Information Sciences 544 (2021) 183–196

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A hybrid deep meta-ensemble networks with application in electric utility industry load forecasting

Shaohui Ma

School of Business, Nanjing Audit University, Nanjing 211815, China

ARTICLE INFO

Article history:
Received 30 September 2019
Received in revised form 13 July 2020
Accepted 20 July 2020
Available online 01 August 2020

Keywords:
Load forecasting
Time series forecasting
Big data
Meta-learning
Deep learning

ABSTRACT

Many industrial applications concern the forecasting of large quantities of related time series. This paper presents a novel hybrid modeling framework, named as Hybrid Deep Meta-Ensemble Networks (HDME-Nets), which combines local and global forecasts using meta-ensemble technology. The proposed framework can be used to generate multiple steps ahead of point and interval forecasts for a large number of time series concurrently. Our proposed framework is composed of four modules: a set of local forecasters that model each time series individually, a global forecaster that captures cross-sectional patterns with data pooling, a feature learner that extracts features from each time series in a supervised fashion, and a meta-combiner that combines the local and global forecasts according to the extracted features. The local forecasters are fitted firstly, and the other three modules are integrated seamlessly into one neural network, which is then jointly trained with a common objective measured with a custom loss function. Through testing two public available electric utility load datasets, we find that the proposed method can achieve improved forecasting performance compared against some state-of-the-art methods.

1. Introduction

As more and more fields involve Big Data problems, ranging from global economy to society administration, and from scientific researches to industry operations, we have entered the era of Big Data [29]. Many big data applications concern the problem of forecasting large sets of related time series. Consider for example in transportation, where real-time (15–40 min) forecasting of traffic flow gives travelers the ability to choose better routes and administrations the ability to manage the transportation system. In the retail industry, retailers need to forecast the daily demand of thousands or even millions of products across thousands of locations [21,23].

In the utility industry where this research is focused, forecasting systems are required to provide load forecasts at hourly or sub-hourly intervals for the subsequent days, up to two weeks. The forecasts are used by all sectors of the utility industry, from generation and transmission to distribution and retail [12]. A conservative estimate by Ref. [10] showed that a decrease of the load forecasting error in terms of Mean Absolute Percentage Error (MAPE) by 1% lowers the variable production cost between 0.6 and 1.6 million USD annually for a 10,000 MW utility with MAPE around 4%.

In this research, we contribute to Industrial Artificial Intelligence (IAI) by proposing a novel meta-ensemble modeling framework based on Deep Learning Neural Networks. The framework is named Hybrid Deep Meta-Ensemble Networks (HDME-Nets), which aims at generating multiple steps ahead of point and interval forecasts for a Large Number of Related

E-mail address: shaohui.ma@nau.edu.cn

https://doi.org/10.1016/j.ins.2020.07.054

Time Series (LNR-TS) concurrently. The HDME-Nets first learn from knowledge of the forecasting performance of a given combination of global and local forecasters, as this relates to the characteristics of the data used to fit these forecasters, and then uses that knowledge to generate an optimal ensemble of forecasts according to specific data history. The HDME-Nets consist of four modules: a set of local forecasters that model each time series individually, a global forecaster that captures time series' cross-sectional patterns using a data pooling strategy, a feature learner that extracts features from each time series in a supervised fashion, and a meta-combiner that combines the local and global forecasts according to the extracted features optimally. The HDME-Nets is extensible and flexible, all the modules in the framework could be further designed according to the data characteristics of the specific forecasting applications. We applied the HDME-Nets into the electric utility industry to forecast hourly utility loads. Through experiments on two real utility load datasets, we found that the proposed method can generate superior multiple steps ahead of point and interval forecasts over state-of-the-art methods.

The outline of the paper is as follows. In Section 2, we review related studies and introduce our innovations. In Section 3 we discuss associated methodological issues in our proposed solution. In Section 4, we describe the data, introduce the experiments design and forecasting accuracy measures, and then present the empirical results. In the last section, we discuss the findings offering conclusions as to forecasting practice and further academic research.

2. Related research

2.1. Forecasting methods for LNR-TS problem

The modeling strategies for LNR-TS forecasting problem can be classified into two extremes: modeling each time series individually or modeling the whole set of time series in a pooled fashion [22]. Januschowski et al. distinguished the two strategies as local and global forecasting methods [16].

Local forecasting methods treat each time series separately and forecast them in isolation, which can fully consider each time series' data characteristics, such as seasonality and temporal trend. Most traditional statistical forecasting methods have been developed mainly in this setting. Simple moving averages, the exponential smoothing family or Autoregressive Integrated Moving Average (ARIMA), are among of most popular local forecasting methods. For LNR-TS forecasting problems, though many related series are available, local methods cannot capture data features common among different time series, such as common seasonal patterns, nonlinear trends, and cross-sectional temporal and spatial interactions, this is especially true for short time series with high noise [6].

The global forecasting method usually trains a complex homogenous model to forecast a pool of time series, this enhances the data availability and has the potential to capture cross time series common patterns; the free parameters are trained jointly thereby improving the robustness of the forecasting [4,39]. For many LNR-TS forecasting applications, using complex machine learning methods to build global models across time series have achieved promising results. Modern gradient boosting trees and deep learning neural networks in recent years have seen their successful application to many LNR-TS forecasting problems in areas, such as retail, transportation, environment, and electricity prices. One potential drawback of the global method is that the heterogeneous characteristics in various time series are easily neglected.

There are also hybrids between local and global forecasting methods, where a part of the parameters are estimated globally and a part locally [16]. For example, by applying a global model to the residuals of a number of local models [31], or by training certain parameters of local models across different time series [24], or by reconciling local models globally via hierarchical forecasting techniques [15]. Especially, Smyl [31] recently proposed a local-global "hybrid" method which was the winning solution of the 100,000 time series forecasting competition named M4. In their method, statistical modeling (Exponential Smooth models) is combined concurrently with Machine Learning algorithms (recurrent neural networks), both global and local parameters are utilized in order to enable cross-learning while also emphasizing the particularities of the time series being extrapolated.

2.2. Utility industry load forecasting

In the utility industry, short term load forecasting is usually required to provide load forecasts over a large number of meters in different locations, which is typically a LNR-TS forecasting problem. Most of the existing load forecasting methods fall either in local or global forecasting categories. Many statistical and artificial intelligence techniques have been applied to load forecasting over the past three decades, including regressions [12,38], stochastic time series analyses [27,33], artificial neural networks (ANNs) [3,5,26], wavelet analyses [19], and Gaussian process (GP) models [20]. The overviews on the recent development of load forecasting have been presented in a number of surveys [9,13,30].

In this research, we proposed a novel hybrid forecasting methodological framework for the LNR-TS forecasting problem. It utilizes a meta-ensemble technique to combine the forecasts from local and global forecasters according to the characteristics of the historical data fitting those forecasters, so as to improve the forecasting accuracy by taking advantage of the strength of the two types of forecasters. The proposed method showed significantly better forecasting performance than the state-of-the-art methods on two public available electric utility load datasets. The innovations of the proposed method can be summarized in four respects as follows:

1) Different to existing hybrid forecasting methods, as it utilizes the meta-ensemble technique based on deep learning neural networks to combine the forecasts from both local and global forecasting methods, to improve the forecasting accuracy.
2) It extracts features from each time series being forecasted automatically and in a supervised fashion while existing meta-learning methods depend totally on judgmentally selected features.
3) It is the first meta-forecasting method that can generate both point and interval forecasts.
4) It is extensible and flexible; all the modules in the framework could be designed according to the characteristics of the specific forecasting applications.

