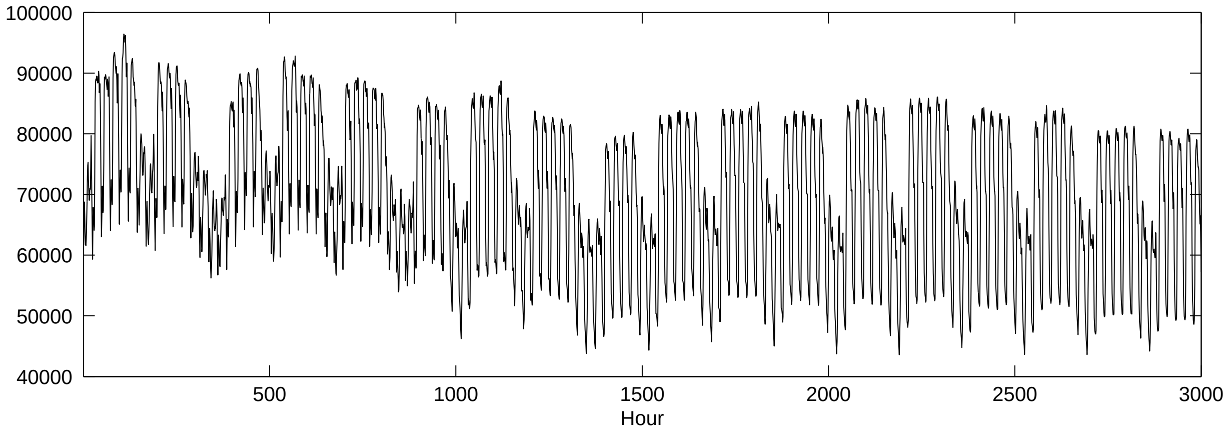

Fig. 1. Plot of the first 2880 observations of the time series.

2007 and 2008 (16080 observations). The validation set has equal number of observations and the training set is the remaining part of the time series.

B. Methods

A single MLP setup is used to model the time series under all methodologies. All model parameters are kept fixed with the exception of the input variables that change per methodology. The network has a single hidden layer. The number of hidden nodes is identified using a grid search from 5 to 40 hidden nodes with a step of 5. The search is stopped at 40 due to computational resources restrictions, as the network is trained using 34,345 observations. Using 40 nodes provides the best performance. All hidden nodes use the hyperbolic tangent function. The lagged observations of the time series are linearly scaled between $[-0.5, 0.5]$, before being inputted to the network. The networks are trained using the Levenberg-Marquardt algorithm, which requires setting the μ_{LM} and its increase and decrease steps. Here $\mu_{LM} = 10^{-3}$, with an increase step of $\mu_{inc} = 10$ and a decrease step of $\mu_{dec} = 10^{-1}$. For a detailed description of the algorithm and the parameters see [28]. The maximum training epochs are set to 1000. Mean squared error is used as a training cost function and is recorded for both training and validation sets. The training can stop earlier if μ_{LM} becomes equal of greater than $\mu_{max} = 10^{10}$ or the validation error increases for more than 25 epochs. This is done to avoid over-fitting. When training is stopped the network weights that give the lowest validation error are used. The limit of a 1000 training epochs is not reached in any of the simulations, due to the early stopping criterion. Each MLP is initialized 30 times with randomized starting weights to accommodate the nonlinear optimization. Once training is finished, the network initialization that exhibits the lowest error on validation set is chosen.

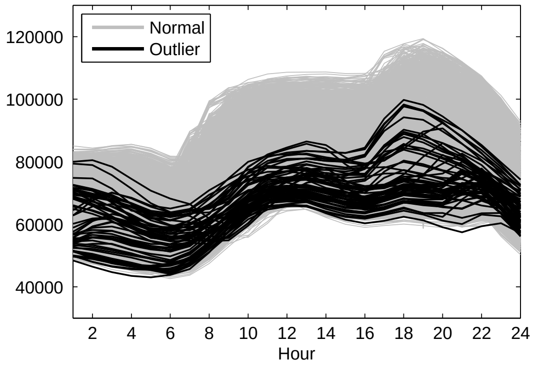

Fig. 3. Identified functional outliers.

