Improving Load Forecasting Based on Deep Learning and K-shape Clustering

Fateme Fahiman*, Sarah M.Erfani†, Sutharshan Rajasegarar*, Marimuthu Palaniswami*, Christopher Leckie†
*Department of Electrical and Electronic Engineering, University of Melbourne, Australia
Email: ffahiman@student.unimelb.edu.au; {sraja, palani}@unimelb.edu.au
†Department of Computing and Information Systems, The University of Melbourne, Australia
Email: {sarah.erfani, caleckie}@unimelb.edu.au

***Abstract*—One of the most crucial tasks for utility companies is load forecasting in order to plan future demand for generation capacity and infrastructure. Improving load forecasting accuracy over a short period is a challenging open problem due to the variety of factors that influence the load, and the volume of data that needs to be considered. This paper proposes a new approach for short term load forecasting using an effective new combination of clustering and deep learning methods, along with a new weighted aggregation mechanism. Our evaluation using smart meter data from a publicly available real-life dataset demonstrates the improved accuracy of our approach over existing methods.**

I. Introduction

The smart grid introduces the two-way flow of data between electricity suppliers and customers. In this new system, customers can play an important role in achieving energy savings by modifying their electricity consumption pattern according to the dynamic electricity market [1], and selling renewable energy on an open market [2] through demand response mechanisms. The demand response mechanism depends on the convergence of accurate electric power load measurements, renewable energy, and price forecasting. Therefore, a crucial task for utility companies in this contest is load forecasting. With accurate prediction of future demand, electric utilities can make important decisions such as generating and purchasing electric load, distributing and developing their infrastructure, and optimizing load switching in order to manage their power grid.

Any error in load forecasting can result in significant cost increases for electricity suppliers and increase the chance of unexpected blackouts or brownouts. In contrast, accurate load forecasting can result in improvements to network reliability. Therefore, trying to produce more accurate load forecasting models has become a major research challenge for energy suppliers, energy marketers, financial markets, and other parties that contribute to electric power generation, distribution and transmission.

Load forecasting has been widely studied since the 1970s. Traditional methods introduced linear models for time-series load forecasting, such as autoregressive (AR) and autoregressive with moving average (ARMA) models [3], [4]. A comprehensive review on short term load forecasting can be found in [5].

The Global Energy Forecasting Competition 2012 (GEFCom2012) brought together the state-of-the-art techniques for energy forecasting and introduced the hierarchical forecasting track. In [6], the authors summarize the methods used by the participants in the GEFCom2012. It should be noted that these forecasting competitions have focused on zonal level demand forecasting, i.e., when predictions are made based on aggregated data from large regions or zones of the power grid, which comprise a large number of households. In contrast, our focus is on demand forecasting for household level data, where data is available from meters at individual households. In particular, this raises the interesting challenge of whether it is possible to improve forecasting accuracy by constructing finer grained models for sets of similar households, compared to coarse grained prediction models based on aggregate data at zonal level.

Currently, nonlinear forecasting models have generally obtained better accuracy than linear models. These nonlinear models are based on machine learning methods such as neural networks [7], support vector machines [8], and k-nearest neighbours approaches [9]. Neural networks have been used widely in load forecasting due to their ability to approximate complex nonlinear relationships. However, neural network methods have some potential drawbacks such as overfitting of the model, sensitivity to random weight initialization, and tendency to convergence to local optima. To address these limitation of traditional neural networks, recently new approaches called deep neural networks have been proposed. Deep belief networks with many nonlinear levels can represent complex features from their inputs [4], and obtain a more general model with the ability to learn these complex features from the data. In this paper, we propose a method for load prediction using Hinton and Salakhutdinov's deep belief nets (DBN), which are a type of probabilistic generative neural network composed by multiple layers of restricted Boltzmann machines (RBM) [10]. In this study, two forecasting methods, based on deep neural networks and traditional neural networks, are proposed and evaluated in terms of their prediction accuracy on real life residential load data.

Key components of smart grid technology are smart meters, which provide fine-grained energy consumption information at sampling intervals of 15, 30 or 60 minutes. One of the most promising applications for such large volumes of data

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

from smart meters is to improve the accuracy of electrical load forecasting. Typical methods for improving load forecasts using the data generated from smart meters of individual customers is based on the use of clustering [11], [12]. In this approach, the knowledge about load consumption behaviour of customers is used to improve the accuracy of forecasting. Instead of developing a single forecasting model for the accumulated load consumption of all customers, clustering can be used to divide the customers into sub-populations with similar demand profiles. Then, a forecasting model can be provided for each cluster of customers according to their load profile. Thus, a more accurate model can be provided for each cluster, and then the load forecast for all consumers can be obtained by combining the forecasts of these models. Applying clustering as an initial step in electric load forecasting has been the focus of several studies [13], [14]. The K-means clustering method has been widely used in previous works for this purpose [11], [15]. However, little attention has been paid in previous studies to the choice of clustering method with the aim of improving the aggregate level of forecasting accuracy. In this paper, we show that the accuracy of load forecasting depends not only on the load forecasting method, but also on the accuracy of the clustering method and the accuracy of extracting of the load pattern of each cluster. In addition, we show that careful clustering of customers can result in smaller forecasting error. We apply a time-series clustering algorithm (k-shape clustering [16]) that aims to reduce the error in load forecasting through more accurate assignment of individuals to clusters based on their load consumption profile.

