Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 2636 – 2653
© TÜBİTAK
doi:10.55730/1300-0632.3960

Load2Load: Day-ahead load forecasting at aggregated level

Mustafa Berkay YILMAZ*

Department of Computer, Faculty of Engineering, Akdeniz University, Antalya, Turkey

Received: 03.04.2022 • **Accepted/Published Online:** 05.10.2022 • **Final Version:** 28.11.2022

Abstract: A reliable and accurate short-term load forecasting (STLF) helps utilities and energy providers deal with the challenges posed by supply and demand balance, higher penetration of renewable energies and the development of electricity markets with increasingly complex pricing strategies in future smart grids. Recent advances in deep learning have been successively utilized to STLF. However, there is no certain study that evaluates the performances of different STLF methods at an aggregated level on different datasets with different numbers of daily measurements.

In this study, a deep learning STLF architecture called Load2Load is proposed for day-ahead forecasting. Different forecasting methods have been evaluated and compared on two datasets with different temporal resolutions and features. An additive ensemble method as well as a selective ensemble method that selects the outputs of different forecasters in an hourly manner are proposed. Moreover; a modified sequential forward feature selection algorithm is proposed, resulting in better performance with a much smaller number of features.

Numerical results show that the proposed Load2Load architecture has a competing performance compared to other advanced forecasters. When used together with the proposed ensemble methods, Load2Load can significantly improve the forecasting performance. The proposed feature selection algorithm results in better performance for the majority of the cases while reducing the dimensionality. According to the results with two different datasets; the proposed methods are shown to be robust to temporal resolutions, feature types and sequence lengths.

Key words: Day-ahead electrical load forecasting, aggregated-level forecasting, deep learning, forecaster ensemble, generative adversarial network, feature selection

1. Introduction

The accurate estimation of electrical load demand can be greatly beneficial in the optimal planning of energy systems and in making proper operational decisions. Electricity load forecasting can be classified into three main categories, according to the period as long-term: 1-50 years; mid-term: one month to one year; short-term: estimates of the day or week ahead of electricity consumption. Huge number of operating decisions require STLF. In general, STLF supports power system operation management planning, scheduling of distributed generation plants, contingency and load flow analysis at power systems and maintenance activities. STLF is particularly important for maintaining power supply-demand balance for electric energy utilities. STLF can mainly be carried out in two categories: the aggregated level and the building (individual) level. The aggregated level forecasting gives an estimate of the total load demand for a group of users in a particular level such as system level and regional level.

*Correspondence: berkayyilmaz@akdeniz.edu.tr

1.1. Related work

A variety of STLF methods are used for electrical load forecasting which is generally based on statistical or machine learning approaches [1–8]. Time series methods [9] and multiple linear regression (MLR) methods [10] are within the scope of the statistical methods. In the early times of load forecasting studies, time series approaches were widely applied especially at the peak load forecasting studies [11]. MLR is still preferred today, however usually outperformed by more advanced machine learning techniques.

Machine learning techniques such as artificial neural network (ANN), which has been started to be used in the early 90's [12], regression tree [13], support vector machine (SVM) [14], k-nearest neighbors (kNN) algorithm [15], recurrent neural network (RNN) [16], convolutional neural network (CNN) [17] and different machine learning-based ensemble models have been gaining much interest recently due to suitability to learn complex and nonlinear relationships hidden in data and their satisfying performance especially in high accuracy, particularly in the field of STLF [18].

It could be seen from the past studies that, an insufficient number of aggregated-level load estimation studies have been performed when compared with household-level studies until recent times. With the development of smart meters and thus the increase in data sizes in the smart grid, aggregated load forecasting based on smart meters data has become one of the most popular topics of recent years [19].

A review and analysis study of regression and machine learning models on STLF of Kensington Campus and Tyree Energy Technologies Building (TETB) at the University of New South Wales (UNSW) which are considered as commercial buildings can be found in [6]. In that study, regression models have been reviewed. The models have also been tested on the data of commercial buildings. It has been seen that most of the machine learning models outperformed the MLR models. However, regression models have shown better performance compared to machine learning models on forecasting daily peak electricity demand. In [20], a feed-forward deep neural network (DNN) model has been compared with some of the trend machine learning models like random forest and gradient boosting machine models. It has been shown that the deep model outperformed these models in terms of forecasting accuracy for three Chinese cities' electricity demand data.

In [18], a comparison between three forecast methods; namely MLR, random forest (RF), and gradient boosting (GB) on hourly load forecasting of southern California has been performed. It is concluded that GB model has generally outperformed the MLR and RF models. A review of the most widely used machine learning models such as SVM, kNN, random forest and ANN on STLF at microgrid level has been presented in [21]. A comprehensive comparison study of 24-h-ahead STLF [7] groups the methods into three main categories; namely the time series model, classical regression models, and the deep learning models. This study has been realized as a 24-h-ahead STLF study of provincial load in Jiangsu Province, China. The results interestingly showed that linear models and their variants; which are MLR, multivariate adaptive regression splines (MARS) and support vector regression (SVR) with the linear kernel outperformed other models. Moreover, a layer-wise trained DNN has forecasted more accurately than shallow ANN but still could not provide a higher accuracy than linear models.

ML methods such as ANN, MLR, adaptive neuro-fuzzy inference system (ANFIS) and SVM have been investigated to forecast electricity demand in Cyprus for both short- and long-term period [22]. The results have indicated that ANN has showed better performance in prediction errors (0.97%, 1.67%) and root mean square error (RMSE) of (7.67, 14.91) than other models in short-term period. In [23], the authors have developed an LSTM-RNN-based univariate model to forecast the aggregated demand of France metropolitan's in short- and medium-term time horizons. They have compared their model with common machine learning approaches and have indicated that LSTM-based model outperformed other models which is optimized with hyper-parameter tuning. Moreover, by utilizing the best features, optimal lags, layers, and training various LSTM configurations they have increased the accuracy even more.

Choi et al. proposed an STLF method based on residual network (ResNet) and LSTM [24]. ResNet is utilized to extract latent features of daily and weekly load data. Then, LSTM is applied to train the encoded feature vector with and make prediction suitable for volatile load data. The proposed model is compared with other deep learning models, which are multi-layer perceptron (MLP), ResNet, LSTM and ResNet/MLP combined model. The results show that the proposed model has 21.3% mean absolute percentage error (MAPE) improvement overall. In this proposed study, ResNet is directly used for regression to forecast the load, different from [24].

A method based only on input data for consumption in Czech Republic is described by Uher et al. [25]. Additional influences such as temperature, wind, gross domestic product are ignored. Local polynomial regression, ANN, Gaussian process, linear regression, and polynomial regression techniques are used to create a predictive model. The best results are obtained with a local polynomial regression algorithm. Daily prediction RMSE is reported as 5.77%.

A probabilistic forecasting framework based on Bayesian neural networks (BNNs) with optimized initialization for one-day ahead forecasting of operational demand across the National Electricity Market (NEM) of Australia is studied in [26]. MAPE values are reported for different regions separately such as NSW (5.18%), VIC (6.40%), QLD (4.06%), SA (8.91%), TAS (4.21%). Average MAPE for all regions is 5.75%.

A recent study presents an adaptive hybrid ensemble, CMKP-EG-SVR for STLF problem [27]. For this purpose, median filtering is utilized at an initial phase of data preprocessing to reduce the high-frequency parts while detecting and correcting the outliers and missing data. Pearson's correlation, kNN algorithm and a particular calendar grouping based on day-type encoding are used for selecting relevant features and extracting similar patterns. The electricity demand data from the Australian state of New South Wales are employed for an extensive evaluation under both normal and anomalous load conditions. MAPE values in the range of 1.03%-4.26% are reported.