In total, ten different sets of inputs are considered. First a $Control$ set of autoregressive lagged inputs is identified using stepwise regression. The high frequency nature of the time series (hourly observations) makes it challenging to automatically choose the relevant input variables, as most of the automatic input identification methodologies have been developed for lower frequency time series and either are associated with prohibitive computational cost or will not produce valid results [2], [19]. In [19] and [29] an extensive empirical evaluation of alternative input variables selection methodologies, on low and high frequency time series, showed that stepwise regression is a robust method to select variables for MLPs, superior to different forms of autocorrelation and partial autocorrelation analysis, selection by mutual information criterion, spectral analysis and random field regression. Based on this finding, regression is used in this study to identify the relevant autoregressive inputs. The resulting $Control$ set of input variables uses 36 lags, $X_{Control} = \{y_{t-1}, y_{t-2}, y_{t-3}, y_{t-5}, y_{t-6}, y_{t-8}, y_{t-9}, y_{t-10}, y_{t-11}, y_{t-12}, y_{t-13}, y_{t-15}, y_{t-16}, y_{t-17}, y_{t-19}, y_{t-20}, y_{t-21}, y_{t-22}, y_{t-23}, y_{t-24}, y_{t-48}, y_{t-72}, y_{t-120}, y_{t-144}, y_{t-164}, y_{t-165}, y_{t-166}, y_{t-167}, y_{t-168}, y_{t-169}, y_{t-170}, y_{t-171}, y_{t-172}, y_{t-8736}, y_{t-8760}, y_{t-8784}\}$. For this set of inputs the outliers are not modeled and will be used as a benchmark in assessing how much the proposed methodologies increase the accuracy of predicting both functional outliers and normal observations.

The first methodology employs a single binary dummy variable as described in section II and is named $Binary(1)$. The second methodology uses multiple binary dummies and four variants are created; $Binary(S)$ uses an equal number of dummy variables to the length of the functional outliers, in this case 24, while $Binary(S-1)$ uses one less. $Binary(Step)$ and $Binary(Back)$ use stepwise and backward regression to identify the number of useful binary dummy variables, starting from $Binary(S)$, resulting in 4 and 7 inputs respectively. Note that in this case stepwise and forward regression resulted in the same selection of variables; hence only one is evaluated here. The next methodology uses a single integer dummy variable and is named $Integer$, followed by $Profile$ that codes an archetypal functional outlier profile in a single dummy variable, which is scaled between -1 and 1. Methodology $SinCos$ uses a pair of trigonometric dummy variables. Finally the time series is separated into two, one for normal observations and one for outliers. To replace the outliers in the normal time series the following technique is used. First, the seasonal component of the time series is identified, through time series decomposition. Then the neighboring four seasons (of the highest frequency, daily in this case), two before and two after the functional outlier if available, are averaged to construct a local average profile, which is used to recreate a set of normal observations to replace the outlier. Replacing the outliers instead of removing them allows retaining the dynamics of the time series. A second time series is created and forecasted separately from all functional outliers. The forecasts of the latter series are used to replace the forecasts of the constructed local average profiles. The methodology is named $Replace$ and is the most complex, requiring to forecast two separate time series. To identify the input variables for the outlier time series stepwise regression is used, resulting in the following inputs: $X_{Replace} = \{y_{t-1}, y_{t-2}, y_{t-15}, y_{t-23}, y_{t-24}, y_{t-166}, y_{t-168}, y_{t-169}, y_{t-172}\}$. Note that all these methodologies use $Control$ to capture the underlying structure of the time series. Table I summarizes the described methodologies.

TABLE I

FUNCTIONAL OUTLIER MODELING METHODOLOGIES.