A. Contributions made in this paper

In this paper we propose a new method for load forecasting which leads to improved accuracy of short term load forecasting. We show that by appropriate choice of features we can significantly improve the accuracy of forecasting. Moreover, we study a time-series clustering method (K-shapes) that can cluster consumers into the groups based on their load consumption profiles, such that the accuracy of models based on each K-shape cluster is more accurate than models based on traditional K-means clustering. In addition, we implement our proposed approaches on publicly available real-life data and demonstrate the biggest improvement in accuracy is achieved by applying both new approaches, i.e., applying the new clustering method and the new forecasting method. Moreover, in contrast to previous work [11], [15], [17] that simply summed the forecasting results of each cluster to obtain the final aggregate forecast, we introduce a new weighted summation method for accumulating the forecasting results of each cluster according to the size of their membership.

B. Paper Outline

In the next section, we briefly explain the existing methods for aggregate-level load forecasting, and our new approach for this problem. In Section III we discuss the new clustering method (K-shape) that we use to identify sub-populations of customers with similar behaviour. In Section IVwe explain the use of shallow vs deep neural network methods for learning load forecasting models. Then, in Section V we show our experimental results based on real-life smart meter data, and give a comparison of the results of our proposed method against existing methods. Finally, in Section VI we conclude our study and discuss some possible directions for future work.

II. Methods

A. Smart Meter Dataset

The real-world smart meter records from the Commission for Energy Regulation [18] are used in this paper. This dataset was recorded in Ireland from July 14, 2009 to December 31, 2010 with more than 6000 Irish consumers at a resolution of 30 min. The dataset is divided in to three groups of consumers: residential, small-to-medium enterprises, and others. In this paper only the residential group of consumers are studied, which include 3176 homes and 17 months of 30 minutes interval data from August 2009 to December 2010. The first 14 months were used as the training set. One month was used for validation, and the model was tested through the last two months of data. Through data pre-processing, consumers with a large number of successive missing records were eliminated. For consumers with single missing records, linear interpolation between the records in the vicinity of the missing ones was applied to complete the dataset. Moreover, temperature data for the mentioned time periods was collected from the website: wunderground.com. Using these data sets, and applying the time-series clustering (K-shape) and deep neural network methods, we aim to obtain the best prediction accuracy for forecasting time horizons ranging from 30 min to one day ahead.

B. Problem Statement

The smart meter records consist of a set of M consumers x_1, x_2, ..., x_M. The consumption history of consumer i can be defined as $x_i = \{\ x_i^1, x_i^2, ..., x_i^T\ \}$, $\forall i \in \{1, 2, ..., M\}$, where T is the total number of historical time periods and M is the total number of consumers. Our aim is to predict the electrical load consumption for the aggregated load of the network. The total load consumption of the network can be represented as:

$$X_{aggr}^t = \sum_{i=1}^{M} x_i^t, \qquad \forall t \in \{1, 2, ..., T\}. \tag{1}$$

Three approaches for load prediction at the network level have been reported in the literature. The first approach is a completely aggregated method [17], which accumulates the load consumption of all consumers that belong to the network into a vector $X_{aggr} = (X_{aggr}^1, X_{aggr}^2, ..., X_{aggr}^T)$ and take this as the input feature vector for forecasting. The second method is a completely disaggregated method [17], which applies load forecasting for individual consumers, and then adds the individual predictions to obtain the prediction at the aggregated level. The third method is a clustering based forecast, where the consumers are divided into specific clusters and then the sum of the load consumption over each cluster is

taken, and finally a forecast for each group is generated and these predictions for each cluster are accumulated. These three models are illustrated in Fig. 1(a), (b) and (c). We propose a different approach in this paper as a clustering based weighted forecasting model, which is illustrated in Fig. 1(d).

The insight of using clustering to group the individual customers is as follows. It is known that load forecasting for individual customers is complicated because each household's load depends on a variety of factors. At the granularity of half an hour, the energy consumed by a single house can fluctuate widely, due to the dependency on the number of residents in the home, the appliances that are used at the particular time, the lifestyle of the people staying in the home, and so on. Thus, accurate load prediction for an individual household can be difficult due to the individual variations in consumption patterns. To improve the overall accuracy of forecasting, an alternative approach is to group households, and analyse the aggregate level of consumption so that individual variations in consumption will tend to cancel each other out. This grouping can be achieved by the application of an appropriate clustering method, such that households with a similar load time series are grouped together, where the total noise in each cluster would tend to cancel out.

