IJCNN 2019. International Joint Conference on Neural Networks. Budapest, Hungary. 14-19 July 2019

Robust and Accurate Short-Term Load Forecasting: A Cluster Oriented Ensemble Learning Approach

Fateme Fahiman*, Sarah M.Erfani† and Christopher Leckie‡
School of Computing and Information Systems,
The University of Melbourne, Australia
Email: *ffahiman@student.unimelb.edu.au, †sarah.erfani@unimelb.edu.au, ‡caleckie@unimelb.edu.au

***Abstract*—One of the most critical tasks for operating a power system is load forecasting in order to keep balance between demand and supply and for planning infrastructure. Errors in load forecasting can result in significant cost increases for electricity suppliers and increase the chance of unexpected blackouts or brownouts. Improving the accuracy of short term load forecasting is a challenging open problem. This paper proposes a novel framework for short-term load forecasting using an effective new combination of c-Shape clustering, LSTM networks and Xgboost methods. In particular, our proposed approach introduces an ensemble process together with novel features that lead to improved accuracy of the load forecasting model. The performance of the proposed framework is validated with publicly available real-life data from Australian Energy Market Operator, as well as through on-site deployment, which has led to substantially higher accuracy over existing methods.**

***Index Terms*—Short-term load forecasting, LSTM networks, Xgboost, Clustering, Ensemble learning, Time-series forecasting, Smart grids, Unsupervised learning.**

I. Introduction

Short term load forecasting (STLF), which ranges from 30 minutes to one week ahead, is a challenging task of crucial importance for power system operation due to the variety of factors that influence the load and the volume of data that needs to be considered. STLF plays an important role in the reliability, security, market operation, and scheduling of reasonable dispatching plans for smart power grids. With increasing integration of renewable energy sources and the introduction commencement of different demand response programs in the energy market, the load has become more volatile and less predictable than ever before [1]–[4]. Thus, 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 and a variety of methods have been proposed. These methods can be categorized into traditional statistical methods and Artificial Intelligence (AI) based methods. Traditional methods introduced linear models for load forecasting, such as regression and time series analysis [5]–[7]. These methods are not suitable for non-linear problems like STLF, thus they often have poor prediction accuracy. AI based methods such as neural networks [8]–[10], support vector machines [11], k-nearest neighbors [12], and fuzzy models [13] are able to approximate nonlinear relationships and have been extensively applied to the STLF problem.

Although a wide range of studies have been conducted into load forecasting, producing an accurate STLF is an open research problem due to the variety of factors that influence the load, the volume of data that needs to be considered, and the non-stationary characteristic of the load data.

Recently, Recurrent Neural Network (RNN) models have been recognized as one of the most effective methods in time series forecasting due to their ability to learn functions of arbitrary complexity and capture the time varying dynamics of the underlying system [14], [15]. In this paper we use Long Short Term Memory (LSTM) [16], which is a special type of RNN architecture. LSTM can learn the order dependence between items in a sequence and it has the potential to learn the context required to make predictions in time series forecasting problems, rather than having this context pre-specified and fixed [17], [18]. We use this capability of LSTM to forecast the morning and afternoon peak load for the day-ahead time horizon. We then use these two values as features inputs to a day-ahead forecasting model that generates load forecast for 24 hours ahead at intervals of 30 minutes granularity. We show that these two features have a major influence in improving the day-ahead forecast accuracy. Moreover, in order to produce a robust and accurate forecasting model we have used an unsupervised data mining approach through using c-Shape clustering [19] and ensemble learning. The proposed framework is able to produce multiple historical models for different temporal variations, and achieves a high accuracy of forecast by applying ensemble learning techniques. In addition, it is able to handle the challenge of unbalanced training data sets, which arise due to the frequency of abnormal situations like extreme weather conditions, which are critical events that require highly accurate forecasts.

There are many challenges associated with load forecasting. Load demand depends on different exogenous factors like temperature, humidity, wind speed, seasonal patterns related to human activities and cyclic information like time of year/season/week/day, and type of day (holiday/working day/ weekend). The selection of exogenous factors and modeling the relationship between the selected variables and load data plays a crucial role in load forecasting. In this work we introduce a novel factor, apparent temperature, which has a major impact on the accuracy of the prediction model.

