Predicting Electricity Consumption in Toronto: A Machine Learning Approach

Greta Avetisian Cesare Spinoso-Di Piano

Abstract

We attempt to accurately predict hourly electricity demand (kWh) in the Greater Toronto Area with Machine Learning models while contrasting them with traditional time series forecasting methods. We also attempt to determine feature importance. We use classical time series models like exponential time smoothing, ARIMA and seasonal ARIMA and contrast them with linear models with feature transformations, vanilla as well as recurrent neural networks such as the long short-term memory network and the gated recurrent unit network. The linear model performs the best and provides us with satisfactory prediction accuracy and feature importance. Experiments show that in general classical time series methods may not be appropriate for this type of densely packed data.

1. Introduction

In the following report, we will describe and analyze the methods used to observe hourly electricity demand patterns in the Greater Toronto Area (GTA). Our primary goal will be to accurately predict the future demand of electricity within the region for a fixed period of half a year on previously unseen data. In this context, we define a model as being accurate if it has a low (out-of-sample) Root Mean Squared Error (RMSE) between its prediction and the corresponding true value. As a secondary goal, we will determine if Machine Learning (ML) methods presented throughout the course can surpass traditional time series forecasting methods. This part heavily relies on time series based external sources such as [2]. We will also show which features are the most important in driving down the validation RMSE i.e. feature importance.

2. Methodology & Experimental Results

2.1. Data Preparation

The electricity data is provided by the Independent Electricity System Operator (IESO) in Ontario and can be found here. Their online data directory consists of hourly electricity demand (in kilowatt-hour or kWh) and prices (CAD) in the GTA. In the process of merging the relevant datasets we observed some outliers in 2016 where demand would drop to 0. We were unable to determine whether this was the result of a power outage. Thus, we modified the values to take on the demand for the corresponding hour of the previous day. Although we are aware this is not the most sophisticated outlier handling method, the scarcity of such data points made this method acceptable for our purposes. To this resulting dataset, we added hourly weather data generated here. We also added day related characteristics as features. These included whether the given day was a weekend or a local holiday. We had originally planned to use data from 2004 to 2017 but the dataset size was computationally prohibitive for most non-trivial models. As a result we limited our dataset to the most recent period of 2012-2017 inclusively (a discussion of how the data was split for training is detailed in further sections). Thus, the data used was in a standard tabular form (Table 5).

2.2. Baselines and Time Series Models

We first describe our baseline models which are used to assess the usefulness of all our other models. Also, we describe our traditional time series models. All the models in this section (including the dummy models) were trained on years 2012 to 2016 inclusively using 10-fold time series aware cross validation. Ten folds were chosen since, for a period of 5 years, each validation fold corresponds to approximately half a year. Thus, our validation score faithfully estimates the test time performance related to our primary goal. Dummy models included an average estimator (taking the average electricity demand of previous folds and using it to predict the next), a median estimator (similar to the average estimator, but with respect to the median) and a lagged estimator where the previous half-year was used to estimate the following one. These models provided baseline validation RMSEs of 1053, 1054 and 1083 respectively.

For the time series specific models, we used the exponential time smoothing (ETS) and ARIMA models which according to [2] are widely used in time series forecasting. In both cases, the models observe the electricity demand accumulated in previous folds and then predict the demand for the next fold. The ETS model, which relies on a weighted average of previous observations, was trained for every possible configuration (Table 2). The best model was the ETS with weekly periodicity and damped trend component which achieved an RMSE of 678, but produced poor behaviour on out-of-sample estimates (Figure 2). We notice that models with seasonality always perform better than those without. This can be explained by the intuitive seasonality that we understand electricity demand patterns to have. The ARIMA(p, d, q) model was trained for various configurations of p, d and q. We observed that none of the configurations performed better than our baseline estimators (Table 3). This is not surprising since the ARIMA is non-seasonal. We attempted to train various configurations of the seasonal ARIMA (SARIMA). However, every configuration of period 24 or greater failed to train due to memory overflow. We suspect that this is related to the way the SARIMA is trained [3]. Thus, in general, we already see that traditional time series forecasting methods fail or are unable to predict accurately for the given data.

2.3. Linear Models

To obtain a more accurate prediction, we turned to linear regression and feature basis transformations. Training and testing was done under the same conditions as the previous section. As described in [2], for time series data with seasonality, if observations are indexed with a variable t (where t runs from 0 to $N-1$, where N is the number of observations) and we suspect the data to have a seasonality of period length m then a suitable feature basis transformation for this setting is

$$\boldsymbol{\phi}_t = \left[\sin(\tfrac{2\pi t}{m}), \cos(\tfrac{2\pi t}{m}), \ldots, \sin(\tfrac{2\pi P t}{m}), \cos(\tfrac{2\pi P t}{m})\right]^T$$

where $P = \frac{m}{2}$. We decided to include transformations to account for daily, weekly and yearly seasonality. Thus our resulting model had the following functional form

$$\begin{aligned} Y_t = \beta_0 &+ \sum_{j=1}^{P_d} \beta_{2j-1} \sin(\frac{2\pi jt}{m_d}) + \beta_{2j} \cos(\frac{2\pi jt}{m_d}) \\ &+ \sum_{k=1}^{P_w} \beta_{2k-1} \sin(\frac{2\pi kt}{m_w}) + \beta_{2k} \cos(\frac{2\pi kt}{m_w}) \\ &+ \sum_{l=1}^{P_y} \beta_{2l-1} \sin(\frac{2\pi lt}{m_y}) + \beta_{2l} \cos(\frac{2\pi lt}{m_y}) \end{aligned}$$