3. Methodology

3.1. Overview of the hybrid deep Meta-Ensemble networks

We here adopt the definition of the LNR-TS forecasting problem proposed by Ref. [36]. We denote$y_{:t,i}$ as the observed data up to time t of the ith time series, $\mathbf{x}^{(v)}_{:t,i}$ the temporal covariates available in history, $\mathbf{x}^{(f)}_{i,:t}$ the knowledge about the future, and $\mathbf{x}^{(s)}_{i}$ the static time-invariant features, the goal being to predict

$$p\left(y_{t+1,i},\cdots,y_{t+k,i}|y_{:t,i},\mathbf{x}^{(v)}_{:t,i},\mathbf{x}^{(f)}_{i,:t},\mathbf{x}^{(s)}_{i}\right),\ i\in I$$

where I is the set of time series to be forecasted. The HDME-Nets is proposed to solve the LNR-TS forecasting problem using a novel meta-ensemble technology based on newly developed deep learning neural networks. Fig. 1 presents an overview of the proposed HDME-Nets architecture. The HDME-Nets consists of four modules: (i) a set of local forecasters, (ii) a global forecaster, (iii) a feature extractor, and (iv) a meta- combiner.

Local forecasters are used to fit each time series individually so as to consider their respective features when generating forecasts. Local forecasters should be chosen according to the characteristics of the specific forecasting problem. They could be a set of classical univariate time series models, a set of linear or nonlinear regression models if exogenous explanatory variables are available, or a mixture of the two types of methods. The output of the module is a matrix of the multiple steps ahead of forecasts on all the time series, which is named as local forecasts in the HDME-Nets.

The global forecaster is a neural network module which is used to capture cross series patterns by training the networks with the whole set of the time series. It generates global forecasts which will be optimally combined with the local forecasts by the meta-combiner. Any neural network which is capable of scalable time series forecasting, e.g. Long Short-Term Memory (LSTM) [11], Sequence to Sequence Neural Networks (seq2seq) [32], or WaveNet [28], etc., could be employed here as the global forecaster.

The feature extractor is also a neural network module and is used to extract a feature representation from each time series. The extracted features are the inputs to the meta-combiner to generate optimal weights to combine the global and local forecasts. The feature extractor could be any neural networks that can extract longitudinal features from time series. Usually, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), or any of their variants and combinations, are capable of accomplishing such task.

The meta-combiner is the last module. For each time series, it transforms the extracted features into a set of weights. The weights are used to combine the local and global forecasts using a weighted combination. The principle of the meta-combiner is it first learn the knowledge during training stage on the forecasting performance of a given combination of global and local forecasts as this relates to the characteristics of the data used to fit these forecasters, and then using the knowledge to generate optimal weights for each time series during the forecasting stage. Therefore, meta-combiner allows different ensemble models to be used to forecast different time series.

The training process of the HDME-Nets includes two phases. During the first phase, for each time series to be forecasted, it is transformed into pairs of <input, output> patches using the moving window strategy. Local forecasters are fitted separately with the 'input' patches and then the fitted models are used to generate the local forecasts for each patch on the following 'output' window. In the second phase, all the local forecasts, together with all the pairs of patches, are then inputted into the other three modules, i.e., global forecaster, feature extractor, and the meta-combiner. The three modules are conjointly trained with a common objective which is evaluated by a custom loss function with the true values in the 'output' patches, and then the parameters of the networks are updated through gradient descent backpropagation optimization algorithm.

3.2. A concrete realization of HDME-Nets for forecasting electric utility loads

The utility load time series usually contains a trend, multiple seasonal variations, and stochastic irregular components. It often can be observed annual, weekly, and daily cycles. The noise level in a load time series depends on the system size and customer structure. The trend, amplitude of the annual cycle, weekly and daily profiles, and noise intensity may change considerably from zone to zone. We designed two network structures under HDME-Nets framework for point and interval load forecasting tasks respectively, which are illustrated in Figs. 2 and 3 respectively.

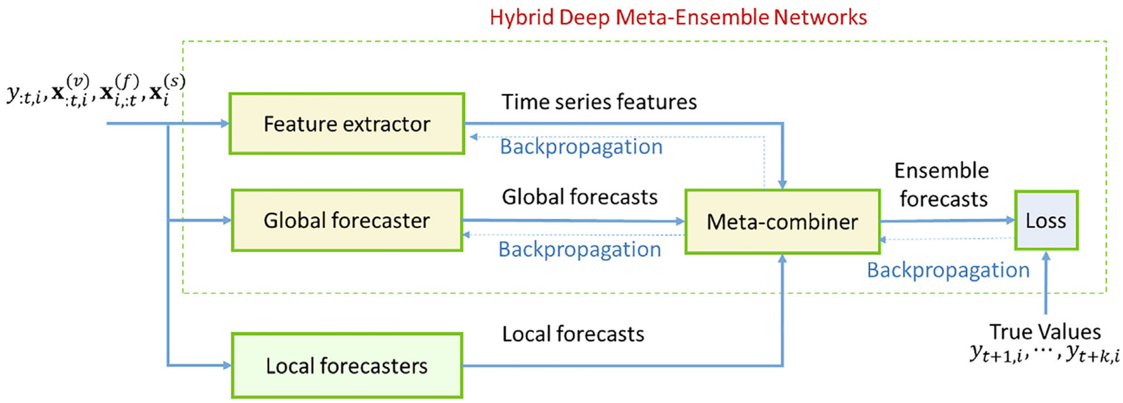

Fig. 1. Framework of the hybrid deep meta-ensemble networks.

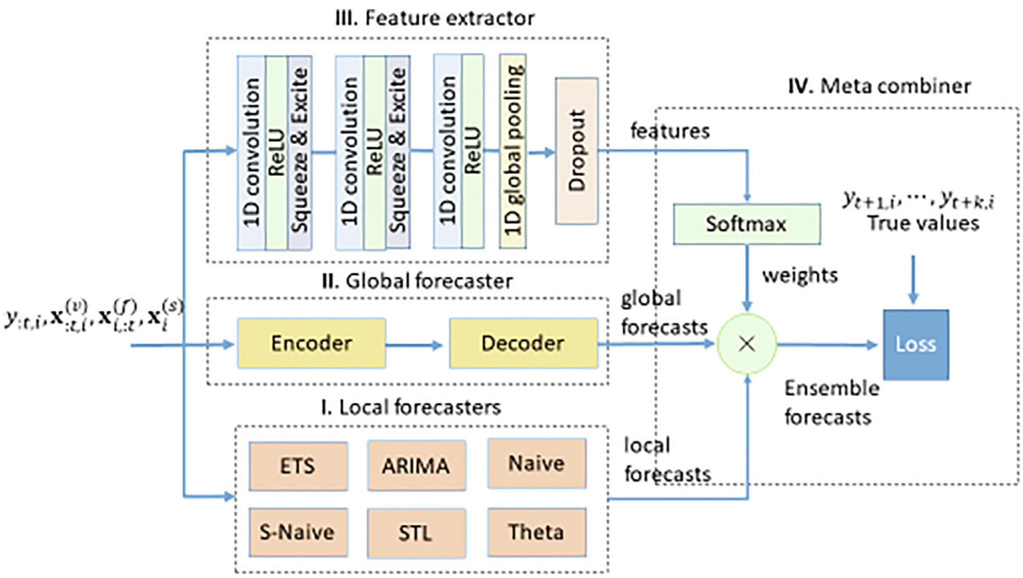

Fig. 2. The HDME-Nets designed for load point forecasting.

3.2.1. Local forecasters

We selected seven classical univariate time series forecasting methods as the local forecasters, including Naïve, Seasonal Naïve (S-Naive), Exponential Smoothing state space model with seasonality and non-damped trend (ETS), Theta, ARIMA, Random Walk (RW), and Seasonal and Trend decomposition using Loess (STL). They have been chosen from those considered in the M3 competition in which a large number of series were analyzed and a large number of extrapolation methods have been compared. All are practical alternatives in commercial applications and have demonstrated significant performance in past forecasting exercises. For point forecasting tasks, the mean forecasts from each local forecaster are used as local point forecasts. For interval forecasting tasks, if a is the expected significance level of the interval, then the $a/2$ and 1-$a/2$ quantiles of forecasts generated by the fitted local forecasters are used as the local lower and upper prediction limits respectively.

