



Short Term Load Forecasting of Industrial Electricity Using Machine Learning

Rodrigo Porteiro^{1(✉)}, Sergio Nesmachnow^{2(✉)}, and Luis Hernández-Callejo^{3(✉)}

¹ UTE, Montevideo, Uruguay
rporteur@ute.com.uy

² Universidad de la República, Montevideo, Uruguay
sergion@fing.edu.uy

³ Universidad de Valladolid, Valladolid, Spain
luis.hernandez.callejo@uva.es

Abstract. Forecasting the day-ahead electricity load is beneficial for both suppliers and consumers. The reduction of electricity waste and the rational dispatch of electric generator units can be significantly improved with accurate load forecasts. This article is focused on studying and developing computational intelligence techniques for electricity load forecasting. Several models are developed to forecast the electricity load of the next hour using real data from an industrial pole in Spain. Feature selection and feature extraction are performed to reduce overfitting and therefore achieve better models, reducing the training time of the developed methods. The best of the implemented models is optimized using grid search strategies on hyperparameter space. Then, twenty four different instances of the optimal model are trained to forecast the next twenty four hours. Considering the computational complexity of the applied techniques, they are developed and evaluated on the computational platform of the National Supercomputing Center (Cluster-UY), Uruguay. Standard performance metrics are applied to evaluate the proposed models. The main results indicate that the best model based on ExtraTreesRegressor obtained has a mean absolute percentage error of 2.55% on day ahead hourly forecast which is a promising result.

1 Introduction

Decision making in the energy sector was historically supported by information that allows predicting, with certain degree of uncertainty, the variables that affect these decisions [8, 22]. Much of the useful information is related to natural variables (e.g., temperature, wind speed, humidity). Other information is related to the energy consumption profile of users. In recent years, the sources of energy generation have diversified in the world. Many renewable sources that are directly related to natural variables have been incorporated [16].

All of aforementioned issues implies that for making decisions is necessary to take into account a large number of stochastic variables, to ensure that they are feasible/optimal from the economic point of view. The increase in complexity

associated with the number of variables to be considered is mitigated by two factors. On the one hand, the sources of data on the variables have multiplied, since many technological components of measurement have emerged in all disciplines and the hardware infrastructure that supports these components has developed strongly. On the other hand, multiple new uses for energy have emerged.

The new reality presents the challenge of developing new tools that allow taking advantage of available data as much as possible. Classic statistical models that were always useful for making predictions have clear limitations in this new context. Computational intelligence algorithms have shown in recent years to perform excellently for forecasting in different areas [11, 13, 15]. These methods are able to learn the most relevant features of the data to be taken into account in order to provide a precise forecast, thus providing excellent results by excluding information of little relevance and considering the most relevant one.

In this line of work, this article presents the application of several prediction algorithms based on computational intelligence to forecast the electricity demand of an industrial pole for the next hour. The modeled scenario is based on historical demand data of an industrial pole in Spain from 2014 to 2017. From the study and comparison of the results of the algorithms developed for the next hour, a model is constructed to forecast the next 24 h. This model is based on optimizing the algorithm that presented the best results for the one hour forecast and extending it to 24 h forecast. The major contributions of this research are: (i) the evaluation and comparison of computational intelligence models applied to forecasting the demand of an industrial pole in Spain, and (ii) the optimization of the model using the infrastructure of the National Supercomputing Center, in Uruguay.

The article is organized as follows. Section 2 presents the formulation of the day ahead forecasting problem and a review of related works. Section 3 describes the proposed approach to solve the problem proposed. Section 4 presents Experimental Analysis of the problem. In Sect. 4.5 analysis of the best method and extension to 24 h load forecast is presented. Finally, Sect. 5 formulates the main conclusions and lines for future work.

2 Load Forecasting

This section introduces the load forecasting problem, describes forecasting techniques, and reviews related works.

2.1 General Considerations

The load forecasting problem is usually approached applying mathematical methods using historical data to predict the demand of electric power. In general, there is no method that can be used in all types of load forecasting. Thus, an appropriate method must be found for each load profile. Using historical data of a particular load profile is common in practice to determine the most effective algorithm. Electric load forecasting can be classified by time horizon to forecast:

(i) ultra short-term load forecasting: up to a few minutes ahead; (ii) short-term load forecasting: up to few days ahead; (iii) medium-term load forecasting: up to few month ahead; and long-term load forecasting: years ahead. Different techniques are applied when considering each time horizon. This work focuses in short-term load forecasting using historical data.