where Y_t is the electricity demand at time (or index) t, $m_d = 24, m_w = 7 \cdot 24, m_y = 365 \cdot 24$ and P_d, P_w, P_y can go up to $\frac{m_d}{2} - 1$, $\frac{m_w}{2} - 1$, $\frac{m_y}{2} - 1$ respectively. To find the optimal setting for P_d, P_w and P_y, we considered every possible combination from the ranges $\{0, 1, \ldots, 11\}$ for P_d, $\{0, 12, \ldots, 72\}$ for P_w and $\{0, 1, \ldots, 10, 243, 486, 729\}$ for P_y. That is, a feature matrix X was constructed with each of the mentioned configurations using index t for the period of 2012 to 2016 and evaluated via cross-validation. We did not consider the full range for P_y since preliminary experiments showed that very few of these bases were useful in the prediction. The optimal model found, $\hat{Y_t}^*$, had a configuration of $(P_d = 20, P_w = 144, P_y = 4)$ with RMSE 605. This is better than the optimal ETS model found in the previous section. More importantly, we were able to introduce seasonality with very little computational overhead which leads us to believe that data with dense periodicity could benefit from this sort of ML approach.

To further improve our results, we introduced the original features that we had initially extracted. After analyzing a pairwise plot (Figure 3), we determined that temperature and air density were highly correlated leading us to drop the latter variable. We also noticed that electricity demand presented a possible nonlinear relation with temperature leading us to introduce a Temperature^2 feature. A best subset approach for all possible combinations of features was taken on the resulting 11 features. That is, we added every possible combination of features to the feature matrix for the current best model $\hat{Y_t}^*$, trained on an accumulation of folds, predicted the electricity demand for the subsequent fold and computed the validation RMSE. The observed result was that Temperature, Temperature^2, Snowfall, Snow Mass, Radiation Toa, Cloud Cover, isWeekend, and isHoliday were added to the optimal model $\hat{Y_t}^*$ by the exhaustive search and provided an out-of-sample RMSE of 392 (Figure 4).

A limitation in our approach is that although we considered every possible subset of features, we did not consider every possible combination of features with different settings of (P_d, P_w, P_y). As a result, we also explored an L1 and L2 regularization on the full feature matrix consisting of all transformations with $\sin$'s and $\cos$'s as well as all the features used in the previous section. In both cases we used penalty weights in the range $10^{-2}, 10^{-1}, \ldots, 10^5$. The optimal configuration for the ridge regression was $\lambda = 10^2$ with an RMSE of 512. The optimal configuration for the lasso was $\lambda = 10$ with an RMSE of 422. As we can see, regularization does not provide an improved RMSE so we believe this shows that considering all possible interactions would not have significantly improved our prediction results.

Another form of regularization that we considered was dimensionality reduction where we considered the full feature matrix, reduced its dimensionality to a certain number M and determined the performance of the reduced feature matrix via cross-validation. We perform Principal Component Analysis (PCA) for the range of reduced number components in $\{10, 60, 110, \ldots, 990\}$ and performed linear regression (with an added intercept) with the reduced feature matrix (Figure 5). We obtained an optimal configuration with $M = 660$ components with an RMSE of 742.

This is worse than the lasso model and thus fails to regularize as we intended it to. We also performed an autoencoder (AE) dimensionality reduction with an architecture of 1000-750-M-750-1000 where $M \in \{50, 100, \dots, 500\}$ and performed linear regression with the reduced feature matrix. Since the feature bases are nonlinear we expected AEs would preserve the nonlinearity of features better than PCA. Due to the computational overhead of AE training, we limited ourselves to the matrix that consisted of only the first 730 yearly $\sin$ and $\cos$ basis functions as in the initial linear regression training (Figure 5). We also needed to limit the number of epochs to 500 and kept a batch size of 500. The optimal configuration was $M = 150$ with an RMSE of 932.

Finally, to boost the performance of our best linear model $\hat{Y}_t^*$, we performed an ensemble where the training errors of the model were predicted by an ARIMA(p, d, q) [2]. That is, supposing we have $Y_t = \hat{Y}_t^* + \epsilon$, we trained an ARIMA model on the residual $Y_t - \hat{Y}_t^*$ such that $\epsilon_t = \hat{\epsilon_t} + \delta_t$ for some error component δ_t. To train the ARIMA, we first trained $\hat{Y}_t^*$ on the whole training set, computed the residuals $\epsilon_t = Y_t - \hat{Y}_t^*$ and fed these to the ARIMA model so that it could train on the errors of accumulated folds and predict the errors for the subsequent fold. We used the same configuration for (p, d, q) as in the time series section and obtained the best validation RMSE for $(p, d, q) = (1, 0, 0)$ with an RMSE of 348 (Table 4). Thus using our optimal model $\hat{Y}_t^*$ added to the ARIMA provides the best out-of-sample error of this section (Figure 6).

2.4. Neural networks

In another attempt to obtain better prediction accuracy, we implemented several neural networks. The first was a vanilla feed-forward neural network with input corresponding to a year's worth of hourly electricity demand data and output corresponding to the prediction for the following year. That is, we artificially constructed a "time series" neural network with input and output size 8760 (365 days $\cdot$ 24 hours). The year's worth of hourly electricity demand was fed to it in increments of 24 hours (to keep a manageable batch size) starting from 2012 and ending in 2015. Similarly, the corresponding output sequence started in 2013 and ended in 2016 (Table 1). This way of shifting the data provided us with 1098 batches.

Validation was done using a validation set rather than cross-validation to cut down on computation time. Once the network was trained, it was fed the last available year of training (2016) and predicted the hourly electricity demand for 2017. To keep the validation score representative of test time performance, we computed the RMSE only for the first half of the predicted validation set. Tuning the batch size in the range $\{183, 549, 1098\}$ (to fit perfectly within the total number of batches), the hidden layer in the range $x \in \{(10), (50), (100), (200), \dots, (1500), (10, 10), \dots, (1500, 1500)\}$ (where the resulting architecture would be 8760-x-8760) and the number of epochs in the range $\{100, 200, \dots, 500\}$, we obtained the optimal configuration with a batch size of 183, 2 hidden layers with 1200 hidden units each and 100 epochs. The validation RMSE was 467, thus outperforming all of the time series models as well as the linear models with only periodic feature bases. We notice however that the behaviour of the network on the validation set is a lot less smooth than the linear models (Figure 7).