Methodology	No. of Inputs
$Control$	36
$Binary(1)$	37
$Binary(S)$	60
$Binary(S-1)$	59
$Binary(Step)$	40
$Binary(Back)$	43
$Integer$	37
$Profile$	37
$SinCos$	38
$Replace$	36, 9

C. Experimental Setup

The different methodologies are evaluated on forecasting the next 24 hours, considering the aggregate error from $t+1$ to $t+24$. Rolling origin evaluation is used, i.e. from each observation a trace of 24 consecutive forecasts is produced. This evaluation methodology has several advantages over the commonly employed fixed origin, where only a single out-of-sample measurement is done, collecting several error measurements, thus providing a richer and more reliable distribution of errors. For a discussion of rolling origin evaluation and its advantages see [30]. The forecasting accuracy is measured in Mean Absolute Percentage Error (MAPE) that is $MAPE = \sum_{t=1}^{h} (|y_t - f_t|/y_t)$, where y_t is the actual and f_t is the forecast at time t and h is the forecast horizon. MAPE is preferred to the commonly used MSE being more robust [31], but also due to its relevance to practice. Due to the high positive values of the time series none of the key issues of this metric are relevant here, while enjoying its very intuitive interpretation. For a detailed discussion on the choice of error measures for forecasting purposes see [30]. Once the MAPE has been calculated for each forecast origin it is average across origins to produce a single figure for each training, validation and test subsets. Furthermore, the MAPE is measured across all types of observations, only normal ones and only outlying ones. This allows tracking the performance of each methodology in improving forecasting accuracy in either normal or outlying observations.

IV. RESULTS

Table II provides the MAPE results for the best MLP training initialization for each methodology and in brackets the mean MLP across all initializations. The results are broken down in three categories, All, $Normal$ and $Outlier$ showing the errors filtered by the type of observation. The lowest error by column is highlighted in boldface. Any results worse than the benchmark $Control$ are underlined.

We will focus on the results after initialisation selection. First the results across all observations are analyzed. All methods outperform significantly the $Control$ in both training and validation sets. The $Replace$ methodology ranks first, while the remaining approaches follow with small differences among themselves. In the test set $SinCos$ is performing the best, closely followed by $Replace$. Note that $Binary(1)$ and $Binary(Step)$ fail to outperform the benchmark $Control$, pointing to over-fitting in the training set and poor generalization. It can also be observed that using a large number of binary dummy variables is preferable, with $Binary(S)$ being more accurate than $Binary(S-1)$, followed by $Binary(Back)$ and lastly by $Binary(Step)$. Both $Integer$ and $Profile$ that use only one additional, non-binary, dummy variable, perform better than the benchmark, with $Profile$ being best.

Looking at the performance only across normal observations we can see the impact of modeling adequately the outliers on the model coefficients and consequently on how well each methodology fits and predicts the time series. The $Replace$ methodology significantly outperforms all other approaches across all training, validation and test sets. In training and validation sets all methods perform better than the $Control$ benchmark, however once the test set is considered, all but $Replace$, $SinCos$ and $Profile$

