Received January 23, 2021, accepted January 31, 2021, date of publication March 2, 2021, date of current version March 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063650

Past Vector Similarity for Short Term Electrical Load Forecasting at the Individual Household Level

HARIS MANSOOR[1], HUZAIFA RAUF[1], (Student Member, IEEE), MUHAMMAD MUBASHAR[2], MUHAMMAD KHALID[3,4], (Senior Member, IEEE), AND NAVEED ARSHAD[2], (Associate Member, IEEE)

[1]Department of Electrical Engineering, Lahore University of Management Science (LUMS), Lahore 54792, Pakistan
[2]Department of Computer Science, Lahore University of Management Science (LUMS), Lahore 54792, Pakistan
[3]Electrical Engineering Department, King Fahd University of Petroleum & Minerals (KFUPM), Dhahran 31261, Saudi Arabia
[4]K. A. CARE Energy Research and Innovation Center, Dhahran 31261, Saudi Arabia

Corresponding author: Naveed Arshad (naveedarshad@lums.edu.pk)

This work was supported from the LUMS Energy Institute, LUMS.

ABSTRACT Demand side management (DSM) programs are an integral part of the modern grid. Most of these DSM programs are designed to work at household and hour level. The optimization problems in these DSM programs are guided by the forecasted load. An error in the hour ahead load forecasting results in a suboptimal solution entailing economic cost to both the utility and the customers. Predicting loads at a fine granularity (e.g., households) is challenging due to a large number of (known or unknown) factors affecting power consumption. At larger scales (e.g., clusters of consumers), since the inherent stochasticity and fluctuations are averaged out, the problem becomes substantially easier. Many techniques have been proposed to predict loads of clusters of consumers in various localities with great accuracy. Also, these techniques generally utilize sociological and weather information and work better on data from a particular locality. In this paper, a new technique called Past Vector Similarity (PVS) has been proposed to predict electricity load one hour ahead at the level of an individual household. The proposed strategy is based on load information only and does not require calendar, weather or any other attributes. In fact, the idea is to extract the exact load patterns of individual households corresponding to their routine and socio-economic values. Consequently, the technique makes use of the recent past vector and generate similar patterns for the prediction of future load profiles. Furthermore, the ensemble of these similar loads is an efficient prediction of electricity. PVS has just two parameters due to which it can be applied to the smaller dataset without overfitting issue. Moreover, due to the parallel nature of PVS, it can be scaled for a large number of customers without computation burden. The proposed PVS has been assessed empirically for two distinct datasets from Australia and Sweden. The simulation results demonstrate that the PVS significantly outperforms other state-of-the-art forecasting methods in terms of accuracy.

INDEX TERMS Short term load forecasting (STLF), household load forecasting, past vector similarity (PVS), data transformation, hour ahead load forecasting, long short-term memory (LSTM), random forest (RF).

NOMENCLATURE

ADF	Augmented Dickey-Fuller test
AI	Artificial Intelligence
AIC	Akaike Information Criterion
AMI	Advanced Metering Infrastructure
ARIMA	AutoRegressive Integrated Moving Average
CLSTM	Cycle Long Short-Term Memory
CNN	Convolution Neural Network
DBSCAN	Density Based Spatial Clustering Of Applications With Noise
DNN	Deep Neural Network

The associate editor coordinating the review of this manuscript and approving it for publication was S. Ali Arefifar.

kNN	k-Nearest Neighbours
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MRC	Multi Resolution Clustering
MSE	Mean Square Error
NN	Neural Network
NRMSE	Normalized Root Mean Square Error
PVS	Past Vector Similarity
RBF	Radial Basis Function
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SARIMA	Seasonal AutoRegressive Integrated Moving Average
STLF	Short Term Load Forecasting
TDCNN	Time Dependency Convolution Neural Network
j	Customer Index
T	Hour Index
A	Training Dataset
B	Testing Dataset
d_1	Length of Training Dataset
d_2	Length of Testing Dataset
k	Length of Past Vector in PVS
L_p	p_{th} Vector Norm
L_{inf}	Infinity Vector Norm
m	Averaging Parameter of PVS
n_1	Number of Customers in Dataset
n_2	Number of Hours in Dataset
P_T^j	Prediction of T_{th} Hour for j_{th} Customer
q	Data Transformation Parameter
R	Set of Real Numbers
X	Whole Dataset
$Xtest_T^j$	Feature Vector of T_{th} Hour for j_{th} Customer for Testing Dataset
$Xtrain_T^j$	Feature Vector of T_{th} Hour for j_{th} Customer for Training Dataset
$Ytest_T^j$	Label of T_{th} Hour for j_{th} Customer for Testing Dataset
$Ytrain_T^j$	Label of T_{th} Hour for j_{th} Customer for Training Dataset

I. INTRODUCTION

A fundamental task in power planning is matching electricity supply with demand. Accurate demand forecasting is crucial to ensure efficient management in the power sector. Both overestimating or underestimating the demand entail huge economic costs. A load forecast error of 1% translates to several hundred thousand dollars per GWh [1].Long Term Load Forecasting is needed for power system infrastructure planning. However, operational decisions for smart grids have to be made within a short time and require Short Term Load Forecasting (STLF) of a few hours to days [2]. Most of the renewable energy resources are highly variable and intermittent in nature. Therefore, to effectively integrate these resources in the grid, an hour ahead load forecasts and at finer scales for one or a few households are required. Accurate fine-scale load forecasts thus result in optimal resource allocations and local energy generation, thereby reducing costs associated with transmission and distribution [3], [4]. Short-term load forecasting at household level is also pivotal for design of demand side management programs (DSM) such as demand response programs (DR) [5], peak shaving [6]–[8], dynamic pricing [9], [10] and soft load-shedding [11], [12]. Most of these DSM programs are designed to work at each hour. The optimization problems in these DSM programs are guided by the data from household level hourly load forecasting. An error in the hour ahead load forecasting results in a suboptimal solution entailing economic cost to both the utility and the customers.

There has been significant progress towards short-term load forecasting using statistical and machine learning techniques [13]. However, Reducing the error of load forecasting at the household level is still an open problem [14]. The household level forecast is also more challenging due to the interplay of many different factors. Most of these factors are unknown or very complex to quantify and evaluate such as demographics of occupants and their daily routine [13]. There is also inherent stochasticity in an individual load consumption pattern. Since machine learning techniques depend upon the information available due to this reason a lot of machine learning techniques do not perform well on this problem. Some studies use both past year's load data and results of sociological surveys [15], [16]. Such information, is not readily available, thus limiting the applicability of the method to a particular locality. Recently, deep learning techniques are producing prominent results on this problem [17]–[19]. However, a major limitation of deep learning techniques is the availability of a large amount of data to train the model. To overcome these challenges some studies have used clustering and household aggregation for short-term load forecast [20]. Aggregating the loads smooth out the inherent variability as shown in Figure 1, which makes forecasting relatively simpler [13], [21]. However, these methods do not work at the finer spatial granularity of the household level.

In this study, we propose a novel approach for the hour ahead electricity forecasting at a household level called Past Vectors Similarity (PVS). PVS is based on two assumptions. The first assumption is that the electricity load itself gives information about user patterns such as the number and demographics of occupants, their daily schedules, and socio-economic data. This assumption is based on the observation that a house with a different number of occupants and daily routine has a different load pattern. Thus, the past load (past vector) of an hour is an effective feature vector. The second assumption states that if the past of a load is similar to another load, then the future will also be similar. This assumption is analogous to Markov property [22]. In this way, the feature vector of every hour is created as a past vector. To predict the load of an hour, its past vector is compared with all past

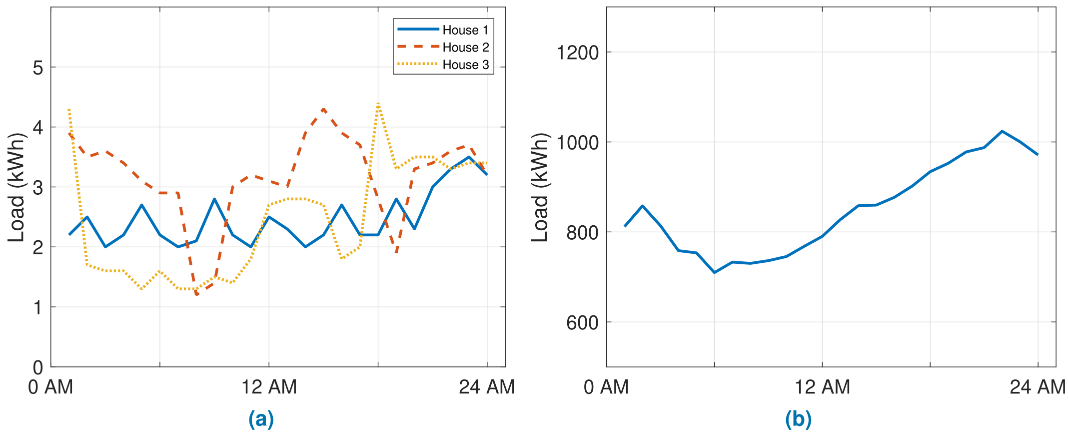


FIGURE 1. (a) The hourly load of three randomly selected houses for a single day (Thursday 1_{st} January 2004) in Sweden Dataset. (b) Total combined hourly load of all customers for a single day (Thursday 1_{st} January 2004).

vectors to evaluate similar vectors. Since these vectors are similar to each other, they are representing a unique routine of the customer. According to Markov property, the customer routine in a future hour will also be similar. Thus the next hour values of similar past vectors are a potential forecast of the hourly household load. Averaging as an ensemble measure of similar vectors is considered as the final forecast. The averaging operation is important as it removes the stochastic nature of daily routine and demographics, and gives a robust prediction. A graphical example involving the averaging operation of the proposed technique is shown in Figure 2. The grey lines are representing similar past vectors, while the blue line is the original vector. The blue dot is the hour ahead original load, whereas the red dot is the ensemble average predicted load.