3.2.2. Global forecasters

Different to local forecasters, the global forecaster is designed to capture the cross-sectional patterns across time series and to generate forecasts complementary to the local forecasts.

To investigate the impact of the selection of global forecasters on the forecasting performance of the HDME-Nets, we designed and tested three HDME-Nets each using a different global forecaster. The first is named as HDME-Nets-MLP which uses a Multi-Layer Perceptron (MLP) networks as the global forecaster (Fig. 4a). Specifically, it consists of three layers: input, hidden, and output layers. For the output layer, ReLU ($ReLU(z) = \max(0, z)$) activation is used to keep the load forecasts as positive values. A dropout layer is inserted after the output layer to avoid overfitting. For interval forecasting, we used

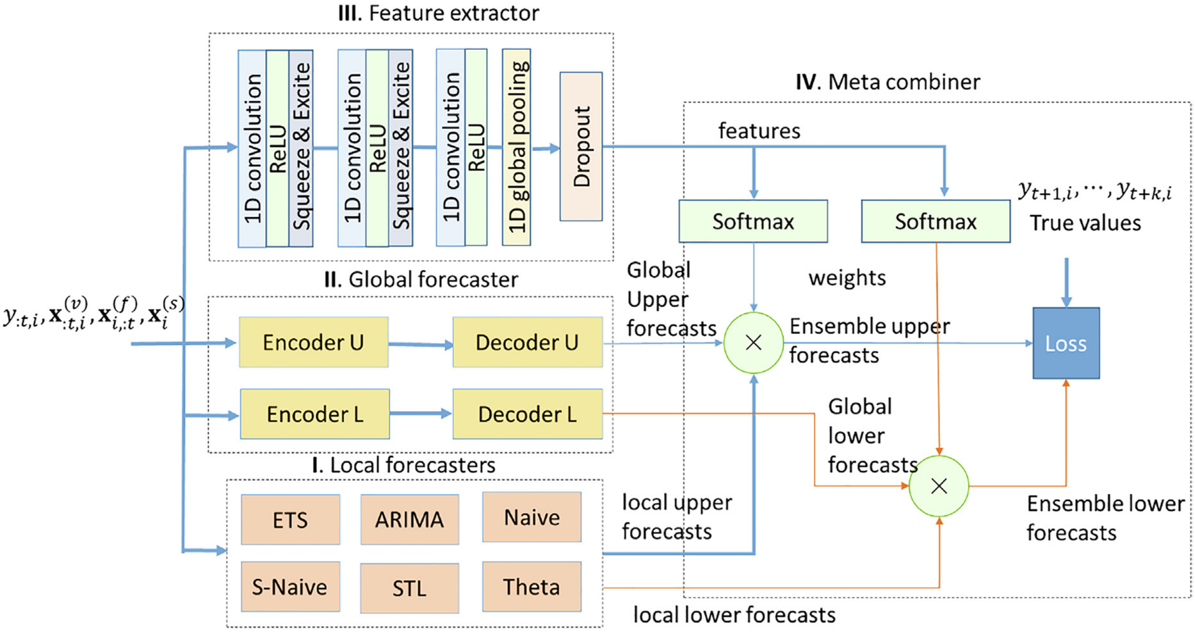

Fig. 3. The HDME-Nets designed for utility load interval forecasting.

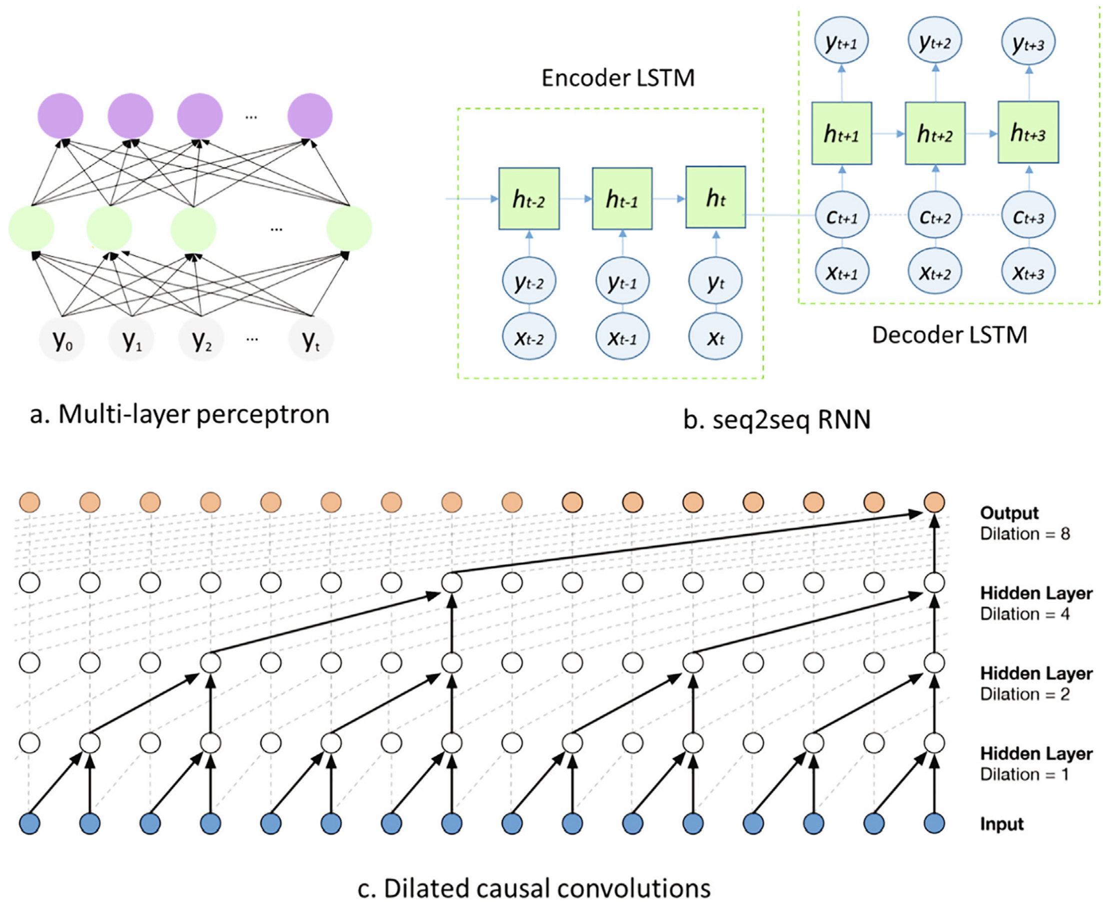

Fig. 4. Structures of the global forecasters.

two MLPs to generate the lower and upper forecasts simultaneously. MLP is a well-known neural network for its absolute generalization ability, and has been widely adopted in utility loads forecasting practice [30].

The second HDME-Nets is named as HDME-Nets-seq2seq. Its global forecaster is selected to be a Sequence-to-Sequence Recurrent Neural Network (Seq2Seq RNN) (Fig. 4b). Seq2Seq RNN is chosen as it has recently demonstrated state-of-the-art performance in various time series forecasting applications [7,37]. A vanilla LSTM (Long Short-Term Memory) network is used to encode all history information into hidden states h_t, and then hidden states are used as inputs for a decoder LSTM. LSTM were designed to cope with the vanishing gradient problem, which is essential to capturing long-term dependency. For interval forecasting tasks, we used one encoder LSTM and two decoder LSTMs to generate lower and upper bound forecasts.