A load forecasting model to introduce a feature extraction module that is combined with the variational mode decomposition (VMD) and the variational autoencoder (VAE) is proposed in [28]. In this combination, VMD is utilized for decomposing the load series and VAE is used to eliminate the redundant information from each decomposed series. The proposed model is shown to achieve accurate predictions with two real data sets from China, reporting MAPEs for one-step-ahead prediction to be 1% (Nanjing) and 0.8% (Taixing), respectively.

Turkey's 24-h-ahead load forecasting without meteorological data is studied in [29]. ANN, wavelet transform and ANN, wavelet transform and radial basis function (RBF) neural network (NN), empirical mode decomposition (EMD) and RBF NN structures are used for STLF procedures. The average MAPE result over 2009 and 2010 for ANN is 3.74, wavelet transform and ANN is 3.96, wavelet transform and RBF NN is 2.95, EMD and RBF NN is 3.58.

Deep learning has been successfully used for STLF in recent times. A survey of deep learning studies on solar load forecasting has been carried out by Akinola et al. [30]. A comparison of stacked denoising auto-encoder (SDAE) and CNN deep learning architectures is available by Chelabi et al. [31]. Feedforward and recurrent neural networks, sequence-to-sequence models and temporal CNNs along with architectural variants on four real-world datasets are experimented for one-day-ahead prediction in [32].

Electrical load data is proposed to be transformed from 1D series to 3D images and the problem is transformed from future series forecasting to missing patch inpainting in [33] for STLF. An RNN is presented to model the temporal trends in the series by convolutional operations on the spatial neighbourhood in the images.

Generative adversarial networks (GAN) have recently been studied by a number of works for electrical load forecasting. Different GAN architectures are compared in [34] such as conditional GAN (cGAN), deep convolutional GAN, least squares GAN and Wasserstein GAN (WGAN) for day-ahead electric demand forecasting. The proposed GAN models provided an average MAPE of 4.99%. Comparison of normalizing flows with GANs and variational autoencoders is provided in [35] for weather-based photovoltaics, wind power and load forecasting scenarios. A new method called E-GAN which combines a physics-based model (EnergyPlus) and a data-driven model (GAN) to predict the daily power demand for buildings at a large scale is proposed in [36]. E-GAN is reported to predict power demand with 5% MAPE. DeepAR and Wavenet architectures are investigated as the representative of deep generative models for the purpose of STLF in [37]. Deep generative models are reported to perform better compared to other baseline models such as ARIMA, machine learning and baseline neural networks. Huang et al. [38] utilize Wasserstein GAN for multinodes interval electric vehicle day-ahead charging load forecasting. MAPE value of 17.7% is reported.

A Bayesian training method to enhance the robustness of deep-learning-based load forecasting models towards adversarial attacks is proposed in [39]. Bayesian training is reported to improve the load forecasting robustness against various attacking objectives without compromising the prediction performance.

In the author's previous study [40], a deep RNN Bi-LSTM model was developed to forecast 24-h-ahead aggregated load profile for the purpose of operational scheduling in a reconfigurable microgrid (MG). The results showed that the RNN Bi-LSTM model outperformed other methods in the recent literature such as feed-forward neural network (FFNN), deep CNN, Copula deep belief network (DBN), and parallel CNN-RNN.

2. Proposed work

In this study, day-ahead electrical load forecasting is in focus, by implementing and comparing the performance of numerous forecasting techniques. Contributions with respect to the author's previous work [40] is summarized in what follows.

A new STLF architecture called Load2Load based on a GAN model is proposed. To the best of our knowledge, this is the first time such an architecture is used for numerical regression, specifically STLF. Different model designs that are better suitable for different kinds of forecasters have been proposed and investigated.

In addition to Czech data consisting of 24 measurements per day, Australian data consisting of 48 measurements per day is utilized for evaluation. Because each dataset has its own dynamics with the features; a modified sequential forward feature selection algorithm is proposed, resulting in better performance for some of the forecasters, with much smaller number of features.

Ensembles of the same kind of forecasters with different hyper-parameters, as well as different kinds of forecasters are examined. A selective ensemble technique is proposed, which selects the outputs of different forecasters in an hourly manner. According to the experimental results; the proposed ensemble method shows that, Load2Load can improve the results when used together with other architectures.

The rest of the paper is organized as follows. Section 2.1 describes the forecaster architectures and Section 2.2 depicts the model designs of those architectures as utilized in this work. Section 2.3 describes the proposed ensemble strategy. Feature selection is explained in Section 2.4. Experimental protocol and the results are given in Section 3. Finally, conclusions are provided in Section 4.

2.1. Forecaster architectures

Forecasters take as input last six days' hourly data of the features, where the sequence length s is 144 for hourly measured data (Czech data) or 288 for half-an-hourly measured data (Ausdata). Load2Load is proposed in this work and compared with several different forecasters whose architectures are designed and optimized for STLF. H depicts the number of forecasts per day where $H = H_{measured}$, $H_{measured}$ is the number of measurements per day in the database (24 for hourly measured Czech data or 48 for half-an-hourly measured Ausdata). Adjustable parameters of the forecasters are determined according to the protocol defined in Section 3.2, using a separate subset of data.

2.1.1. Load2Load

A network called Load2Load which takes as input a sequence image and outputs the forecasted sequence as an image is proposed. To the best of our knowledge, this is the first time such an architecture is used for numerical regression, specifically STLF. Load2Load network both learns a mapping from the input sequence image to the output sequence image and a loss function to train this mapping. This is done by learning a structured loss that penalizes a structure that differs from the current output and the desired output. The idea of GAN in general is learning to generate new data with the same statistics as the training set [41]. There are two components in a GAN. The generative network generates new samples while the discriminative network discriminates them. The two models are trained simultaneously in an adversarial process where the generator seeks to create better forgery samples and the discriminator seeks to better identify the forgery samples.

The discriminator is a deep CNN that performs conditional-image classification. It can take the input sequence and the corresponding forecast sequence and then predicts the likelihood of whether the target image is a correct or incorrect forecast sequence. Input and forecast sequence images are fed as two channels of a single image. Because only one feature sequence is to be forecasted, forecast input sequence is repeated vertically to be compatible with the first channel input number of features. During forecasting, generated image row-base average is calculated to decide final forecasts through repeated forecasts in the image. The discriminator model is trained on real and generated sequence images.

The generator is an encoder-decoder model using a U-Net architecture. It takes a source image and generates a target image. This is achieved by downsampling the input to a bottleneck layer (encoder), then upsampling the bottleneck representation to the size of the output (decoder). The U-Net model uses skip-connections between the encoding layers and the corresponding decoding layers. The generator model is trained from the discriminator model. It is updated to minimize the loss predicted by the discriminator for generated sequences that are marked as real. As a result, it learns to generate more realistic sequence images. The generator is updated to minimize the loss between the generated sequence image and the target sequence image as well.

Overall Load2Load model stacks the generator on top of the discriminator. A source sequence is an input to the generator and to the discriminator, while the output of the generator is connected to the discriminator as the corresponding target sequence. The discriminator finds the probability that the output is a correct forecast sequence of the source.

The trained Load2Load function is $F_{Load2Load}(\hat{x})$ for some test sample $\hat{x}$. Figure 1 demonstrates the discriminator and Figure 2 demonstrates the generator of the proposed Load2Load components.