There are many advantages of our proposed PVS technique. PVS consists of two parameters, which assist in its application to a relatively less amount of data without overfitting. Other deep learning techniques, require a large amount of data to learn [18]. PVS does not require weather, sociological or other data and works with just load consumption data. Due to the parallel nature of PVS, it can be scaled better for large number of houses. The performance evaluation of the proposed technique is carried out by using two widely available benchmark datasets, which comprise of hourly consumption data from households in Eskilstuna, Sweden [23] and New South Wales, Australia [13]. Three error measures mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE) are used to validate our proposed technique. The results demonstrate that PVS substantially outperforms other techniques. To the best of our knowledge, this technique has not been used earlier in the literature for the prediction of load demand of individual household customers.

The rest of the paper is organized in five different sections as: The related work is presented in Section II. In Section III, the methodology of the proposed PVS technique for the hour ahead STLF at the household level is presented. Section IV describes the datasets and experimental setup in detail. Experimental results are reported in Section V followed by the conclusion in Section VI.

II. RELATED WORK

The existing work in the domain of STLF can be divided into three categories namely (i) STLF at system/subsystem level in which aggregated load of all households at city/country level is forecasted, (ii) STLF for the cluster of customers in which customers are intelligently divided into clusters/communities and load for those clusters is forecasted, and (iii) STLF for individual customers in which load is forecasted separately for each customer.

A. STLF AT SYSTEM/SUBSYSTEM LEVEL

Short term load forecasting at a system or subsystem level is well explored in the literature. In [24], a radial basis function (RBF) neural network (NN) is proposed for STLF for households, which are grouped based on location, nature, and size of loads. In [25], k-nearest neighbors (KNN) based algorithm is used to forecast day-ahead loads of groups of consumers. Another framework using wavelet transform and Bayesian neural network for STLF at the system level is proposed in [26]. A time-series method using intra-day and intra-week seasonal cycles is proposed in [27] for forecasting country loads few minutes ahead. In [28], stochastic characteristics of electric consumption in France are utilized to predict short term aggregated load. A kernel-based support vector regression (SVR) combination model for the STLF at the system level is proposed in [29]. Several authors proposed hybrid methods involving data preprocessing along with effective use of classification/regression algorithms for STLF at system level [30], [31]. As we have explained earlier aggregation causes the load to smooth out at system level Figure 1, making forecasting easier. These methodologies are not applicable at the household level due to the highly stochastic nature of the household load.

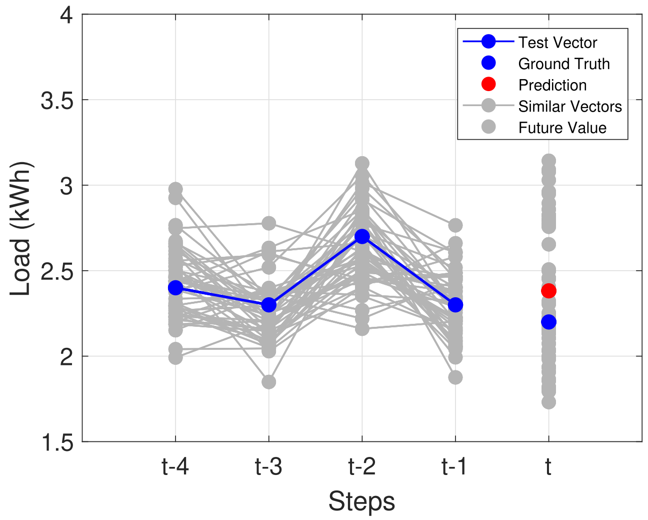


FIGURE 2. PVS example with past vectors, original and predicted load.

B. STLF FOR CLUSTER OF CUSTOMERS

A wide variety of methods utilize the increasingly available AMI data from individual consumers to enhance STLF at the system level. The general idea is to cluster similar consumers into groups, to predict the total loads of each group, and aggregate the predictions to obtain the total load forecast. For cluster loads prediction machine learning techniques such as *Random Forest* (RF), *Neural Networks* (NN) and deep learning are used. Clustering is accomplished based on similarities in the load profiles (consumers' AMI readings) [32] and consumers demographic information [20]. Practice theory of human behavior is incorporated for improved clustering [33], resulting in accuracy boost for day-ahead system level load forecast. A deep neural network (DNN) based model for STLF at the individual, as well as subsystem level, is proposed in [34], which learns complicated relations between weather variables, dates, and previous consumptions for individual customers. A hybrid approach consists of convolutional neural network (CNN) and k-means clustering algorithm is proposed in [35], which is used to forecast hourly load of clusters of customers. In another study, two deep learning methods, time-dependency convolutional neural network (TD-CNN), and cycle-based long short-term memory (C-LSTM) network are proposed to improve the forecasting performance of short-term load forecasting [36]. Li *et al.* in [37] evaluated a multi-resolution clustering (MRC) method to forecast half hourly load of the cluster of customers. The relationship between group/cluster size and forecast accuracy is studied in [38] using two forecasting methods namely Holt-Winters and Seasonal Naive. A major drawback of Deep learning methods is that they require a large amount of data to learn.

C. STLF FOR INDIVIDUAL CUSTOMERS

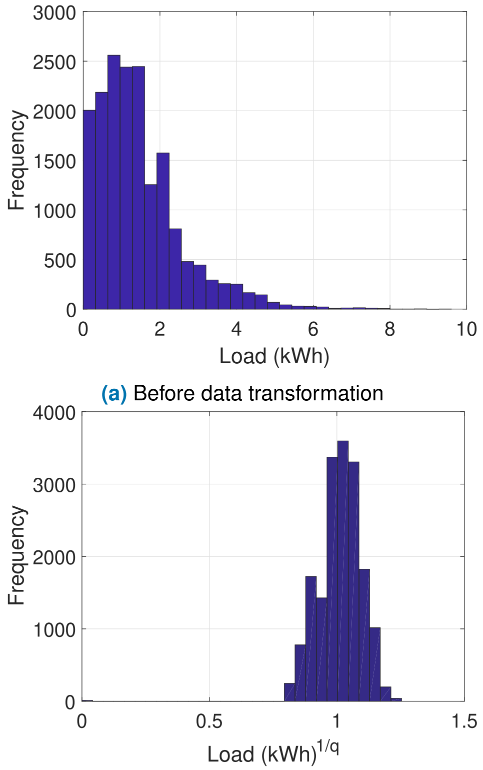
STLF at individual consumers level is significantly more challenging due to high volatility and variability in load profiles [39]. Methods for STLF at the household level can be divided into time series based or artificial intelligence-based methods. Time series based methods are classical techniques that mostly originates due to statistical problem formulation. Most of the classical methods treat data as a stationary time series. These simplistic assumptions are unable to capture the complex nonlinear behavior between electricity consumption and periodic routines of household residents [15]. Andreas *et al.* in [40] used the autoregressive integrated moving average (ARIMA), neural networks (NN), and exponential smothering for the household forecast. They showed that these methods hardly beat persistent forecasts in terms of mean absolute percentage error (MAPE). Mahmoud *et al.* in [41] utilized the Kalman filter and reported MAPE. Many statistical methods like linear regression, stochastic and time series based methods are also given in [27], [28], [42]–[45].

Recently artificial intelligence (AI) based methods are dominating short-term load forecasting. This domination is due to their less mathematical complexity and better results. In [18], pooling based deep recurrent neural network (RNN) are used to predict household load. Pooling based Recurrent neural network shows better results as compared to the autoregressive integrated moving average (ARIMA), support vector regression(SVR), and simple RNN. The error metrics used are the root mean square error (RMSE), normalized root mean square error (NRMSE), and mean absolute error (MAE). Mean absolute percentage error (MAPE) which is a more suitable error metric for household level load forecast is not given. Similarly, in [17] Weicong *et al.* used LSTM along with density-based spiral clustering of application with noise (DBSCAN) for household load forecast. This technique shows better results in terms of mean absolute percentage error (MAPE). Elena Mocanu in [46] used a factored conditional restricted Boltzmann machine for load forecasting of residential homes and showed improvement as compared to simple support vector machine (SVM) and neural network (NN). Clustering household load based methods are proposed in [20], [47]. These methods try to predict hourly aggregated load for each cluster rather than each household separately. This makes their techniques achieve better results. However, our goal is to predict individual household level hourly load with minimum possible error.

III. PROPOSED METHODOLOGY

In this section, the methodology of the proposed technique is presented, i.e., PVS for STLF at household level. PVS takes as input the load matrix with rows and columns corresponding to n_1 hours and n_2 consumers, respectively, i.e., $X \in \mathbb{R}^{n_1 \times n_2}$. The entry $X(T, j)$ is the electricity consumption of consumer j at hour T. PVS performs the following steps:

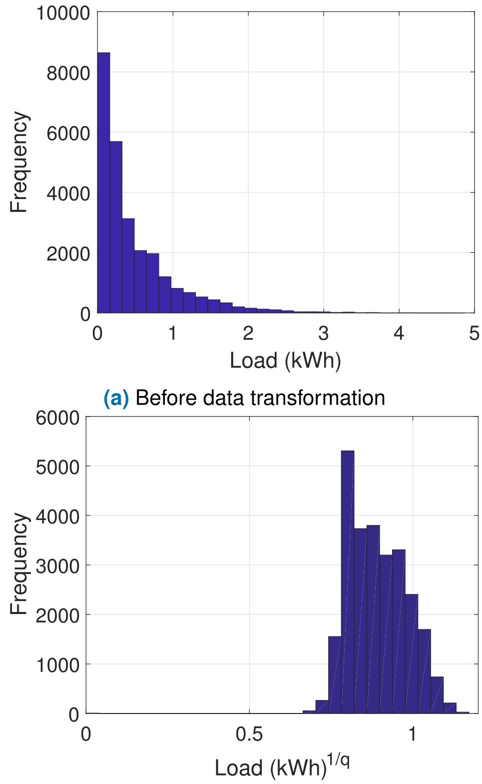
- It first preprocesses the data to improve its statistical properties.
- It then represents features of each timestamp by the load values of the past hours.
- Lastly, to predict a load, the past vector is compared to find similar past vectors.