To improve our results, we introduced our extracted features. Because it was unclear how to easily integrate features to the existing neural network, we created a new feed-forward network with the features at time t as the inputs and the residual error from the "time series" neural network at time t as the output. That is, the "feature" neural network had input size 11 (for the 11 extracted features) and output size 1. This is similar to how we used the ARIMA model to train on the errors of our optimal linear model. The residuals were computed from the trained optimal "time series" neural network on the years 2013 to 2016 (since 2012 was the first observed sequence and thus could not be predicted). The network was tuned on batch size, hidden layer configuration as well as dropout to reduce overfitting since, as already observed in the linear models section, not all features might be useful. The batch size was tuned within the range $\{240, 480, 730\}$ (again to make it fit within the total number of batches), the hidden layer was tuned in the range $\{(100), (200), \dots, (1000), (100, 100), \dots, (1000, 1000)\}$ and each model was trained with and without dropout, using a dropout rate of 0.2 for the input units and 0.5 for the hidden units as suggested by [1]. The optimal configuration had a batch size of 480, a 11-900-1 architecture and used dropout. The predicted residuals were validated on the true residual errors from the prediction of the optimal "time series" neural network (8760-1200-1200-8760 architecture) on the validation set. We obtained an RMSE of 460 (Figure 8). Note that although the "time series" network did better than the linear model that was trained only on periodic features, it did worse when integrating features. We suspect that this may be related to the way we

Batch 1	$X_1, \dots, X_{8760}$	$X_{8761}, \dots, X_{17520}$	
Batch 2		$X_{24}, \dots, X_{8784}$	$X_{8785}, \dots, X_{17544}$

Table 1. Illustration of the training process for the "time series" neural network. The first batch uses $X_1, \dots, X_{8760}$ as the input and $X_{8761}, \dots, X_{17520}$ as the output where X_t is the electricity demand at time t and 8760, 17520 are the time indices for the first and second years of training. In the subsequent batch everything is shifted by 24.

introduced features to the neural network.

To try to further improve our prediction using neural networks, we used the "sequence aware" long short-term memory (LSTM) and gated recurrent units (GRU) neural networks. We immediately went to these rather than starting with a vanilla recurrent neural network (RNN) because we suspected that the length of our sequence would prevent the RNN from learning long term patterns as it would most likely suffer from vanishing and/or exploding gradients. The LSTM was trained under the same conditions as the vanilla neural network passing it a sequence corresponding to a year's worth of hourly data in increments of 24. The LSTM was tuned for its batch size ($\{183, 549\}$), its number of layers ($\{2, 3, 4\}$) and whether it would be bidirectional. An important observation to make here was that because the sequence length of training data was long (8760) and because RNNs lose a degree of parallelism since they need to be trained sequentially, the LSTM was extremely slow to train. This intractability made us unable to tune the LSTM as finely as previous models. As a result, most of the observed validation scores were much worse than our baseline estimators. The optimal configuration corresponded to a batch size of 183, 4 layers and no bidirectionality. It had a validation RMSE of 726, worse than the best traditional time series forecasting method. The GRU, which we thought might have faster computation speeds, was just as slow to train under the same conditions. It reported a slightly better validation RMSE of 627 with a batch size of 183, 2 layers and using bidirectionality. In both cases we did not pursue a vanilla "feature" neural network to improve residuals like we did in the previous architecture. The poor results obtained from the LSTM and GRU as well as their computational overhead tell us more work needs to be done in implementing RNNs for datasets with dense periodicity like ours.

2.5. Model selection and test time performance

Based on our results of validation RMSE, it appears that the linear model with ARIMA errors performs the best out of all the models that were trained. We selected this one as our final model and provided an unbiased estimate of the test time performance by retraining it again on all the training set as well as the first half of 2017 and predicting the electricity demand for the second half of 2017. We obtained an RMSE of 410 (Figure 1).

We have been dealing with RMSE throughout this report as it is the simplest metric to tune different models. However, it is not a very interpretable metric. To resolve this, we also calculated the mean absolute percent error (MAPE) for the resulting prediction and obtained a percent error of 5.9%. Thus, given its low RMSE and percent error, we can conclude that our best model performs well for unseen data.

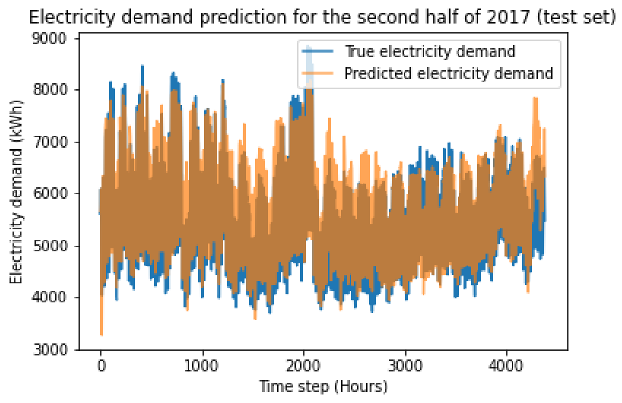

Figure 1. Hourly electricity demand predicted on the test set.