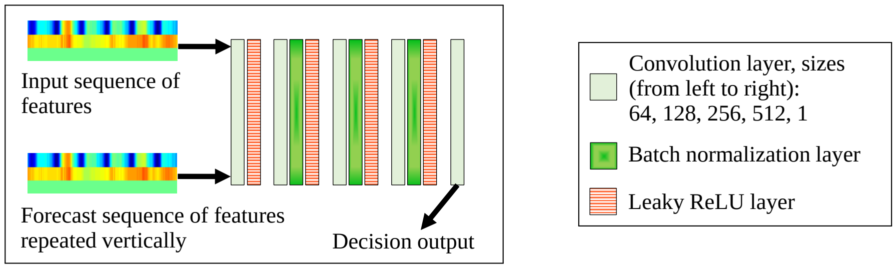


Figure 1. Discriminator of Load2Load architecture.

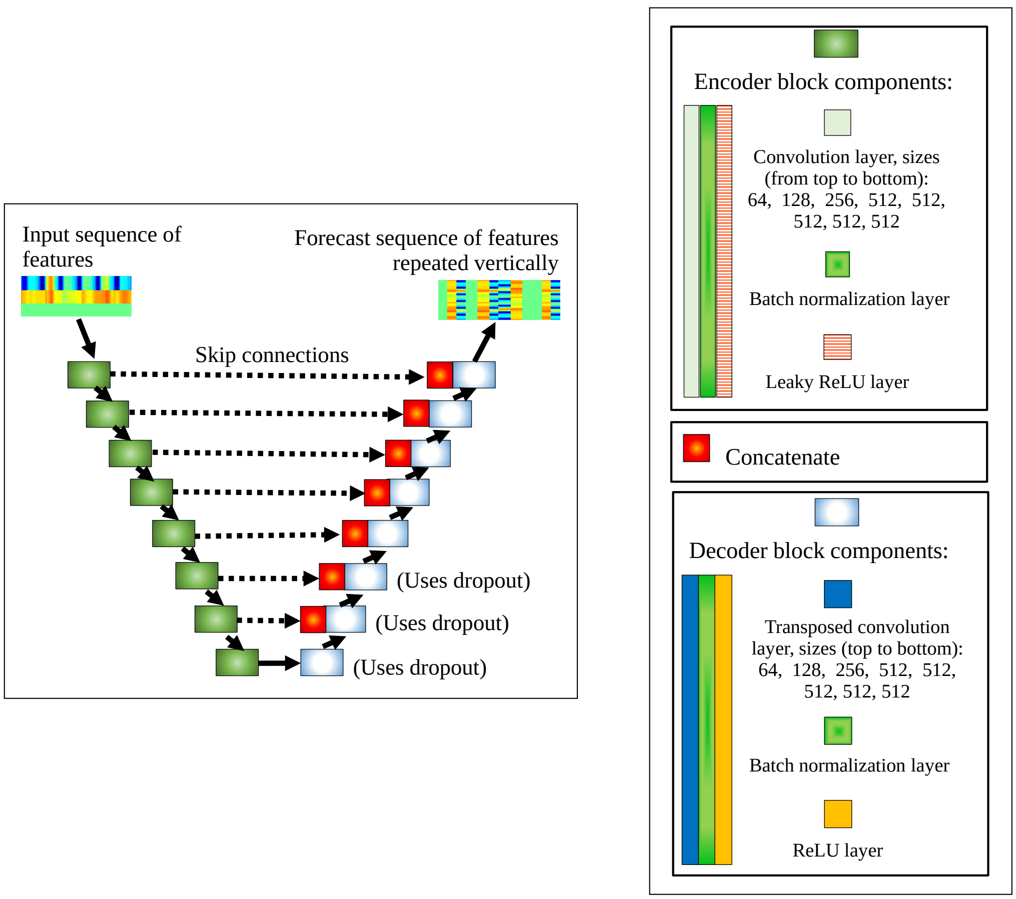


Figure 2. Generator of Load2Load architecture.

2.1.2. Other architectures

Logistic regression model Logistic regression model (LRM) describes a linear relationship between a dependent variable y and one or more independent variables (features) matrix X of observations.

K-nearest neighbors kNN is a nonparametric supervised machine learning algorithm that can be used to solve both classification and regression problems. In kNN regression, the output value is the average of the values of k nearest neighbors. In this work, the simplest kNN architecture of $k = 1$ and Euclidean distance as the metric is utilized.

Support vector regression SVM works well for support vector classification (SVC) or SVR. For SVR, ν-SVR [42] is chosen as a linear SVM solver which is faster to train with a less number of support vectors.

Regression tree Regression tree is a decision tree for which the output can take continuous values. Regression tree provides promising results where rule-based decisions can be drawn from the data.

Convolutional neural network CNNs have shown great performance on signal processing, classification, representation learning as well as many other tasks. Two different CNNs CNN_1 and CNN_2 are considered and additive ensemble of them (Section 2.3.1) is investigated, shown as $CNNs = CNN_1 \boxplus CNN_2$ in the results.

Throughout the text; CL denotes the convolution layer, REL denotes rectified linear unit (ReLU) layer, APL denotes average pooling layer, MPL denotes max pooling layer, BL denotes batch normalization layer, DL denotes dropout layer, FL denotes fully connected layer and RL denotes regression layer.

CNN_1 consists of the following layers: CL (8 filters), BL, REL, CL (16 filters), BL, REL, MPL, CL (32 filters), BL, REL, MPL, CL (64 filters), BL, REL, FL (5096), FL (512), FL (144), RL.

CNN_2 consists of the following layers: CL (8 filters), BL, REL, APL, CL (16 filters), BL, REL, APL, CL (32 filters), BL, REL, CL (32 filters), BL, REL, DL, FL (144), RL.

Recurrent neural network RNN is a kind of neural network that allows previous outputs to be used as inputs while having hidden states. RNN is suitable for sequence inputs because they can process input of any length. Two RNNs of bidirectional long short-term memory layers (BiLSTM) with similar structures but different numbers of hidden units are considered. In RNN_1, number of units in BiLSTM layer is 30 whereas in RNN_2 number of units in BiLSTM layer is 100. Additive ensemble of RNN_1 and RNN_2 (Section 2.3.1) is investigated, shown as $RNNs$ in the results.

Throughout the text; SL denotes the sequence input layer, BiL denotes BiLSTM layer. The proposed RNN consists of the following layers: SL, BiL, FL, RL.

Feedforward artificial neural network Feedforward ANN is a simple type of ANN where the connections between the units do not form a cycle. Two different ANNs are proposed. ANN_1 is a simple ANN with one hidden layer of 12 hidden units, whereas ANN_2 is a more complex ANN with five hidden layers with a decreasing number of hidden units as 50, 40, 30, 20 and 10. Additive ensemble of ANN_1 and ANN_2 (Section 2.3.1) is investigated, shown as $ANNs$ in the results.

Residual network ResNet consists of the main branch with convolutional, batch normalization, and ReLU layers connected sequentially; in addition, residual connections that bypass the convolutional units of the main branch [43]. The outputs of the residual connections and convolutional units are added element-wise. In the skip connections where the layer activations in the convolutional units change the size, the activations in the skip connections also change size. The width of the proposed ResNet model is three. Figure 3 demonstrates the proposed ResNet model.