(a) Before data transformation

(b) After data transformation

FIGURE 3. **Households loads histograms of Australia dataset for houshold number 25 (a) before data transformation and (b) after data transformation.**



(a) Before data transformation

(b) After data transformation

FIGURE 4. **Households loads histograms of Sweden dataset for household number 79 (a) and (b).**

The final forecast is an ensemble (average) forecast of the future load of similar past vectors. Each step of the methodology is discussed in detail as follow:

A. DATA PREPROCESSING

The load values at household level in X, comprise of very low magnitude values. Hence, there is a significant skewness in the data. This causes a huge problem in the load prediction. Since predicting zero or closer to zero loads is a complex task. Thus, the standard q^{th} root transformation is performed X as a preprocessing step i.e. every value $X(T,j)$ is replaced with $X(T,j)^{\frac{1}{q}}$. The skewness effect of transformation is depicted in Figures 3 and 4 which show the load histogram of the household 79 and 25 of the Sweden and Australia dataset respectively. Figures 3a and 4a are the load histograms before transformation whereas Figures 3b and 4b are the load histograms after the transformation. It is observed that a large number of values are very close to 0 before transformation and the transformed data is more normally distributed. We apply the corresponding reverse transformation after forecasting the load and report our predictions and their errors in the original units and scales.

B. PAST VECTOR SIMILARITY

In this section, the proposed PVS technique is discussed in detail. The prepossessed data is first divided into training data A of d_1 hours and testing data B of d_2 hours, such that $d_1 + d_2 = n_1$. The training and testing data is converted into feature vectors. The feature vector of each timestamp of each consumer consists of the following two parts:

(i) Future electricity load part, which is the load of the next timestamp.
(ii) A past electricity load part called the past vector.

The past vector is a k-length vector which contains load consumption for the k previous hours. The past feature vector of T_{th} comprised of $[T\ T-1\ T-2 \cdots T-k]$ load. A feature vector is the load of $[T\ T-1\ T-2 \cdots T-k]$, where $[T+1]$ is the label or future value to be predicted, and k is the parameter of the PVS model.

$$Xtrain^{j}_{(T)} = A[T\ \ T-1 \cdots T-k, j] \quad T \in d_1,\ j \in n_1 \tag{1}$$
$$Ytrain^{j}_{(T)} = A[T+1, j] \quad T \in d_1,\ j \in n_1 \tag{2}$$
$$Xtest^{j}_{(T)} = B[T\ T-1\ \cdots\ T-k, j] \quad T \in d_2,\ j \in n_1 \tag{3}$$
$$Ytest^{j}_{(T)} = B[T+1, j] \quad T \in d_2,\ j \in n_1 \tag{4}$$

Hence, there are roughly d_1 and d_2 total vectors for training and testing set respectively for each customer. All vectors of the training set of a customer are collected and are called the pool of training past vectors $\{(Xtrain^{j}_{1}, Ytrain^{j}_{1}), \ldots, (Xtrain^{j}_{d_1}, Ytrain^{j}_{d_1})\}$. Similarly, all vectors of the testing set of a customer are collected and are termed as the pool of testing past vectors

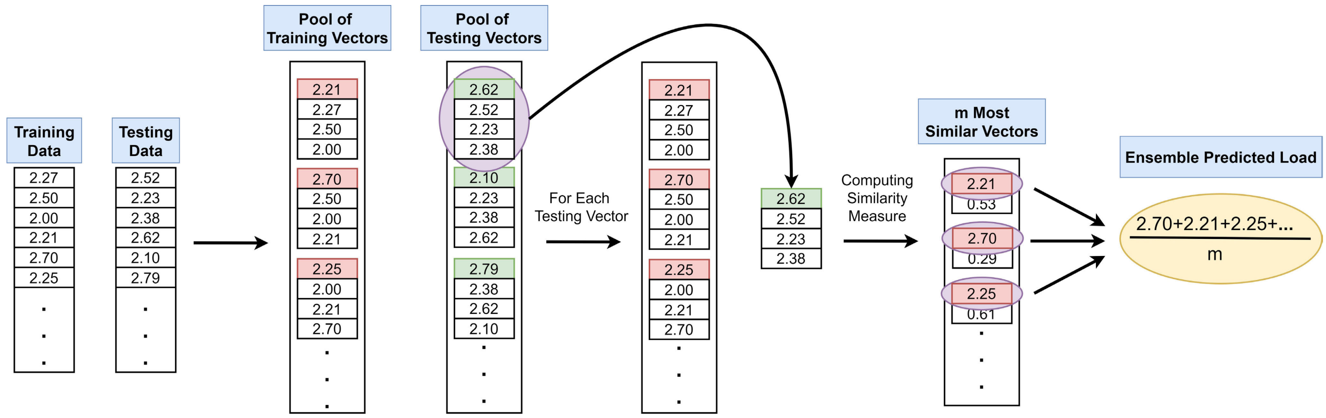


FIGURE 5. Flow diagram of PVS technique.

$\{(Xtest_1^j, Ytest_1^j), \ldots, (Xtest_{d_1}^j, Ytest_{d_1}^j)\}$. The whole process is shown in Figure 5.

Now, consider a distance metric $\|\cdot\|$ on R^k. For a test point $Xtest_T^j$, let

$\{(Xtrain_1^j, Ytrain_1^j), \ldots, (Xtrain_{d_1}^j, Ytrain_{d_1}^j)\}$ be a reordering of the training data such that

$$\|Xtrain_1^j - Xtest_T^j\| \leq \cdots \leq \|Xtrain_{d_1}^j - Xtest_T^j\| \quad (5)$$

To predict the load of a time stamp T_{th}, its past vector is considered and its similarity is computed with the pool of training past vectors, i.e., equation 5. The m most similar past vectors are now taken into account, each one of these vector has a similar past in terms of electricity load and their future value is a potential forecast for load. An average forecast is taken into account for the m most similar vectors.

$$P_T^j = \frac{1}{m}\sum_{i=1}^{m} Ytrain_i^j \quad T \in d_2,\ j \in n_1 \quad (6)$$

An example of the past vector for $k = 3$ is shown in Figure 5. The highlighted first values are the load to be predicted (labels), whereas the remaining vector is the past load vector. The red highlighted load values are known for the training data. For testing data, the green highlighted values are the loads to be predicted. The process is shown for a single household. It is to be noted here that for a different customer its own load profile is used to create the pool of past vectors. The data is transformed before the whole process and inverse transformed before the final prediction.

To compute the similarity between vectors, different similarity measures are used and their errors are reported. The value of K and m are the parameters of the model which are tuned on the training set. The PVS model is simple enough to optimize these parameters on the training set without overfitting issue. Due to the distributed nature of the algorithm, it can be easily parallelized for any number of customers and implemented on a simple computer without memory overflow issues.

The complete process is shown in Algorithm 1. The algorithm works in a distributed way predicting one hour load of a single customer at a time. First, the data transformation is performed to reduce skewness, and then the past vectors are created from the dataset. The first 12 months (8760 hours) of past vectors are converted into training vector pool and the next 6 months (4380 hours) to the testing vector pool. Each vector is taken from the testing pool and its similarity with the all vectors in the training vector pool is found using the euclidean norm. The mean of m most similar hours is used as a prediction, while the inverse transform is performed to produce the final prediction. The process is repeated for each customer for each hour in the testing pool.

IV. DATASETS AND EXPERIMENTS

In this section, datasets and the experimental setup is described, which is used to validate the PVS technique. The error measures and other STLF techniques are also presented, which are further used for comparison purpose.

A. DATASET AND PERFORMANCE MEASURES

Statistics of these datasets are given in Table 1. The measure of central tendency is shown by the mean load. Australian dataset has 33 customers with the mean load of 0.79 kWh, whereas Sweden dataset has 194 customers with the mean load of 2.52 kWh. This mean is the average of all hours overall customers. The measure of dispersion is represented by the standard deviation. Sweden dataset is more disperse than the Australian dataset. The skewness represents the symmetry of distribution around the mean. As we can observe the Australian dataset is more skewed than Sweden dataset. The measure of tailedness is shown by kurtosis, which gives information about the outliers in the distribution. The Australian dataset has a larger kurtosis as compared to Sweden dataset Table 1.

The average weekday and weekend load of both datasets is shown in Figure 7. This curve is computed by taking the average of all weekdays of all customers for the whole dataset and similarly weekends for all customers of both datasets. We can observe a change in the pattern of electricity consumption. During the weekend the load curve is delayed

Algorithm 1 Past Vector Similarity

Input: Training matrix A, Testing matrix B
Output: Predicted load Matrix P

```
A = A^(1/q)   // training data transformation
B = B^(1/q)   // testing data transformation
P = [ ]
for i ∈ 1:n   // n = number of customers
do
    Data_Pool_train = [ ]
    Data_Pool_test = [ ]
    y = [ ]
    counter = 1
    for j ∈ k + 1:d1   // k = 3 (Sweden),4 (Australia). d1 = number of training hours
    do
        Data_Pool_train(j) = A[j-1, j-2··· j-k]
        // create training past vectors
        y(counter) = A[j]
        counter ++
    for j ∈ 1:d2   // d2 = number of testing hours
    do
        Data_Pool_test(j) = B[j-1, j-2 ··· j-k]
        // create testing past vectors
    Sim = [ ]
    for j ∈ 1:d2 do
        for w ∈ 1:d1 − k do
            g = Data_Pool_test(j)
            h = Data_Pool_train(w)
            Sim(j) = SIMILARITY(g, h)   // Finding similarity
            [index val] = SORT(Sim(j))
            // descending order
            P(i, j) = MEAN(y[index[1 : m]])
P = P^q   // Inverse Transform
```

TABLE 1. Statistics of datasets: Only the first 12864 hours data is used for both datasets.