The energy management and operation of grids becomes highly difficult and uncertain, particularly when new technologies were incorporated. The power demand of end customers is versatile and is changing on hourly, daily, weekly, and seasonally basis. Hence, there is a real need of developing a model for precise and accurate forecasting at different time horizons, depending on the management goals. Day ahead hourly power load prediction is considered a short term forecasting problem, and it is very important to develop very precise models for solving this particular problem.

This work focuses on industrial power consumption. Residential (domestic) power profiles are usually very variable, mainly dependent on the time of the day and the day of the week, but it also dependent on occasional vacations and other particular factors. On the other hand, industrial users power profile tends to be more stable due to the needs of the industrial process itself.

There are two classes of forecasting models for predicting power profile: statistical and physical models. The main purpose of both classes of models is to predict the power profile at a future time frame. Statistical models can be built for time series analysis. Computationally, statistical models are less complex than physical models and are suitable for short term prediction. Physical models are based on differential equations for relating the dynamics of the environment and generally are applied for long term forecasting. In the present work, statistical models are selected for short term forecasting due to their very good prediction accuracy and lower complexity.

2.2 Problem Formulation and Strategies

Relation Between One Hour and 24 Hour Forecasting. The main goal of the study reported in this article is to apply computational intelligence methods to develop a model for electricity load 24 h ahead forecasting. When historical data are available with hourly frequency is natural to develop a model that predicts next hour. From that model, a multi-step time forecasting model can be constructed, in this case 24 steps in the future.

Four strategies are typically applied for multi-step forecasting starting from a one-step model:

- *Direct strategies* develop a different model for each time step to be predicted. Assuming past observations of the variable to be predicted are used, this strategy implies, in case of 24 steps, developing 24 models with the structure defined in Eq. 1, where $pred_t$ is the prediction of time t value and obs_t is the observed value at time t .

$$\begin{aligned}
pred_{(t+1)} &= model_1(obs_t, obs_{(t-1)}, \dots, obs_{(t-n)}) \\
pred_{(t+2)} &= model_2(obs_t, obs_{(t-1)}, \dots, obs_{(t-n)}) \\
&\dots \\
pred_{(t+24)} &= model_{24}(obs_t, obs_{(t-1)}, \dots, obs_{(t-n)})
\end{aligned} \tag{1}$$

Unfortunately, a direct strategy implies developing a model for each time step to be predicted and consequently is very expensive computationally. In addition, temporary dependencies are not explicitly preserved between consecutive time steps.

- *Recursive strategies* apply a one-step model (recursively), multiple times. The predictions for previous time steps are used as input for making a prediction on the following time step. The structure to develop for a recursive strategy is presented in Eq. 2.

$$\begin{aligned}
pred_{(t+1)} &= model_1(obs_t, obs_{(t-1)}, \dots, obs_{(t-n)}) \\
pred_{(t+2)} &= model_1(pred_{(t+1)}, obs_t, obs_{(t-1)}, \dots, obs_{(t-n+1)}) \\
&\dots
\end{aligned} \tag{2}$$

$$pred_{(t+24)} = model_1(pred_{(t+23)}, pred_{(t+22)}, \dots, pred_{(t+1)}, obs_{(t-n+23)})$$

In this strategy predictions are used instead of observations. A single model is trained, but the recursive structure allows prediction errors to accumulate and the performance of the model can quickly degrade as the time horizon increases.

- *Hybrid strategies* combine the previously described to get benefits from both methods. A separate model is constructed for each time step to be predicted. Each model may use the predictions made by models at prior time steps as input values. For example, using all known prediction, a hybrid strategy produces the structure in Eq. 3.

$$\begin{aligned}
pred_{(t+1)} &= model_1(obs_t, obs_{(t-1)}, \dots, obs_{(t-n)}) \\
pred_{(t+2)} &= model_1(pred_{(t+1)}, obs_t, \dots, obs_{(t-n)}) \\
&\dots
\end{aligned} \tag{3}$$

$$pred_{(t+24)} = model_1(pred_{(t+23)}, pred_{(t+22)}, \dots, obs_t, \dots, obs_{(t-n)})$$

- *Multiple output strategies* develop a model that has as output all time steps to be predicted (in this case 24). Multiple output models are more complex as they can learn the dependence structure between inputs and outputs as well as between outputs. For this reason, they are slower to train and require more

data to avoid overfitting. Equation 4 shows the corresponding structure.

$$pred_{(t+1,\dots,t+24)} = model_1(obs(t), obs(t-1), \dots, obs(t-n)) \quad (4)$$

In this work, hybrid strategies are applied for solving the forecasting problem.