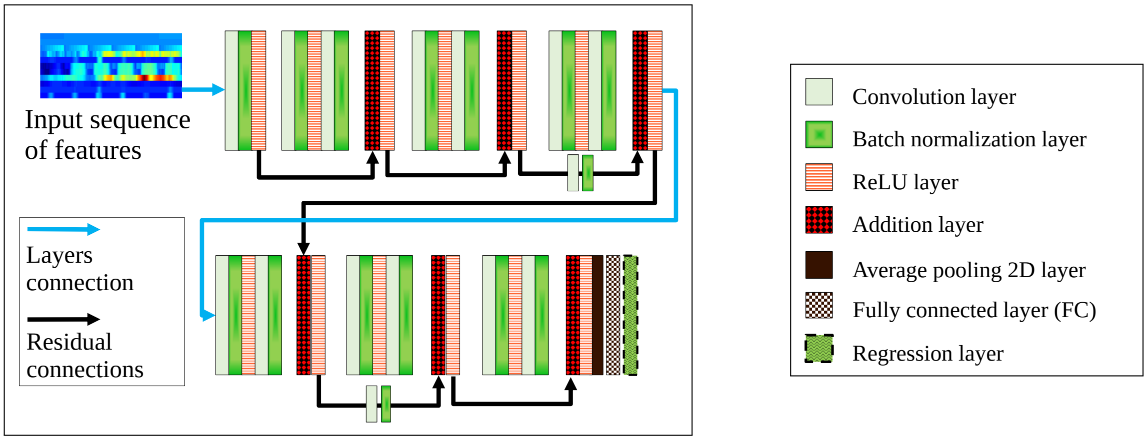


Figure 3. ResNet architecture.

2.2. Forecaster model designs

Forecaster architectures are designed with two different schemes depending on the characteristics of the forecaster. Input sequence of features are either represented in $s \times f$ 2D (for CNN and RNN) or as a stacked vector of length $s \times f$ in 1D (for other forecasters), where s is the sequence length and f is the number of features.

2.2.1. $Model_1$

$Model_1$ forecasts the next day's all hourly values at one shot. Such day-ahead models are trained only to predict the sequences that start the first hour of a new day (i.e. from 00:00 to 23:00), but not in-between sequences (such as from 14:00 to the following day's 13:00) as the goal is to forecast all hourly values of the next day, reducing the amount of training data. Only one forecaster of a kind is trained that constitutes the $Model_1$. Load2Load, ResNet, CNN, and RNN forecasters are built following this approach.

2.2.2. $Model_2$

This model forecasts the next day's all hourly values utilizing specialized forecaster instances for each hour of the next day. In total, H different forecasters $\{Model_{2_1}, Model_{2_2}, ..., Model_{2_H}\}$ are trained that constitute $Model_2$. Like $Model_1$, only the sequences that start the first hour of a new day are utilized in training. MLR, SVR, kNN and tree forecasters are built following this approach.

2.3. Forecaster ensemble

Two different ensemble methods are investigated and found to be useful. The first one is based on weighted sum of the predictions and the second one is specifically proposed for the problem which selects different forecasters based on the hour to be predicted.

2.3.1. Additive ensemble

Additive ensemble fuses the outputs of different forecasters following a weighted sum rule and is denoted with $\boxplus$:

$$F_{add}(\hat{x}) = F_1(\hat{x}) \boxplus F_2(\hat{x}) \boxplus ... \boxplus F_N(\hat{x}) = \sum_{k=1}^{N} w_k F_k(\hat{x}), \tag{1}$$

where F_k is the forecast output of the particular forecaster model k (e.g., CNN, RNN...), w_k is the corresponding weight ($0 \leq w_k \leq 1$ and $\sum_{k=1}^{N} w_k = 1$) and N is the total number of considered forecasters.

The ensemble problem is translated into optimizing the weights w_k for the best forecasting performance. However, one can simply set the weights as $w_k = 1/N$ to obtain the average of forecasts obtained from all forecasters. In this study, fusion weights w_k are optimized according to the validation set performance. Utilized optimization process is based on a brute-force search of the best-performing weight set within the sensitivity of 0.05. This value is heuristically selected as a smaller value would yield to overfitting to the validation set and a slower search; whereas a bigger value may end up missing the best weight set.

The additive ensemble is used both to improve the same kind of forecasters (e.g., ensemble of $F_{RNN_1} \boxplus F_{RNN_2}$ $(RNNs)$) and to improve the performance of different kinds of forecasters (e.g., $F_{RNNs} \boxplus F_{SVR}$).

2.3.2. Selective ensemble

From the hourly error analysis performed on the validation set; it has been observed that among two of the well-performing forecasters (SVR and RNN), SVR performs relatively better for the hours where the values do not deviate much from day to day and RNN performs relatively better for the hours where the values tend to deviate more from day to day. Depending on this observation, a selective ensemble is proposed for the above-mentioned forecasters:

$$F_{sel}(\hat{x}^h) = E_{RNN}(h) F_{RNN}(\hat{x}^h) + E_{SVR}(h) F_{SVR}(\hat{x}^h), \tag{2}$$

where h is an integer denoting the forecasted hour ($0 \leq h < H$), $\hat{x}^h$ is some input test sample which specifically represents an observation at hour h (previous s hours' p features), F_{RNN} is the forecasting function of RNN, F_{SVR} is the forecasting function of SVR, E's are the selective ensemble functions,

$$E_{RNN}(h) = \left\{ \begin{array}{lll} 1 & \text{for} & h \in H_{RNN}^{sel} \\ 0 & & otherwise \end{array} \right. , \tag{3}$$

$$E_{SVR}(h) = 1 - E_{RNN}(h). \tag{4}$$

Set of hours to be selectively forecasted by RNN, H_{RNN}^{sel} is experimentally found from the validation set and depends on the database and the forecasted feature, as detailed in Section 3.4. Selective ensemble is denoted with $\boxdot$ and investigated to further improve the best performing forecasters, for example additive ensemble of RNNs ($F_{RNN_1} \boxplus F_{RNN_2}$) and SVR as $(F_{RNN_1} \boxplus F_{RNN_2}) \boxdot F_{SVR}$.

2.4. Feature selection

Feature selection on the validation set based on a modified sequential forward selection algorithm is proposed. Feature selection is analyzed only with SVR for the speed and performance advantages according to validation set tests with the expectation of generalizing well to other forecasters with the same feature subset. Mean absolute error (MAE) metric is used to compare the performance of candidate features $\hat{k}$ for its simplicity. The proposed algorithm consists of the following steps:

Algorithm 1 Feature selection

Initialize the feature set $Z \leftarrow \emptyset$ and the residual feature set $R \leftarrow \epsilon$ where ϵ is the complete set of features
while $R \neq \emptyset$ **do**
  $k \leftarrow argmin_{\hat{k}} MAE(F_{SVR}(X_{Z \cup \hat{k}}))$ where MAE is the function that measures the MAE of the given model using forecast outputs and X is the observations matrix in the validation set with the denoted feature subset
  if $MAE(F_{SVR}(X_{Z \cup k})) < MAE(F_{SVR}(X_Z))$ **then**
    Update $Z \leftarrow Z \cup k,\ R \leftarrow R \backslash k$
  else
    return the feature set Z
  end if
end while

3. Experimental results

3.1. Datasets

3.1.1. Czech data

Czech data is a one-h-resolution (24 measurements per day) dataset that consists of year, day, month, hour, wind power plant (WPP), photovoltaic power plant (PVPP), pumping load, load, wind speed, temperature, direct horizontal radiation and diffuse horizontal radiation features from 01.01.2012 to 31.12.2016 in the Czech Republic. The goal is to predict the electrical energy demand. Year information is not used in experiments; hence the considered features are:

1) Day 2) Month 3) Hour 4) WPP 5) PVPP 6) Load(pumping) 7) Load 8) windspeed (10m) 9) temperature 10) radiation (direct horizontal) 11) radiation (diffuse horizontal)

