Short-Term Load Forecasting Using AMI Data

Haris Mansoor®, Sarwan Ali®, Imdad Ullah Khan®, Senior Member, IEEE, Naveed Arshad, Associate Member, IEEE, Muhammad Asad Khan, and Safiullah Faizullah, Senior Member, IEEE

Abstract—Accurate short-term load forecasting (STLF) is essential for the efficient operation of the power sector. Forecasting load at a fine granularity such as hourly loads of individual households is challenging due to higher volatility and inherent stochasticity. At the aggregate levels, such as monthly load at a grid, the uncertainties and fluctuations are averaged out; hence predicting load is more straightforward. This article proposes a method called forecasting using matrix factorization (FMF) for STLF. FMF only utilizes historical data from consumers’ smart meters to forecast future loads (does not use any noncalendar attributes, consumers’ demographics or activity patterns information, etc.) and can be applied to any locality. A prominent feature of FMF is that it works at any level of user-specific granularity, both in the temporal (from a single hour to days) and spatial dimensions (a single household to groups of consumers). We empirically evaluate FMF on three benchmark data sets and demonstrate that it significantly outperforms the state-of-the-art methods in terms of load forecasting. The computational complexity of FMF is also substantially less than known methods for STLF, such as long short-term memory neural networks, random forests, support vector machines, and regression trees.

Index Terms—Advanced metering infrastructure (AMI), short-term load forecasting, smart meter.

I. INTRODUCTION

SMART grid is a communication and control network on top of the electric grid to collect and analyze data from IoT devices at various grid elements [1]. A fundamental task in smart grid management is to accurately and timely predict the load. Inaccuracy in demand estimates and the subsequent operational decisions may result in grid instability and suboptimal resource utilization entailing high economic costs.

Load forecasting is categorized by its temporal and spatial granularity [2]. Temporally, forecasting is divided into short-term load forecasting (STLF) (from an hour to a few days), mid-term load forecasting (from a few days up to months), and long-term load forecasting (from months up to years). Similarly, based on spatial granularity, load forecasting is categorized from the fine-scale (e.g., individual consumer or household level) to medium-scale (feeders or grids level) to the large-scale (e.g., a collection of grids or a utility company).

Load forecasting for longer terms and larger scales is mainly used for power generation, transmission, and other infrastructure planning. However, operational decisions for smart grids have to be made within a short time and require STLF [3], [4], [5], [6], [7]. With the increasing integration of renewable energy in the energy mix, which is inherently intermittent in nature [8], accurate STLF becomes even more crucial [9]. Moreover, efficient and accurate load forecasting for short-term and at a fine scale is pivotal for demand response programs [10], [11], peak shaving [5], dynamic pricing [12], and soft load-shedding schemes [13], [14].

Traditionally, short-term aggregate electricity consumption data was available for all or large groups of consumers. For STLF at larger scales, a variety of statistical and machine-learning-based methods analyze the historical aggregate load data to predict the (total) load of a group of consumers [5], [6], [7]. Individual consumer’s short-term load data is increasingly becoming available from IoT-based advanced metering infrastructure (AMI). There is an increasing interest in utilizing this data for STLF to better optimize the grid.

However, STLF for individual consumers is considerably more challenging compared to that for large scale. Electricity consumption at the household level depends on the interplay of many factors and consumer characteristics [2], [9], [15]. Most of these factors are unknown or hard to measure, such as demographics, number, daily schedules of residents in a household, etc. The consumption of a group of consumers, on the other hand, is smoother and less volatile since many fluctuations tend to average out at the group level (see Fig. 1). It is well known that loads of larger groups of consumers are easier to forecast [10], [15]. Some recent works on STLF cluster consumers based on their load profiles (hourly AMI readings) and sociological data (obtained from surveys). Then, the load of each cluster is forecasted using machine learning methods [3], [16], [17], [18]. The survey data is not reliable when available and also limits the applicability of the methods to a specific locality. Moreover, the accuracy of these methods significantly degrades at fine scales (e.g., individual consumers).

An important challenge for STLF at a small scale is the volume of data. As the temporal dimension decreases the volume of data increases. But for STLF at a small scale, the number of consumers could be large. This hinders the applicability of machine learning methods requiring complex training (particularly deep learning methods). Another important challenge for STLF at the individual consumer level is computational
complexity since, for short intervals and fine scales, the volume of data is large. While for decision making the inference time matters the most, however since the data is updated in regular (short) intervals, any machine learning model will require continual retraining to adapt forecasts to fresh trends. Retraining frequency will result in enhanced model accuracy. Thus, models with less complex and efficient training are more desirable. Another issue is that most known methods report their forecasting results for a specific level of spatial and temporal granularity. For a different granularity, the models require significant adjustments. Even moving from one scale to another (e.g., 1–2 h) in STLF requires a new design for deep learning algorithms [neural network (NN) and long short-term memory (LSTM)] and new feature engineering for machine learning algorithms.