978-1-7281-2009-6/$31.00 ©2019 IEEE paper N-20052.pdf

In summary the novel contributions made in this paper are:

- We introduce a robust and accurate framework based on unsupervised ensemble learning, which is able to produce accurate predictions even in difficult-to-predict cases that arise out of volatility, spikes, and abnormal situations.
- We introduce a way to handle unbalanced training data sets, which leads to high accuracy of forecasting in abnormal situations like extreme weather conditions.
- We address the research question of whether combining weather variables that reflect the human perception of temperature results in more accurate load forecasts than using traditional weather variables.
- We introduce novel features corresponding to, evening and morning peak demand estimated by an LSTM, as key features in improving the accuracy of load forecasting.
- The performance of the proposed method is validated with real data from the power system in the Australian National Electricity Market as well as through on-site implementation by the system operator. It is shown that our predictive framework has a substantially higher accuracy over existing methods in both normal and abnormal situations.
- Our proposed framework is flexible and easy to apply, which makes it an effective model to supplement any operational power system.

A. *Paper Outline*

The rest of the paper is organized as follows: Section II presents the load forecasting problem in a formal manner and the machine learning benchmark model. In Section III we explain our approach for day-ahead demand forecasting by introducing a novel clustering based ensemble learning framework and two new input variables for improving forecast accuracy. In Section IV we show our experimental results based on real-life data. Finally, in Section V we conclude and discuss some possible directions for future work.

II. Background

In this section we first formally define the problem, then we introduce a benchmark machine learning model for load forecasting.

A. *Problem Statement*

Producing an accurate electricity load forecasting result is achievable by effectively combining appropriate predictive features. We first describe how to systematically model the point load forecast in each time step, then we explain how to choose appropriate input variables for the model.

We are given a historical load data set $L_T = \{L_1, \ldots, L_T\}$, comprising T observations. The goal is to predict the H future observations $\{L_{T+1}, \ldots, L_{T+H}\}$. We can predict the electric load at time $t+h$, where $h \in \{1, 2, \ldots, H\}$, by producing different model at each forecast horizon h as follows:

$$L_{t+h} = f_h(X_t) + \epsilon_{t+h}, \tag{1}$$

where ϵ_{t+h} denotes the model error, $f_h(.)$ is the model which is based on the conditional expectation $E[L_{t+h} \mid X_t]$, and X_t is defined as follows:

$$X_t = (l_t, g_{t+h}), \tag{2}$$

where l_t is a vector of lagged demand data occurring prior to time $t+1$, and g_{t+h} is a vector of exogenous variables at time $t+h$ which will be discussed in Section II-B2 in detail.

The performance of the model can be evaluated in terms of the Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100}{T} \sum_{t=1}^{T} \mid \frac{\hat{L_t} - L_t}{L_t} \mid, \tag{3}$$

where $\hat{L_t}$ is the forecat value and L_t is the actual observation at time step t.

B. *Benchmark Model*

In this paper we use the eXtreme Gradient Boosting (XGBoost) [20] algorithm as a benchmark machine learning model. XGBoost has been used widely in many data mining and machine learning challenges and gives state-of-the-art results on a wide range of problems [20]. Moreover, there are a range of typical input variables that have been used in the literature for the electricity load forecasting problem during last three decades, which we use with this benchmark model.

1) eXtreme Gradient Boosting: A Scalable Tree Boosting System: XGBoost [20] is an improved implementation of gradient boosted decision trees [21] designed for speed and performance. XGBoost is able to create boosted trees in a well organized way, operates in parallel, and avoid overfitting by using regularization technique. It is called gradient boosting because it uses a gradient descent algorithm to optimize the value of the objective function. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models, and then combined to make the final prediction. An important attribute of XGBoost is its scalability in all scenarios, where parallel and distributed computing can make this algorithm run more than ten times faster than existing popular solutions. In addition, It is able to handle sparse data and instance weights in approximate tree learning.

XGBoost is a supervised learning algorithm where the training data x_i (with multiple features) is used to predict the target variable y_i. The prediction model is defined as follows:

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in F, \tag{4}$$

where K is the number of trees, f is a regression tree that maps the attributes to the scores, and F is the set of all possible classification and regression trees (CART). Then, the objective function can be written as:

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k), \tag{5}$$

where $l(y_i, \hat{y_i})$ is the training loss function which measures how well the model fits the training data, and $\Omega(f_k)$ is a regularization function that measures the complexity of the model. By incorporating these two components in the objective function, XGBoost is able to optimize the training loss function to encourage simple models that tend to have smaller variance and hence more stable predictions.