3.1.2. Ausdata

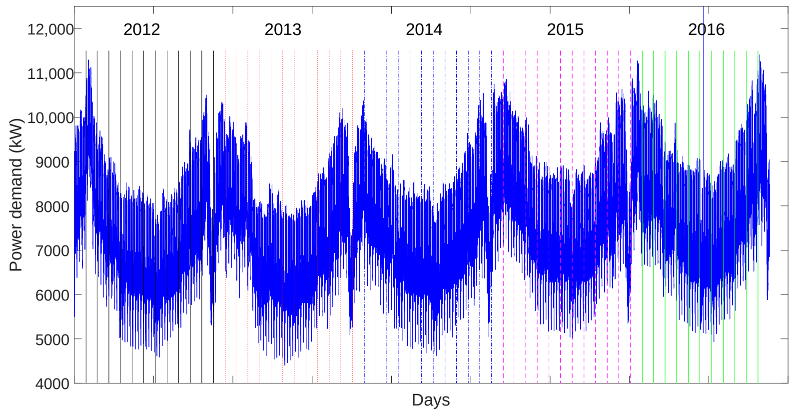
Ausdata is a thirty-min-resolution (48 measurements per day) dataset that consists of day, month, year, hour, dry bulb temperature, dew point, electric price, system load, wet bulb temperature, and humidity features from 01.01.2006 to 31.12.2010 in Australia; provided by Australian Energy Market Operator (AEMO) and Bureau of Meteorology (BOM) for Sydney/New South Wales (NSW). For predicting the electrical energy load using this dataset, the following features are considered:

1) Day 2) Month 3) Hour 4) DryBulb 5) DewPnt 6) ElecPrice 7) SYSLoad 8) WetBulb 9) Humidity

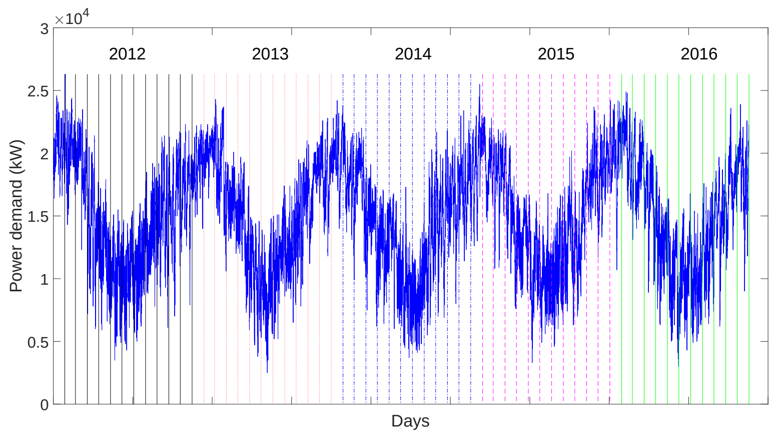
Complete electrical load plots of the two datasets are given in Figure 4. Months are separated with vertical lines for better visualization.

3.2. Cross validation

Days are divided into five equi-sized folds for cross validation. In each test, one of the folds is left out for testing and all other folds are used for training the models. **First fold is devoted for development purposes**; including ensemble weight optimization, selective ensemble hours, model hyper-parameter optimization, and



(a) Complete electrical load plot of Czech data.



(b) Complete electrical load plot of Ausdata.

Figure 4. Complete electrical load plots of the two datasets.

feature selection. Results obtained by testing this fold are only utilized for development purposes, though it is used for training the models for other tests, just like other folds have participated in the training.

Analyses that have been performed for development purposes; namely hourly analysis (Section 3.4, used to decide the selective ensemble hours), feature selection analysis (Section 3.5, used to decide the best feature subset) are shown in the figures as the result of the first fold without cross-validation. Detailed results of selective ensemble and feature selection can be observed in tables under Section 3.6 as the average and standard deviation of the remaining four cross-validation tests. All the forthcoming error metrics consist of the average of all days, in the included test sets or validation set depending on the purpose.

3.3. Error metrics

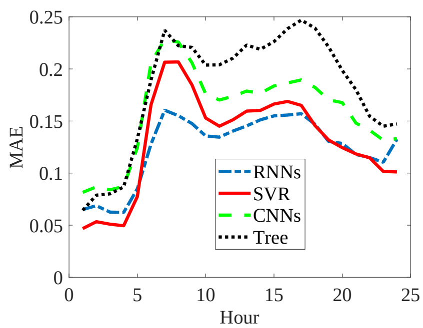
For the sake of training and testing the forecasters, the data is normalized for some observation i of some feature k, X_{ik} by:

$$X_{ik_{normalized}} = (X_{ik} - \mu(X_k))/\sigma(X_k), \tag{5}$$

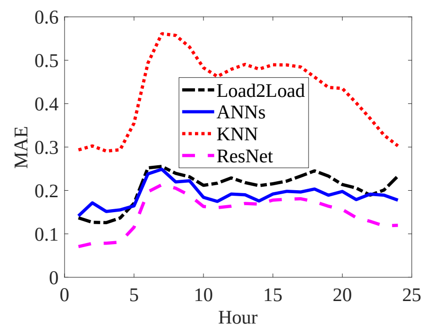
where $\mu(X_k)$ is the mean and $\sigma(X_k)$ is the standard deviation of feature k overall the data. The results with such normalized data are reported.

3.4. Hourly analysis

Hourly error analysis, the baseline for selective ensemble, for day-ahead forecasting of different forecasters is performed on the validation set using the full set of features as shown in Figures 5 and 6 for Czech data and Ausdata load forecasting; respectively. Results consist of the average of all day-ahead predictions on certain hours overall the validation set which covers a complete year.

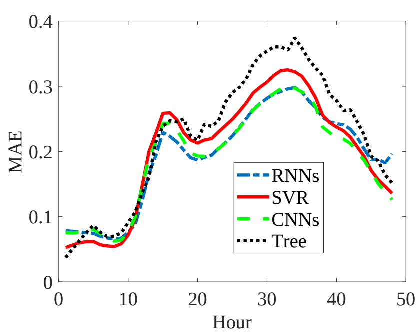


(a) Results for RNN's ensemble, SVR, CNN's ensemble and tree.

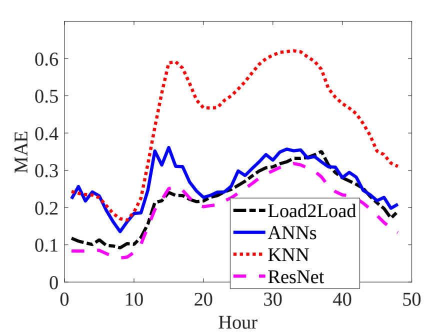


(b) Results for Load2Load, ANN's ensemble and ResNet.

Figure 5. Hourly plot of average day-ahead MAE metrics for load feature on Czech database validation set (1st test fold over 5 folds in total).



(a) Results for RNN's ensemble, SVR, CNN's ensemble and tree.



(b) Results for Load2Load, ANN's ensemble and ResNet.

Figure 6. Hourly plot of average day-ahead MAE metrics for load feature on Ausdata database validation set (1st test fold over 5 folds in total).

It can be observed that in both datasets, load forecasting errors have two peaks centered around 7:00 and 17:00 ($7 \times 2 = 14$ and $17 \times 2 = 34$ for 30-min-resolution Ausdata), starting and ending of work hours where load becomes unstable. Best performing forecasters are Load2Load, RNN, ResNet, CNN and SVR. RNN can perform the best for the most challenging hours although SVR can outperform RNN for more stable hours. Load2Load can perform well especially on Ausdata for unstable hours. According to the results, H^{sel}_{RNN} described in Section 2.3.2 for load feature on Czech database is determined to be $\{6, 7, ..., 21, 22\}$.

3.5. Feature selection analysis

Feature selection is experimented on the Czech dataset and Ausdata. A maximum of 8 features are added to the feature set using the proposed modified sequential forward feature selection algorithm.

Selected features for the Czech dataset with the order of selection are load, pumping load, hour, temperature, PVPP, and radiation diffuse horizontal. Month and day are also considered as the next best features with the order but this increased the errors, so not selected.

Selected features for Ausdata with the order of selection are load, wetbulb, and month. Hour, day, DryBulb, humidity, and DewPnt are also considered as the next best features with the order but increased the errors, so not selected.