In this article, we propose forecasting using matrix factorization (FMF) for STLF. FMF is a general-purpose STLF algorithm, meaning it works for the whole range of durations within the short term without losing performance. FMF can also work at any user-specified level of spatial granularity. Given the historical consumption of each consumer at each timestamp, FMF represents timestamps (hour) and consumers by low-dimensional feature vectors. The new feature vectors for timestamps are based on all consumers’ overall behavior. Similarly, the feature vectors of consumers are determined by their consumption patterns for the total period of the past data. In this feature space, we aggregate hours into meaningful clusters. To predict the load of a consumer in a query hour, we find clusters of hours in the past data that are most similar to the query hour and report an average of the consumer’s loads in those clusters.

We perform an extensive empirical evaluation of FMF to showcase its efficacy as a general-purpose forecasting method that works for user-specified spatial and temporal granularity. We compare FMF with baseline and state-of-the-art (SOTA) methods using their respective experimental setup and evaluation metric. We show that FMF significantly outperforms known methods on STLF at the individual consumer level. We test FMF on forecasting loads for longer durations (up to a day), for groups (clusters) of consumers, and both combined. Results reveal that the performance of FMF surpasses the computationally expensive specialized methods for these tasks.

Key features of FMF are as follows.
1) FMF only utilizes the readily available hourly electricity consumption AMI data and does not require any consumer-specific information. Hence FMF is more generally applicable.
2) FMF works at any level of user-specified granularity, both in the temporal and spatial dimensions.
3) FMF does not require any complex training of the model, unlike LSTM, random forest (RF), support vector machine (SVM), and regression tree (RT). Thus, FMF is more suitable for STLF, where timely decisions need to be made.
4) FMF achieves up to 26.5% and 24.4% improvement in root mean square error (RMSE) over support vector regression (SVR) and RT, respectively, and up to 73.8% and 38% improvement in mean absolute percentage error (MAPE) over RF and LSTM, respectively.
5) The representations learned for consumers and time steps in our model preserve the overall structure of data, reduce the computational complexity of the model, and make the model scalable to big data.

The remainder of this article is organized as follows. We provide a brief review of existing methods for STLF in Section II. In Section III, we describe the FMF scheme. Section IV contains data sets description and experimental setup. Results of empirical evaluation and comparisons of FMF are reported in Section V. Finally, we conclude this article in Section VI.

II. RELATED WORK

The existing work on STLF can be divided into three categories, namely: 1) STLF at system or subsystem level in which aggregated the load of all consumers at a locality is forecasted; 2) clusters level STLF, where consumers are intelligently grouped into clusters and loads for those clusters are forecasted; and 3) STLF for individual consumers in which load is forecasted separately for each customer.

A. STLF at System or Subsystem Level

STLF at a system/subsystem level is well explored in the literature. An NN-based method for STLF is proposed in [19] in which households are grouped based on location, nature, and size of loads. In [20], the k-nearest neighbors-based algorithm is used to forecast day-ahead loads of groups of consumers. A framework using wavelet transform and Bayesian NN for STLF at the system level is proposed in [21]. In [22], stochastic properties of electricity in France are utilized to predict short-term aggregated load. A kernel-based SVR model for the STLF at the system level is proposed in [23]. Several authors proposed hybrid methods involving data preprocessing with classification, regression, and other machine learning-based methods for STLF at system/subsystem level [9], [24].

There are several problems with directly using machine learning models for STLF, such as difficulties in parameter selection and nonobvious selection of input variables [25]. Therefore, these models have to be combined with statistical models and different data preprocessing techniques to reduce the computational overhead. Tayab et al. [25] used a combination of NN and statistical models for STLF at the microgrid level. An unsupervised machine learning model is proposed
in [26], which combines auto correlation function and least-squares SVMs model to forecast short-term load at the system level. The actual deployability of the forecasting algorithms at the country level is studied in [27]. The authors focus on prediction power, the robustness of the model, the dependence of the model on the data set, and storage size. Multiple wavelet convolution NNS are proposed, which balance these measures. Since the impact of variables on demand changes over time, an online continuous learning approach is proposed and tested by [28]. The author used correlation to figure out the effect of variables and then used an NN for prediction in an online setting.