We consider that by applying a clustering algorithm, k clusters C=$\{c_1, c_2, ...,c_k\}$ are obtained, so that each cluster of households in C can then be used to train a neural network. Therefore k prediction models F_C=$\{F_{c_1}, F_{c_2}, ...,F_{c_k}\}$ are generated from the k groups of consumers. To calculate the final prediction for period of interest, some studies [11], [15] simply take the sum over the predictions from the clusters as follows:

$$F_{aggr} = \sum_{i=1}^{k} F_{c_i}, \tag{2}$$

This summation can be used to calculate the accuracy of the forecast based on the mean absolute percentage error (MAPE):

$$MAPE = \frac{100}{T} \sum_{t=1}^{T} \mid \frac{F^t_{agrr} - X^t_{aggr}}{X^t_{aggr}} \mid . \tag{3}$$

This approach works well if the clusters have similar sizes. When the clusters have widely varying sizes, then better results can generally be achieved by using a weighted summation of the forecasts from each cluster. We can define these weights according to the number of members in each cluster. Assume that N_{c_i} is the number of consumers that belong to the cluster c_i, therefore the weight of the cluster c_i can be defined as W_{c_i}= $\frac{N_{c_i}}{M}$, where M is the total number of consumers. We can then modify equations (2) and (3) as follows:

$$F_{agrr,w} = \sum_{i=1}^{k} W_{c_i}.F_{c_i}, \tag{4}$$

$$X_{agrr,w} = \sum_{i=1}^{k} W_{c_i}.X_{agrr,c_i}, \tag{5}$$

where X_{aggr,c_i} is the total load consumption of the consumers that belong to cluster i. Now, we can define the mean absolute percentage error as:

$$MAPE = \frac{100}{T} \sum_{t=1}^{T} \mid \frac{F^t_{agrr,w} - X^t_{aggr,w}}{X^t_{aggr,w}} \mid . \tag{6}$$

C. Feature Selection

The load forecasting accuracy is influenced by a variety of factors. While neural networks can fit highly non-linear models, the selection of input variables, or features, is also very important and has a major impact on the accuracy of load prediction. Fig. 2 demonstrates the influence of human added features on the overall model performance. We considered four sets of features, namely, load variables, temperature variables, time cyclic variables, and day-type variables. A separate neural network is trained for each set of features, and the $MAPE$ of each network is shown in Fig. 2.

1) Load Features: Electrical load consumption usually exhibits a daily periodicity pattern. In addition, load consumption profiles in adjacent weeks exhibit strong positive correlations. According to these characteristic we define six variables: load at time t of the previous day, load at time t-30 minutes of the previous day, load at the same day and same time of the previous week, average load of the previous day, minimum load of the previous day, and maximum load of the previous day.

2) Temperature Features: The weather can have a major impact on the electrical load consumption pattern. We have defined seven features as temperature variables: the temperature at time t and the same time of previous day, maximum and minimum temperature of the previous day, past 3 hours, 6 hours, and 24 hours average temperatures. Note that because the weather data set is available for an hour granularity, linear interpolations should be used for producing temperature data every 30 minutes.

3) Time Cyclic Features: These features reflect the cyclic characteristic of the time of day and the type of day. Cyclic variables are extracted in order to capture the cyclic nature of load time series [19]. By spectral analysis, the half day, day, week and year periods have been identified as the dominant frequencies of the load data [20]. For each frequency, a pair of variables is considered to represent the corresponding cycles:

$$\begin{aligned} c_1(t) &= \sin(\frac{2\pi t}{T}) \\ c_2(t) &= \cos(\frac{2\pi t}{T}), \end{aligned} \tag{7}$$

where t is the time indicator of the half hourly granularity sampling, which extends from 1 to 17520 for one year. The variable T represents the cycle period that is equal 17520, 336, 48, and 24 for a year, a week, a day, and a half-day cycle, respectively.

4) Type of Day Features: Discriminating between working days and non-working days is also an important feature that should be considered. The load analysis has shown that load consumption on non-working days is lower than on working

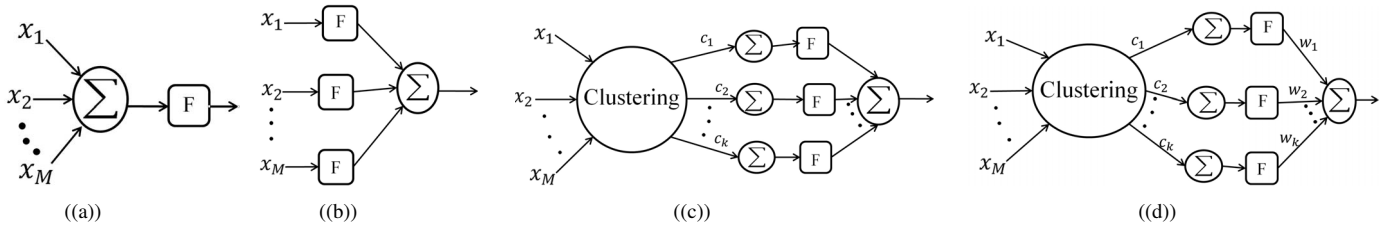

Fig. 1. Different methods for network-level load forecasting. (a) Completely aggregated method. (b) Completely disaggregated method. (c) Clustering based forecasting method. (d) Clustering based weighted forecasting method

days. The non-working days (weekends and national holidays) and working days are represented by 0 and 1, respectively, as the type of day variables.