3.6. Detailed results

Load forecasting results for the Czech dataset are provided in detail in Table 1 for day-ahead full set of features and selected features subset. Load forecasting results for Ausdata are provided in detail in Table 2 for day-ahead full set of features and selected features subset.

Table 1. Czech dataset load forecasting results.

Features:	All features		Selected features	
Model	MAE	MAPE	MAE	MAPE
$Mean$	$87.15 \pm 0.00\%$	$31.21 \pm 0.00\%$	$87.15 \pm 0.00\%$	$31.21 \pm 0.00\%$
$Random$	$117.85 \pm 1.17\%$	$42.57 \pm 0.35\%$	$116.76 \pm 0.66\%$	$42.13 \pm 0.30\%$
ANN_1 (1)	$20.15 \pm 1.24\%$	$9.28 \pm 1.72\%$	$18.95 \pm 1.52\%$	$8.61 \pm 1.73\%$
ANN_2 (2)	$24.06 \pm 1.90\%$	$10.87 \pm 2.12\%$	$19.58 \pm 1.70\%$	$9.01 \pm 1.74\%$
SVR (3)	$13.12 \pm 0.56\%$	$5.48 \pm 0.62\%$	$12.90 \pm 0.47\%$	$5.37 \pm 0.61\%$
kNN (4)	$42.78 \pm 2.19\%$	$19.03 \pm 2.23\%$	$27.23 \pm 1.92\%$	$11.99 \pm 1.12\%$
LRM (5)	$23.35 \pm 1.21\%$	$9.80 \pm 0.92\%$	$15.65 \pm 1.01\%$	$6.55 \pm 0.43\%$
$Tree$ (6)	$18.21 \pm 1.16\%$	$7.49 \pm 0.81\%$	$17.81 \pm 0.90\%$	$7.33 \pm 0.81\%$
RNN_1 (7)	$16.78 \pm 1.43\%$	$7.10 \pm 0.83\%$	$19.42 \pm 1.54\%$	$8.21 \pm 1.29\%$
RNN_2 (8)	$12.46 \pm 1.60\%$	$5.33 \pm 0.30\%$	$12.95 \pm 1.37\%$	$5.75 \pm 1.24\%$
CNN_1 (9)	$15.99 \pm 1.50\%$	$6.85 \pm 0.99\%$	$15.18 \pm 1.23\%$	$6.65 \pm 1.14\%$
CNN_2 (10)	$22.83 \pm 1.66\%$	$9.91 \pm 1.40\%$	$18.03 \pm 2.00\%$	$7.83 \pm 1.01\%$
$Load2Load$ (11)	$20.63 \pm 1.26\%$	$9.58 \pm 1.99\%$	$20.18 \pm 3.21\%$	$8.73 \pm 1.58\%$
$ResNet$ (12)	$14.96 \pm 1.98\%$	$6.25 \pm 0.70\%$	-	-
$1 \boxplus 2$ ($ANNs$) (13)	$18.93 \pm 1.10\%$	$8.74 \pm 1.60\%$	$17.00 \pm 1.24\%$	$7.79 \pm 1.53\%$
$7 \boxplus 8$ ($RNNs$) (14)	$12.46 \pm 1.60\%$	$5.33 \pm 0.30\%$	$12.95 \pm 1.37\%$	$5.75 \pm 1.24\%$
$9 \boxplus 10$ ($CNNs$) (15)	$16.00 \pm 1.49\%$	$6.85 \pm 0.99\%$	$15.26 \pm 1.27\%$	$6.67 \pm 1.12\%$
$3 \boxdot 14$ (16)	$12.02 \pm 1.39\%$	$5.05 \pm 0.30\%$	$12.40 \pm 1.18\%$	$5.33 \pm 1.03\%$
$16 \boxplus 13$	$11.95 \pm 1.14\%$	$5.13 \pm 0.32\%$	$12.04 \pm 0.73\%$	$5.22 \pm 0.86\%$
$16 \boxplus 15$	$11.68 \pm 1.10\%$	$4.95 \pm 0.39\%$	$12.00 \pm 1.03\%$	$5.15 \pm 0.92\%$
$16 \boxplus 11$	$11.96 \pm 1.30\%$	$5.09 \pm 0.37\%$	$12.12 \pm 1.34\%$	$5.21 \pm 0.85\%$
$16 \boxplus 12$	$11.67 \pm 1.37\%$	$4.90 \pm 0.36\%$	-	-
$16 \boxplus 13 \boxplus 15$	$11.61 \pm 1.06\%$	$4.95 \pm 0.38\%$	$11.88 \pm 0.72\%$	$5.14 \pm 0.83\%$
$16 \boxplus 13 \boxplus 15 \boxplus 6$	$11.49 \pm 1.07\%$	$4.87 \pm 0.37\%$	$11.72 \pm 0.70\%$	$5.04 \pm 0.79\%$
$16 \boxplus 13 \boxplus 15 \boxplus 6 \boxplus 11$	$11.46 \pm 1.08\%$	$4.86 \pm 0.36\%$	$11.77 \pm 1.00\%$	$5.07 \pm 0.75\%$
$16 \boxplus 13 \boxplus 15 \boxplus 6 \boxplus 12$	$11.42 \pm 1.22\%$	$4.80 \pm 0.36\%$	-	-

Table 2. Ausdata load forecasting results.