One Hour Forecasting Model Training. Section 2.3 reviews different approaches and methods for short term load forecasting. This work explores the use of machine learning techniques, mainly those based on model ensembles. Feature selection is commonly applied in this kind of problems due to several reasons. Simpler models are easier to interpret, and have shorter training times. Also, the size of the model using less features is smaller, mitigating the *curse of dimensionality* [3]. But the main reason to apply feature selection is to reduce overfitting, enhancing generalization of the model to unseen data.

Once established the strategy to extend the next hour forecasting models to twenty four hours model, the main issue is to obtain the best possible model for the next hour. With this purpose, standard steps are taken: (i) data gathering, (ii) data preparation, (iii) choosing a model, (iv) training, (v) evaluation, (vi) parameter tuning, and (vii) testing. Each of these steps is described in detail in Sect. 3.

Complete Model. After obtaining a one hour model with optimized parameters, it is trained for the next hour taking all steps mentioned. Thus, 24 h different instances of this model are trained, one for each of the next 24 h. Then, the hybrid strategy described in Eq. 3 is applied to build a 24 h forecasting model. The complete model is evaluated on testing data and results are reported.

2.3 Related Works

Several methods support electricity demand forecasting, applying short, medium and long-term predictions. These methods are classified in statistical models and machine learning models. This work focuses on short-term load forecasting using machine learning.

Most used forecasting techniques include auto regressive models (AR), moving average models (MA), auto regressive moving average models (ARMA) and auto regressive integrated moving average (ARIMA) models [24]. These kind of models are easy to implement. ARIMA models for short term load forecasting were initially proposed by Hagan and Behr [12]. Taylor and McSharry [26] compared different ARIMA implementations using load data from multiple countries. Linear regression technique was described by Dudek [10]. However, linear models are inadequate to represent the non-linear behavior of electricity load series and fail to predict the accurate future demand values. Thus, their forecasting accuracy tends to be poor.

Several studies have been conducted on short-term load forecasting using non-linear models. For example, Do et al. [9] described a model for predicting hourly electricity demand considering temperature, industrial production levels, daylight hours, day of the week, and month of the year to forecast electricity consumption. Results suggested that consumption is better modeled considering each hour separately. In our work, this strategy is developed and applied. Son and Kim [25] proposed a method based on support vector regression preceded by feature selection for the short-term forecasting of electricity demand for the residential sector. For feature selection, twenty influential variables were considered and the quality of the model improved substantially.

Peak load estimation is also crucial to determine future demand, in order to assist future investment decisions [21]. In this article, the decision to consider ensemble models was taken based in the work presented by Burger and Moura [5], who applied a gated ensemble learning method for short-term electricity demand forecasting and showed that the combination of multiple models yielded better results than the use of a single model. Silva [23] presented a complex feature engineering to build gradient boosted decision trees and linear regression models for wind forecasting; in our work several similar ideas were developed for demand forecasting. De Felice et al. [7] applied several separate models for each hourly period. Each of those models measure variations in electricity demand based on multiple variables.

The analysis of the related works allowed to conclude that two main issues impact on the forecasting capabilities and the results quality: the model itself and other preparation and pre-processing techniques. Several works applied techniques like data normalization, filtering of outliers, clustering of data or decomposition by transformations [1,2,6,14] in order to improve the results. In our research, several data preparation techniques are applied for building a robust approach for short term energy utilization forecasting. Next section describes the proposed approach.

3 The Proposed Approach for Day Ahead Industrial Load Forecasting

This section describes the proposed approach to solve the day-ahead electricity load forecasting for an industrial pole in Spain, applying the strategies described in Sect. 2.2.

3.1 General Approach

Data Description, Data Preparation, and Metrics. The analysis reported in this article considers historical hourly energy consumption data from an industrial pole in Spain. This data was collected between January 2014 and December 2017. The dataset studied in the research is formed by industrial energy consumption measurements. Each measurement is composed of:

- *Year* (integer), representing the year on which the measure was taken.
- *Month* (integer), indicating the month on which the measure was taken.
- *Day* (integer), indicating the day on which the measure was taken.
- *Hour* (integer), indicating the hour on which the measure was taken.
- *Dayofweek* (integer), indicating the day on which the measure was taken.
- *Workingday* (boolean), indicating whether the measure was taking in a working day or not.
- *Useful* (boolean), indicating whether the measure is valid.
- *Demand* (float), indicating the real power measured.