The third global forecaster we tested is WaveNet, which consists of four layers of dilated CNN with an equal number of filters. WaveNet is one of the recent achievements of deep learning techniques in time series forecasting areas [1,2]. WaveNet can generate the sequences of real-valued data to some conditional inputs. The main idea behind the architecture is dilated causal convolutions (Fig. 4c), in which the filter is applied by skipping certain elements in the input, allow for the receptive field of the network to grow exponentially, thereby allowing the network to access a broad range. Due to the convolutional structure of the network, the number of trainable weights is usually smaller than recurrent-type networks, resulting in a much more efficient training and predicting.

3.2.3. Feature extractor

The feature extractor is designed to extract features from the load time series in an automatic and supervised manner. The learnt features will be used to generate optimal weights to combine the global and local forecasters. Convolutional Neural Networks (CNN) can mine and generate deep features from inputted time series automatically, and have shown a strong robustness protesting against data translation, scaling, and rotation [18]. Earlier researches have highlighted the potential of CNNs on learning features from sequential data showing better performance than traditional algorithms, and these findings have motivated us to investigate the feasibility of using CNN as the feature extractor in the proposed HDME-Nets.

The feature extractor we developed for load forecasting contains three stacked temporal convolutional blocks. Each convolutional block contains a convolutional layer and a ReLU activation. In addition, the first two convolutional blocks conclude with a squeeze and excite block that can improve the quality of representations produced by the convolutional layer by explicitly modelling the interdependencies between the channels of its convolutional feature [14]. The last temporal convolutional block is followed by a global average pooling layer, which is used to reduce the number of parameters in the network. The outputs of the global pooling layer are then fed into a dropout layer to mitigate overfitting.

3.2.4. Meta-combiner

In the meta-combiner, we first design a dense layer with softmax activation to transform the learnt features from the feature extractor into a set of weights. The weights are then used to combine the local and global forecasts with a weighted linear combination layer (Fig. 2). For interval forecasting (Fig. 3), the meta-combiner uses two separate dense layers to transform the learnt features into two sets of weights, which are used to combine lower and upper forecasts respectively. The combined forecasts are finally evaluated with a loss function.

As different load time series from different zones have different scales, for a proper summary of the forecasting performance on all the load time series to be forecasted, we use the symmetric Mean Absolute Percentage Error (sMAPE) as the loss function, which is defined as

$$L(\theta) = \frac{200}{NH} \sum_{i=1}^{N} \sum_{h=1}^{H} \left[\left| \hat{y}_{i,T+h}(\theta) - y_{i,T+h} \right| / \left(\hat{y}_{i,T+h}(\theta) + y_{i,T+h} \right) \right] \tag{1}$$

where N is the number of load time series, H the forecasting horizon, and θ the learnable parameters in the network.

For interval forecasting (Fig. 3), we use the Mean Scaled Interval Score (MSIS) of Ref. [25] to measure the forecasting accuracy of the generated lower and upper bounds with the true values in the output patches, which is described as follows:

$$\mathrm{MSIS} = \frac{100}{N} \sum_{i} \frac{\frac{1}{H}\sum_{t=T+1}^{T+H} \left\{ \left| U_{i,t} - L_{i,t} \right| + \frac{2}{a}\left(L_{i,t} - y_{i,t}\right)\mathbf{1}\left\{y_{i,t} < L_{i,t}\right\} + \frac{2}{a}\left(y_{i,t} - U_{i,t}\right)\mathbf{1}\left\{y_{i,t} > U_{i,t}\right\} \right\}}{\frac{1}{T-1}\sum_{t=2}^{T} \left| y_{i,t} - y_{i,t-1} \right|} \tag{2}$$

where L and U are the Lower and Upper bounds of the prediction intervals, a is the expected significance level of the interval, and $\mathbf{1}$ is the indicator function (being 1 if y is within the postulated interval and 0 otherwise). MSIS is scaled by dividing its value with the mean absolute difference of the series, in order for the measure to become scale independent.

4. Empirical studies

4.1. Data

To investigate the forecasting performance of the proposed HDME-Nets on electric utility load data, we first conducted forecasting experiments on an open obtainable utility load dataset from the Kaggle electric load forecasting competition GEFCom2012.[1] It contains hourly electric load data of 20 zones of a US utility from the 1st hour of 2004/1/1 to the 6th hour of 2008/6/30. An example of such load series is given in Fig. 5.

4.2. Training and test set definition

We used the moving window strategy to transform a load time series into pairs of <input, output> patches. Given a time series $Y_i = \{y_{i1}, \ldots, y_{it}\} \in \mathbb{R}^t$ of length t, step length of s, the moving window strategy converts the Y_i into $[(t - K - H + 1)/s]$ patches, where each patch has a window size of $(K + H)$. Here, K and H represent the sizes of the input window and output window respectively. We set the size of the output window identical to the intended forecasting horizon (H). This enables our model to directly predict all future values up to the intended forecasting horizon. We chose the size of the training input window K as 120 h and H as 48 h. We also set the step length s identical to the maximum forecasting horizon, i.e., 48 h (Fig. 6).

For a robust forecasting performance evaluation, we separated the data into training, validation, and test sets (illustrated in Fig. 6). The training set spans 600 days (14,400 h), from October 4, 2005 to May 27, 2007. The other two data sets each consists of 200 days (4800 h) of load data. The validation set spans from May 27, 2007 to December 13, 2007, and the test set from December 21, 2007, to July 8, 2008. When using the test set for evaluation, the training and validation set are combined to form an updated training set. This is designed to mimic a short term load forecasting job, where the forecaster first builds a model using historical data, then develops the forecasts for the next few days.

4.3. Forecasting evaluation metrics

For point forecasting, we used four error measures to compare the forecasting performance of the models. In addition to sMAPE, Mean Absolute Scaled Error (MASE) which was used as the second error measure. MASE can be considered as a "weighted" arithmetic mean of the MAE based on the variations of the load data in the training period. It is here defined as

$$MASE = \frac{1}{NH} \sum_{i=1}^{N} \sum_{h=1}^{H} \left[\left| \hat{y}_{i,T+h}(\theta) - y_{i,T+h} \right| / \tfrac{1}{T-1} \textstyle\sum_{t=2}^{T} \left| y_{i,t} - y_{i,t-1} \right| \right] \tag{3}$$

MASE is clearly independent of the scale of the load data and very suitable for comparing the forecasts across multiple load time series. The other two criteria were Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) which are traditional and popular scale-dependent error measures. They are easy to calculate, easy to understand, and widely applied in load forecasting literature.

For interval forecasting, since MSIS defined in (2) uses a complex formulae to evaluate concurrently both the coverage rate and the width of the predicted intervals, the Absolute Coverage Difference (ACD) was also used as a supplementary measure of the interval prediction precision. ACD is simply the absolute difference between the average coverage of the method and the target set (here 0.95). Thus, if the future values are on average across the testing load time series outside the bounds specified by a method at the 2% of the times (coverage of 98%), the ACD will be |0.98–0.95| = 0.03.

4.4. Benchmark models

To compare the forecasting accuracy of our proposed methods, thirteen forecasting methods were used as benchmarks. We included all seven local forecasters and three global forecasters as the benchmarks. In addition, one sub-model of the HDME-Nets and two state-of-art time series forecasting models were also used as our benchmarks which are explained as the following.

1) Meta Ensemble Networks (ME-Nets). This model is a meta-ensemble of the forecasts from only local forecasters. So it has only three modules of HDME-Nets: Local forecasters, Feature extractor, and Meta Combiner.
2) Random Forest (RF). Random Forest is based on decision trees and combined with aggregation and bootstrap ideas. In this research, we adopted a fast implementation of random forest, the 'Ranger' package v.0.11.2 in R, which is particularly suited for high dimensional data.

[1] The whole dataset can be downloaded from https://doi.org/10.1016/j.ijforecast.2013.07.001.