Since the functions f_k are trees, XGBoost uses an additive training algorithm to find the structure and the leaf scores of the trees. It starts from constant prediction and adds a new function each time. Then, the objective function is defined as follows:

$$obj^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t), \tag{6}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$, $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$, and $\hat{y}^{(t-1)}$ is the forecast value at time step $t-1$.

The model complexity is defined based on the number of leaves of a tree and the L_2 norm of scores on the leaves. Consider regression tree $f_t(x) = w_{q(x)}, w \in R^T, q : R^d \to \{1, 2, \ldots, T\}$, where W is the vector of scores on leaves, q is a function assigning each data point to the corresponding leaf, and T is the number of leaves. In XGBoost the complexity is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2. \tag{7}$$

2) Input Variables: Load forecasting accuracy is influenced by a variety of factors. While new machine learning methods can fit highly non-linear models, the selection of input variables (features) is also very important and has a major impact on the accuracy of load prediction. The features that are utilized in the benchmark model can be categorized as follows:

Load: Electric load consumption usually exhibits a daily periodic pattern. In addition, load consumption profiles in adjacent days and weeks exhibit strong positive correlations. According to these characteristics we define three variables: load at time t for the previous two days, and load at time t on the same day of the previous week.

Weather: Weather is a major driving factor of electrical load consumption patterns. Temperature is the most frequently used weather variable in the load forecasting literature. In addition, the influence of other weather variables such as wind speed and relative humidity on the accuracy of load forecasting has been recently reported in [22], [23]. Thus, we consider the wind speed and relative humidity as well as temperature in our benchmark model. To incorporate the recency effect [24] of temperature features we define seven variables as follows: the temperature at time t today and the same time of the previous day, maximum and minimum temperatures of the previous day, and average temperature of the past 3 hours, 6 hours, and 24 hours.

Time Cyclic: 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 [25]. By spectral analysis, the half day, day, week and year periods have been identified as the dominant frequencies of the load data [26]. For each frequency, a pair of variables is considered to represent the corresponding cycles:

$$\begin{aligned} c_1(t) &= \sin(\frac{2\pi t}{T}) \\ c_2(t) &= \cos(\frac{2\pi t}{T}), \end{aligned} \tag{8}$$

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 to 17520, 336, 48 and 24 for a year, a week, a day and a half-day cycle, respectively.

Type of Day: Load analysis has shown that load consumption on non-working days is lower than on working days due to the decrease in industrial loads on non-working days. We consider nine variables for the a type of day as follows: "Tuesday, Wednesday, Thursday", "Saturday", "Sunday", "Friday", "Monday", "Holiday", "Day before holiday", "Day after holiday", "Special day" which corresponds to special days which are not public holidays like the first week of January and school holidays, etc.

III. Methodology

In this section, first we introduce our proposed novel input variables to improve the accuracy of load forecasting. Then we introduce our robust and accurate short term load forecasting framework in detail.

A. Apparent Temperature - A Novel Feature for Improving Load Forecast Accuracy

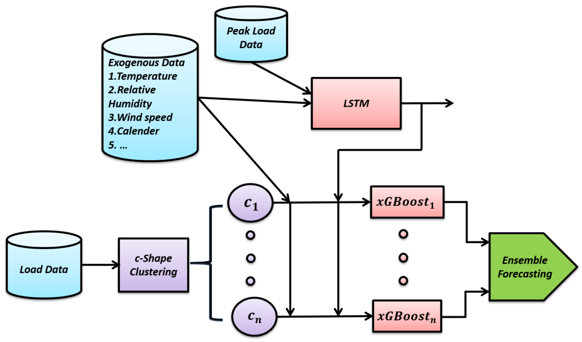
The temperature that we feel determines the amount of energy that we use for cooling or heating. The air temperature is usually used as a measure of how comfortable we feel when we want to use heating or cooling appliances. However, the air temperature is only one of the factors that has an impact on the assessment of thermal stress. Where other factors, principally humidity and wind speed, can vary widely from day to day, we need to consider the effect of all factors to assess the level of comfort realistically.