The data preparation consists in replacing useless measures or outliers using information from neighboring hours. A few useless measures and outliers were found (less than 0.0001%), and none of this measures corresponded to consecutive hours. Thus, useless measures were replaced with the average measure of the previous and next hour. Outliers were replaced by the value of the mean of the measures plus 3 standard deviations. A measure is considered an outlier when its signed number of standard deviations by which is above the mean value of what is being measured is greater than 3. Feature standardization was applied to avoid scale problems. Finally, from the dataset, new features were generated associated with past demand measures to train the models. In particular, the last 48 measures were considered for each record to capture at least two days of consumption pattern directly in the features.

Several visualization analysis were performed to gain an intuitive insight of the information contained in each feature. The most relevant fact confirmed in this preliminary analysis was the daily periodicity of the demand value. The correlation diagram shown in Fig. 1 presents the high correlation between actual demand and the demand of the same hour of two days before. Data preprocessing was performed using pandas library [18]. The dataset from 2014 to 2017 was extended to include all lag features of the last 24 past hours. The training set included all data from 2014 to 2016, and the testing set included data from 2017. A linear regression model M_{sim} was trained using the sklearn toolkit [20], configured with default parameters as benchmark model. New training and test datasets were produced keeping only the relevant features, according to the analysis performed to determine the relative importance of each feature.

Three standard metrics were used for evaluation: Mean absolute percentage error (*MAPE*, Eq. 5), root mean square error (*RMSE*, Eq. 6) and mean absolute error (*MAE*, Eq. 7); $real_i$ represents the measured value for $t = i$, $pred_i$ represents the predicted value and n represents the predicted horizon length.

$$MAPE = 100 \times \frac{\sum_{i=1}^n \left| \frac{real_i - pred_i}{real_i} \right|}{n} \quad (5)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (real_i - pred_i)^2}{n}} \quad (6)$$

$$MAE = \frac{\sum_{i=1}^n |real_i - pred_i|}{n} \quad (7)$$

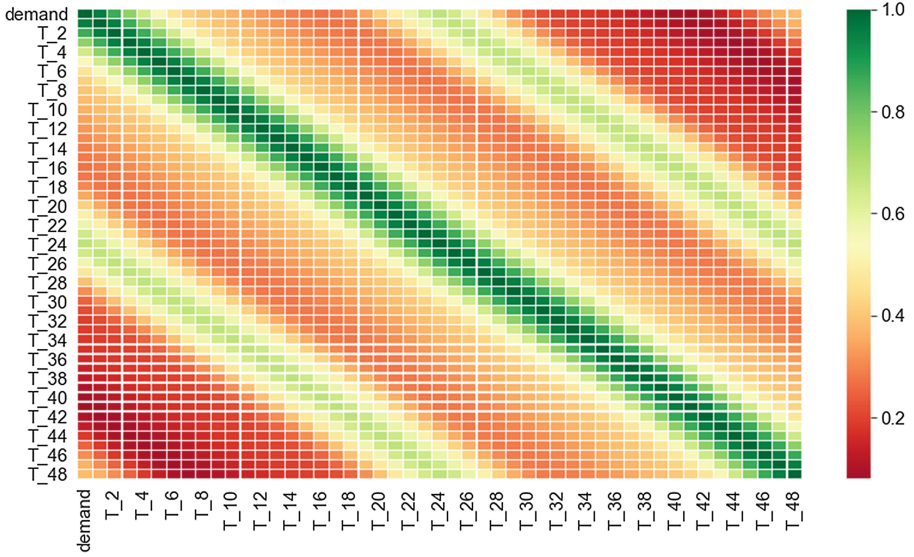


Fig. 1. Correlation diagram between actual demand and 48 last demand measures

Training One Hour Ahead Forecasting Models. Once all data was prepared for model training, a four-step procedure was applied for training and evaluation. The four steps are:

1. Training and test sets were generated in a 3:1 proportion. In this case, the training set considered data from 2014 to 2016 and the test set considered data from 2017.
2. A simple base model was trained for benchmarking. Using the trained model, a recursive feature elimination process was performed. The ten most important features are preserved.
3. Several models were trained and compared with the benchmark model.
4. The best model according to *MAPE*, *RMSE* and *MAPE* metrics was chosen.
5. An optimization of hyperparameters of the best model was performed using grid search techniques.

Finally, the best model found with the optimized hyperparameters was used as a reference to train the 24 h forecasting model.

Twenty Four Hour Model. The best model configured with the best hyperparameters obtained in the previous step, was used to generate twenty four models M_1, M_2, \dots, M_{24} to forecast day ahead hours, applying the following procedure:

1. Training and test sets were generated in a 3:1 proportion. the training set considered data from 2014 to 2016 and the test set considered data from 2017.