The summary of the features that have been used for training the neural networks are given as follows:

Load features = $\{L(t-1dat),\ L(t-1day-30Minutes),\ L(t-1week),\ L_{max}(t-1day),\ L_{min}(t-1day),\ L_{mean}(t-1day)\}$

Temperature features=$\{T(t),\ T(t-1day),\ T_{max}(t-1day),\ T_{min}(t-1day),\ T_{mean}(t-3hours),\ T_{mean}(t-6hours),\ T_{mean}(t-24hours)\}$

Cycle Features = $\{\cos(\frac{2\pi t}{336}),\ \sin(\frac{2\pi t}{336}),\ \cos(\frac{2\pi t}{48}),\ \sin(\frac{2\pi t}{48}),\ \cos(\frac{2\pi t}{24}),\ \sin(\frac{2\pi t}{24}),\ \cos(\frac{2\pi t}{17520}),\ \sin(\frac{2\pi t}{17520})\}$

Type of day features = $\{weekdays,\ weekends\ and\ holidays\}$

where T refers to temperature and L refers to load.

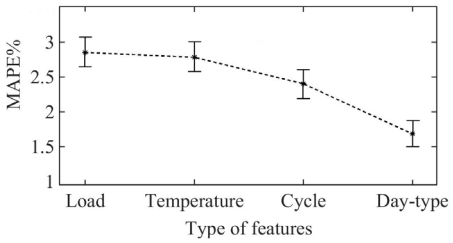

Fig. 2. Mean absolute percentage error achieved using different types of features

III. Grouping Smart Metering Data Based on Consumption Profiles

Recently, there have been a number of studies that segment customers based on their electrical consumption as a prelude to forecasting demand [5], [21]. This segmentation can be performed using clustering methods on the smart meter data from customers. As we shall see, this helps to achieve a deeper understanding of the load profile characteristics as well as greater prediction accuracy.

In contrast to these previous works, we apply a time series approach to clustering the load consumption patterns of each consumer based on the shape of their weekly profile. By clustering the time series data, we can analyze its variations over time in detail, and search for common profiles that can be potentially useful for load prediction. Such profiles can be extracted according to the shape of the load profiles. We search for groups of customers that show similar behavior over time. We use a time-series clustering approach that aims to reduce the error in the aggregate forecast through suitable assignment of individuals to clusters based on the shape of their load profiles. In contrast to traditional clustering methods such as K-means [6], [16], which tend to cluster customers with similar average load levels, we require an approach that clusters customers based on the similarity of their load profile over the course of the day. These groups of customers with similar consumption patterns can be used to achieve a more accurate aggregate load forecast. We next describe the clustering method we have used, which is called K-shape clustering.

A. K-shape Clustering

Numerous clustering algorithms have been used for the load prediction problem. Determining the best algorithm depends greatly on the nature, purpose and mining objectives of the dataset. In this paper, we use K-shape, an algorithm for time-series clustering that can conserve the shapes of time-series sequences [16]. This new time-series clustering algorithm creates consistent and well separated groups of time-series. K-shape considers the shape of the time series during clustering, in contrast to traditional methods such as K-means, which treat the observations in a time series as independent attributes. This time-series clustering method uses a normalized, domain-independent form of cross correlation as its distance measure. Using this method, the K-shape algorithm derives a shape-based distance measure for comparing the time series efficiently and effectively. Then, based on the properties of the shape-based distance measure, K-shape computes cluster centroids, which are used in each iteration to capture shared characteristics of the underlying data and update the assignment of time series to clusters.

The extraction of a representative centroid for each cluster is a challenging task that critically depends on the choice of distance measure. By extracting the centroid, the clustering algorithm can effectively summarize a set of time series in terms of only one sequence, and extract the most representative shape from the underlying data. Then, these extracted shapes or centroids are used for clustering the time-series. The robustness of K-shape clustering has been experimentally evaluated against partitional methods like K-means, K-medoid, hierarchical and spectral clustering methods, with combinations of the most competitive distance measures [16]. It has been

shown [16] that K-shape outperforms all of these approaches in terms of accuracy on time series data. In this paper we compare our results using K-shape and K-means methods, and it will be shown in the experimental results section that K-shape significantly outperforms K-means with respect to clustering accuracy, and as a result forecasting accuracy.

In summary, the K-shape method groups sequences exhibit similar patterns into the same cluster according to their shape similarity, regardless of differences in amplitude and phase. K-shape clustering is a domain-independent, accurate, and scalable algorithm that uses a distance measure which is invariant to scaling and shifting. This clustering method can thus preserve the shapes of time-series sequences. The K-shape clustering algorithm consists of three main components: (1) a shape-based distance measure, (2) time series shape extraction, and (3) shape-based time series clustering. We now give an overview of each of these key components.