3. Conclusions

We believe that the relatively low RMSE and MAPE computed on an unseen test set for our best model shows that we have successfully achieved satisfactory prediction accuracy. We also believe that the poor performance of classical time series methods compared to our ML models as well as the ARIMA's inability to allow long term periodicity show that ML models such as linear models and vanilla neural networks are more suited for this type of dataset. However, we cannot conclude the same thing about the LSTM and GRU that we trained. In fact, the difficulty to train and tune these models prevent us from stating any conclusive results. More work such as applying convolutional networks to reduce the sequence length as well as trying different sequence length settings is necessary. Finally, in terms of feature importance, our best model being the linear model conveniently allows us to identify which features most decrease the out-of-sample RMSE.

References

[1] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. 3

[2] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Australia, 2nd edition, 2018. 1, 2, 3

[3] Robin John Hyndman. Forecasting with long seasonal periods, Sep 2010. 2

Appendix: Extra Figures and Results

Period	Damped	MSE	RMSE
day	True	947811	974
week	True	459656	678
monthly	True	93200226	9654
day	False	10576475	3252
week	False	1281704601846	1132124
monthly	False	52546163660	229230

Table 2. Validation RMSE for different variations of the exponential time smoothing. Weekly seasonality with damped trend component performs the best. The extremely poor performance for undamped ETS can be explained by the model predicting a negative electricity demand.

p	d	q	MSE	RMSE
0	0	0	1109593	1053
0	0	1	1109557	1053
0	0	2	1109531	1053
0	1	0	1734259	1316
0	1	1	1626736	1275
0	1	2	1530334	1237
1	0	0	1113016	1054
1	0	1	1112120	1054
1	0	2	1111110	1054
1	1	0	2292332	1514
1	1	1	1639008	1280
1	1	2	1541018	1241
2	0	0	1108018	1052
2	0	1	1108778	1052
2	0	2	1108749	1052
2	1	0	1457136	1207
2	1	1	1471527	1213
2	1	2	1465475	1210

Table 3. Validation MSE and RMSE for the different orders of the ARIMA model. Notice how the lowest RMSE is the same as the dummy estimator.

p	d	q	MSE	RMSE
0	0	0	121815	349
0	0	1	121796	348
0	0	2	121781	348
0	1	0	231941	481
0	1	1	231634	481
0	1	2	231918	481
1	0	0	121545	348
1	0	1	121596	348
1	0	2	121612	348
1	1	0	236098	485
1	1	1	231789	481
1	1	2	149400	386
2	0	0	121638	348
2	0	1	121608	348
2	0	2	121600	348
2	1	0	228877	478
2	1	1	229042	478
2	1	2	230719	480

Table 4. Validation MSE and RMSE for the different order of the ARIMA model trained on the residuals of the best linear model.

Datetime	Demand	HOEP	temp.	precip.	snowfall	snowmass	airdensity	rad.surf	rad.toa	cloud
16-09-05 22:00	5652	17.06	21.35	0	0	0	1.17	0	0	0.017
16-09-05 23:00	5225	3.69	20.84	0	0	0	1.18	0	0	0.413
16-09-06 0:00	4896	-0.25	20.42	0	0	0	1.18	0	0	0.923
16-09-06 1:00	4706	-0.5	20.15	0	0	0	1.18	0	0	0.934
16-09-06 2:00	4621	-1.03	19.95	0	0	0	1.18	0	0	0.92
16-09-06 3:00	4619	-0.25	19.72	0.008	0	0	1.18	0	0	0.653

Table 5. A set of sample data used in our models. The prediction is on Demand and all other columns are related to features. Please refer to the original source for a description of each variable. We omit the isWeekend and isHoliday column to fit within the margin. Also note that since this is time series data the electricity demand is indexed by a Datetime column.