B. STLF for Clusters of Consumers

A wide variety of methods utilize the increasingly available AMI data to intelligently group the households into clusters (based on their consumption patterns) and forecast the load of these clusters. For cluster loads prediction, machine learning models, such as RF, NNS, and deep learning, are commonly used. Clustering is accomplished based on similarities in load profiles (consumers’ AMI readings) [18] and consumers demographic information [17]. Practice theory of human behavior is incorporated for improved clustering [29], resulting in an accuracy boost for day-ahead system-level load forecast. A deep NN-based model for STLF at an individual and subsystem level is proposed in [30], which learns complex relations between weather, calendar, and previous consumption for individual households. A hybrid approach consisting of a convolutional NN and k-means clustering algorithm is proposed in [31] to forecast the hourly load of clusters of households. Li et al. [32] proposed a multiresolution clustering method to forecast half-hourly load for households. The relationship between cluster size and forecast accuracy is studied in [33] using two forecasting methods, namely, Holt-Winters and Seasonal Naive.

C. STLF for Individual Consumers

STLF at individual consumers’ level is significantly more challenging due to high volatility and variability in load profiles [34]. The classical methods treat each consumer’s data as a stationary time series for prediction. Time-series methods for STLF use Kalman filter [35], advanced statistical techniques [36], and the standard auto-regressive integrated moving average (ARIMA) forecasting models [5]. These time-series approaches, however, do not capture the complex nonlinear relationship between electricity consumption and periodic routines of household residents [16]. It has been shown in [37] that time-series-based methods hardly beat persistent forecast (PF) (using the previous hour load value as the forecast for the next hour). Ignatiadis et al. [38] proposed a regression-based method to forecast the monthly aggregated load of individual households. However, they do not consider the temporal order of the historical loads in which the load values were observed. This limitation restricts the application of the method in any realistic scenario. Recently, deep-learning-based methods have dominated the forecasting problems in smart grids like wind speed and load forecasting [39]. A Pooling-based deep recurrent NN method is proposed in [5] to predict individual household loads and improve upon the accuracy of ARIMA, and other machine learning models. Similarly, [10] uses LSTM network along with density-based clustering for household load forecasting. Mocanu et al. [40] used a factored conditional restricted Boltzmann machine for load forecasting of buildings and showed improvement over the SVM and NN. An STLF model for individual household level is proposed in [34], which uses standard machine learning models, such as NNS and SVM for load forecasting. Yildiz et al. [41] proposed a model for STLF, which uses historical loads and weather information along with the information contained in typical daily consumption profiles (loads in mornings, evenings, and nights, etc.) for load forecasting. A predicted model (sparse high-dimensional partially linear additive models) for STLF at individual households level is used in [15], which forecasts half-hourly electricity load for one day ahead. Hybrid methods for household level STLF incorporate additional activity patterns information (survey, demographic information, etc.) to improve household load forecast [3]. However, the activity patterns information is not readily available in many cases.

III. PROPOSED APPROACH

In this section, we describe the detailed algorithm of FMF. FMF takes the load matrix \( X \in \mathbb{R}^{m \times n} \) as input, with rows and columns corresponding to \( m \) hours and \( n \) consumers, respectively. The entry \( X(i,j) \) is the electricity consumption of consumer \( j \) at hour \( i \). FMF broadly performs the following steps: We first preprocess the data to improve its statistical properties. Then, we split \( X \) into two submatrices \( A \) and \( B \). The submatrix \( A \) consists of the first \( m_1 \) rows and is used as training data. While submatrix \( B \), consisting of the last \( m_2 = m - m_1 \) rows, is used for testing. Thus, the dimensions of \( A \) and \( B \) are \( m_1 \times n \) and \( m_2 \times n \), respectively. Next, we map hours of \( A \) into a low-dimensional feature space based on the overall consumption patterns. In this feature space, hours of \( A \) are clustered. The forecast for \( B(i,j) \) is an average of consumer \( j \)’s loads (along with the load of consumers similar to \( j \)) in the \( t \) clusters of hours of \( A \) that are the most similar to the query hour \( i \). However, since clusters are in a load-based (abstract) feature space and hour \( i \) is in the testing period, we cannot use load values at hour \( i \). Therefore, we find a common representation both for testing hours and clusters of training hours based only on the calendar attributes of hours. Similarities between query hours and clusters are evaluated with this representation. We provide details of each step below.

A. Data Preprocessing

First, we min–max normalize columns of \( A \) to make the values unitless and scale them to \([0,1]\). The load values in \( X \) are for individual consumers and a short duration; most are very low values. Hence, there is a significant right skew and variation in the data. We apply the standard \( q \)th root transformation on \( X \) as a preprocessing step, i.e., every value \( X(i,j) \) is replaced with \( X(i,j)^{1/q} \). For notational convenience, we still
denote the transformed matrix by $X$. The skew and effect of transformation are depicted in Fig. 2, showing load distributions at a randomly chosen hour. It is clear from the figure that a large number of values are very close to 0 before transformation and that the transformed data is more normally distributed. The optimum value of $q$ is selected empirically as 4, 5, and 3 for Sweden, Ireland, and Australia data sets, respectively.