Dataset	Houses	Hours	Mean Load	Std. Dev	Skew	Kurtosis
Sweden	582	17544	2.52	0.81	1.4	6.07
Australia	34	26304	0.79	0.29	2.9	16.24

which is because people tend to wake up late during the weekend. A similar pattern of delay can be observed for both datasets.

In each dataset, there are customers with missing values. Since the Sweden dataset has a large number of customers, all the data points with missing entries are removed. This leads to 194 customers with no missing values. Since there are only 34 customers in Australia dataset, we just remove one customer (number 19) with more than 1000 missing entries. For the remaining customers in Australia dataset, the missing values are replaced by the mean of the immediate future and past hour load of that customer.

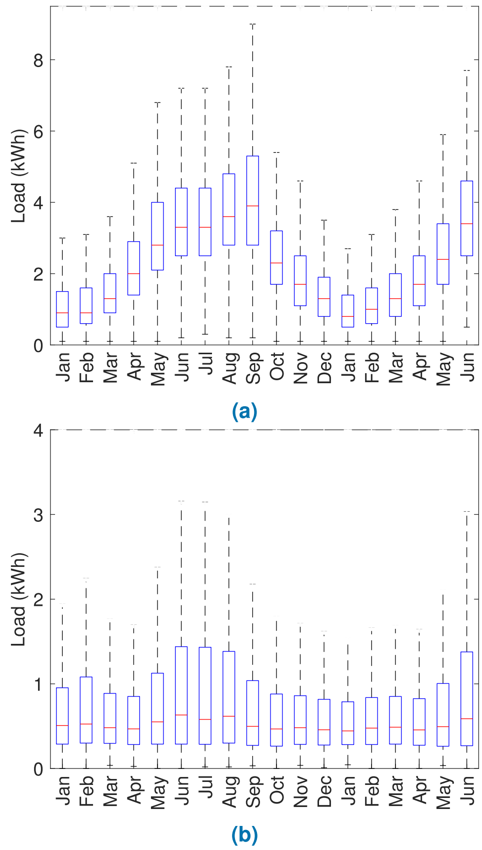


FIGURE 6. (a) Sweden and (b) Australia; dataset boxplot from January-2004 to June-2005 and January-2011 to June-2012.

The reason to select two datasets for performance evaluation is to validate the general performance of *PVS*. The statistics of a dataset depend upon the weather, demographic and economic properties of the area. That is why the statistics of both datasets are quite different. Australia has summer in December-February, where average temperatures range between 20 C° to 35 C°. In winter June-August, the average temperature range is 3 C° to 20 C°. Sweden has summer in June-August, with an average temperature range between 10 C° to 18 C°. During winter November-March the average temperature varies from 0 C° to −15 C°. The vast difference in the statistics of these datasets are helpful in generalizing the performance of our technique.

The boxplot of both datasets is also shown in Figure 6. It can be noted that Sweden customers have a much higher median load in almost every month as compared to Australian customers. The first 12 months are used for training purpose while the last 6 are used for testing purpose. September shows the highest median load for Sweden while June shows the highest median load for the Australia dataset. It is a well-known fact that predicting smaller loads is a very difficult

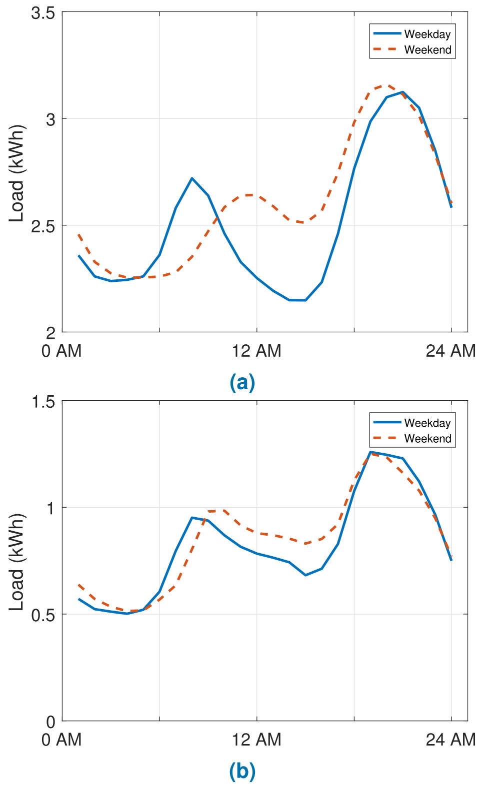


FIGURE 7. (a) Sweden and (b) Australia; average load of weekdays and weekend for all customers.

task due to high variations. Thus, we can expect a large error for the Australia dataset as compared to Sweden dataset.

Different studies in the literature report different evaluation approaches. Three of them are reported to be able to compare different methods with PVS. The error metrics are Mean Absolute Error (MAE) [18], Mean Absolute Percentage Error (MAPE) [17] and Root Mean Square Error (RMSE) [13]. These error metrics are used to depict different aspects of a technique. MAPE is a relative error measure and is very sensitive to low load profiles. Since the hourly household load is quite small in magnitude, MAPE is an ideal error measure for the STLF task and is quite difficult to minimize. MAE on the other hand provides an absolute difference measure of forecasting error. RMSE acts as a standard deviation of overall prediction error. In other words, it tells you how concentrated the prediction points are closer to the original load.

B. EXPERIMENTAL SETUP

In this section, different experiments are performed to show the effectiveness of the PVS technique. Different similarity measures are used, related to euclidean and non-euclidean measures. Since the PVS technique has just three parameters, these parameters can be learned on the training set without overfitting issues. As mentioned earlier, the first 12 months of data is used for training and last 6 months of data for testing purpose. Python is used as a coding environment, and experiments are performed on a core i7 system. Except for parameters search, which is performed on free computation platform, Google Colab. The details of each experimental setup are given below.

There are three parameters of the PVS technique (k, m, q), which are optimized through the search on the training set. The value of k, m, and q are learned using the search on the training set. It is assumed that k, m and q are all positive integers. Of course, q can take a fraction value, but for the sake of simplicity, it is assumed that q is also a positive integer. The values of q and k are supposed to be from set $\{1, 2, 3 \ldots 10\}$ and $\{1, 2, 3 \ldots 20\}$ respectively. Since m can take any value up to the length of training set d_1, to computationally make search feasible, m from a set of $\{2, 4, 6 \ldots 100\}$ is considered. For similarity measure, 15 different measures are used. In this setup, the grid search requires fitting 150000 models, which is computationally infeasible. To make parameter tuning computationally reasonable, one parameter at a time is searched through. In this way, the size of searchable models reduces to 95. The parameters tuning to find the optimal k, m and q for Australia and Sweden datasets is shown in Figure 8 and 9, respectively. First, $k = 5$, $q = 5$, Similarity=Euclidean Norm is selected and different values of m from the set $\{2, 4, 6 \ldots 100\}$ are tested. This leads to the selection of an optimal m as shown in Figures 8a and 9a. On the second stage, optimal $m = 24$, $q = 5$, Similarity=Euclidean Norm is selected and searched for optimal k as shown in Figures 8b and 9b, which corresponds to Australia and Sweden dataset, respectively. At the third stage, optimal $m = 24$, $k = 4$, Similarity = Euclidean Norm is fixed and different q values are tested as shown in Figures 8c and 9c. At the fourth stage using searched optimal parameters, different similarity measures are tested which is shown in Figures 8d and 9d for the corresponding Australia and Sweden datasets. A single model is trained for all customers using MAPE on the training set to tune these parameters. As observed from the Figures 8d and 9d, there is an optimal value for m and k, whereas the error reduces as we increases q. However, q is not tested beyond 10, since increase in q decreases the rate of improvement as shown in Figures 8c and 9c.

Since the PVS technique is based on finding similarity between vectors, different similarity measures like L_1, L_2, L_3, L_4, L_5, L_7, L_{10}, L_{inf}, Average Distance, Mean Character Difference, Canberra Metric, Coefficient of Divergence, Mahalanobis distance, Cosine Similarity, Index of Association and Pearson Coefficient are tested to compute the similarity between past vectors. As it approaches towards infinity norm, the emphasis is shifted towards larger values. As the norm becomes smaller, the more the vectors become similar. The effect of these similarity measures on *MAPE* for Australia and Sweden is shown in Figures 8 and 9, respectively. There is no significant effect of any specific similarity measure. As the value of L_P norm is increased, the error is also increased. This is because increasing p moves emphasis to the single largest element of the vector while ignoring other elements. Fortunately, the L_2 norm which is chosen for parameter tuning performs the best. The tuned parameters are shown in Table 2.