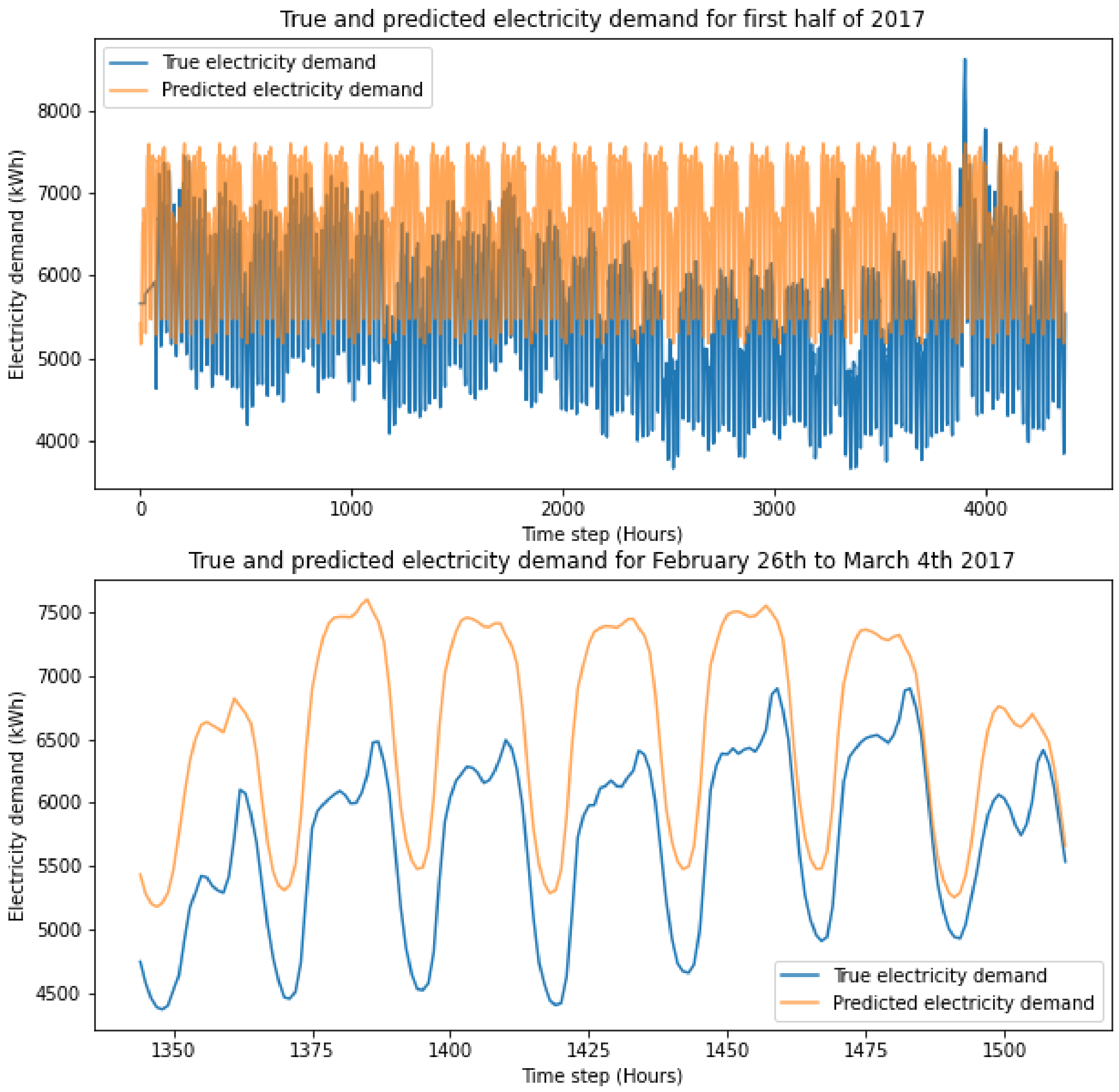

Figure 2. Prediction of the best ETS model on a held out portion of the data corresponding to the first half of 2017. We see that the ETS is unable to observe long term periodicity since it is designed only for weekly seasonality.

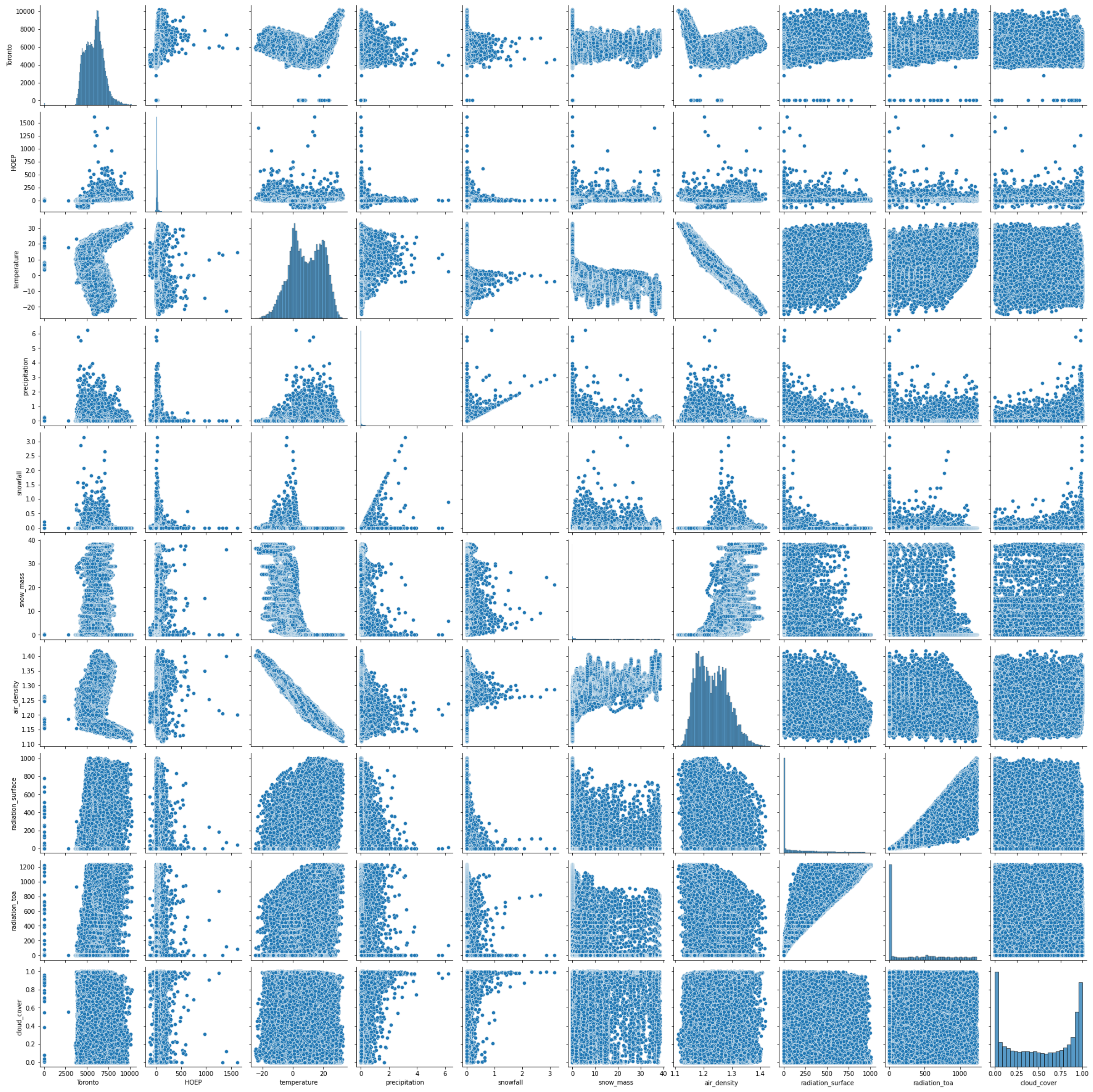

Figure 3. Pairplot for all of the continuous features as well as the the electricity data. Observe that air density and temperature (3rd from the top and 7th from the left) are highly correlated. Moreover electricity demand and temperature (1st from the top and 3rd from the left) follow a non-linear relationship.