2. Model M_i was trained using y_i as output, where y_i consists of the demand value corresponding to i hours ahead, and input X is enriched for models $M_i, i > 2$ with a new column consisting of the $i - 1$ prediction obtained by the trained model M_{i-1}
3. Models M_i are assembled to get a complete model M to forecast the next 24 h altogether.

3.2 Implementation

This section describes the implementation of the approach described in Sect. 3.1.

Computational Platform and Software Environment. Experiments were performed in an HP ProLiant DL380 G9 server with two Intel Xeon Gold 6138 processors (20 cores each) and 128 GB RAM, from the high performance computing infrastructure of National Supercomputing Center Cluster-UY [19].

The proposed approach was implemented in Python. Several scientific packages were used to handle data, train models and visualize results. Used packages included pandas, sklearn, and keras. A generic module was implemented to train various type of models following a pipeline processing. Parameter tuning of the studied models were performed using RandomizedSearchCV and GridSearchCV modules from sklearn. The main details of the implementation of the studied models are provided in the following subsections.

Implementation of One Hour Model. Data preprocessing was already described in Sect. 3.1. All one hour models described in this section use a training set containing data from 2014 to 2016 and a test set containing data from 2017.

Base Model: Linear Regression. A linear regression model was trained to be used as benchmark for the results comparison. A recursive feature selection strategy [4] was also applied on this model to determine the most important features (the rest of features were removed from the dataset).

Ten features were selected based on their relative importance:

- T_1, T_2, T_{24}, T_{25} : demand values lagged.
- *workingday*: flag indicating whether the day of measured value is a working day
- *month*: month on which the measure was taken.
- *hour*: hour of the day on which the measure was taken.
- *dayofweek*: day of the week on which the measure was taken.
- *day*: day of month on which the measure was taken.
- *year*: year on which the measure was taken.

The most relevant past demand values are T_1, T_2, T_{24} , and T_{25} because the current demand is highly correlated with the immediate past demands and also with the demands of the previous day at the same time due to the daily periodicity. The full analysis is presented and discussed in Sect. 4.1.

Selection of the Best Method. Seven regression models were trained including the base model considering the ten most important features, and default parameters, using the scikit-learn API [4]: Linear Regression, MLP, Extra Trees, Gradient Boosting, Random Forest, K-Neighbors and Ridge. These models were evaluated using the *MAPE* metric and the linear regression model was used to determine a baseline performance value. The most accurate method was chosen for further evaluation (this method is called M_{best}).

Optimization of the Best Method. Parameter search techniques were applied to optimize a model based on the best method obtained (M_{best}). The model M_{best} trained with default parameters was optimized using two standard sklearn tools:

- GridSearchCV: The user specify a parameters grid selecting a discrete set of values for each parameter and a model. The tool trains the model in each point of the multidimensional grid generated and finds the best parameters setting according to a predetermined metric.
- RandomizedSearchCV: The user specify a parameter probability distribution and the number of points that must be draw. The tool samples according to the distribution and train the model in each of this points. Then finds the best parameters setting according to a predetermined metric.

The best parameter set obtained for M_{best} results in an optimal model M_{opt} . The main details of the implementation of the complete model based on M_{opt} are described in the next subsection.

3.3 Implementation of the Complete Model

Model M_{opt} was optimized for predicting the next hour and used for predicting any of the following 24 h to build the complete model. This decision was adopted assuming that the forecasting quality of the parameter setting obtained in the previous phase is independent of the hour used as output.

To build the complete model, 24 instances of the optimized model M_{opt} were trained. These instances are called $M_{opt,i}$, defining the model trained to forecast the i_{th} hour ahead. The output y_i used to train the model consisted in the demand value for the i -th hour ahead. For $i > 2$, the input X_i is enriched with a new set of columns consisting of all predictions obtained by models $M_{opt,1}, \dots, M_{opt,i-1}$. Equation 8 describes the hybrid strategy applied to M_{opt} .

$$\begin{aligned}
 pred_{(t+1)} &= M_{opt,(t+1)}(obs_t, obs_{(t-1)}, \dots, obs_{(t-n)}) \\
 pred_{(t+2)} &= M_{opt,(t+2)}(M_{(t+1)}, obs_t, \dots, obs_{(t-n)}) \\
 &\dots \\
 pred_{(t+24)} &= M_{opt,(t+24)}(M_{(t+23)}, M_{(t+22)}, \dots, obs_t, \dots, obs(t-n))
 \end{aligned} \tag{8}$$

The complete model M_{opt} is computed by Eq. 9. Output of the model is a 24 valued vector, one prediction for each hour.

$$M_{opt}(t) = (pred_{(t+1)}, pred_{(t+2)}, \dots, pred_{(t+24)}) \quad (9)$$

4 Experimental Analysis

This section presents the results of the experimental analysis of the proposed computational intelligence methods for day ahead industrial electricity load forecasting.