TABLE II
MAPE RESULTS

Methodology	Trn	Val	Tst
All			
Control	1.83% (1.92%)	1.91% (1.98%)	1.92% (2.10%)
Binary(1)	1.72% (1.74%)	1.73% (1.83%)	1.93% (1.97%)
Binary(S)	1.60% (**1.70%**)	1.71% (**1.83%**)	1.86% (**1.96%**)
Binary(S-1)	1.64% (1.78%)	1.73% (1.90%)	1.86% (2.06%)
Binary(Step)	1.75% (1.89%)	1.80% (1.96%)	1.96% (2.10%)
Binary(Back)	1.73% (1.85%)	1.80% (1.93%)	1.89% (2.07%)
Integer	1.66% (1.74%)	1.71% (1.85%)	1.91% (1.97%)
Profile	1.74% (1.76%)	1.77% (1.86%)	1.86% (1.99%)
SinCos	1.70% (1.88%)	1.73% (1.93%)	**1.80%** (2.06%)
Replace	**1.49%** (1.81%)	**1.70%** (2.05%)	1.82% (2.08%)
Normal			
Control	1.70% (1.76%)	1.78% (1.86%)	1.78% (1.96%)
Binary(1)	1.68% (1.70%)	1.68% (1.78%)	1.88% (1.92%)
Binary(S)	1.59% (1.67%)	1.67% (1.77%)	1.82% (1.91%)
Binary(S-1)	1.62% (1.74%)	1.68% (1.83%)	1.82% (2.01%)
Binary(Step)	1.66% (1.76%)	1.72% (1.86%)	1.87% (1.98%
Binary(Back)	1.64% (1.75%)	1.73% (1.85%)	1.82% (1.98%
Integer	1.62% (1.68%)	1.66% (1.77%)	1.88% (1.91%
Profile	1.66% (1.70%)	1.68% (1.79%)	1.77% (1.93%
SinCos	1.65% (1.80%)	1.67% (1.85%)	1.75% (1.98%
Replace	**1.37%** (**1.57%**)	**1.59%** (**1.72%**)	**1.69%** (**1.81%**
Outlier			
Control	7.76% (8.78%)	7.75% (7.43%)	8.75% (9.01%
Binary(1)	3.21% (3.53%)	3.78% (**4.00%**)	4.51% (**4.18%**
Binary(S)	**1.98%** (**3.09%**)	**3.51%** (4.38%)	3.78% (4.43%
Binary(S-1)	2.38% (3.47%)	4.03% (4.79%)	3.77% (4.89%
Binary(Step)	5.76% (7.23%)	5.49% (6.13%)	6.17% (7.63%
Binary(Back)	5.52% (6.18%)	5.20% (5.49%)	5.65% (6.32%
Integer	3.54% (4.39%)	4.14% (5.10%)	**3.72%** (4.91%
Profile	4.97% (4.28%)	5.74% (4.93%)	6.16% (4.91%
SinCos	4.04% (5.11%)	4.33% (5.18%)	4.31% (5.63%
Replace	6.52% (11.40%)	6.53% (17.60%)	8.37% (15.50

Best MLP initialization error. Values in bracket are mean MAPE over a initializations. The lowest error in each column is in boldface.

have higher errors in comparison to $Control$ demonstrati again lack of generalization. Similar to the ranking for a observations, $SinCos$ is marginally better than $Profil$ Considering the accuracy of the methods only for the func onal outliers all methods outperform the benchmark signi cantly. $Binary(1)$, $Binary(S)$, $Binary(S-1)$, $Integ$ and $SinCos$ perform very well with errors in all se lower than 5%, while $Control$'s errors are around 8%. I sample $Binary(S)$ performs the best, however we can see significant degradation of accuracy between the training ar the validation and tests sets, implying potential over-fittin Note that this model has the highest degrees of freedor as it can be seen in table I. The same behavior to a less extend can be observed for $Binary(S-1)$ and $Binary(1$ $Integer$ gives the lowest test set error, demonstrating th the MLP can accurately map the functional outlier profi using a simple sawtooth waveform input, while there no evidence of over-fitting. $SinCos$ results in marginal higher errors, again with no evidence of over-fitting. $Profi$ has mediocre performance that degrades to the test set ar $Replace$, though still better than the $Control$ benchmar has poor performance. The latter can be explained by tl relatively small size of the outlier time series (1512 hour observations) and how erratic these are.

Across both normal values and outliers $SinCos$ has consistently good performance resulting in a superior overall accuracy. $Replace$ is the best method to predict the normal observations, but fails to improve the accuracy of outliers significantly. Due to the high number of normal observations (more than 97% of all values), its poor performance on outliers is masked when considering the aggregate accuracy across all observations, however it should be avoided, unless the objective is to focus only on the normal observations. $Profile$ performs in both cases better than $Control$, but worse than $SinCos$, while all other methods should be avoided as they lead to inferior performance in predicting the normal observation in comparison to the benchmark $Control$.