Features:	All features		Selected features	
Model	MAE	MAPE	MAE	MAPE
$Mean$	$79.24 \pm 0.00\%$	$46.21 \pm 0.00\%$	$79.24 \pm 0.00\%$	$46.21 \pm 0.00\%$
$Random$	$111.88 \pm 0.41\%$	$59.42 \pm 0.18\%$	$111.60 \pm 0.58\%$	$59.37 \pm 0.30\%$
ANN_1 (1)	$29.38 \pm 2.41\%$	$12.06 \pm 0.60\%$	$22.42 \pm 1.87\%$	$9.42 \pm 0.93\%$
ANN_2 (2)	$32.69 \pm 4.70\%$	$14.06 \pm 1.85\%$	$26.79 \pm 1.29\%$	$11.60 \pm 0.95\%$
SVR (3)	$20.12 \pm 1.85\%$	$7.83 \pm 0.65\%$	$18.05 \pm 1.34\%$	$7.02 \pm 0.63\%$
kNN (4)	$43.60 \pm 2.17\%$	$17.97 \pm 1.43\%$	$31.06 \pm 1.84\%$	$12.40 \pm 1.32\%$
LRM (5)	$418 \pm 161\%$	$155 \pm 57\%$	$22.03 \pm 1.90\%$	$8.29 \pm 0.77\%$
$Tree$ (6)	$22.34 \pm 1.59\%$	$8.54 \pm 0.84\%$	$21.67 \pm 1.53\%$	$8.25 \pm 0.83\%$
RNN_1 (7)	$20.71 \pm 1.42\%$	$8.18 \pm 0.77\%$	$24.24 \pm 0.97\%$	$9.65 \pm 0.66\%$
RNN_2 (8)	$19.37 \pm 1.10\%$	$7.69 \pm 0.77\%$	$20.12 \pm 0.90\%$	$7.90 \pm 0.81\%$
CNN_1 (9)	$19.10 \pm 2.40\%$	$7.58 \pm 1.01\%$	$18.76 \pm 1.95\%$	$7.44 \pm 0.92\%$
CNN_2 (10)	$22.11 \pm 1.39\%$	$8.97 \pm 0.55\%$	$20.63 \pm 1.18\%$	$8.25 \pm 0.62\%$
$Load2Load$ (11)	$22.48 \pm 2.27\%$	$8.85 \pm 0.91\%$	$21.02 \pm 0.74\%$	$8.19 \pm 0.03\%$
$ResNet$ (12)	$20.13 \pm 1.44\%$	$8.00 \pm 0.73\%$	-	-
$1 \boxplus 2$ (ANNs) (13)	$26.86 \pm 2.00\%$	$11.37 \pm 0.68\%$	$21.85 \pm 1.39\%$	$9.27 \pm 0.66\%$
$7 \boxplus 8$ (RNNs) (14)	$19.34 \pm 1.25\%$	$7.62 \pm 0.74\%$	$20.12 \pm 0.90\%$	$7.90 \pm 0.81\%$
$9 \boxplus 10$ (CNNs) (15)	$19.08 \pm 2.41\%$	$7.58 \pm 1.01\%$	$18.73 \pm 1.93\%$	$7.43 \pm 0.92\%$
$3 \boxdot 14$ (16)	$18.73 \pm 1.34\%$	$7.32 \pm 0.75\%$	$18.95 \pm 0.99\%$	$7.36 \pm 0.82\%$
$16 \boxplus 13$	$18.43 \pm 1.31\%$	$7.28 \pm 0.72\%$	$18.22 \pm 0.98\%$	$7.30 \pm 0.63\%$
$16 \boxplus 15$	$17.46 \pm 1.89\%$	$6.89 \pm 0.84\%$	$17.46 \pm 1.55\%$	$6.84 \pm 0.82\%$
$16 \boxplus 11$	$18.15 \pm 1.50\%$	$7.10 \pm 0.72\%$	$18.02 \pm 0.82\%$	$6.97 \pm 0.62\%$
$16 \boxplus 12$	$17.50 \pm 1.39\%$	$6.89 \pm 0.71\%$	-	-
$16 \boxplus 13 \boxplus 15$	$17.47 \pm 1.91\%$	$6.90 \pm 0.84\%$	$17.35 \pm 1.43\%$	$6.83 \pm 0.76\%$
$16 \boxplus 13 \boxplus 15 \boxplus 6$	$17.23 \pm 1.84\%$	$6.77 \pm 0.80\%$	$17.19 \pm 1.41\%$	$6.75 \pm 0.74\%$
$16 \boxplus 13 \boxplus 15 \boxplus 6 \boxplus 11$	$17.26 \pm 1.90\%$	$6.79 \pm 0.82\%$	$17.12 \pm 1.38\%$	$6.72 \pm 0.71\%$
$16 \boxplus 13 \boxplus 15 \boxplus 6 \boxplus 12$	$17.15 \pm 1.78\%$	$6.75 \pm 0.78\%$	-	-

Table 3. Total numbers of learnt parameters.

Features:	All	Selected	All	Selected
Forecaster	Czech data		Ausdata	
ANN1	19K	11K	32K	11K
ANN2	84K	48K	134K	48K
SVR	33,321K	16,642K	115,450K	32,388K
SVR (#SVs)	877	803	928	781
kNN	35K	35K	70K	70K
LRM	152K	83K	498K	166K
Tree	6K	8K	7K	9K
RNN1	10K	9K	10K	8K
RNN2	90K	86K	88K	83K
CNN1	26,196K	14,454K	49,752K	14,505K
CNN2	347K	181K	1343K	665K
Load2Load (gen)	61,426K	61,426K	61,426K	61,426K
Load2Load (total)	64,184K	64,184K	64,184K	64,184K
ResNet	27K	-	89K	-

3.7. Computational complexity

The total numbers of learnt parameters for each model in each case are shown in Table 3. Average numbers of support vectors (SVs) are provided as well. Forecaster networks having FLs before the output are affected by sequence length s, similarly, networks having SL or input FL are affected by the number of features f in terms of the number of parameters.

3.8. Discussions

It can be observed that among the forecasters for day-ahead forecasting; best single forecasters are generally SVR and RNN, followed by CNN and tree (Table 1). The selective ensemble is shown to further improve the forecasters that usually provide the best results (namely RNNs and SVR). Overall best results are obtained with the additive ensemble of many useful forecasters, e.g., $((F_{RNN_1} \boxplus F_{RNN_2}) \boxdot F_{SVR}) \boxplus F_{tree} \boxplus (F_{CNN_1} \boxplus F_{CNN_2}) \boxplus (F_{ANN_1} \boxplus F_{ANN_2})$. According to Table 2; the best performing forecasters are CNN (forecasters 9, 10, and 15), RNN (forecasters 7, 8, and 14) and SVR (forecaster 3) on Ausdata. RNN is performing slightly worse than CNN in this case. Overall, the proposed ensemble methods are shown to reduce errors. The same trend follows regardless of feature selection.

The proposed Load2Load model, although originally designed for image generation purposes, can give competing results when utilized for STLF. It can help improve the results when used together with the proposed ensemble techniques. Using the mean values or random values for forecasting results in around four to five times higher errors, showing the significance of the proposed forecasters.

According to the feature selection experiments over Czech data for load, individual forecaster errors can be decreased with a reduced (six) number of features. However, RNN has a slightly increased error which results in a slightly increased ensemble error overall. Most individual forecasters can still provide better results with the selected features. Feature selection is found most useful in Ausdata, resulting in an overall decrease in error rate only with three selected features. As can be seen from the tables; ResNet model can not be trained with much fewer features, using the exact same model.

Compared with the previous work [40], better results are obtained with a very similar experimental setup on the same test set using Czech data. The only difference is that in the proposed work the first test fold is assigned for development purposes and thus excluded from cross-validation test results. For simplicity, only normalized test data MAPE is considered in the proposed work column of the comparison table (Table 4). It should be noted that a direct comparison is impossible with previous studies other than [40] as the data, considered date interval, experimental protocol and even the reported error metrics can have significant variability. A detailed comparison with similar works can be found in [40].

Table 4. Day-ahead STLF MAPE results comparison.

Model	Dataset	Proposed work	Previous work
RNN	Czech	$\mathbf{5.33 \pm 0.30}$%	5.59 ± 0.62% [40]
ANN	Czech	8.74 ± 1.60%	$\mathbf{8.61 \pm 1.15}$% [40]
Best ensemble	Czech	$\mathbf{4.80 \pm 0.36}$%	5.77% (RMSE) [25]
Best ensemble	Australia	6.75 ± 0.78%	4.06% to 8.91%[26]

4. Conclusions and future work

In this work, the performance of different forecasting techniques for day-ahead load forecasting is evaluated and compared. Different deep architectures as well as conventional machine learning methods are investigated. The proposed Load2Load model, although originally designed for image generation purposes, can give competing results. Ensemble of the same kind of forecasters with different hyper-parameters, as well as different kinds of forecasters are examined. A selective ensemble technique is proposed, which selects the outputs of different forecasters in an hourly manner. Load2Load model is shown to be complementary when used within the ensembles. A modified sequential forward feature selection algorithm is proposed, resulting in better performance for some kinds of forecasters, with a much smaller number of features.

Performance is evaluated on two different and independent datasets with different features, one consisting of 24 measurements per day and the other consisting of 48 measurements per day; investigating the effect of resolution and sequence length. According to the results, the number of measurements and sequence length do not have a negative impact on the results. The significance of the proposed work has been shown such that the performance of the proposed architectures has been up to ten times better than random forecasting and mean forecasting.

More kinds of deep forecasters are planned to be utilized in the future to obtain further improved results. In addition to short-term forecasting, medium- and long-term forecasting will be investigated using other models of forecasters.

References

[1] Bhattacharyya S, Thanh L. Short-term electric load forecasting using an artificial neural network: Case of Northern Vietnam. International Journal of Energy Research 2004; 28 (4): 463 – 472.