4.1 Recursive Feature Elimination

A feature selection analysis was performed using the recursive feature elimination tool in sklearn. A model and a number of features are selected, and the tool works by recursively removing features and building a new model (of the type selected) on those remaining features. The accuracy of the new model is used to identify the features or combination of features that contribute the most to predicting the target attribute. The recursive feature selection tool was applied over the linear regression method described in Subsect. 3.2 and studying up to ten features. Figure 2 presents the results of the analysis, reporting the relative importance of the ten most important features.

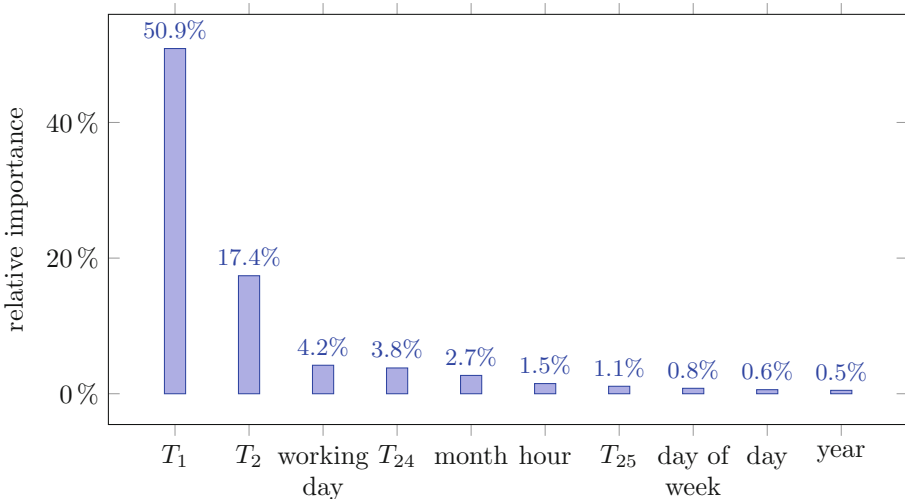


Fig. 2. Relative importance of most important features (percentage values)

4.2 Experimental Results on Preliminary Models

Performance metrics defined in Sect. 3.1 were used to evaluate implementation of one hour models as described in Sect. 3.2. Table 1 reports the obtained results for the studied forecasting models. The best results are reported in cells with green background. Results reported in Table 1 indicate that three methods achieved the best results regarding the analyzed metrics. Focusing on *MAPE*, Extratreesregressor improved over MLP by 4.16% and over RandomForest by 6.54%. Additionally, the training time of Extratreesregressor was approximately three times shorter than RandomForest and six times smaller than MLP. Overall, ExtraTreesRegressor was the most effective model for forecasting the next hour, outperforming all the other methods regarding the three standard metrics studied. According to this result, ExtraTrees was selected as the best method for showing the best performance and a low training time. $M_{best} = ExtraTreesRegressor$.

Table 1. Results for each regression method.

Regression method	MAE	MAPE	RMSE	Score	Time (s)
LinearRegression	127.63	3.60	176.00	0.96	1.72
Ridge	127.63	3.60	176.00	0.97	0.09
KNeighbors	180.54	5.03	253.20	0.93	0.07
RandomForest	180.20	3.21	151.54	0.98	3.1
GradientBoosting	121.97	3.38	166.17	0.97	1.99
MLP	111.08	3.13	154.23	0.97	6.21
ExtraTrees	105.44	3.00	148.61	0.99	1.2

4.3 Parameter Tuning

Parameter tuning techniques described in Sect. 3.2 were applied on the best model M_{best} . The following grid was generated as input for both studied techniques: *n_estimators*: [10, 50, 75, 100, 150], *max_features*: [auto, sqrt, log2], and *max_depth*: [50, 100, 150, 200, 250]. GridSearchCV achieved the best results. The best parameter setting found by the algorithm was *n_estimators* = 50, *max_features* = auto and *max_depth* = 250, improving 14% on the *MAPE* results over the second best configuration.

4.4 Experimental Results After Parameter Tuning

Table 2 reports results of the *ExtraTreesRegressor* model before and after parameter tuning. The best results are highlighted (cells with green background).

Results show that the numerical results improved considerably for the three studied metrics. In particular, *MAPE* reduced from 3.00% to 1.79%. The performance improvement just demanded a negligible increase on training time increases after parameter tuning from 1.2s to 1.7s.