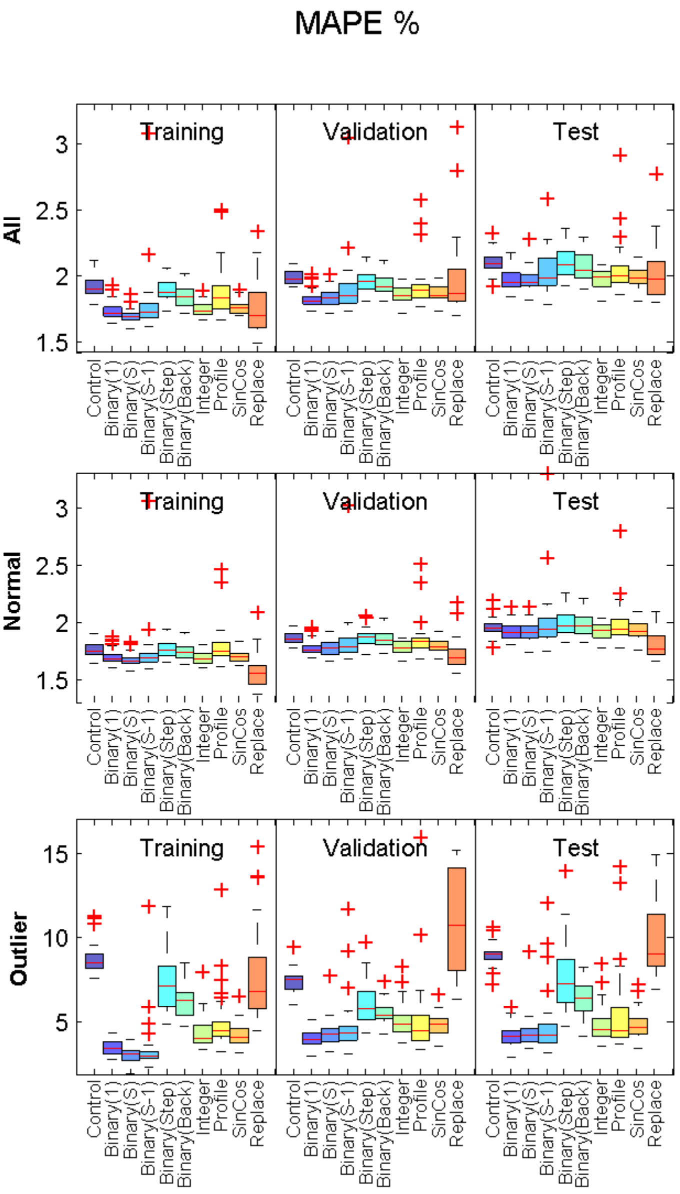

Fig. 4. Summer and winter consumption profiles for Thursday.

Figure 4 provides MAPE errors for each method across all initializations. This allows us to evaluate the stability and robustness of each method, reflecting the mean error provided in table II. The rankings of the models are in agreement with table II, though it is easier to assess the significance of the differences. Considering the results across all observations (1^{st} row in figure 4) we can see that the main body of the distributions of all methods but $Binary(Step)$ and $Binary(Back)$ and in some cases $Binary(S-1)$ and $Replace$, are well below the distribution of $Control$, indicating significant differences. Furthermore, across errors for normal and outlying observations the contrast in the performance of $Replace$ is clearly shown, as well as its wide distribution, implying more variability in the results, i.e. less robustness, in contrast to the other methods. On the other hand, note that with the exception of $Binary(Step)$, $Binary(Back)$ and $Replace$ the other methods result in tight distributions, meaning that the different training initializations resulted in similar accuracy, i.e. the methods are not sensitive to the initial training weights of the MLP.

Overall, $SinCos$ is the best compromise in performance across both normal and outlying observations, while $Replace$ has significantly superior accuracy in modeling the normal values.

V. Conclusions

In this paper alternative methodologies to model functional outliers with neural networks in high frequency time series are proposed in the context of electricity load forecast. A novel trigonometric coding of the outliers performs the best, improving the accuracy of both normal and outlying observations. Depending on the modeler's objective other approaches may perform well specifically in improving the performance of the network for normal values or outliers.

References

[1] R. F. Engle, "The econometrics of ultra-high-frequency data," *Econometrica*, vol. 68, no. 1, pp. 1–22, 2000.

[2] C. W. J. Granger, "Extracting information from mega-panels and high-frequency data," *Statistica Neerlandica*, vol. 52, no. 3, pp. 258–272, 1998.

[3] A. J. Conejo, J. Contreras, R. Espinola, and M. A. Plazas, "Forecasting electricity prices for a day-ahead pool-based electric energy market," *International Journal of Forecasting*, vol. 21, no. 3, pp. 435–462, 2005.