[2] Santos P, Martins A, Pires AJ, Martins J, Vilela-Mendes R. Short-term load forecast using trend information and process reconstruction. International Journal of Energy Research 2006; 30 (8): 811 – 822.

[3] Sousa JC, Jorge HM, Neves LP. Short-term load forecasting based on support vector regression and load profiling. International Journal of Energy Research 2014; 38 (3): 350–362.

[4] Srivastava AK, Pandey AS, Singh D. Short-term load forecasting methods: A review. 2016 International Conference on Emerging Trends in Electrical Electronics Sustainable Energy Systems (ICETEESES), pp. 130–138.

[5] Baliyan A, Gaurav K, Mishra SK. A review of short term load forecasting using artificial neural network models. Procedia Computer Science 2015; 48: 121 – 125. International Conference on Computer, Communication and Convergence (ICCC 2015).

[6] Yildiz B, Bilbao J, Sproul A. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renewable and Sustainable Energy Reviews 2017; 73: 1104 – 1122.

[7] Liu X, Zhang Z, Song Z. A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: From classical data mining to deep learning. Renewable and Sustainable Energy Reviews 2020; 119: 109632.

[8] Hsu CC, Chen XT, Chen YS, Chang A. Short-term load forecasting by machine learning. 2020 International Symposium on Community-centric Systems (CcS), pp. 1–6.

[9] Lee YS, Tong LI. Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. Knowledge-Based Systems 2011; 24 (1): 66 – 72.

[10] Ahmed T, Vu D, Muttaqi K, Agalgaonkar A. Load forecasting under changing climatic conditions for the city of Sydney, Australia. Energy 2018; 142: 911 – 919.

[11] Amjady N. Short-term hourly load forecasting using time-series modeling with peak load estimation capability. IEEE Transactions on Power Systems 2001; 16 (4): 798–805.

[12] Harun MHH, Othman MM, Musirin I. Short term load forecasting (stlf) using artificial neural network based multiple lags and stationary time series. 2010 4th International Power Engineering and Optimization Conference (PEOCO), pp. 363–370.

[13] Kumar Srivastava A, Singh D, Pandey AS. Short term load forecasting using regression trees: Random forest, bagging and m5p. International Journal of Advanced Trends in Computer Science and Engineering 2020; 9 (2): 1898 – 1902.

[14] Li Y, Che J, Yang Y. Subsampled support vector regression ensemble for short term electric load forecasting. Energy 2018; 164: 160 – 170.

[15] Fan GF, Guo YH, Zheng JM, Hong WC. Application of the weighted k-nearest neighbor algorithm for short-term load forecasting. Energies 2019; 12 (5): 916.

[16] Jian Z, Cencen X, Ziang Z, Xiaohua L. Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. 2017 51st Annual Conference on Information Sciences and Systems (CISS), pp. 1–6.

[17] Li L, Ota K, Dong M. Everything is image: CNN-based short-term electrical load forecasting for smart grid. 2017 14th International Symposium on Pervasive Systems, Algorithms and Networks 2017 11th International Conference on Frontier of Computer Science and Technology 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC), pp. 344–351.

[18] Zhang N, Li Z, Zou X, Quiring SM. Comparison of three short-term load forecast models in southern California. Energy 2019; 189: 116358.

[19] Chen H, Wang S, Wang S, Li Y. Day-ahead aggregated load forecasting based on two-terminal sparse coding and deep neural network fusion. Electric Power Systems Research 2019; 177: 105987.

[20] Guo Z, Zhou K, Zhang X, Yang S. A deep learning model for short-term power load and probability density forecasting. Energy 2018; 160: 1186 – 1200.

[21] Mele E. A review of machine learning algorithms used for load forecasting at microgrid level. Sinteza 2019 - International Scientific Conference on Information Technology and Data Related Research, pp. 452–458.

[22] Solyali D. A comparative analysis of machine learning approaches for short-/long-term electricity load forecasting in cyprus. Sustainability 2020; 12 (9): 3612.

[23] Bouktif S, Fiaz A, Ouni A, Serhani M. Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches †. Energies 2018; 11 (7): 1636.

[24] Choi H, Ryu S, Kim H. Short-term load forecasting based on Resnet and LSTM. In 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) 2018: 1–6.

[25] Uher V, Burget R, Dutta MK, Mlynek P. Forecasting electricity consumption in Czech Republic. In 2015 38th International Conference on Telecommunications and Signal Processing (TSP) 2015: 262–265.

[26] Mahdavi N. Probabilistic forecasting of operational demand in Australia. In 2020 International Conference on Smart Grids and Energy Systems (SGES) (2020), pp. 139–144.

[27] Laouafi A, Laouafi F, Boukelia TE. An adaptive hybrid ensemble with pattern similarity analysis and error correction for short-term load forecasting. Applied Energy 2022; 322: 119525.

[28] Yang Y, Wang Z, Gao Y, Wu J, Zhao S et al. An effective dimensionality reduction approach for short-term load forecasting. Electric Power Systems Research 2022; 210: 108150.

[29] Esener II, Yüksel T, Kurban M. Short-term load forecasting without meteorological data using AI-based structures. Turkish Journal of Electrical Engineering and Computer Sciences 2015; 23 (2): 370 - 380.

[30] Akinola SA, Thakur P, Sharma MS, Kumar K, Singh G. Deep learning approach to load forecasting: A survey. In Communication, Networks and Computing (Singapore, 2021), Springer Singapore, pp. 250–262.

[31] Chelabi H, Khadir MT, Chikhaoui B, Telmoudi AJ. Comparison of deep learning architectures for short-term electrical load forecasting based on multi-modal data. Cybernetics and Systems 2022; 53 (1): 186–207.

[32] Gasparin A, Lukovic S, Alippi C. Deep learning for time series forecasting: The electric load case. CAAI Transactions on Intelligence Technology 2021; 7 (1): 1–25.

[33] Liu Y, Dutta S, Kong AWK, Yeo CK. An image inpainting approach to short-term load forecasting. IEEE Transactions on Power Systems 2022: 1–1.

[34] Bendaoud NMM, Farah N, Ahmed SB. Comparing generative adversarial networks architectures for electricity demand forecasting. Energy and Buildings 2021; 247: 111152.

[35] Dumas J, Wehenkel A, Lanaspeze D, Cornélusse B, Sutera A. A deep generative model for probabilistic energy forecasting in power systems: normalizing flows. Applied Energy 2022; 305: 117871.

[36] Tian C, Ye Y, Lou Y, Zuo W, Zhang G et al. Daily power demand prediction for buildings at a large scale using a hybrid of physics-based model and generative adversarial network. Building Simulation 2022; 15: 1685–1701.

[37] Zhu L, Huang Y. Research on deep generative model application for shortterm load forecasting of enterprise electricity. IOP Conference Series: Earth and Environmental Science 2021; 687 (1): 012113.

[38] Huang N, He Q, Qi J, Hu Q, Wang R et al. Multinodes interval electric vehicle day-ahead charging load forecasting based on joint adversarial generation. International Journal of Electrical Power & Energy Systems 2022; 143: 108404.

[39] Zhou Y, Ding Z, Wen Q, Wang Y. Robust load forecasting towards adversarial attacks via Bayesian learning. IEEE Transactions on Power Systems 2022: 1–1.

[40] Yaprakdal F, Yılmaz MB, Baysal M, Anvari-Moghaddam A. A deep neural network-assisted approach to enhance short-term optimal operational scheduling of a microgrid. Sustainability 2020; 12 (4): 1653.

[41] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D et al. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (Cambridge, MA, USA, 2014), NIPS'14, MIT Press, p. 2672–2680.

[42] Chang CC, Lin CJ. Training v-support vector regression: Theory and algorithms. Neural Computation 2002; 14 (8): 1959–1977.

[43] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016: 770–778.