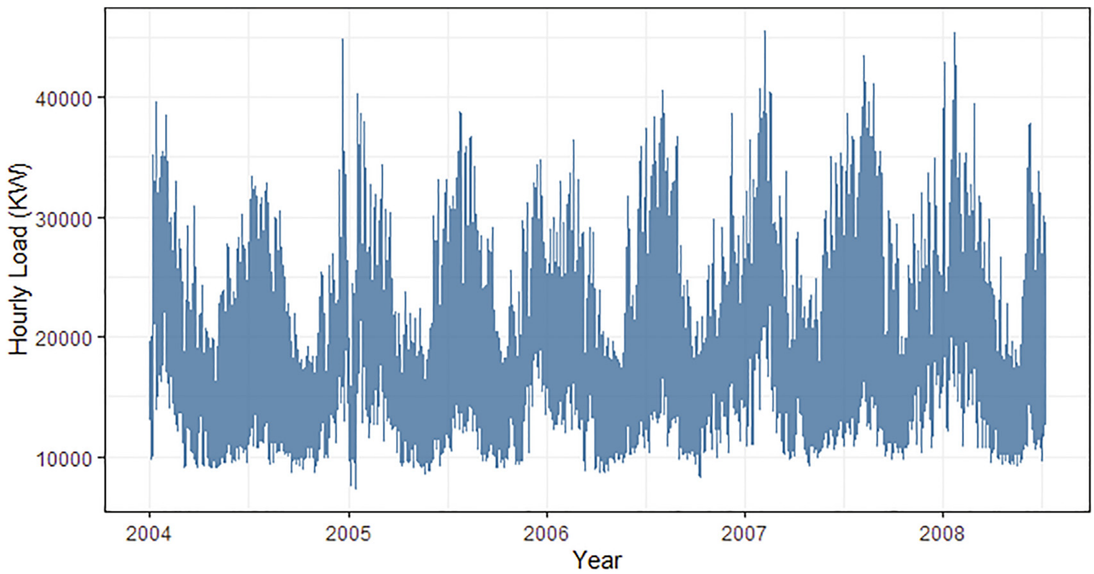

Fig. 5. An example series of the hourly electric load in GEFCom2012.

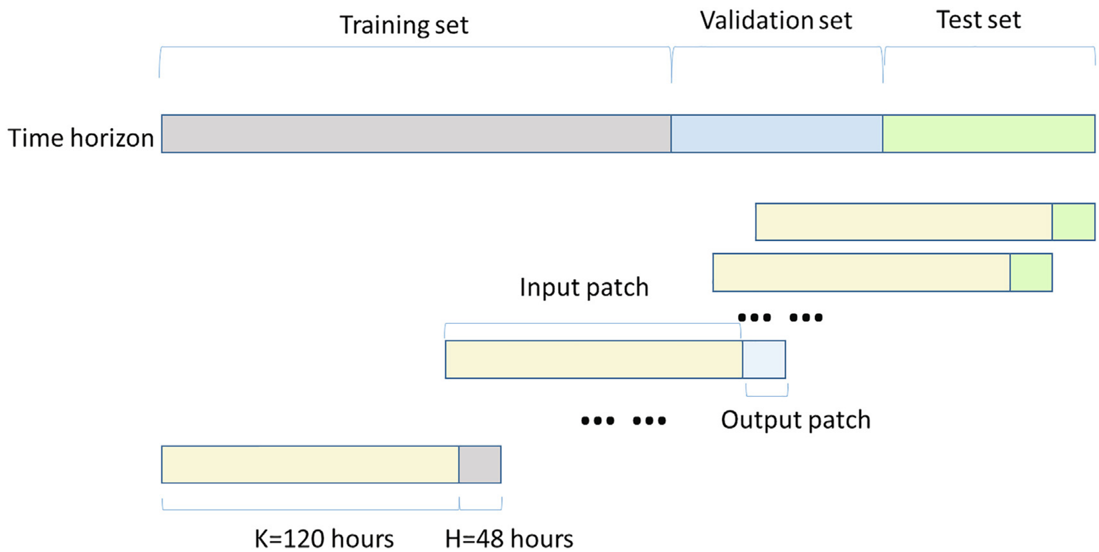

Fig. 6. An illustration of the moving window strategy in data preprocessing where a time series is transformed into many pairs of patches.

3) Gradient Boosting Regression Trees (GBRT). GBRT is one of most popular gradient boosting algorithms, which uses a regression tree as the base weak learner [8]. GBRT has empirically proven itself to be highly effective for a vast array of classification, ranking, and regression problems. It is one of the most preferred choices in data analytics competitions such as Kaggle and KDD Cup. In this research, we implemented the GBRT with the 'Xgboost' package v. 0.82.1 in R.

4.5. Training local forecasters

For each pair of <input, output> time series patches, we first in turn trained seven local forecasters using the input patches, and then generate 1 to 48 h ahead of forecasts with each trained model. All the methods are implemented in the 'forecast' package in R. In Table 1, we summarize the details for the training process, including the software tools and the settings for hyper-parameters.

We used the mean forecasts from each local forecaster as the inputs of the HDME-Nets for point forecasting task, and used 95% lower and upper prediction limits as the inputs of the HDME-Nets for interval forecasting task. If any function returned an error when fitting the series (e.g. a series is constant), the 'SNaive' method was used instead.

Table 1
Training settings for local forecasters.

Base forecasters	Software tools	Hyper-parameters
1 ETS	'ets' function in the 'forecast' R package v.8.4	Under default settings, level = 95, frequency = 24
2 ARIMA	'Arima' function in the 'forecast' R package v.8.4	Non-seasonal order = (0,1,1), seasonal order = (0,1,1), other parameters are under default settings, level = 95, frequency = 24
3 Naive	'naive' function in the 'forecast' R package v.8.4	Under default settings, level = 95
4 SNaive	'snaive' function in the 'forecast' R package v.8.4	Under default settings, level = 95, frequency = 24
5 Theta	'thetaf' function in the 'forecast' R package v.8.4	Under default settings, level = 95, frequency = 24
6 RW	'rwf' function in the 'forecast' R package v.8.4	Drift = TRUE, level = 95
7 STL	'stlm' function in the 'forecast' R package v.8.4	Model function = stats::ar, other parameters are under default settings, level = 95, frequency = 24

4.6. Training hybrid deep meta-ensemble networks and benchmarks

To train HDME-Nets, we first used a local normalization process at each moving window steps. The mean value of each time series in the input window is calculated and each data point in the corresponding input window is then unscaled by the mean value using the division operator. After the global forecasts are generated from the global forecasting module, the forecasts are rescaled by their respective series means using the multiplication operator.

The number of filters in the three convolution blocks of the feature extractor are set to be 64, 128 and 64 respectively, these values are set according to existing time series classification literature (e.g., [35]) and the consideration on the scale of the dataset. To avoid over-fitting, the dropout rate in the feature extractor module was set to be 0.8 and 0.3 for point and interval forecasting tasks respectively. A smaller dropout rate for interval forecasting is reasonable, as the extracted features will be used to generate weights for both upper and lower quantile forecasts.

HDME-Nets with different global forecasters and for different forecasting tasks have various hyper-parameters. We limited the hyper-parameter search space based on some initial results and rules-of-thumb and explored it using Bayesian optimization algorithm (implemented in the R package '*rBayesianOptimization*') measuring performance on the validation set. Bayesian optimization has been widely used in the machine learning community. The basic principle of the optimization algorithm is to optimize a black-box function, e.g. the performance of a neural network as a function of the hyper-parameters, by iteratively estimating an approximation of the function and exploring the function space using the local minima of the approximation.