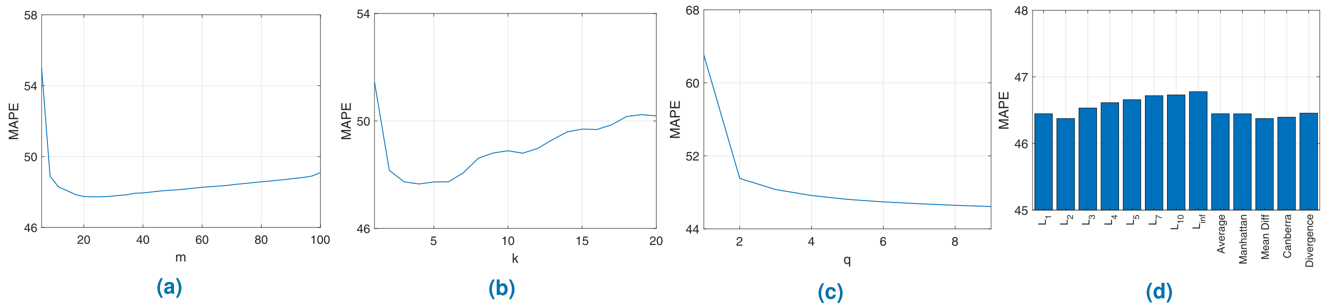


FIGURE 8. Parameters tuning of Australian dataset (a) Parameter 'm' (b) Parameter 'k' (c) Parameter 'q' (d) Different norms.

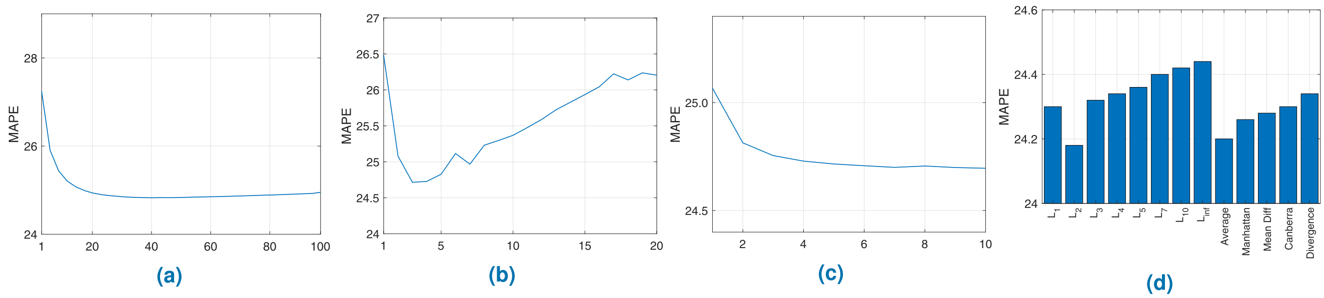


FIGURE 9. Parameters tuning of Sweden dataset (a) Parameter 'm' (b) Parameter 'k' (c) Parameter 'q' (d) Different norms.

TABLE 2. Optimal parameters of PVS technique.

Dataset	m	k	q	Similarity
Australia	24	4	10	Euclidean
Sweden	42	3	10	Euclidean

C. COMPARISON WITH OTHER STLF TECHNIQUES

To compare the PVS technique with baseline and existing classification algorithms such as Persistence, ARIMA, LSTM, and Random forest(RF) algorithms are used. First, the technical details of these models are described, then the results are presented. In all of these techniques, a separate model is learned for each household except for LSTM, as due several parameters and features used in learning separate LSTM model results in overfitting and poor performance. PVS technique does not require a calendar, weather, or any demographic information. However, since weather and calendar data is easily available, calendar and weather attributes are used to train LSTM and RF. Demographics or any other information is not used for training purposes. Definitely, the accuracy can be improved by using more information. But such kind of information is not easily available and is practically difficult to process.

1) PERSISTENCE FORECAST

Persistence forecast is used as a baseline in most of the forecasting problems. In the persistence forecast, the load value of the previous hour is used as a forecast for the next hour. As we can see persistence forecast has no parameters to learn.

2) AUTO-REGRESSIVE INTEGRATED MOVING AVERAGE

A popular and widely used statistical method for time series forecasting is the Auto-Regressive Integrated Moving Average ARIMA which consists of p, q, and d parts. p shows the Auto-Regressive part of ARIMA which indicates that the variable of interest depends upon its past values. q part indicates that the error of forecast depends upon previous time instances errors, and d indicates the integration part, which is used to make time-series stationary.

Since each dataset has a large number of households, fitting individual model on each household is computationally very expensive. However, each household represents an individual time series, and a separate model must be learned for each time series. To solve this problem, '*auto.arima*' function is used which takes maximum p, minimum p, maximum q, minimum q, maximum d, minimum d as parameters and returns the best ARIMA model fitted according to *Akaike Information Criterion (AIC)*. Here, the selection of hyper-parameters of *auto.arima* is described for Australia and Sweden dataset, which makes it possible to fit a different model on each household.

The first step is to check if the time series is stationary or not. This is done by performing *Augmented Dickey-Fuller test (ADF)*. The null hypothesis of the ADF test is that the time series is non-stationary. Therefore, if the p-value is less than the significance level (e.g., 0.05) then the series is stationary, otherwise, it's non-stationary. ADF test is performed on all households separately. Since all the p-values are below 0.05, and $d = 0$, so the value of p and q is selected, based on partial autocorrelation (PACF) and auto-correlation test (ACF), respectively. It is also observed for the ACF test that there is seasonality in the timer series after every 24 hour. Based on the results computed from ACF and PACF tests, the minimum $p = 1$, maximum $p = 5$ is selected. Seasonal ARIMA (SARIMA) is fitted, which is more suitable due to capturing seasonal information. SARIMA consists of parts (p,q,d) and (P,Q,D,S). The value of S is taken as 24 because of the seasonality effect. Maximum P and maximum Q values are taken as 2, and the maximum D value is taken as 0.

3) LONG SHORT-TERM MEMORY

Long Short-term memory (LSTM) is a deep learning architecture. LSTM is a sequential model, which captures the temporal correlation between the previous and the current time step. LSTM contains memory and forgets gates, which are used to deal with vanishing gradient problem [48]. The decision made by LSTM at time step T depends upon the decision made at time step $T - 1$. Such characteristics of LSTM are perfect for individual household load forecasting problem. Since the load at the time, T depends upon the routine of the household at previous time instances.

For LSTM, previous studies suggest that the performance of the network is relatively insensitive to any combination of some layer and layer size [49]. Previously, LSTM based household short term load forecasting is proposed by Weicong *et al.* in [17]. Therefore, the same architecture is re-implemented with similar hyper-parameters as described in [17]. The architecture consists of 2 hidden layers with 20 hidden nodes in each layer. The number of previous time steps used in the prediction is 6 as given in [17]. In the end, a simple neural network with a sigmoid activation function is utilized to ensemble the final prediction.

The features used in LSTM are shown in Table 3. Since LSTM is sensitive to magnitude, the load consumption matrix and features (temperature, wind speed, and humidity) are normalized by min-max normalization. Mean Square Error (MSE) with Adam optimizer [50] is used to carry out training for 100 epochs. Since LSTM contains a large number of weights, learning these weights requires a large quantity of data. We tested a separate LSTM model on individual households and a single model on all households. Due to less amount of data, a separate model on individual households performs poorly as compared to a single model on all households. Therefore, only the results of a single model on each dataset are included.

4) RANDOM FORESTS

Random forest is a regression technique that works by combining multiple decision trees. It takes a subset of variables to build the decision trees. These multiple decision trees are then used to produce an ensemble output. In the case of regression, the output is taken as the mean of all trees. Two of the most important hyper-parameters on which the performance of random forest depends are the maximum features used in the individual tree and the total number of trees. Both of these hyperparameters are tuned using the validation set. As stated earlier, the train-validation-test split, in this case, is 9-3-6 month respectively. It is established that the error decreases as the number of trees are increased. This can also be confirmed from Figure 10b, in which mean square error (MSE) is shown with the number of trees. Due to computational constraints, 100 number of trees are selected. Maximum features used in each decision tree are 29. This feature number is selected using a grid search on the validation set, as shown in Figure 10a. The error metric used to find this feature number is MSE. Since the random forest is invariant to scaling, features of Table 3 are used without scaling. A separate model is learned for each household in each dataset, and this model is tuned with its separate hyperparameters.

TABLE 3. Input parameters for the classification algorithms.

Input Size	Variable	Description
1-24	Hours	One hot encoding vector representing hours
25-31	Day of Week	One hot encoding representing day of week
32-62	Day of Month	One hot encoding representing date
63-74	Month	One hot encoding representing month
75-85	Lagged Input	3 previous hours of same day, 4 hours of previous day including hour to be predicted, 4 hours of same day of previous week including hour to be predicted
86	Public Holiday	Boolean input representing Public holiday
87	Temperature	Single numeric input representing temperature
88	Wind Speed	Single numeric input representing wind speed
89	Humidity	Single numeric input representing humidity

V. RESULTS AND DISCUSSION

In this section, results of PVS are reported for hourly load prediction and performance comparison is carried out with the earlier mentioned *STLF* methods.

Tables 4 and 5 demonstrate the overall performance measures of persistence forecast, ARIMA, RF, LSTM and our proposed PVS technique. The error shown is averaged over all customers and testing hours of 6 months. As stated earlier, the hourly household load is quite small in magnitude and MAPE is a relative measure of error that is sensitive to small numbers, so even a slight prediction error results in a higher percentage error. This is the reason higher MAPE is observed for all techniques. However, both RMSE and MAE are quite small which shows that the predicted loads are closer to the actual load.