Apparent Temperature (AT) is a useful index which condenses all the factors of perceived temperature into a single value. In fact, it is the temperature equivalent perceived by humans caused by the combined effects of air temperature, relative humidity and wind speed [27]. In this paper we adopt the AT formula based on the definition introduced by Robert et al. in [28] as follows:

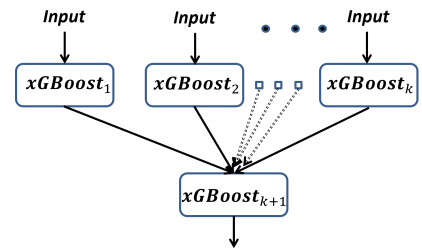
$$AT = Ta + 0.33e - 0.7ws - 4, \tag{9}$$

where Ta is the temperature ($^\circ$C), ws is the wind speed (m/s), and e is water vapour pressure (hPa) which can be calculated using the following formula:

$$e = \frac{6.105rh}{100} \exp(\frac{17.27Ta}{237.7 + Ta}), \tag{10}$$



((a)) Framework



((b)) Horizontal ensemble learning

Fig. 1. (a) The cluster oriented ensemble learning framework for robust and accurate day ahead load forecasting. (b) Horizontal ensemble learning system based on XGBoost predictors.

where rh is the relative humidity ($\%$).

After defining AT we can use this value to define two other weather dependent features, Heating Degree Days (HDD) and Cooling Degree Days (CDD). HDD is an indicator to quantify the amount of energy required to heat a building, while the CDD is a measure designed to indicate how much energy is needed to cool a building. Both HDD and CDD are defined relative to a based comfort temperature. We define the HDD and CDD based on AT as follows:

$$HDD = max(0, (mean(AT_t, AT_{t-1}, \ldots, AT_{t-6}) - 16.5)) \tag{11}$$

$$CDD = max(0, (18 - mean(AT_t, AT_{t-1}, \ldots, AT_{t-6}))) \tag{12}$$

B. Our Proposed Framework

In this section we first introduce our robust and accurate short term load forecasting framework, then we explain each part of the framework in detail. Figure 1(a) shows the block diagram of the proposed framework.

We propose a cluster oriented ensemble learning framework to handle the different temporal variations and unbalanced aspects of the training load data set. The proposed frame work produces multiple historical models for different temporal variations, and uses ensemble learning techniques to forecast the day-ahead load with high accuracy for both extreme weather conditions as well as normal days.

As shown in Figure 1(a), the data set for load forecasting consists of historic load data, exogenous data, and peak load data. We partition the each of these data sets into three parts: training, validation, and test data sets. The training data is used to train the models, validation is used for tuning the parameters for each model, and the test data is used to evaluate the accuracy of the proposed framework.

First we cluster the load data (training part) based on the daily load demand pattern by using the fuzzy c-Shape clustering algorithm [19]. Fuzzy c-Shape clustering is a very fast and accurate clustering model that is able to group time series data based on their shape. Then we produce the estimated morning and evening peak load for the target day by using an LSTM model. The input variables to the LSTM algorithm are the weather features, type of day, and peak load information. In the next step we produce the forecast for the target day by using the information that is related to each cluster. As a result, we produce n different forecasts for the target day. In the final stage we use the ensemble learning algorithm to make the final forecast. Next we describe each part of the framework in detail.

1) Fuzzy c-Shape Clustering: Numerous clustering algorithms have been used for the load prediction problem [29]–[31]. Determining the best algorithm depends heavily on the nature, purpose and mining objectives of the dataset. In this paper, we use fuzzy c-Shape [19], a novel algorithm for time-series clustering that can conserve the shapes of time-series sequences. It uses a normalized, domain-independent form of cross correlation as its distance measure. Using this method, the c-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, c-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 c-Shape clustering has been experimentally evaluated against popular methods like k-Shape [32], K-means, K-medoid, hierarchical and spectral clustering methods, with combinations of the most competitive distance measures [19]. It has been shown [19] that c-Shape outperforms all of these approaches in terms of accuracy on time series data. The fuzzy c-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. For a fuller explanation of the c-Shape clustering algorithm, interested readers are referred to reference [19].