1) Shape-Based Distance: The first component is the Shape Based-Distance, which is based on a cross-correlation measure. This algorithm derives a scale- and shift-invariant time-series distance measure in a computationally efficient way. Consider two sequences $\overrightarrow{x} = (x_1, x_2, ..., x_m)$ and $\overrightarrow{y} = (y_1, y_2, ..., y_m)$. The similarity of these two sequences can be determined as follows:

$$CC_w(\overrightarrow{x}, \overrightarrow{y}) = R_{w-m}(\overrightarrow{x}, \overrightarrow{y}), \tag{8}$$

where $CC_w(\overrightarrow{x}, \overrightarrow{y})$ is the cross-correlation sequence with length $2m-1$, and $R_{w-m}(\overrightarrow{x}, \overrightarrow{y})$ is computed as:

$$R_k(\overrightarrow{x}, \overrightarrow{y}) = \begin{cases} \sum_{l=1}^{m-k} x_{l+k}.y_l & : k \geq 0 \\ R_{-k}(\overrightarrow{y}, \overrightarrow{x}) & : k < 0 \end{cases} \tag{9}$$

For scaling invariance, this algorithm uses the Z-normalization method, where each sequence $\overrightarrow{x}$ is transformed into $\overrightarrow{x}' = \frac{\overrightarrow{x} - \mu}{\sigma}$, where μ is the mean and σ is the standard deviation. In order to address the shift invariance problem this algorithm useS coefficient normalization, which generates values between $[-1, 1]$. By coefficient normalization, the cross-correlation sequence is divided by the 5^{th} geometric mean of the autocorrelation of the individual sequences. So, we can define the normalized cross-correlation coefficient NCC_c as follows:

$$NCC_c(\overrightarrow{x}, \overrightarrow{y}) = \frac{CC_w(\overrightarrow{x}, \overrightarrow{y})}{\sqrt{R_0(\overrightarrow{x}, \overrightarrow{x}).R_0(\overrightarrow{y}, \overrightarrow{y})}}. \tag{10}$$

The goal is to find the position w where $NCC_c(\overrightarrow{x}, \overrightarrow{y})$ is maximized. Thus the Shape Based Distance (SBD) can be calculated as:

$$SBD(\overrightarrow{x}, \overrightarrow{y}) = 1 - \max_w(NCC_c(\overrightarrow{x}, \overrightarrow{y})), \tag{11}$$

which yields values between 0 to 2, with 0 indicating perfect similarity for time-series sequences.

2) Time-Series Shape Extraction: The second component of K-shape clustering is defining a centroid that summarizes a set of time series. Extracting the shape of a set of time series, which is able to capture the shared characteristics of the underlying data, is a difficult task that depends on the SBD distance measure. To achieve accurate shape extraction, this algorithm casts centroid computation as an optimization problem, where the objective function is to find the maximizer μ_k^* of the squared similarities to all other time-series sequences. So, we can formulate the optimization problem as follows:

$$\overrightarrow{\mu}_k^* = \arg\max_{\overrightarrow{\mu}_k} \sum_{\overrightarrow{x_i} \in P_k} (NCC_c(\overrightarrow{x}, \overrightarrow{\mu_k}))^2, \tag{12}$$

where P_k is the k^{th} partition and $\overrightarrow{\mu}_k$ is the initial centroid for the k^{th} partition. This equation uses the previously computed centroid as a reference and aligns all time-series according to this reference sequence in each iteration of the clustering algorithm. To align the sequences with the reference sequence, this algorithm uses the SBD measure, which identifies an ideal shift for every $\overrightarrow{x_i} \in P_k$.

3) Shape-based Time-Series Clustering: The third and final component is K-shape clustering, which relies on the SBD distance measure and the shape extraction method to efficiently produce clusters of time series. The K-shape clustering algorithm takes the time series set X and the number of clusters K as inputs. Then, this algorithm assigns the time series in the set X to the K clusters randomly. In the next step, it computes each cluster's centroid using the Shape-Extraction algorithm. Then once the centroids are computed, it refines the memberships of the clusters by using the SBD distance measure. The K-shape clustering algorithm repeats these steps until the algorithm converges or reaches the maximum number of iterations (usually a small number, such as 100). The outputs of this algorithm are the assignment of the time-series to clusters, and the centroids for each cluster which represent the shared characteristics from the underlying data. In addition, each centroid depicts the most representative shape of each group, which is the load profile for each cluster.

IV. Deep-Learning vs Shallow Neural Network Based Load Forecasting

Neural networks have been widely used for electric load forecasting. this approach attempts to learn a complex nonlinear relationship between the inputs and the outputs. Recently, deep learning neural network theory has developed rapidly and achieved improvements over traditional shallow neural networks. Three aspects, namely, better parameter initialization techniques, having a large number of hidden layers and better learning algorithms, have influences the success of deep learning methods [22].

Shallow architectures with large hidden layers can suffer from the over-fitting problem [23], [24]. In contrast, deep architectures with a large number of hidden layers and many levels of non-linearity can represent complex features from their input, and learn these features with models that have better generalization. Consequently, a shallow architecture can

result in a model that does not generalise well, unless trained with very large data sets, thus increasing the computational resources required. Recent research has shown that when the problem has highly nonlinear properties, deep networks can be more efficient to train than shallow networks [22], [25], [26]. In general, deep architectures can include multiple layers of representation and abstraction to help model and generalise from a complex dataset. This enables deep architectures to provide a compact representation for a much wider range of functions than is possible using shallow architectures. A thorough description of shallow and deep architectures is presented in [27], and the interested reader is referred to that paper for further information.

In this paper we use both the feed forward Multilayer Perceptron (MLP) and the Restricted Boltzman Machine (RBM) models for load prediction, and compare their results in terms of prediction accuracy. In the following, we first briefly introduce neural networks, and then introduce the deep learning method based on the Restricted Boltzman Machine (RBM).