It is observed that there is a significant difference in the results of both datasets. From Tables 4 and 5, it is observed

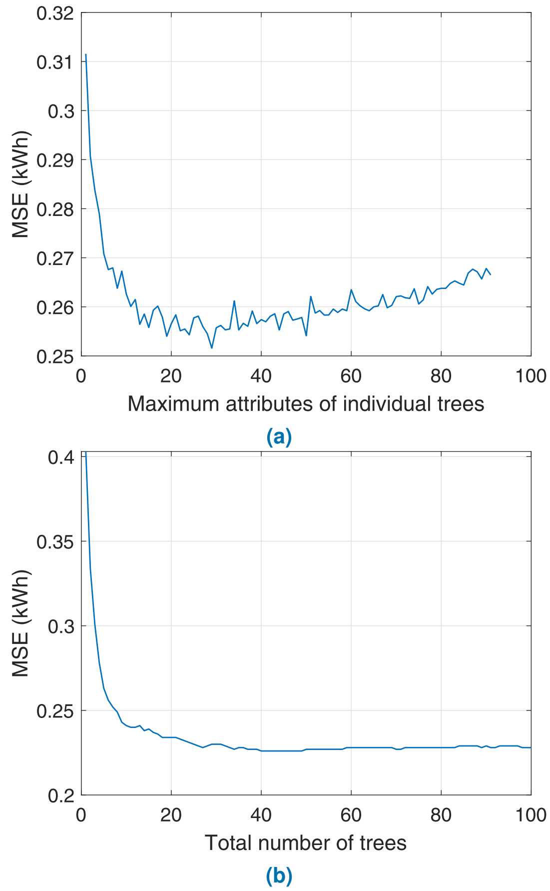


FIGURE 10. Hyper parameter tuning of Random Forest for a single customer of Sweden dataset. (a) MSE of the maximum attributes of the individual trees. (b) MSE of the total number of trees.

TABLE 4. Performance evaluation of different techniques on Sweden dataset.

Technique		RMSE (kWh)	MAE (kWh)	MAPE (%)
Persistence		2.13	0.81	38.7
ARIMA		1.8	1.5	93.6
RF		2.1	0.98	46.1
LSTM		1.25	0.7	31.8
PVS (Proposed Technique)		**0.86**	**0.57**	**25.1**
PVS % Improvement over	Persistence	59%	29%	38%
	ARIMA	52%	62%	73%
	RF	57%	41%	84%
	LSTM	31%	18%	21%

that almost all algorithms perform better on Sweden dataset as compared to Australian dataset. The main reason for this improvement is the magnitude of the forecasting load. Table 1 shows the average and standard deviation of the load of both datasets. It is observed that the average load is the largest for the Sweden dataset, and the minimum MAPE is achieved for it using our proposed PVS technique as shown in Table 4. This also points to the widely accepted phenomenon that forecasting larger aggregated loads is substantially easier than smaller individual loads. It is also observed that PVS outperforms all other STLF techniques. It is to be noted here that PVS is trained on just household data that, while other algorithms are trained using additional parameters like weather and calendar attributes.

TABLE 5. Performance evaluation of different techniques on Australia dataset.

Technique		RMSE (kWh)	MAE (kWh)	MAPE (%)
Persistence		0.83	0.53	61.2
ARIMA		0.89	0.71	199.6
RF		0.98	0.72	234.2
LSTM		0.65	0.40	90.6
PVS (Proposed Technique)		**0.62**	**0.32**	**44.5**
PVS % Improvement over	Persistence	25%	39%	27%
	ARIMA	77%	54%	30%
	RF	36%	55%	80%
	LSTM	4.6%	20%	50%

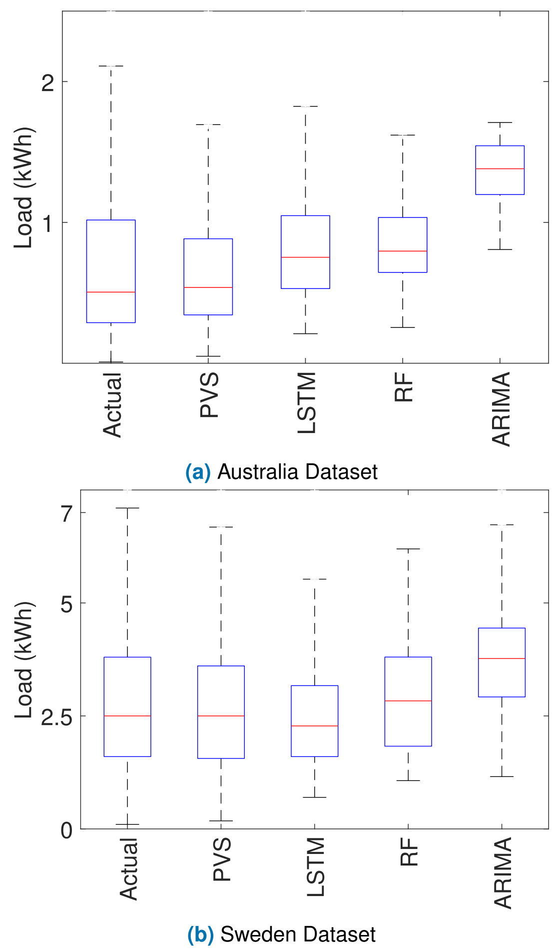


FIGURE 11. Boxplot of actual and predicted load for Sweden and Australia datasets.

ARIMA performed worst on both datasets, followed by RF, persistence, and LSTM. LSTM performed better on Sweden dataset as compared to Australia dataset. The reason behind

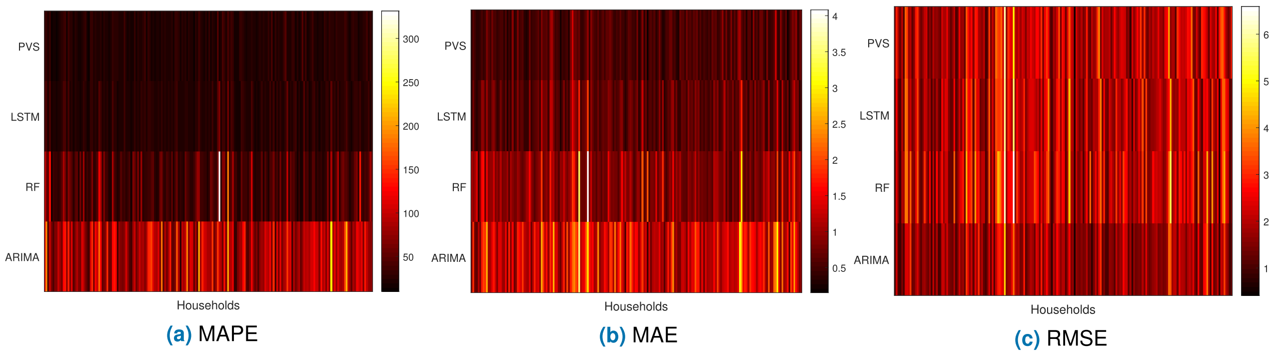


FIGURE 12. Error comparison of individual household for Sweden datasets.

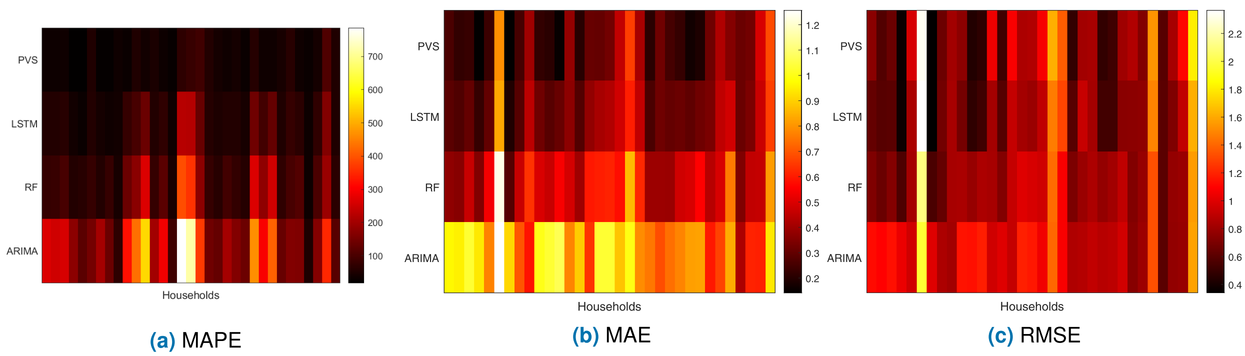


FIGURE 13. Error comparison of individual household for Australia datasets.

this is due to the fact that the average load of Sweden customers is higher than Australia, and LSTM requires a massive amount of data to learn due to the large number of parameters. As Sweden dataset has the large number of customers, i.e., 194, while the Australia dataset has 34 customers. A large amount of data is available to train LSTM for Sweden dataset, so consequently, LSTM performs better for Sweden dataset. PVS has this advantage too over LSTM, that it can be learned for smaller datasets for any number of the customer. It can also be observed in Table 4 that in comparison to LSTM's performance on the Sweden dataset, PVS shows improved performance of 21%, 18% and 31% for MAPE, MAE, and RMSE, respectively. Similarly, in Table 5, 50%, 20% and 4.6% improvement in performance through PVS technique is observed as compared to LSTM on Australia dataset for MAPE, MAE, and RMSE, respectively. In comparison to RF, PVS presents an improvement on both datasets for MAPE, MAE, and RMSE of up to 80%, 40%, and 36%, respectively. A similar improvement of the PVS technique can be seen for ARIMA and persistence forecast.

Since the mean value of error is sensitive to outliers. Recall that due to many actual loads that are close to 0, there could be a large number of outliers in the error. To further analyze the distribution of error, the box plot of actual and predicted load for different techniques is presented in Figure 11. For both datasets, the PVS distribution is closest to the original load. For Sweden household, LSTM slightly under-predicts the load, while RF and ARIMA show over-prediction. In the case of the Australia dataset, LSTM, RF, and ARIMA show overprediction of the load. This can be observed by box plots above the original level. These boxplots are computed using all customers on full testing (6 months) data.