2) Long Short Term Memory Networks: A challenging type of predictive modeling problem is time series prediction. Unlike regression predictive modeling, time series also adds the complexity of a sequence dependence among the input variables. Recurrent Neural Networks (RNN) [33] are a powerful type of neural network, which is sequence based model and designed to learn the temporal correlations between previous information and the current conditions. It is trained by Backpropagation Through Time (BPTT) [34], however, RNN is faced with the vanishing gradient problem [35], [36] which limits its ability to learn long-term temporal correlations. To address this problem, Hochreiter et al. [37] introduced the Long Short Term Memory (LSTM) architecture by including a memory cell, and further improvement was presented by Gers et al. [38] with an extra forget gate. LSTM has been the most successful RNN architecture for creating large recurrent networks. Thus, it has been widely adopted to address difficult sequence problems in machine learning, and achieves state of the art outcomes.

To briefly explain the concept of LSTM, we adopt a similar description introduced by Lipton et al. in [39]. LSTM networks consist of memory blocks, instead of neurons, which are connected in layers. A block contains a memory cell for establishing temporal connections, gates for managing the state of the block, and an output. Consider the input sequence $\{x_1, x_2, \ldots, x_T\}$, where $x_t \in \Re^k$ is a k dimensional vector at time step t. LSTM defines an internal memory cell state and keeps it during the whole life cycle in order to establish the temporal connections. The memory cell state s_t captures the interaction with the intermediate output h_{t-1} and the subsequent input x_t. It uses the information of the outputs of the previous time step and the inputs of the present time step to update, maintain or erase the elements of the internal state vector. LSTM has three types of gates, (1) forget gate f_t: which conditionally decides about the information that should be thrown away from the block; (2) input gate i_t: which conditionally decides about the input values which should be used for updating the memory state; and (3) output gate o_t: which conditionally decides about the output values based on the inputs and the block's memory. The formulation of all mentioned nodes are as follows:

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \tag{13}$$

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \tag{14}$$

$$g_t = \phi(W_{gx}x_t + W_{gh}h_{t-1} + b_g), \tag{15}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o), \tag{16}$$

$$s_t = g_t \odot i_t + s_{t-1} \odot f_t, \tag{17}$$

$$h_t = \phi(s_t) \odot o_t, \tag{18}$$

where W_{gx}, W_{gh}, W_{ix}, W_{ih}, W_{fx}, W_{fh}, W_{ox}, and w_{oh} are weight matrices for the corresponding inputs of the network activation functions. $\odot$, σ, ϕ represent element wise multiplication, sigmoid activation function and tanh function, respectively. In Figure 2 the LSTM block and the sequential architecture of an unrolled LSTM are shown.

For training a simple one-layer LSTM recurrent network, the hyperparameter of the hidden output, s_t and h_t should be initialized. Usually these parameters are initialized to zero, i.e., $h_0 = 0$ and $s_0 = 0$. The three sigmoid functions in the LSTM block work as soft switches to decide which signals are allowed to pass the gates. These decisions for the forget f, input i, and output o gates are made based on the information of the current input x_t and the previous output h_{t-1}. The input gate determines the information that should be preserved in the internal state. The forget gate controls the information that should be forgotten from the previuos state s_{t-1}. Finally, when the internal state is updated, the output gate determines the information that should be passed as the LSTM output h_t. These steps are repeated in each time step to learn the weights and biases based on minimization of the differences between the LSTM outputs and the actual training samples.

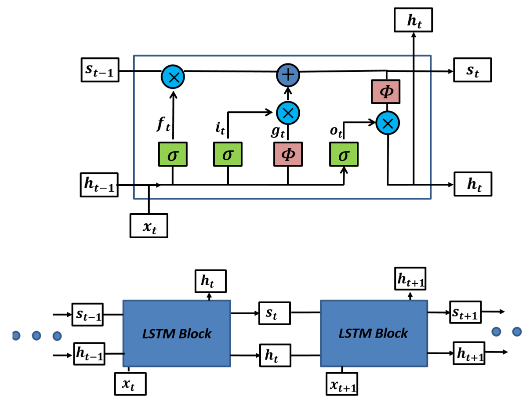


Fig. 2. The top figure shows the architecture of an LSTM block and the bottom one shows the unrolled LSTM sequential architecture

3) Ensemble Learning: Ensemble systems consist of an integration of two or more learning methods. The main goal is to integrate the capabilities of individual learning methods into a single comprehensive system. There are two types of ensemble learning systems - horizontal and vertical [40]. Vertical ensemble learning systems are systematized in a vertical order such that the output of one learning method is used as an additional input to the next learning method. In contrast, horizontal ensemble learning systems are organized in a horizontal order such that the results produced by each method are combined to produce a single response to a given input. Horizontal ensemble learning systems integrate the predictive capabilities of the underlying individual forecasts. Therefore, horizontal ensembles are insensitive to errors caused by a single method and are hence more appropriate for critical applications such short term load forecasting.