In Table 2, we summarize the bounds of the hyper-parameter values used throughout the training processes, and the picked hyper-parameter set from the top solutions of the exploration. For the HDME-Nets-MLP, we jointly tuned three hyper-parameters, including the number of neurons in the hidden layer, dropout rate of the hidden layer and the number of training epochs. Similarly, for HDME-Nets-seq2seq, we tuned the dimensionality of the encoder LSTM, the dimensionality of the decoder LSTM, the dropout rate in the encoder LSTM, and the number of training epochs. For HDME-Nets-wavenet, we tuned the number of the filters in the dilated convolution layers, the dropout rate after the convolution layers, and the number of training epochs. For the other benchmark models, for a fair comparison, we also tuned their hyper-parameters in a similar way. For all the benchmark models, the parameters that are not tuned were set as their default values.

We used Keras with Tensorflow as the backend to implement the proposed HDME-Nets, and used a stochastic gradient descent algorithm Adam [17] for network optimization. The batch size for training all HDME-Nets in our experiment are set to be 1024. All the experiments were run on a workstation with one NVIDIA Titan XP GPU. Neural networks are inherently parallel algorithms, and GPUs can take advantage of this parallelism to accelerate the training process. In our experiments, the training time of the HDME-Nets on GEFCom2012 data were in between 51 and 245 s, depending on the network structure of the HDME-Nets. The HDME-Nets-seq2seq usually needs more training time than the HDME-Nets with MLP or wavenet. Once the models are trained, the forecasts could be generated in seconds.

The hyper-parameters tuning algorithm need to repeat the training process many times to search for optimal hyper-parameters, so it consumed more computing time. The HDME-Nets-seq2seq on interval forecasting was the most time consuming networks for tuning, which took about 16 h on 300 Bayesian iterations, while the tuning procedure for HDME-Nets-MLP took about 3 h. Though time consuming, hyper-parameters once being tuned need not to be updated frequently in practice.

4.7. Results

The point forecasting performance of the proposed methods together with that of the benchmarks on the validation and test sets are reported in Tables 3 and 4 respectively. Among all the methods used, the proposed HDME-Nets clearly provided

Table 2
Hyper-parameters for training HDME-Nets.

Models	Hyper-parameters	Range	Picked value for point forecasting	Picked value for interval forecasting
ME-Nets	Number of epochs	(3L,15L)*10	100	130
HDME-Nets-MLP	Neurons in the hidden layer of MLP	(1L,3L)*48	144	144
	Dropout of the hidden layer of MLP	(2L,9L)/10	0.5	0.5
	Number of epochs	(3L,15L)*10	100	50
HDME-Nets-seq2seq	Units of encoder	(1L,3L)*48	48	144
	Units of decoder	(1L,3L)* 48	48	48
	Encoder dropout	(2L,9L)/10	0.1	0.7
	Number of epochs	(3L,15L)*10	60	80
HDME-Nets-wavenet	Dilute convolutional filters	(1L,3L)*48	48	48
	Dropout rate	(2L,9L)/10	0.5	0.7
	Number of epochs	(3L,15L)*10	150	150
MLP	Neurons in the hidden layer of MLP	(1L,3L)*48	144	96
	Dropout of the hidden layer of MLP	(1L,5L)/10	0.5	0.1
	Number of epochs	(3L,15L)*10	150	150
Seq2Seq	Units of encoder	(1L,3L)*48	144	144
	Units of decoder	(1L,3L)*5	48	48
	Encoder dropout	(2L,9L)/10	0.3	0.7
	Number of epochs	(3L,15L)*10	150	80
Wavenet	Dilute convolutional filters	(1L,3L)*48	48	48
	Dropout rate	(2L,9L)/10	0.1	0.1
	Number of epochs	(3L,15L)*10	150	150
GBRT	Learning rate	(1L,6L)/100	0.03	0.03
	Max depth	(8,12)	12	12
	Number of epochs	(5L,10L)*100	800	800
RF	Number of trees	(3L,6L)*100	500	500

L in the table indicates that the number is constrained to be integer.

Table 3
Point forecasting results for GEFCOM2012 utility load data on validation sets.

Methods	Horizon = 1 to 24				Horizon = 1 to 48			
	sMAPE	MASE	RMSE	MAE	sMAPE	MASE	RMSE	MAE
HDME-Nets-MLP	**8.13**	**1.45**	**12,055**	**6430**	**10.44**	**1.85**	**14,978**	**8312**
HDME-Nets-Seq2seq	8.59	1.54	12,511	6953	10.90	1.94	15,724	8907
HDME-Nets-wavenet	8.45	1.52	12,430	6822	10.82	1.93	15,635	8807
ME-Nets	8.86	1.59	12,872	7191	11.15	1.99	16,045	9128
MLP	9.10	1.61	13,025	7167	11.11	1.97	15,668	8850
Seq2Seq	9.08	1.59	12,519	7199	11.30	1.99	15,674	9084
Wavenet	8.98	1.58	12,605	7096	11.22	1.97	15,660	8983
RF	11.26	1.95	15,302	8847	12.86	2.25	17,649	10,222
GBRT	9.15	1.62	13,093	7328	11.03	1.95	15,585	8825
ARIMA	10.21	1.76	14,372	7952	14.53	2.48	20,220	11,367
ETS	13.24	2.14	22,916	9735	16.29	2.64	26,094	12,103
Naive	21.17	3.61	27,504	16,840	22.08	3.79	28,854	17,520
SNaive	10.61	1.86	14,826	8422	12.74	2.24	17,863	10,251
RW	21.13	3.58	27,010	16,624	22.56	3.83	28,890	17,662
STL	10.15	1.84	15,063	8359	12.29	2.20	17,684	10,082
Theta	10.41	1.87	15,526	8459	13.24	2.37	19,472	10,895

The best performance models are bolded.

the more accurate point forecasts over benchmarks on all the accuracy metrics over all the horizons and data sets. Among three HDME-Nets tested, HDME-Nets-MLP shows superior forecasting performance than the other two alternatives. It's a somewhat surprising result since the Seq2Seq and WaveNet are the more sophisticated networks reported in recent time series forecasting literature This indicates that the MLP could work well at least in forecasting hourly load data. HDME-Nets-wavenet is inferior to HDME-Nets-MLP but superior to HDME-Nets-Seq2seq. The same performance ranks are also found among MLP, Wavenet, and Seq2Seq, the three global forecasters. This indicates that the forecasting performance of the HDME-Nets is closely related to that of its global forecaster.

GBRT shows competitive point forecasting accuracy compared to the three global forecasters, but inferior to all three HDME-Nets. Though seven local forecasters showed poor performance, the ME-Nets, a simple meta-ensemble of the local forecasts, obtained similar forecasting accuracy with the three global forecasters and GBRT. The results provide evidence that the ensemble of global and local forecasts can improve the forecasts over either global or local forecasters.

Table 4
Point forecasting results for GEFCOM2012 utility load data on test set.

Methods	Horizon = 1 to 24				Horizon = 1 to 48			
	sMAPE	MASE	RMSE	MAE	sMAPE	MASE	RMSE	MAE
HDME-Nets-MLP	**8.66**	**1.67**	**12,373**	**6656**	**10.65**	**2.08**	**14,936**	**8401**
HDME-Nets-Seq2seq	9.17	1.78	13,001	7217	11.19	2.20	15,824	8985
HDME-Nets-wavenet	9.13	1.77	12,941	7170	11.07	2.17	15,464	8835
ME-Nets	9.74	1.90	13,828	7757	11.65	2.30	16,440	9436
MLP	9.36	1.80	13,014	7215	11.24	2.19	15,444	8850
Seq2Seq	9.82	1.89	13,560	7702	11.52	2.25	15,952	9261
Wavenet	9.76	1.88	13,416	7596	11.63	2.28	15,980	9290
RF	10.99	2.11	14,722	8481	12.37	2.41	16,747	9718
GBRT	9.81	1.90	13,539	7563	11.39	2.24	15,805	8997
ARIMA	10.66	2.05	14,874	8303	14.40	2.78	20,077	11,371
ETS	16.48	2.80	26,287	11,752	19.43	3.30	29,095	13,890
Naive	16.31	3.10	21,422	12,951	17.69	3.40	23,526	14,181
SNaive	12.07	2.36	17,119	9670	13.93	2.75	19,595	11,314
RW	16.65	3.14	21,593	13,113	18.76	3.56	24,614	14,833
STL	10.91	2.12	15,416	8689	12.66	2.48	17,882	10,270
Theta	11.78	2.28	16,684	9261	13.98	2.73	19,676	11,171

The best performance models are bolded.