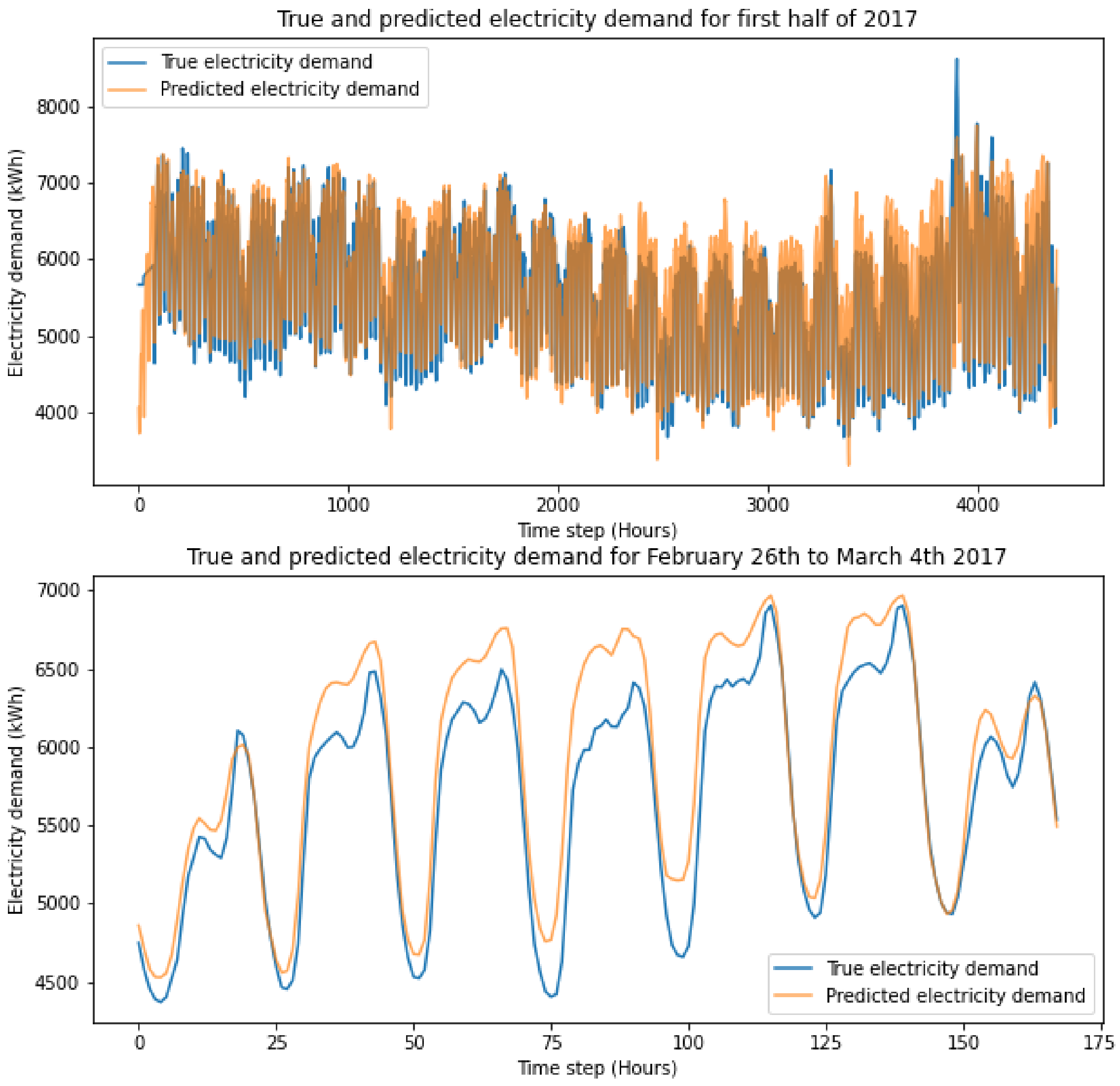

Figure 4. Prediction performed by the linear model tuned for periodicity and features on the first half of 2017.

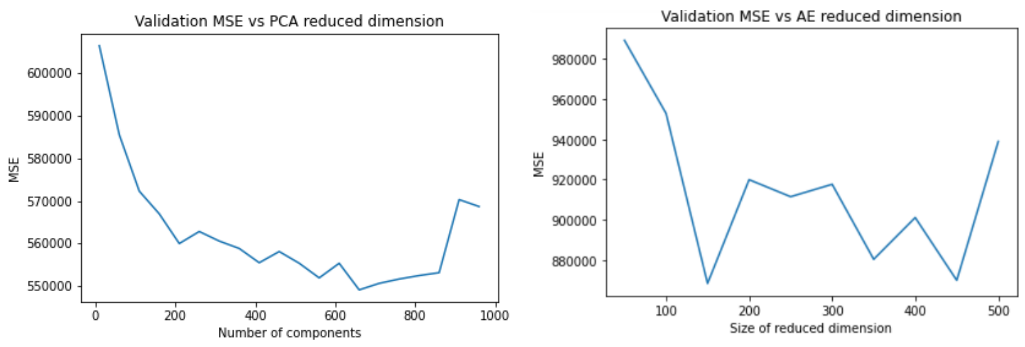

Figure 5. Validation MSE computed on the reduced feature matrix for both PCA and AE.