The error comparisons of all techniques for the individual household of Sweden and Australia dataset are presented in the form of heatmaps, in Figures 12 and 13, respectively. The heat maps correspond to MAPE, MAE, and RMSE error of all techniques. There are vertical lines and horizontal patches in each figure. Each vertical line corresponds to a single user error and each horizontal patch corresponds to a STLF technique. Each line is computed by the error of a single user over the testing period. The dark color in the figures refers to less error and improved performance. In the case of MAPE and MAE, PVS performs slightly better than LSTM. The error is increased as the technique is switched from PVS to ARIMA. The improvements of MAPE, MAE, and RMSE measures follow the irregular pattern in terms of results for different households. In the case of RMSE, no significant difference between techniques is observed as can be observed in Figure 12c. The main reason is the square root and averaging operation performed in RMSE, which smooths out the differences. Moreover, the RMSE is a nonlinear operator and individual household RMSE does not accumulate to

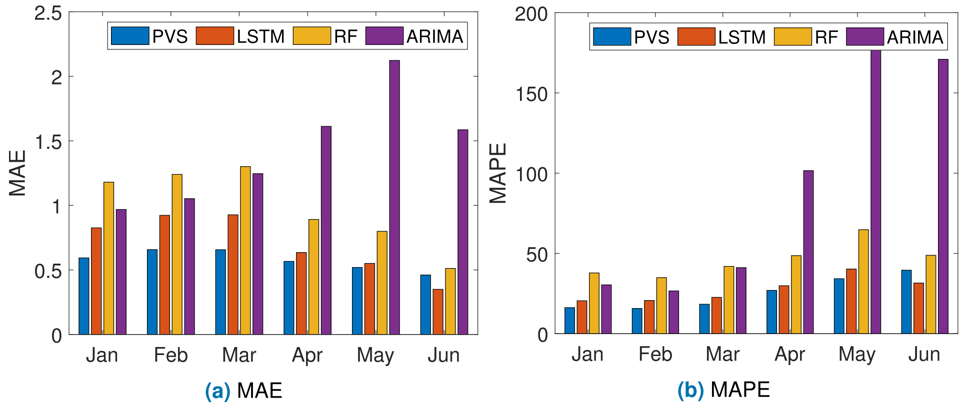


FIGURE 14. Monthly MAE and MAPE for all techniques for Sweden dataset.

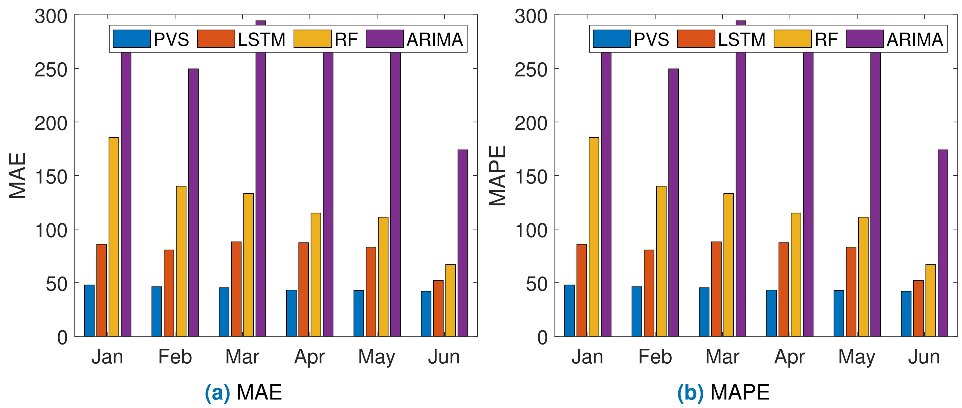


FIGURE 15. Monthly MAE and MAPE for all techniques for Australia dataset.

overall RMSE. Due to these factors, almost all techniques produce similar results in terms of RMSE on individual households.

To analyze seasonal variations of Australian and Sweden dataset, monthly MAE and MAPE of all techniques for the respective dataset are given in Figures 14 and 15. The MAE is calculated over all households for each month. The testing data has the values for the last six months and their box plot is shown in Figure 6. Referring to the box plot as we move from January to June, the median load increases. PVS, LSTM, and RF show consistent behaviors as their MAE decreases as we move towards June. These techniques show the highest MAE in March and February. As discussed earlier that smaller loads are stochastic and variable in nature which makes them difficult to predict. ARIMA shows an opposite behavior and it's MAE increases towards June, with the highest value in May. From Figures 11, 14, 15, we infer that the main reason for this behaviour is due to the overprediction of load. As we approach June, this over-prediction increases further, causing very high MAE in April, May, and June.

VI. CONCLUSION AND FUTURE WORK

Short-term load forecasting is an integral part of a reliable and economical power system. At the household level, short term load forecasting is pivotal for the design of demand side management (DSM) programs, such as demand response (DR), peak shaving, dynamic pricing, and soft load shedding. Most DSM programs are designed to work at an hourly level. The optimization problems in these DSM programs are guided by the data from an hour ahead household load forecasting. Various statistical and machine learning methods are used for the problems associated with STLF at the household level, and most of the techniques require several features for efficient performance. In this work, we have proposed a new technique in this domain called 'Past Vector Similarity' (PVS), in which the hour ahead load forecasting problem at the household level is explored. The operational structure of PVS uses past vector load data to find similar patterns in the load data, and then report the ensemble of the similar loads as a prediction. It is discussed throughout the study that there are many advantages of our proposed PVS technique. PVS consists of two parameters, which assist in its application to a relatively less amount of data without overfitting. Other deep learning techniques require a large amount of data to learn. PVS does not require weather, sociological or other data and works with just load consumption data. Due to the parallel nature of PVS, it can be scaled better for a large number of houses.

Along with the proposed PVS technique, different deep learning techniques used for STLF like ARIMA, LSTM, and Random Forest are evaluated and tested on the household load data of Australia and Sweden. Results of these techniques are then used to validate and compare the results and performance of the proposed PVS technique with the other techniques. State-of-the-art persistence forecast, ARIMA, LSTM, and RF techniques are trained using calendar and weather information, while the proposed technique has only analyzed the load data for prediction. The reported MAPE of the PVS technique has shown a significant improvement over persistence forecast, ARIMA, RF, and LSTM techniques, respectively. We have observed up to 80% improvement over MAPE for Sweden dataset, while up to 200% improvement in MAPE for the Australia dataset. A similar improvement is also seen in terms of MAE and RMSE. To further validate the PVS technique we have analyzed errors in terms of boxplots, monthly basis, and heat maps. All the error analysis support our proposed technique. Although the proposed work has devised a technique that performs better than the other state-of-the-art deep learning techniques; however, a potential future direction is to integrate calendar, weather, and other information into the PVS model to evaluate its performance. Another possible direction is to extend the proposed PVS approach towards a general approach, the applications of which can be aimed towards any regression problem.

ACKNOWLEDGMENT

This research was supported by LUMS Energy Institute (LEI) and Energy Informatics Group (EIG). The authors would also like to extend thanks to the members of LEI and EIG at LUMS for the feedback on the manuscript throughout its preparation, and the support from K.A.CARE.

REFERENCES

[1] A. Ahmed and M. Khalid, "A review on the selected applications of forecasting models in renewable power systems," *Renew. Sustain. Energy Rev.*, vol. 100, pp. 9–21, Feb. 2019.

[2] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, "Dynamic mobile crowdsourcing selection for electricity load forecasting," *IEEE Access*, vol. 6, pp. 46926–46937, 2018.

[3] A. S. Pandey, D. Singh, and S. K. Sinha, "Intelligent hybrid wavelet models for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1266–1273, Aug. 2010.

[4] M. Khalid, R. P. Aguilera, A. V. Savkin, and V. G. Agelidis, "On maximizing profit of wind-battery supported power station based on wind power and energy price forecasting," *Appl. Energy*, vol. 211, pp. 764–773, Feb. 2018.

[5] B. Haley, J. Gaede, M. Winfield, and P. Love, "From utility demand side management to low-carbon transitions: Opportunities and challenges for energy efficiency governance in a new era," *Energy Res. Social Sci.*, vol. 59, Jan. 2020, Art. no. 101312.

[6] X. Li, Y. Tan, X. Liu, Q. Liao, B. Sun, G. Cao, C. Li, X. Yang, and Z. Wang, "A cost-benefit analysis of V2G electric vehicles supporting peak shaving in shanghai," *Electr. Power Syst. Res.*, vol. 179, Feb. 2020, Art. no. 106058.

[7] C. Jankowiak, A. Zacharopoulos, C. Brandoni, P. Keatley, P. MacArtain, and N. Hewitt, "Assessing the benefits of decentralised residential batteries for load peak shaving," *J. Energy Storage*, vol. 32, Dec. 2020, Art. no. 101779.

[8] J. Shen, C. Cheng, S. Wang, X. Yuan, L. Sun, and J. Zhang, "Multiobjective optimal operations for an interprovincial hydropower system considering peak-shaving demands," *Renew. Sustain. Energy Rev.*, vol. 120, Mar. 2020, Art. no. 109617.

[9] H. Mansoor and N. Arshad, "Market model for demand response under block rate pricing," 2020, *arXiv:2009.00439*. [Online]. Available: http://arxiv.org/abs/2009.00439

[10] G. Dutta and K. Mitra, "A literature review on dynamic pricing of electricity," *J. Oper. Res. Soc.*, vol. 68, no. 10, pp. 1131–1145, Oct. 2017.

[11] T. Aslam and N. Arshad, "Soft load shedding: An efficient approach to manage electricity demand in a renewable rich distribution system," in *Proc. 7th Int. Conf. Smart Cities Green ICT Syst.*, 2018, pp. 101–107.

[12] S. Ali, H. Mansoor, I. Khan, N. Arshad, S. Faizullah, and M. A. Khan, "Fair allocation based soft load shedding," in *Proc. SAI Intell. Syst. Conf.* Cham, Switzerland: Springer, 2020, pp. 407–424.