A. Artificial Neural Networks

We executed the tests using a feed forward multilayer perceptron architecture with one input layer, two or three hidden layers and one output layer. The input consists of a vector of various features extracted from the load, temperature, and calendar variables. Each hidden layer receives a vector of inputs from the previous layer and converts it to its output vector using a linear transformation followed by a nonlinear activation function. Given a set of training data with corresponding target values $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(T)}, y^{(T)})\}$, for each pair $(x^{(i)}, y^{(i)})$, $x^{(i)}$ is the input to the network and $y^{(i)}$is its label. The output state for each neuron, with N_c input connections, can be calculated as:

$$y = f_a \left(\sum_{j=0}^{N_c} W_i X_i \right), \tag{13}$$

where W_i and X_i are the weight and input vectors respectively, and f_a is the activation function, which is the sigmoid function in this study, $f_a(z) = sigmoid(z) = \dfrac{1}{1+e^{-z}}$.

Let vector $o^{(i)}$ be the corresponding output of the neural network and vector $x^{(i)}$ be the input feature vector. In training the neural network, the output value $o^{(i)}$ should be close to the actual label $y^{(i)}$. The network adjusts its weights to obtain a good estimation of $y^{(i)}$ by minimizing the square error function $E = \dfrac{1}{2}\sum_{t=1}^{T}(o^{(t)} - y^{(t)})^2$. In the optimization process, the resilient backpropagation algorithm [28] is used, which is based on the traditional backpropagation algorithm. In contrast to traditional backpropagation, the resilient method can find better network parameters and accelerate the optimization process. In order to modify the weights of networks for finding a local minimum of the error function, resilient backpropagation allocates a separate learning rate for each weight, which can be changed during the training process.

B. Deep-Learning Based on Restricted Boltzman Machine and Deep Belief Networks

Neural networks with multiple hidden layers can be very sensitive to initialization, which can lead to much better or much worse results after training [29]. To reduce this sensitivity, Hinton et al. introduced a greedy layer-wise pre-training method [27], [30] by a combination of Restricted Boltzman Machines (RBM) to address the drawbacks of traditional neural networks.

A Restricted Boltzmann Machine (RBM) can be defined as a generative stochastic neural network model that learns a probability distribution over the input vectors. The standard type of RBM consists of binary-valued hidden units, visible units, a matrix of weights W_{j_k} corresponding to the link between hidden unit h_j and visible unit v_k, and bias weights a_k and b_j for the visible and hidden units, respectively. In addition, the RBM is a special type of energy based model that can be defined as:

$$\begin{aligned} E(v,h) &= -h^T W_v - a^T v - b^T h \\ &= -\sum_k a_k v_k - \sum_j b_j h_j - \sum_{jk} W_{jk} v_k h_j, \end{aligned} \tag{14}$$

where the joint distribution over the visible and hidden units is defined by $p(v,h) = \dfrac{e^{-E(v,h)}}{Z}$, where Z is a normalization factor.

The visible and hidden units are conditionally independent given one another. Accordingly, the conditional probabilities $p(h \mid v)$ and $p(v \mid h)$ can be written as:

$$\begin{aligned} p(h \mid v) &= \prod_j p(h_j \mid v) \\ p(v \mid h) &= \prod_k p(v_k \mid h). \end{aligned} \tag{15}$$

If the values of v and h are limited to the set $\{0,1\}$, the conditional probabilistic model can be defined as:

$$\begin{aligned} p(h_j = 1 \mid v) &= sigmoid(b_j + \sum_k W_{jk} h_k) \\ p(v_k = 1 \mid h) &= sigmoid(c_k + \sum_j W_{jk} h_j). \end{aligned} \tag{16}$$

DBNs are multi-layer generative models that learn one layer of features at a time from unlabelled data. The extracted features are then treated as the input for training the next layer. This efficient, greedy learning can be followed by fine-tuning the weights to improve the generative or discriminative performance of the whole network. There are a range of advantageous properties that have been identified for DBNs [27]: they can learn higher-level features that yield good classification accuarcy; they are parametric models, whose training time scales linearly with the number of records; and they can use unlabelled data to learn from complex and high-dimensional datasets.

DBNs are multilayer neural networks, where each layer is typically made of RBMs and can be carefully pre-trained in

an unsupervised, layer-by-layer method. Each larger of the hierarchical network is trained in an unsupervised manner, where the output of one layer becomes the input of the next layer in a bottom-up manner. After the unsupervised pre-training. The DBN uses supervised learning to precisely adjust the network. DBN defines a joint probability distribution between the input x and the hidden variables at layer i, h^i, as follows:

$$p(x, h^1, \ldots, h^l) = (\prod_{j}^{l-1} p(h^{i-1} \mid h^i))p(h^{l-1}, h^l), \qquad (17)$$

V. Experimental Results

In this paper, we investigate the best clustering method for grouping the load consumption data that comes from smart meters at the household level, in order to improve the accuracy of load forecasting at the network level. In addition, we study the application of deep learning to improve the performance of load forecasting. Moreover, we introduce a new method for aggregating the forecast load for each cluster to derive a network level load forecast that is more accurate than previous methods [11], [15], [31]. To evaluate the effectiveness of our method, a real-life smart meter dataset was used for testing our approach. The Neural Network and Deep Neural Network based forecasting models with K-shape and K-means clustering methods were applied to the dataset.