Table 5
95% interval forecasting results for GEFCOM2012 utility load data.

	Validation set				Test set			
	Horizon = 1 to 24		Horizon = 1 to 48		Horizon = 1 to 24		Horizon = 1 to 48	
	MSIS	ACD%	MSIS	ACD%	MSIS	ACD%	MSIS	ACD%
HDME-Nets-MLP	**10.60**	0.27	**13.14**	0.52	13.21	0.92	15.26	0.62
HDME-Nets-Seq2seq	11.45	0.14	13.72	0.48	14.05	0.73	16.18	0.21
HDME-Nets-wavenet	10.67	1.20	13.19	0.52	**12.96**	0.13	**15.02**	0.03
ME-Nets	12.53	0.65	14.70	0.42	14.45	0.69	16.61	0.06
MLP	12.09	2.40	13.87	1.46	14.82	0.44	16.54	0.79
Seq2Seq	15.04	1.64	16.66	2.41	16.09	1.17	17.42	1.87
Wavenet	11.80	0.45	14.09	0.63	13.94	1.30	15.98	1.22
RF	14.31	3.29	15.97	1.00	16.01	2.72	17.51	0.98
GBRT	17.25	4.16	18.29	3.21	17.28	2.84	18.70	1.38
ARIMA	14.10	0.05	19.44	1.02	16.44	1.18	22.61	2.37
ETS	15.76	3.32	19.35	1.66	19.77	3.44	25.47	1.13
Naive	19.92	1.22	25.08	2.86	20.48	1.95	26.28	3.24
SNaive	16.31	4.65	18.92	4.60	19.42	3.86	21.10	2.90
RW	20.46	2.83	28.12	3.84	21.73	2.87	29.94	3.86
STL	18.55	12.83	24.23	17.24	20.64	11.37	25.19	14.59
Theta	15.87	6.25	19.52	5.74	18.34	3.11	22.00	1.34

The best performance models are bolded.

Table 5 presents the interval forecasting results. Most of the results on relative forecasting performance of the tested methods are similar to that in Tables 3 and 4. Once again, three HDME-Nets are all among the most correct interval forecasts evaluated by MSIS, while the traditional time series methods display significantly worse performance. Among three HDME-Nets under tested, HDME-Nets-MLP shows the best forecasting performance in the validation set, but HDME-Nets-wavenet is the most accurate model in the test set. The improvements in terms of MSIS by HDME-Nets over the benchmarks (including ME-Nets and global forecasters) are ranging from 50% to at least 5%. Moreover, the coverages achieved by HDME-Nets and ME-Nets are very close to a perfect score of 0.95 (only about 0.5% differences on average), which means that both of them did an excellent job of estimating the forecasting uncertainty. These findings demonstrate that the traditional time series forecasting methods fail to estimate the uncertainty properly.

4.8. Robustness check

To investigate the robustness of the forecasting performance of the proposed HDME-Nets showed in Tables 3–6, we further tested the method together with the tuned hyper-parameters listed in Table 2 to another electric utility load dataset which was used in GEFCom2017[2] [13]. The dataset contains 165 hourly load time series of delivery point meters, is at a larger

[2] The whole dataset can be downloaded from https://doi.org/10.1016/j.ijforecast.2019.02.006.

Table 6
Point forecasting results for GEFCOM2017 utility load data.

Methods	Horizon = 1 to 24				Horizon = 1 to 48			
	sMAPE	MASE	RMSE	MAE	sMAPE	MASE	RMSE	MAE
HDME-Nets-MLP	8.67	1.31	1189	489	11.22	1.70	1489	634
HDME-Nets-Seq2seq	8.73	1.31	1176	488	11.57	1.74	1510	651
HDME-Nets-wavenet	8.92	1.35	1225	504	11.57	1.75	1525	655
ME-Nets	9.85	1.49	1335	558	12.32	1.86	1624	699
MLP	9.26	1.35	1226	503	11.73	1.72	1501	641
Seq2Seq	10.61	1.55	1313	582	12.82	1.89	1569	709
Wavenet	9.15	1.34	1225	501	11.80	1.75	1516	652
RF	11.09	1.65	1457	642	13.25	1.98	1690	759
GBRT	10.26	1.54	1323	587	12.51	1.88	1586	712
ARIMA	11.53	1.67	1613	634	16.20	2.37	2315	907
ETS	18.52	2.30	2394	866	22.14	2.78	2692	1042
Naive	22.39	3.20	2899	1373	23.85	3.44	3021	1439
SNaive	12.08	1.80	1590	665	14.71	2.22	1891	817
RW	22.63	3.21	2867	1367	24.90	3.57	3060	1474
STL	11.69	1.75	1525	657	13.94	2.10	1754	782
Theta	12.48	1.88	1676	707	15.66	2.35	2037	884

scale than the load forecasting problems of GEFCom2012. The data spans 7 years from 2005/01/01/ to 2011/12/31. For each load series, we used its latest 20% of the periods as the test set, the other 80% of the data as the training set. Similarly, with the data processing process on GEFCom2012 data, we used the same moving window strategy to transform a time series into pairs of <input, output> patches. The width of the input and output window is also set to be 120 and 48 h respectively. The patches that contain missing values are simply dismissed.

The point and interval forecasting results on GEFCom2017 data are presented in Tables 6 and 7 respectively. On the point forecasting, we got similar relative forecasting performance results with that on GEFCom2012. Three HDME-Nets still dominate all benchmark methods, and HDME-Nets-MLP is still the most accurate model among the three HDME-Nets. On the interval forecasting, three HDME-Nets still show improved forecasting performance over their global and local forecasters, as well as other benchmarks, and HDME-Nets-MLP and HDME-Nets-wavenet still are the top performing two models.

5. Conclusions

We have presented a novel meta-ensemble modeling framework based on deep learning neural networks for forecasting a large number of related time series. To improve the forecasting accuracy, it assigns weights using a meta-ensemble technique to the forecasts from both local and global forecasters according to the characteristics of each time series. Different from existing methods, it extracts features from each time series automatically in a supervised manner, and it can generate both point and interval forecasts using meta-ensemble. It is also extensible and flexible: all the modules in the framework could be designed according to the characteristics of the specific time series forecasting applications.

We tested HDME-Nets on two public obtainable utility load datasets, and compared their point and interval forecasting performance. Compared to all the benchmarks, the proposed HDME-Nets clearly provide the most accurate point and

Table 7
95% interval forecasting results for GEFCOM2017 utility load data.

Methods	Horizon = 1 to 24		Horizon = 1 to 48	
	MSIS	ACD	MSIS	ACD
HDME-Nets-MLP	8.88	1.11	11.08	0.06
HDME-Nets-Seq2seq	9.35	1.13	11.75	2.35
HDME-Nets-wavenet	9.11	1.04	11.27	0.25
ME-Nets	10.76	0.75	12.98	0.29
MLP	8.93	1.12	11.60	2.94
Seq2Seq	10.87	3.37	13.98	6.25
Wavenet	10.39	0.52	12.78	1.61
RF	13.18	3.91	14.53	1.74
GBRT	15.84	4.03	16.86	2.88
ARIMA	14.18	1.35	20.24	2.09
ETS	16.19	2.48	24.66	0.47
Naive	18.15	1.94	23.31	3.28
SNaive	17.43	4.95	19.87	4.84
RW	19.39	2.85	26.54	3.79
STL	16.84	10.74	21.54	14.13
Theta	14.55	3.84	17.90	2.47

interval forecasts on all the accuracy metrics over all the horizons. The results provided evidence that the meta-ensemble on global and local forecasts can provide more accurate forecasts than either global or local forecasters. This improvement is entirely due to the proposed hybrid meta-ensemble approach. Among the three HDME-Nets we designed for forecasting load time series, we found that HDME-Nets-MLP shows relatively better forecasting performance compared with the other two alternatives. The results are robust as the proposed methods showed similar forecasting performance on different data sets.