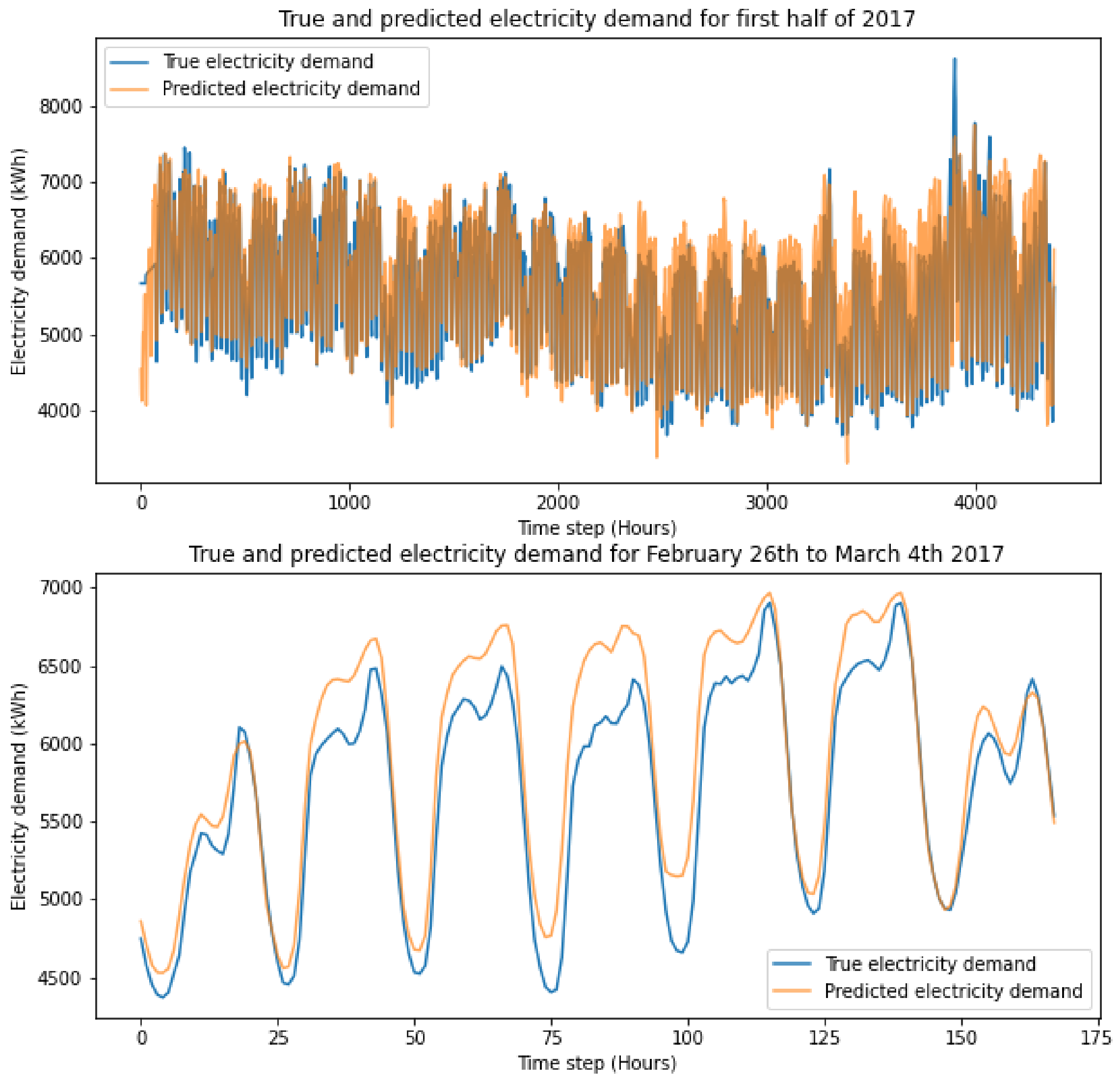

Figure 6. Prediction performed by the linear model tuned for periodicity and features with ARIMA errors on the first half of 2017.