For K-shape and K-means clustering, the average consumption for each day of the week during the training period of interest were obtained. In addition, for preprocessing the dataset, the standard score normalization method was applied to normalize the load into the range $[0, 1]$. Then, the data points were arranged in a 3176-by-336 data matrix, where the dimension of the matrix refers to the number of customers and the number of features (48 segments per day for seven days of a typical week). To determine the optimal number of clusters, the MAPE at each granularity of the day for each day of the testing period was calculated with the number of clusters varying from 2 to 8. Then, the average MAPE was calculated during the whole testing period for varying numbers of clusters. Figs. 3, 4 show that the best accuracy for our dataset is obtained when $K = 5$ clusters are used in both clustering methods (K-shape and K-means). Using this value of $K = 5$, to compare the accuracy of K-means with NN, K-shape with NN, K-means with DNN and K-shape with DNN. In Fig. 5, it is shown that K-shape with the DNN based forecasting model has the best accuracy among the four proposed methods.

For both Neural Network and Deep Neural Network methods, the features that were described in the section on feature selection have been used. Tabel I shows the comparative results between the proposed methods. We show the forecasting reults for 24 hours ahead for each of the four methods, and it can be seen that the method using K-shape clustering with DNN has the highest forecasting accuracy for the aggregate load. Based on the forecasting results, we have demonstrated that by using our approach, load prediction accuracy can be improved by effectively clustering the customers based on their consumption pattern, and applying the forecasting model that can better generalize over the dataset. Fig. 6 depicts the average of weekly consumption at each cluster level during 14 months from August 2009 to September 2010. In this figure, it is clear that each cluster has the specific load profile in terms of the shape extraction in the K-shape clustering method.

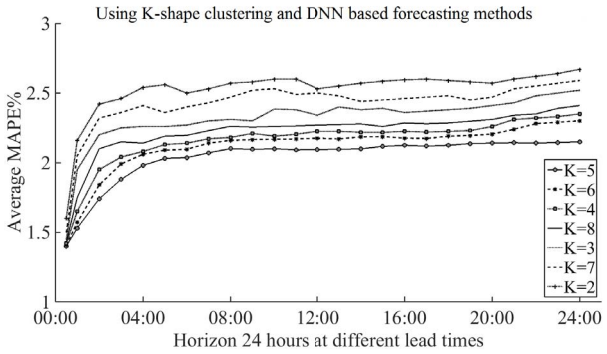

Fig. 3. Average MAPE for one month (November 2010-December 2010) vs lead times of 30 min ahead, 1 h ahead, 2 h ahead, ..., 24 h ahead for different numbers of cluster.

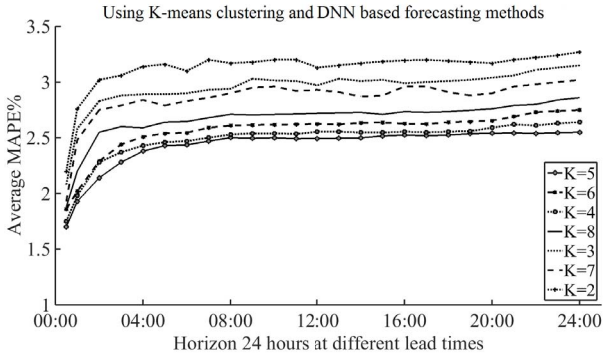

Fig. 4. Average MAPE for one month (November 2010-December 2010) vs lead times of 30 min ahead, 1h ahead, 2h ahead, ..., 24h ahead for different numbers of cluster.

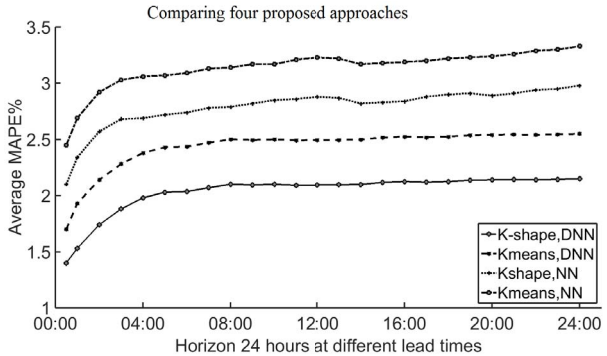

Fig. 5. Average MAPE for one month (November 2010-December 2010) vs lead times of 30 min ahead, 1h ahead, 2h ahead, ..., 24h ahead for different proposed methods.

TABLE I. Mean Absolute Percentage Error for different clustering and forecasting methods for lead time 24hours ahead in November 2010 - December 2010

Methods	Average MAPE%
K-shape clustering+DNN	2.15%
K-means clustering+DNN	2.55%
K-shape clustering+NN	2.98%
K-means clustering+NN	3.33%

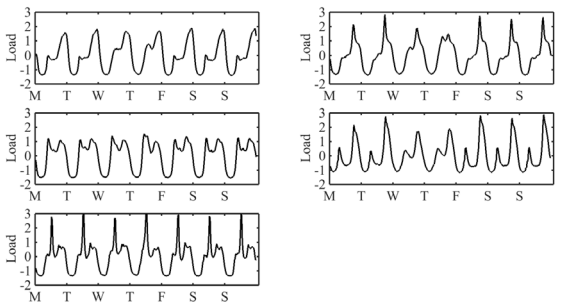

Fig. 6. Normalized load profiles extracted by K-shape clustering when K=5, the optimal number of clusters.