In this paper we apply the horizontal ensemble learning approach as shown in Figure 1(b). In the first step the c-Shape clustering is trained to group the electricity demand curves. Since c-Shape clustering is a shape based clustering algorithm, similar or related demand curves fall into the same clusters. In the next step, one predictor is trained for each cluster. The proposed system is able to significantly improve the accuracy of prediction since it no longer relies on a single predictor, the data set is grouped into subsets that share the similar patterns, and each predictor is an expert in handling a type of energy market condition.

IV. CASE STUDIES

In this section we show the comparison results for the three introduced models in this paper: (1) the benchmark model introduced in Section II-B, called the "Benchmark" model; (2) the benchmark model with the addition of the new proposed features in Section III-A based on Apparent Temperature, HDD and CDD, called the "BenchmarkPlus" model; (3) our proposed framework in Section III-B, called the "Framework" model. We evaluate the accuracy of the forecasts of the proposed framework, especially for difficult-to-predict days such as holidays and very hot days (which corresponds to extreme load conditions).

A. Data

The demand data is provided by the Australian Energy Market Operator. The data is aggregated smart meter data from all over five regions in Australia: Victoria, South Australia, New South Wales, Tasmania, and Queensland. The granularity of data is 30 minutes and the history of 5 years data $(2014-01-01$ to $2018-12-28)$ have been used. Three years of data have been used for training, one year for validating and parameter tuning, and one year for testing the accuracy of different models presented in this paper. The weather data is provided by the Bureau of Meteorology (BoM) [41] in Australia with a granularity of 30 minutes.

B. Experiments

Table I shows the comparison results in terms of mean absolute percentage error during the test period $(2018-01-01$ to $2018-12-28)$ for the three different methods and five regions in Australia. As shown, our proposed framework improved the accuracy of day-ahead forecasts over each region by 1%. Based on the research by Hong et al. [42], every 1% reduction in mean percentage absolute error leads to savings of around $\$300,000$ per year for a utility with $1GW$ peak. Note that the aggregate peak load in these five region is $33GW$ So, any small improvement in the accuracy of load forecasting leads to substantial savings for utility companies.

Figure 3 shows the monthly comparison analysis of the day-ahead forecast accuracy among the three proposed methods during the test period in Victoria. As shown, the accuracy of the BenchmarkPlus model is slightly better than the Benchmark model, which means that by using apparent temperature instead of temperature the accuracy of the model is improved. Also, it is clear that the Framework model is more accurate than the two other models, especially during the summer periods in Victoria (January, February, March, November, December), when the difficult to predict days such as new year eve, school holidays and high temperature conditions are occur. Figure 4 shows how the Framework model produces a stable prediction in the first two weeks of 2018, which contains the difficult to predict days which arising from new year and high temperature days, which impact on the consumption of the energy by customers. As shown, during this period of high demand days $(2018-01-05$, $2018-01-06$, $2018-01-11$, $2018-01-12)$ the proposed framework has accurately predicted the peak. The

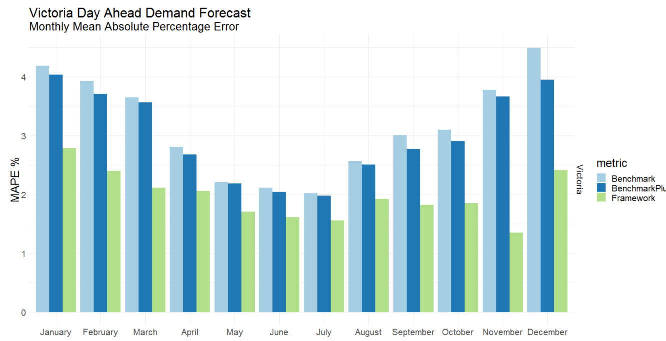


Fig. 3. Monthly analysis of day-ahead forecast accuracy in terms of MAPE during the test period 2018-01-01 to 2018-12-28, in Victoria