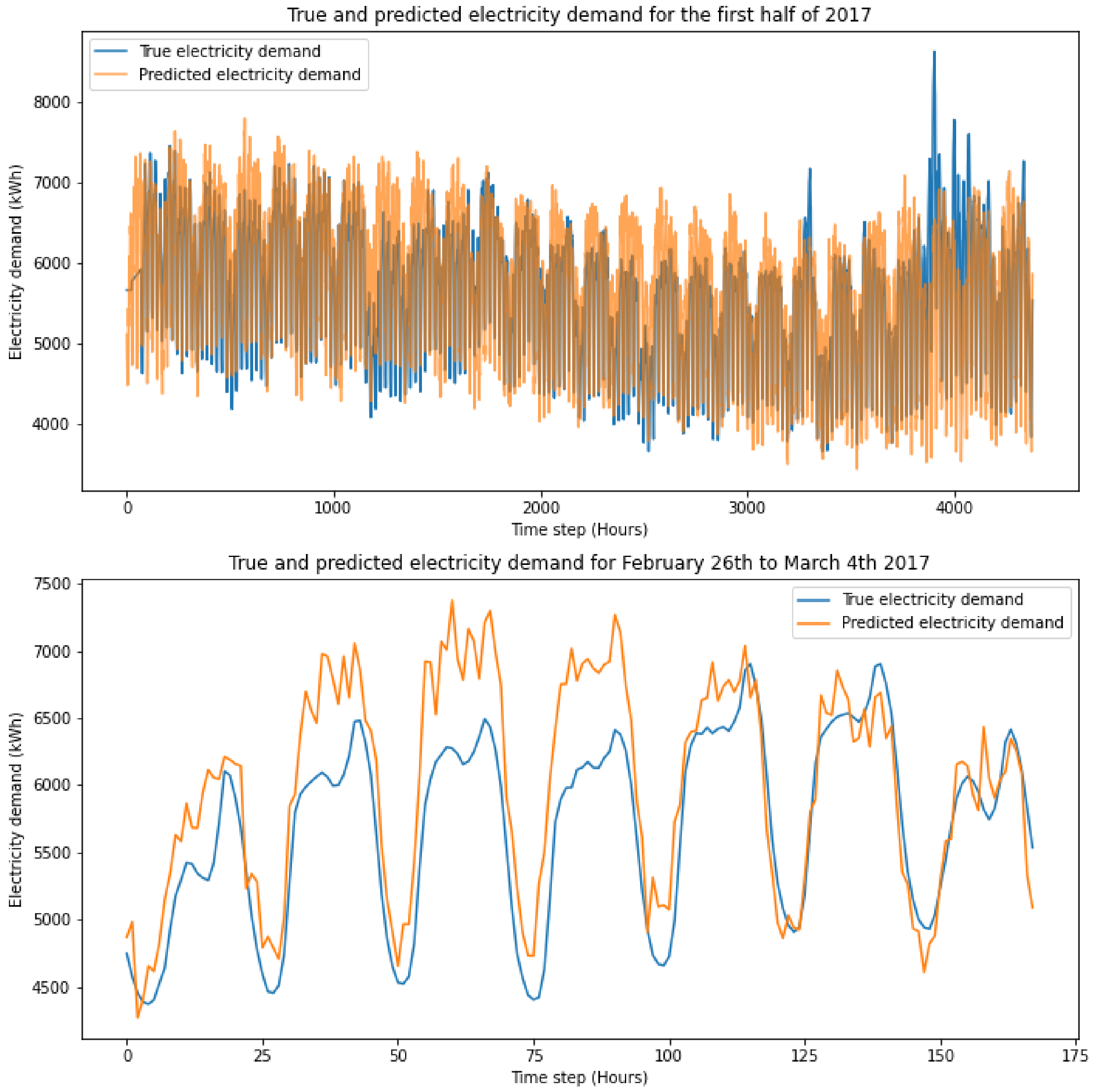

Figure 7. Prediction by the “time series” neural network based only on previous electricity observations on the first half of 2017.

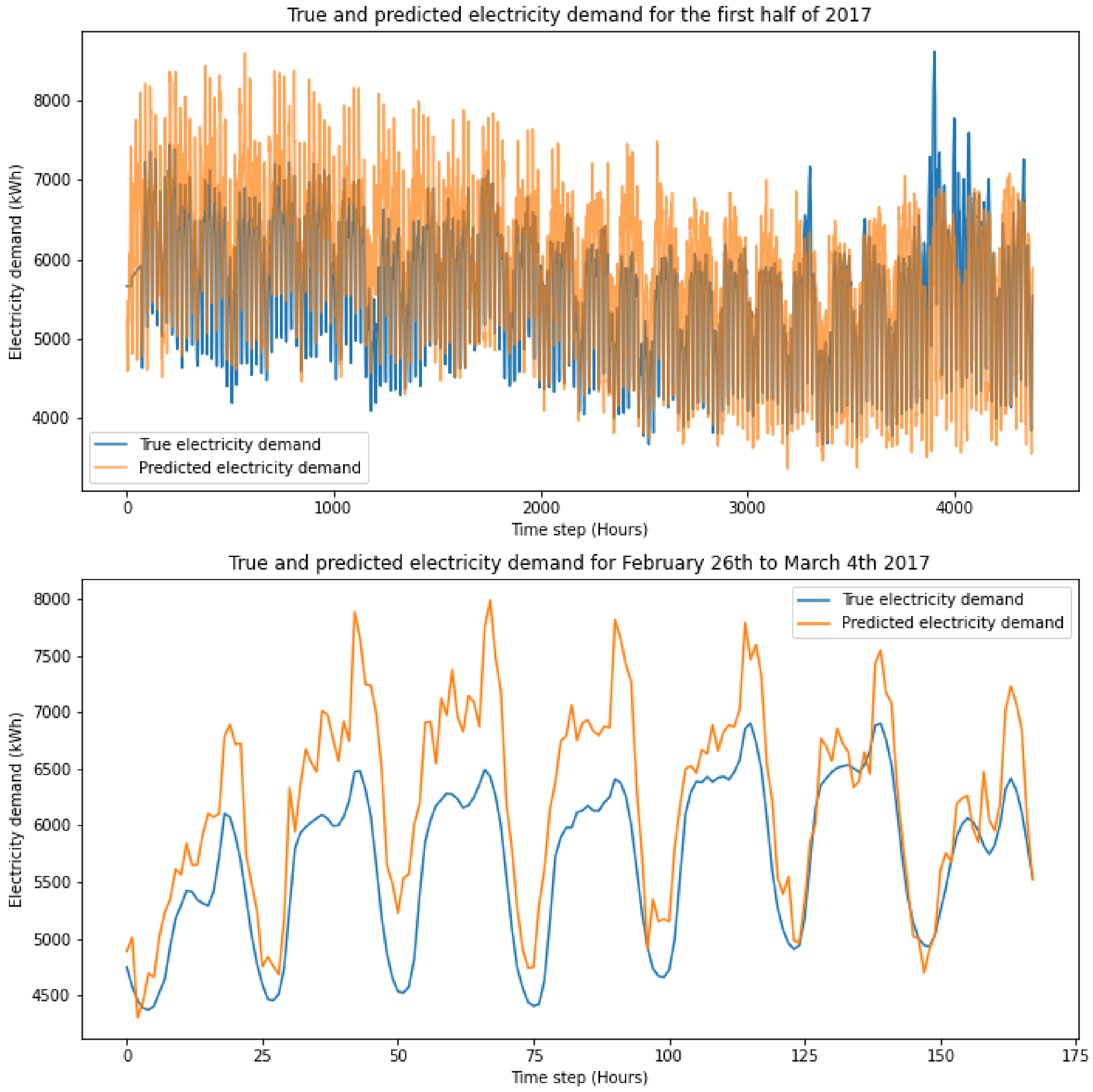

Figure 8. Prediction by the “feature” neural network added to the “time series” neural network on the first half of 2017.