VI. Conclusion

We present a new approach to the problem of short term load forecasting by using deep neural networks, which provide better generalization than traditional neural network models. In addition, we show that by using an appropriate clustering method we can further improve forecasting accuracy. Based on an evaluation using real-life smart meter data, we have shown that the best accuracy can be achieved when K-shape clustering is used in combination with deep neural network methods for forecasting. In future work, we will focus on developing dynamic clustering methods for real-time forecasting.

References

[1] S. Paoletti, M. Casini, A. Giannitrapani, A. Facchini, A. Garulli, and A. Vicino, "Load forecasting for active distribution networks," in *Proceedings of IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, 2011, pp. 1–6.

[2] C. Cecati, C. Citro, and P. Siano, "Combined operations of renewable energy systems and responsive demand in a smart grid," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 4, pp. 468–476, 2011.

[3] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*, 2015.

[4] F. Martínez-Álvarez, A. Troncoso, G. Asencio-Cortés, and J. C. Riquelme, "A survey on data mining techniques applied to electricity-related time series forecasting," *Energies*, vol. 8, no. 11, pp. 13 162–13 193, 2015.

[5] S. Haben, C. Singleton, and P. Grindrod, "Analysis and clustering of residential customers energy behavioral demand using smart meter data," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 136–144, 2016.

[6] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[7] K.-H. Kim, J.-K. Park, K.-J. Hwang, and S.-H. Kim, "Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1534–1539, 1995.

[8] P.-F. Pai and W.-C. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Conversion and Management*, vol. 46, no. 17, pp. 2669–2688, 2005.

[9] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[10] G. E. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[11] A. Shahzadeh, A. Khosravi, and S. Nahavandi, "Improving load forecast accuracy by clustering consumers using smart meter data," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[12] D. Ilić, P. G. da Silva, S. Karnouskos, and M. Jacobi, "Impact assessment of smart meter grouping on the accuracy of forecasting algorithms," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 673–679.

[13] M. Misiti, Y. Misiti, G. Oppenheim, J.-M. Poggi *et al.*, "Optimized clusters for disaggregated electricity load forecasting," *Revstat*, vol. 8, no. 2, pp. 105–124, 2010.

[14] P. Goncalves Da Silva, D. Ilic, and S. Karnouskos, "The impact of smart grid prosumer grouping on forecasting accuracy and its benefits for local electricity market trading," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 402–410, 2014.

[15] F. L. Quilumba, W.-J. Lee, H. Huang, D. Y. Wang, and R. L. Szabados, "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 911–918, 2015.

[16] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1855–1870.

[17] S. Bandyopadhyay, T. Ganu, H. Khadilkar, and V. Arya, "Individual and aggregate electrical load forecasting: One for all and all for one," in *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*, 2015, pp. 121–130.

[18] "Smart metering trial data publication," November 2013. [Online]. Available: http://www.cer.ie/en/information-centre-reportsand-publications.aspx?article=5dd4bce4-ebd8-475e-b78d-da24e4ff7339

[19] I. Drezga and S. Rahman, "Input variable selection for ann-based short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1238–1244, 1998.

[20] N. Ding, Y. Besanger, F. Wurtz, G. Antoine, and P. Deschamps, "Time series method for short-term load forecasting using smart metering in distribution systems," in *Proceedings of the IEEE Trondheim PowerTech*, 2011, pp. 1–6.

[21] M. Chaouch, "Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves," *Smart Grid, IEEE Transactions on*, vol. 5, no. 1, pp. 411–419, 2014.

[22] T. Tieleman, "Training restricted boltzmann machines using approximations to the likelihood gradient," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008, pp. 1064–1071.

[23] V. Pacelli, V. Bevilacqua, M. Azzollini *et al.*, "An artificial neural network model to forecast exchange rates," *Journal of Intelligent Learning Systems and Applications*, vol. 3, no. 02, p. 57, 2011.

[24] G. Panchal, A. Ganatra, Y. Kosta, and D. Panchal, "Behaviour analysis of multilayer perceptronswith multiple hidden neurons and hidden layers," *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, p. 332, 2011.

[25] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.

[26] Y. Bengio and O. Delalleau, "Justifying and generalizing contrastive divergence," *Neural Computation*, vol. 21, no. 6, pp. 1601–1621, 2009.

[27] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[28] F. Günther and S. Fritsch, "Neuralnet: Training of neural networks," *The R Journal*, vol. 2, no. 1, pp. 30–38, 2010.

[29] Y. Bengio, "Deep learning of representations: Looking forward," in *Proceedings of the Statistical Language and Speech Processing*, 2013, pp. 1–37.

[30] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[31] S. Ramos, J. Soares, and Z. Vale, "Short-term load forecasting based on load profiling," in *Proceedings of the IEEE Power and Energy Society General Meeting (PES)*, 2013, pp. 1–5.