[4] H. Hahn, S. Meyer-Nieberg, and S. Pickl, "Electric load forecasting methods: Tools for decision making," in *International Conference on Information Systems, Logistics and Supply Chain*, vol. 199, Lyon, France, 2009, pp. 902–907.

[5] J. R. Trapero and D. J. Pedregal, "Frequency domain methods applied to forecasting electricity markets," *Energy Economics*, vol. 31, no. 5, pp. 727–735, September 2009.

[6] 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*, vol. 22, no. 1, pp. 1–16, 2006.

[7] C. Chatfield, *The Analysis of Time Series: An Introduction*, 6th ed. Chapman & Hall/CRC, 2004.

[8] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*. Springer Science+Business Media Inc., 2002.

[9] R. J. Hyndman and H. L. Shang, "Rainbow plots, bagplots and boxplots for functional data," Monash University, Department of Econometrics and Business Statistics, Monash Econometrics and Business Statistics Working Papers 9/08, Nov. 2008.

[10] Y. Sun and M. G. Genton, "Functional boxplots," *Journal of Computational and Graphical Statistics*, vol. to appear, 2011.

[11] P. K. Chan and M. V. Mahoney, "Modeling multiple time series for anomaly detection," in *Proceedings of the Fifth IEEE International Conference on Data Mining*, ser. ICDM '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 90–97.

[12] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *SIGMOD Rec.*, vol. 29, pp. 427–438, May 2000.

[13] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, pp. 85–126, October 2004.

[14] N. Kourentzes and S. F. Crone, "Semi-supervised monitoring of electric load time series for unusual patterns," in *Proceedings of the 2011 International Joint Conference on Neural Networks*, ser. IJCNN 2011, Forthcoming.

[15] H. S. Hippert, D. W. Bunn, and R. C. Souza, "Large neural networks for electricity load forecasting: Are they overfitted?" *International Journal of Forecasting*, vol. 21, no. 3, pp. 425–434–, 2005.

[16] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. New Jersey: Prentice Hall Inc., 1994, vol. 3rd.

[17] S. F. Crone and N. Kourentzes, "Segmenting electrical load time series for forecasting? an empirical evaluation of daily uk load patterns," in *Proceedings of the 2011 International Joint Conference on Neural Networks*, Forthcoming.

[18] G. Q. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[19] N. Kourentzes, "Input variable selection for time series forecasting with artificial neural networks an empirical evaluation across varying time series frequencies," Ph.D. dissertation, Department of Management Science, Lancaster University, 2009.

[20] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[21] G. P. Zhang, "An investigation of neural networks for linear time-series forecasting," *Computers and Operations Research*, vol. 28, no. 12, pp. 1183–1202, 2001.

[22] G. P. Zhang, B. E. Patuwo, and M. Y. Hu, "A simulation study of artificial neural networks for nonlinear time-series forecasting," *Computers and Operations Research*, vol. 28, no. 4, pp. 381–396, 2001.

[23] G. P. Zhang, "Neural networks for classification: a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 30, no. 4, pp. 451–462, Nov 2000.

[24] P. H. Franses and van Dijk D., *Non-linear time series models in empirical finance*. Cambrid, 2006.

[25] N. Kourentzes and S. F. Crone, "Inference for neural network predictive models with impulse interventions," in *Proceedings of the 2010 International Conference on Data Mining*, ser. DMIN10, Las Vegas, USA, July 2010.

[26] S. F. Crone and N. Kourentzes, "Forecasting seasonal time series with multilayer perceptrons an empirical evaluation of input vector specifications for deterministic seasonality," in *Proceedings of the 2009 International Conference on Data Mining*, ser. DMIN09, Las Vegas, USA, July 2009, pp. 232–238.

[27] E. Ghysels and D. R. Osborn, *The econometric analysis of seasonal time series*. Cambridge: Cambridge University Press, 2001.

[28] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*. Boston: PWS Publishing, 1996.

[29] S. F. 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*, ser. IJCNN'09, Atlanta, USA, 2009, pp. 3221–3228.

[30] L. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," *International Journal of Forecasting*, vol. 16, pp. 437–450, 2000.

[31] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, pp. 679–688, 2006.