The results, as with any empirical study, still suffer from the limitations of using a particular data source. Further research could test our approach on many other practical forecasting scenarios, e.g., transportation, retail, or financial forecasting applications, which would be helpful to generalize the findings we reported. Further research could also design domain specific HDME-Nets according to the data characteristics, to consider more global forecasters (e.g. Stochastic configuration networks [34]), to explore the composition of various local forecasters, as well as to develop more sophisticated automatic feature extraction methods for HDME-Nets.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The author acknowledges the support of the National Natural Science Foundation of China under grant no. 71571089.

References

[1] B. Anastasia, B. Sander, W.O. Cornelis, Dilated convolutional neural networks for time series forecasting, J. Comput. Finance 22 (2019) 73–101.
[2] A. Borovykh, S. Bohte, K. Oosterlee, Conditional time series forecasting with convolutional neural networks, Lect. Notes Comput. Sci. 729–730 (2017).
[3] C. Cecati, J. Kolbusz, P. Różycki, P. Siano, B.M. Wilamowski, A novel RBF training algorithm for short-term electric load forecasting and comparative studies, IEEE Trans. Ind. Electron. 62 (2015) 6519–6529.
[4] M. Dekker, K. van Donselaar, P. Ouwehand, How to use aggregation and combined forecasting to improve seasonal demand forecasts, Int. J. Prod. Econ. 90 (2004) 151–167.
[5] G. Dudek, Neural networks for pattern-based short-term load forecasting: a comparative study, Neurocomputing 205 (2016) 64–74.
[6] G.T. Duncan, W.L. Gorr, J. Szczypula, Forecasting Analogous Time Series, Springer, US, 2001.
[7] V. Flunkert, D. Salinas, J. Gasthaus, DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks, arXiv:1704.04110, 2017.
[8] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (2001) 1189–1232.
[9] H. Hahn, S. Meyer-Nieberg, S. Pickl, Electric load forecasting methods: tools for decision making, Eur. J. Oper. Res. 199 (2009) 902–907.
[10] B.F. Hobbs, S. Jitprapaikulsarn, S. Konda, V. Chankong, K.A. Loparo, D.J. Maratukulam, Analysis of the value for unit commitment of improved load forecasts, IEEE Trans. Power Syst. 14 (1999) 1342–1348.
[11] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[12] T. Hong, P. Pinson, S. Fan, Global energy forecasting competition 2012, Int. J. Forecast. 30 (2014) 357–363.
[13] T.J. Hong, J. Xie, Black Global energy forecasting competition 2017: hierarchical probabilistic load forecasting, Int. J. Forecast. 35 (2019) 1389–1399.
[14] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. (2019), 1–1.
[15] R.J. Hyndman, A.J. Lee, E. Wang, Fast computation of reconciled forecasts for hierarchical and grouped time series, Comput. Stat. Data Anal. 97 (2014) 16–32.
[16] T. Januschowski, J. Gasthaus, Y. Wang, D. Salinas, V. Flunkert, M. Bohlke-Schneider, L. Callot, Criteria for classifying forecasting methods, Int. J. Forecast. 36 (2019) 167–177.
[17] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: 3rd International Conference for Learning Representations, arXiv:1412.6980, San Diego, 2015.
[18] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time-series, in: The handbook of brain theory and neural networks, MIT Press, Arbib, M.A., 1995.
[19] C.M. Lee, C.-N. Ko, Short-term load forecasting using lifting scheme and ARIMA models, Expert Syst. Appl. 38 (2011) 5902–5911.
[20] L.L. Li, J. Sun, C.H. Wang, Y.T. Zhou, K.-P. Lin, Enhanced Gaussian process mixture model for short-term electric load forecasting, Inf. Sci. 477 (2019) 386–398.
[21] S. Ma, R. Fildes, A retail store SKU promotions optimization model for category multi-period profit maximization, Eur. J. Oper. Res. 260 (2017) 680–692.
[22] S. Ma, R. Fildes, Forecasting third-party mobile payments with implications for customer flow prediction, Int. J. Forecast. 36 (2020) 739–760.
[23] S. Ma, R. Fildes, T. Huang, Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra- and inter-category promotional information, Eur. J. Oper. Res. 249 (2016) 245–257.
[24] D.C. Maddix, Y. Wang, A. Smola, Deep factors with Gaussian processes for forecasting, in: Third Workshop on Bayesian Deep Learning (NeurIPS 2018), Montreal, Canada, 2018.
[25] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The M4 Competition: 100,000 time series and 61 forecasting methods, Int. J. Forecast. 36 (2019) 54–74.
[26] H.A. Malki, N.B. Karayiannis, M. Balasubramanian, Short-term electric power load forecasting using feedforward neural networks, Expert Syst. 21 (2004) 157–167.
[27] J.Z. Nowicka, R. Weron, Modeling electricity loads in California: ARMA models with hyperbolic noise, Signal Process. 82 (2002) 1903–1915.
[28] A.V.D. Oord, S. Dieleman, H. Zen, K. Simonyan, K. Kavukcuoglu, WaveNet: a generative model for raw audio, arXiv:1609.03499, 2016.
[29] C.L. Philip Chen, C.Y. Zhang, Data-intensive applications, challenges, techniques and technologies: a survey on Big Data, Inf. Sci. 275 (2014) 314–347.
[30] M.Q. Raza, A. Khosravi, A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings, Renew. Sust. Energy Rev. 50 (2015) 1352–1372.
[31] S. Smyl, A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting, Int. J. Forecast. 36 (2019) 75–85.
[32] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 2, MIT Press, Montreal, Canada, 2014, pp. 3104–3112.
[33] J.W. Taylor, Short-term load forecasting with exponentially weighted methods, IEEE Trans. Power Syst. 27 (2012) 458–464.
[34] D. Wang, M. Li, Stochastic configuration networks: fundamentals and algorithms, IEEE Trans. Cybern. 47 (2017) 3466–3479.
[35] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: a strong baseline, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1578–1585.

[36] R. Wen, K. Torkkola, B. Narayanaswamy, A multi-horizon quantile recurrent forecaster, in: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.
[37] R. Wen, K. Torkkola, B. Narayanaswamy, A multi-horizon quantile recurrent forecaster, arXiv:1711.11053, 2017.
[38] J. Xie, T. Hong, Temperature scenario generation for probabilistic load forecasting, IEEE Trans. Smart Grid 9 (2018) 1680–1687.
[39] G. Zotteri, M. Kalchschmidt, A model for selecting the appropriate level of aggregation in forecasting processes, Int. J. Prod. Econ. 108 (2007) 74–83.

Shaohui Ma received M.Sc. and Ph.D. degrees in Management Science from Tianjin University, China. In 2007, he joined the Jiangsu University of Science & Technology as an assistant professor and became a professor in management science in 2016. From 2014 to 2015, he was a visiting researcher at Forecasting Centre in Lancaster University. In 2018, he joined the Nanjing Audit University and worked as the director of the Centre for management science. Dr. Ma has more than 40 papers published in peer-reviewed journals and conferences. His research has been ongoing funded by National Natural Science Foundation of China. His research interests include machine learning, data mining and time series processing with applications in retail, energy, and transportation.