Table 2. Comparative results of ExtraTrees before and after parameter tuning.

Regression method	MAE	MAPE	RMSE	Score	Time (s)
ExtraTrees before tuning	105.44	3.00	148. 61	0.99	1.2
ExtraTrees after tuning	87.52	1.79	111.08	0.99	1.7

4.5 Experimental Results of the Complete Model

The forecast accuracy of the final model was validated by applying a metric that extends *MAPE*. Let $MAPE_h$ be the *MAPE* value for a predicted horizon h , the extension of *MAPE* to the complete testing set is defined by Eq. 10.

$$MAPE_{tot} = \frac{\sum_{i=1}^k MAPE_h}{k} \tag{10}$$

Table 3 reports the results for each of the 24 models. The expected behaviour is that the models trained for highly correlated hours in the future respect to the current hour, perform best. This fact is due to predictability, and it is enhanced when the correlation between input features and predicted values is higher. According to Fig. 1, highly correlated demand values correspond to the immediately preceding hours and from the same hours of the day before.

Analyzing the obtained results for the $MAPE_{tot}$ metric for each one of the 24 hourly models, the performance got worse from $i = 1$ to $i = 17$ and then improved from $i = 18$ to $i = 24$. These results show that highly correlated demand values performed better, as expected.

Table 3. $MAPE_{tot}$ score for each $ET_{opt,i}$ single hour model.

	hour											
	1	2	3	4	5	6	7	8	9	10	11	12
$MAPE_{tot}$	1.79	1.84	1.90	1.97	2.09	2.19	2.39	2.52	2.68	2.75	2.80	2.86
	hour											
	13	14	15	16	17	18	19	20	21	22	23	24
$MAPE_{tot}$	2.93	3.02	3.05	3.08	3.09	3.02	2.88	2.77	2.63	2.49	2.32	2.17

Finally, the complete model ET_{opt} was applied. A day-ahead hourly forecast load curve was generated for each time window for the testing set and the $MAPE_{tot}$ value was calculated.

The final result for the complete model was $MAPE_{tot} = 2.55\%$. This result implies that the model obtained for the day ahead demand forecasting of the industrial pole analyzed incurs in an error that is considered very low for most of the studies that rely on these types of models [13,15]. Figure 3 presents an example of the real demand curve and the predicted demand curve using the best model, for the testing set considered in the experiments.

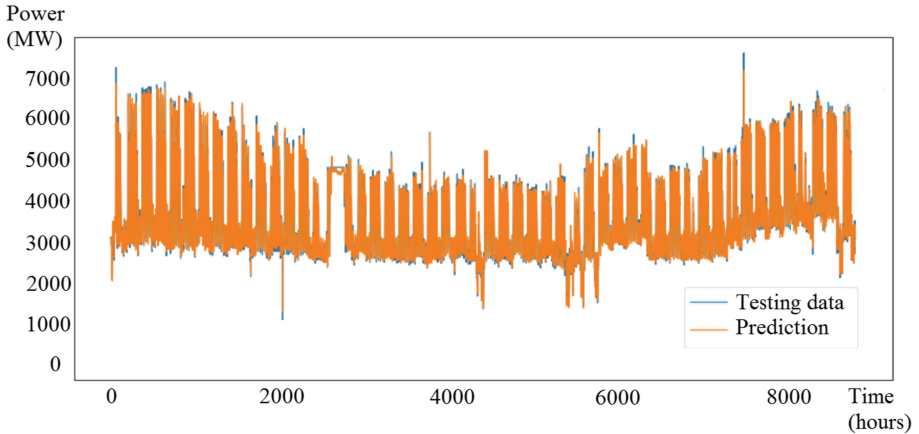


Fig. 3. Predicted demand and testing data curves

5 Conclusions and Future Work

This article presented an approach to address the problem of day ahead electricity load forecasting. Several machine learning models was presented and studied for next hour forecasting. Recursive feature selection was applied to select most relevant features to train the studied models. After a comparative evaluation, the best model was optimized using random search and grid search techniques. With the optimized model for single hour prediction, an hybrid strategy (direct and recursive) was applied to build a complete day ahead electricity load hourly forecasting model.

An extension of $MAPE$ metric was used to evaluate this complete model for the testing set, obtaining a value of $MAPE_{tot} = 2.55\%$. This result shows that the proposed algorithm is effective for addressing the problem of day-ahead industrial demand forecasting.