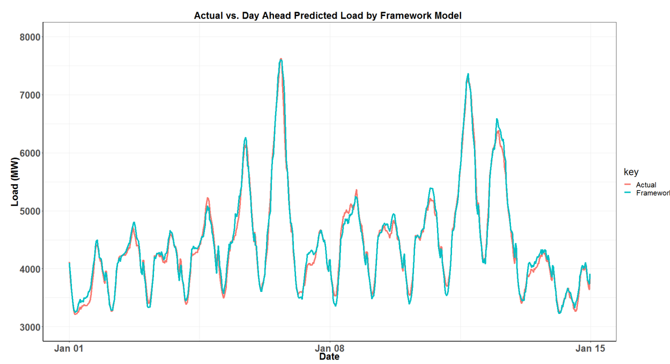


Fig. 4. Actual vs. day-ahead forecast by Framework model for period 2018-01-01 to 2018-01-14, in Victoria

accuracy of the proposed models are also analyzed in the extreme weather as well as normal weather conditions. The extreme weather conditions are defined based on two standard deviations from the mean temperature as follows:

$$Weather = \begin{Bmatrix} Extreme & Temperature > \mu + 2\sigma \quad or \\ & Temperature < \mu - 2\sigma, \\ Normal & Otherwise \end{Bmatrix}.$$

where μ and σ are the average and standard deviation of the temperature during the test period, respectively. Figures 5 and 6 show the box plot of extreme and normal weather conditions in the test period in Victoria for the three proposed models. In this study the temperatures above 28°C and below 3°C are considered as extreme weather situations in Victoria, where the average of temperature during the test period is 15°C and the standard deviation is 6°C. As shown in Figures 5 and 6, the Framework model has lower absolute error than the Benchmark and Benchmarkplus models in both normal and extreme weather situations, which demonstrates the robustness of the proposed framework.

TABLE I. MAPE value for day ahead forecast during test period ("2018-01-01" to "2018-12-28") for each five states in Australia

Method	States				
	Victoria	Tasmania	South Australia	New South Wales	Queensland
Benchmark	3.2%	3.5%	4.5%	3.7%	3%
BenchmarkPlus	2.9%	3.1%	4%	3.4%	2.8%
Framework	**1.9%**	**2.2%**	**3.3%**	**2.5%**	**2%**

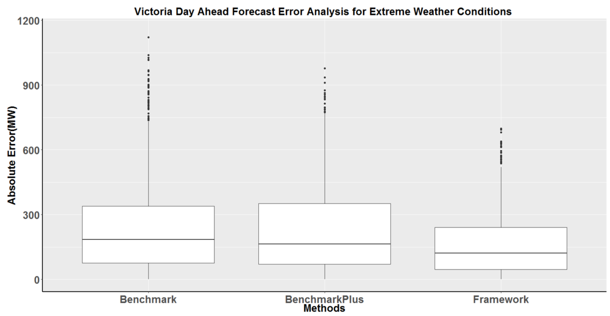


Fig. 5. Day-ahead forecast Absolute Error box plot during the extreme weather conditions in the test period 2018-01-01 to 2018-12-28, in Victoria

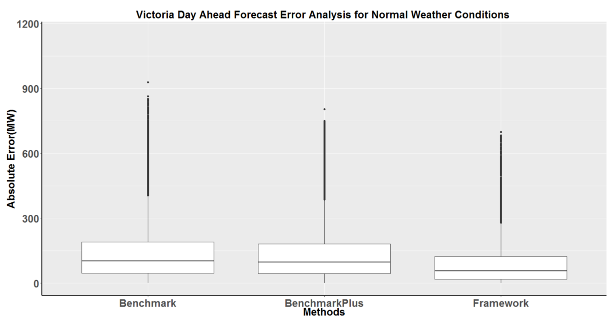


Fig. 6. Day-ahead forecast Absolute Error box plot during the normal weather conditions in the test period 2018-01-01 to 2018-12-28, in Victoria

V. Conclusion

In this work we introduced a novel framework for improving day-ahead load forecasting in the electricity grid. We investigated how adding new features to the load forecasting system can result in improvement of forecasting accuracy. In addition, because of the nonstationary nature of the demand data, we introduced an unsupervised ensemble learning approach, which can achieve high accuracy in situations that have not seen before by the forecasting system. We tested our proposed model for the most difficult prediction times in five regions in Australia. The results demonstrate the robustness and accuracy of our model even for these unusual and extreme load conditions.