[13] P. Lusis, K. R. Khalilpour, L. Andrew, and A. Liebman, "Short-term residential load forecasting: Impact of calendar effects and forecast granularity," *Appl. Energy*, vol. 205, pp. 654–669, Nov. 2017.

[14] R. Patel, M. R. Patel, and R. V. Patel, "A review: Introduction and understanding of load forecasting," *J. Appl. Sci. Comput.*, vol. 4, pp. 1449–1457, Apr. 2019.

[15] Y.-H. Hsiao, "Household electricity demand forecast based on context information and user daily schedule analysis from meter data," *IEEE Trans. Ind. Informat.*, vol. 11, no. 1, pp. 33–43, Feb. 2015.

[16] K. Gajowniczek and T. Ząbkowski, "Electricity forecasting on the individual household level enhanced based on activity patterns," *PLoS ONE*, vol. 12, no. 4, Apr. 2017, Art. no. e0174098.

[17] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[18] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.

[19] M. De Felice and X. Yao, "Short-term load forecasting with neural network ensembles: A comparative study [application notes]," *IEEE Comput. Intell. Mag.*, vol. 6, no. 3, pp. 47–56, Aug. 2011.

[20] A. Kell, A. S. McGough, and M. Forshaw, "Segmenting residential smart meter data for short-term load forecasting," in *Proc. 9th Int. Conf. Future Energy Syst.*, Karlsruhe, Germany, Jun. 2018, pp. 91–96.

[21] A. Marinescu, C. Harris, I. Dusparic, S. Clarke, and V. Cahill, "Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods," in *Proc. 2nd Int. Workshop Softw. Eng. Challenges Smart Grid (SE4SG)*, May 2013, pp. 25–32.

[22] S. Lu, G. Lin, H. Liu, C. Ye, H. Que, and Y. Ding, "A weekly load data mining approach based on hidden Markov model," *IEEE Access*, vol. 7, pp. 34609–34619, 2019.

[23] F. Javed, N. Arshad, F. Wallin, I. Vassileva, and E. Dahlquist, "Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting," *Appl. Energy*, vol. 96, pp. 150–160, Aug. 2012.

[24] H. Li, Z. Zhang, Y. Zhao, and X. Hu, "Short-term load forecasting based on the grid method and the time series fuzzy load forecasting method," in *Proc. Int. Conf. Renew. Power Gener. (RPG)*, 2015, pp. 1–6.

[25] R. Zhang, Y. Xu, Z. Y. Dong, W. Kong, and K. P. Wong, "A composite K-nearest neighbor model for day-ahead load forecasting with limited temperature forecasts," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Jul. 2016, pp. 1–5.

[26] M. Ghofrani, M. Ghayekhloo, A. Arabali, and A. Ghayekhloo, "A hybrid short-term load forecasting with a new input selection framework," *Energy*, vol. 81, pp. 777–786, Mar. 2015.

[27] J. W. Taylor, "An evaluation of methods for very short-term load forecasting using minute-by-minute British data," *Int. J. Forecasting*, vol. 24, no. 4, pp. 645–658, Oct. 2008.

[28] Y. Chakhchoukh, P. Panciatici, and L. Mili, "Electric load forecasting based on statistical robust methods," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 982–991, Aug. 2011.

[29] J. Che and J. Wang, "Short-term load forecasting using a kernel-based support vector regression combination model," *Appl. Energy*, vol. 132, pp. 602–609, Nov. 2014.

[30] A. Ghasemi, H. Shayeghi, M. Moradzadeh, and M. Nooshyar, "A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management," *Appl. Energy*, vol. 177, pp. 40–59, Sep. 2016.

[31] S. Li, L. Goel, and P. Wang, "An ensemble approach for short-term load forecasting by extreme learning machine," *Appl. Energy*, vol. 170, pp. 22–29, May 2016.

[32] 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 Trans. Smart Grid*, vol. 6, no. 2, pp. 911–918, Mar. 2015.

[33] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1591–1598, Jul. 2017.

[34] S. Ryu, J. Noh, and H. Kim, "Deep neural network based demand side short term load forecasting," *Energies*, vol. 10, no. 1, p. 3, Dec. 2016.

[35] X. Dong, L. Qian, and L. Huang, "Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 119–125.

[36] L. Han, Y. Peng, Y. Li, B. Yong, Q. Zhou, and L. Shu, "Enhanced deep networks for short-term and medium-term load forecasting," *IEEE Access*, vol. 7, pp. 4045–4055, 2019.

[37] R. Li, F. Li, and N. D. Smith, "Multi-resolution load profile clustering for smart metering data," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4473–4482, Nov. 2016.

[38] P. G. D. 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 Trans. Smart Grid*, vol. 5, no. 1, pp. 402–410, Jan. 2014.

[39] K. Gajowniczek and T. Ząbkowski, "Short term electricity forecasting using individual smart meter data," *Procedia Comput. Sci.*, vol. 35, pp. 589–597, Jan. 2014.

[40] A. Veit, C. Goebel, R. Tidke, C. Doblander, and H.-A. Jacobsen, "Household electricity demand forecasting: Benchmarking state-of-the-art methods," in *Proc. 5th Int. Conf. Future Energy Syst.*, Jun. 2014, pp. 233–234.

[41] M. Ghofrani, M. Hassanzadeh, M. Etezadi-Amoli, and M. S. Fadali, "Smart meter based short-term load forecasting for residential customers," in *Proc. North Amer. Power Symp.*, Aug. 2011, pp. 1–5.

[42] P. P. A. W. Athukorala and C. Wilson, "Estimating short and long-term residential demand for electricity: New evidence from Sri Lanka," *Energy Econ.*, vol. 32, pp. S34–S40, Sep. 2010.

[43] J. W. Taylor, "Short-term load forecasting with exponentially weighted methods," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 458–464, Feb. 2012.

[44] A. Gerossier, R. Girard, G. Kariniotakis, and A. Michiorri, "Probabilistic day-ahead forecasting of household electricity demand," *CIRED-Open Access Proc. J.*, vol. 2017, no. 1, pp. 2500–2504, Oct. 2017.

[45] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecasting*, vol. 32, no. 3, pp. 914–938, Jul. 2016.

[46] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Deep learning for estimating building energy consumption," *Sustain. Energy, Grids Netw.*, vol. 6, pp. 91–99, Jun. 2016.

[47] E. Barbour and M. González, "Enhancing household-level load forecasts using daily load profile clustering," in *Proc. 5th Conf. Syst. Built Environ.*, Nov. 2018, pp. 107–115.

[48] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw. (ICANN)*. Edinburgh, U.K.: IET, Sep. 1999.

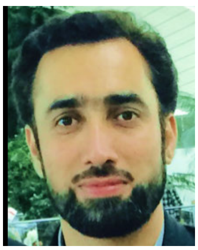
[49] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *Proc. IECON-42nd Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2016, pp. 7046–7051.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

HARIS MANSOOR received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Lahore (UET), and the master's degree in control systems from the University of Engineering and Technology, Taxila. He is currently pursuing the Ph.D. degree in electrical engineering with the Lahore University of Management Sciences (LUMS), Lahore, Pakistan. At LUMS, he is also working with the Energy Informatics Group. His research interests include machine learning, deep learning techniques for modeling sequential data, distributed optimization, and reinforcement learning.

HUZAIFA RAUF (Student Member, IEEE) received the B.S. degree in electrical engineering from the National University of Computer and Emerging Science (FAST-NUCES), Lahore, Pakistan, in 2016, and the M.S. degree in electrical engineering from the Lahore University of Management Sciences (LUMS), Lahore, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, School of Science and Engineering (SSE), Lahore University of Management Sciences (LUMS). He is also working as a Graduate Research Assistant with the Energy Informatics Group (EIG), LUMS Energy Institute. His research interests include smart grids, optimization, energy data analytics, machine learning, power systems planning and operation, renewable integration technologies, electric vehicles, and energy management.

MUHAMMAD MUBASHAR is currently pursuing the bachelor's degree in computer science with the School of Sciences and Engineering (SSE), Lahore University of Management Sciences (LUMS), Lahore, Pakistan. He is actively involved in research with LUMS and is also working in collaboration with the Torr Vision Group, Oxford. His research interests include computer vision, machine learning, and deep learning.



MUHAMMAD KHALID (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the School of Electrical Engineering and Telecommunications (EE and T), University of New South Wales (UNSW), Sydney, NSW, Australia, in 2011. He was a Postdoctoral Research Fellow for a period of three years and a Senior Research Associate with the School of EE and T, Australian Energy Research Institute, UNSW, for a period of two years. He is currently an Associate Professor with the Electrical Engineering Department, King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia. He is also a Researcher with the King Abdullah City for Atomic and Renewable Energy (K. A. CARE) Energy Research and Innovation Center, Dhahran. He has authored/coauthored several journal articles and conference papers in control and optimization for renewable power systems. His current research interests include optimization and control of battery energy storage systems for large-scale grid-connected renewable power plants (particularly wind and solar), distributed power generation and dispatch, hybrid energy storage, EVs, and smart grids. He was a recipient of the Highly-Competitive Postdoctoral Writing Fellowship from UNSW in 2010. He also received the Prestigious K. A. CARE Fellowship. He serves as a reviewer for numerous international journals and conferences.

NAVEED ARSHAD (Associate Member, IEEE) received the M.S. and Ph.D. degrees in computer science from the University of Colorado at Boulder. He is currently an Associate Professor of computer science with the Lahore University of Management Sciences (LUMS), Lahore, Pakistan. He is also the Director of the Energy Informatics Group (EIG), LUMS. His research interests include short, medium, and long term forecasting of energy demand; renewable energy generation; forecasting for wind and solar resources; demand-side management in agricultural, residential, and industrial sectors; energy efficiency; and renewable energy integration in the existing building stock.