The main lines for future work are related to extend the analysis to other data sets of industrial poles with different demand profiles, and apply the proposed approach to residential demand forecasting, including other relevant features (e.g., related to weather, such as temperature, humidity, and wind speed, which have impact on residential demand [17]). Deep learning techniques (e.g., recurrent/long-short term memory neural networks) should be considered for future work, since they can provide accurate results in scenarios that are difficult for other simpler methods, i.e. when handling large volumes of historical data.

References

1. Amjady, N., Keynia, F.: Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm. *Energy* **34**(1), 46–57 (2009)
2. Bashir, Z., El-Hawary, M.: Applying wavelets to short-term load forecasting using pso-based neural networks. *IEEE Trans. Power Syst.* **24**(1), 20–27 (2009)
3. Bellman, R.: *Dynamic Programming*. Princeton University Press, Princeton (1957)
4. Buitinck, L., et al.: API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122 (2013)
5. Burger, E., Moura, S.: Gated ensemble learning method for demand-side electricity load forecasting. *Energy Build.* **109**, 23–34 (2015)
6. Chen, Y., Luh, P., Rourke, S.: Short-term load forecasting: similar day-based wavelet neural networks. In: *7th World Congress on Intelligent Control and Automation*, pp. 3353–3358 (2008)
7. De Felice, M., Alessandri, A., Ruti, P.M.: Electricity demand forecasting over italy: potential benefits using numerical weather prediction models. *Electr. Power Syst. Res.* **104**, 71–79 (2013)
8. Diniz, A., Costa, F., Maceira, M., dos Santos, T., Dos Santos, L., Cabral, R.: Short/mid-term hydrothermal dispatch and spot pricing for large-scale systems—the case of Brazil. In: *Power Systems Computation Conference*, pp. 1–7 (2018)
9. Do, L., Lin, K.H., Molnár, P.: Electricity consumption modelling: a case of germany. *Econ. Model.* **55**, 92–101 (2016)
10. Dudek, G.: Pattern-based local linear regression models for short-term load forecasting. *Electr. Power Syst. Res.* **130**, 139–147 (2016)
11. Fan, S., Chen, L., Lee, W.J.: Machine learning based switching model for electricity load forecasting. *Energy Convers. Manag.* **49**(6), 1331–1344 (2008)
12. Hagan, M., Behr, S.: The time series approach to short term load forecasting. *IEEE Trans. Power Syst.* **2**(3), 785–791 (1987)
13. Karrthikeyaa, S., Vijayaraghavan, V.: A two-fold machine learning approach for efficient day-ahead load prediction at hourly granularity for NYC. In: *Advances in Information and Communication*, vol. 2, p. 84 (2019)
14. Kim, C., Yu, I., Song, Y.: Kohonen neural network and wavelet transform based approach to short-term load forecasting. *Electr. Power Syst. Res.* **63**(3), 169–176 (2002)
15. Lahouar, A., Slama, J.: Day-ahead load forecast using random forest and expert input selection. *Energy Convers. Manag.* **103**, 1040–1051 (2015)

16. Lazos, D., Sproul, A., Kay, M.: Optimisation of energy management in commercial buildings with weather forecasting inputs: a review. *Renew. Sustain. Energy Rev.* **39**, 587–603 (2014)
17. Luján, E., Otero, A., Valenzuela, S., Mocskos, E., Steffanel, L., Nesmachnow, S.: An integrated platform for smart energy management: the CC-SEM project. *Revista Facultad de Ingeniería* (2019)
18. McKinney, W.: Data structures for statistical computing in python. In: *Proceedings of the 9th Python in Science Conference*, pp. 51–56 (2010)
19. Nesmachnow, S., Iturriaga, S.: Cluster-UY: scientific HPC in Uruguay. In: *International Supercomputing in Mexico* (2019)
20. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
21. Qamber, I.: Peak load estimation studies in several countries. *Electr. Power Syst. Res.* **1**(2) (2017)
22. Resende, L., Soares, M., Ferreira, P.: Electric power load in Brazil: view on the long-term forecasting models. *Production* **28**, 83–99 (2018)
23. Silva, L.: A feature engineering approach to wind power forecasting: GEFCom 2012. *Int. J. Forecast.* **30**(2), 395–401 (2014)
24. Soliman, A., Al-Kandari, A.: *Electrical Load Forecasting: Modeling and Model Construction*. Elsevier, Amsterdam (2010)
25. Son, H., Kim, C.: Short-term forecasting of electricity demand for the residential sector using weather and social variables. *Resour. Conserv. Recycl.* **123**, 200–207 (2017)
26. Taylor, J., McSharry, P.: Short-term load forecasting methods: an evaluation based on European data. *IEEE Trans. Power Syst.* **22**(4), 2213–2219 (2007)