References

[1] J. M. Morales, A. J. Conejo, H. Madsen, P. Pinson, and M. Zugno, *Integrating renewables in electricity markets: operational problems*. Springer Science & Business Media, 2013, vol. 205.

[2] A. Motamedi, H. Zareipour, and W. D. Rosehart, "Electricity price and demand forecasting in smart grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 664–674, 2012.

[3] P. Pinson, "Wind energy: Forecasting challenges for its operational management," *Statistical Science*, pp. 564–585, 2013.

[4] J. Nowotarski, B. Liu, R. Weron, and T. Hong, "Improving short term load forecast accuracy via combining sister forecasts," *Energy*, vol. 98, pp. 40–49, 2016.

[5] J. Nowicka-Zagrajek and R. Weron, "Modeling electricity loads in california: Arma models with hyperbolic noise," *Signal Processing*, vol. 82, no. 12, pp. 1903–1915, 2002.

[6] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 2015.

[7] 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.

[8] 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.

[9] D.-x. Niu, H.-f. Shi, and D. D. Wu, "Short-term load forecasting using bayesian neural networks learned by hybrid monte carlo algorithm," *Applied Soft Computing*, vol. 12, no. 6, pp. 1822–1827, 2012.

[10] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 44–55, 2001.

[11] 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.

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

[13] L. Suganthi, S. Iniyan, and A. A. Samuel, "Applications of fuzzy logic in renewable energy systems–a review," *Renewable and Sustainable Energy Reviews*, vol. 48, pp. 585–607, 2015.

[14] A. M. Schaefer and H.-G. Zimmermann, "Recurrent neural networks are universal approximators," *International Journal of Neural Systems*, vol. 17, no. 4, pp. 253–263, 2007.

[15] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] F. A. Gers and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.

[18] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, no. Aug, pp. 115–143, 2002.

[19] F. Fahiman, J. C. Bezdek, S. M. Erfani, M. Palaniswami, and C. Leckie, "Fuzzy c-shape: A new algorithm for clustering finite time series waveforms," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–8.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[21] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[22] J. Xie, Y. Chen, T. Hong, and T. D. Laing, "Relative humidity for load forecasting models," *IEEE Transactions on Smart Grid*, 2016.

[23] J. Xie and T. Hong, "Wind speed for load forecasting models," *Sustainability*, vol. 9, no. 5, p. 795, 2017.

[24] P. Wang, B. Liu, and T. Hong, "Electric load forecasting with recency effect: A big data approach," *International Journal of Forecasting*, vol. 32, no. 3, pp. 585–597, 2016.

[25] 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.

[26] 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.

[27] R. G. Steadman, "A universal scale of apparent temperature," *Journal of Climate and Applied Meteorology*, vol. 23, no. 12, pp. 1674–1687, 1984.

[28] ——, "Norms of apparent temperature in australia," *Australian Meteorological Magazine*, vol. 43, no. 1, pp. 1–16, 1994.

[29] P. G. 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.

[30] 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.

[31] F. Fahiman, S. M. Erfani, S. Rajasegarar, M. Palaniswami, and C. Leckie, "Improving load forecasting based on deep learning and k-shape clustering," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 4134–4141.

[32] 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*. ACM, 2015, pp. 1855–1870.

[33] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.

[34] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[35] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[36] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.(2001)," *Cited on*, p. 114, 2001.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," 1999.

[39] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[40] R. Avnimelech and N. Intrator, "Boosted mixture of experts: an ensemble learning scheme," *Neural Computation*, vol. 11, no. 2, pp. 483–497, 1999.

[41] Australian Bureau of Meteorology Weather Stations. http://www.bom.gov.au/climate/cdo/about/sites.shtml.

[42] T. Hong, "Crystal ball lessons in predictive analytics," *EnergyBiz Mag*, vol. 12, no. 2, pp. 35–37, 2015.