Remark 1: Note that we apply the corresponding reverse transformation (4th power) after forecasting the load and report our predictions and their errors in the original scale.

B. Consumption Patterns-Based Feature Map and Clustering of Training Hours

Our goal is to cluster hours of $A$ based on the overall consumption patterns during these hours. Moreover, we want to define a similarity measure between a testing hour and a cluster of training hours. However, every hour of $A$ is (potentially) a very high-dimensional vector ($n$). In such a high-dimensional space, due to curse of dimensionality, no notion of pairwise similarities and thus clustering is significant. Forecasting accuracy, however, critically depends on the quality of clustering. To deal with this problem, we reduce the dimensionality of the matrix $A$, using the following fundamental result on singular value decomposition (SVD) from linear algebra (c.f., [42]).

Theorem 1: Suppose $Z$ is an $a \times b$ real matrix with rank $r$. Then, there exists a factorization $Z = U \Sigma V^T$ such that $U$ is $(a \times r)$ matrix of orthonormal rows, $\Sigma$ is $r \times r$ diagonal matrix of nonnegative real numbers, and $V$ is $b \times r$ matrix of orthonormal rows.

The rows of $U \Sigma$ and columns of $\Sigma V^T$ represent rows and columns of $Z$, respectively, in abstract feature space (latent factors on which the data varies). The relevance of features is quantified by the singular values $(\Sigma)$. For $d \leq r$, let $\Sigma_d$ be the $d \times d$ diagonal submatrix of $\Sigma$ consisting of the largest singular values (these singular values contain maximum weight or energy in $\Sigma$ [42]). Let $U_d$ and $V_d$ be the submatrices of $U$ and $V$ consisting of the $d$ columns corresponding to values in $\Sigma_d$. The truncated matrix $Z_d = U_d \Sigma_d V_d^T$ approaches $Z$ as $d$ approaches $r$ and is called low-rank approximation of $Z$.

We apply SVD on $A$ to get $A = U \Sigma V^T$ and consider its approximation $A_d := U_d \Sigma_d V_d^T$. Let $H := U_d \Sigma_d$, then rows of $H$ are the $d$-dimensional representations of hours of $A$. Each column of $H$ is a feature of hours, based on loads of all consumers in $A$. We cluster the training hours of $H$ into $r$ clusters, $C = \{C_1, C_2, \ldots, C_r\}$. Each cluster $C_i$ contains hours that are substantially similar to each other based on the overall consumption patterns. We choose the values of $d$ (the number of reduced dimensions) and $r$ (the number of clusters) by evaluating the quality of clusters. The chosen values for $r$ are 80, 70, and 70, while those for $d$ are 300, 424, and 34 for Sweden, Ireland, and Australia data sets, respectively.

Remark 2: No dimensionality reduction is applied for the Australia data set as the number of consumers (columns of the load matrix) is already small, i.e., “34.”

Remark 3: We selected the singular values $d$ of $\Sigma$ for SVD that preserve 80%, 80%, and 100% cumulative energy in $\Sigma$ for Sweden, Ireland, and Australia data sets, respectively.

We use the k-means++ algorithm [43] to cluster the rows of $H$. To avoid local minima, we replicate the clustering 1000 times and select the most accurate partition.

Given a customer $j$, we now find other customers similar to $j$. For this, we first take the transpose of matrix $X$ (to represent customers in rows and hours in columns) and apply SVD to reduce the number of columns (hours). However, we perform this task for each month’s hours separately (to capture the seasonality information). To do this, we separate the hours of each month ($\approx 720$ h for each month), which will give us a separate $\approx n \times 720$-dimensional matrix. On each month’s matrix, we apply SVD separately columns (and get ten principal components) to reduce its dimensions (we get $n \times 10$-dimensional matrix). Then, we concatenate values of all months (all $n \times 10$ matrices together for 12 months) to form a single $n \times 120$ dimensional matrix (where $10 \times 12 = 120$). We refer to this as the seasonal SVD approach. The top $k$ similar consumers (rows in this new matrix) to consumer $j$ are then selected using the Euclidean similarity. We use $k = 3$, empirically set using the standard validation set approach [44].

C. Calendar Attributes-Based Feature Map

We use rows of $H$ to represent corresponding hours of $A$. To predict a test matrix value $B(i, j)$, the load of consumer $j$ at hour $i$, we identify hours of $A$ (represented by clusters in $C$) that are most similar to the query hour $i$ and report an average of loads of consumer $j$ and that of its top $k$ neighbors in those clusters. The only information of a query hour we can use is its calendar attributes, i.e., time, day, and month. On the other hand, attributes of clusters of training hours (columns of $H$) are based on overall consumption at those hours.

For a common representation of an hour and a cluster of hours, we use a 75-D vector, $\mathbf{v}(\cdot)$. The first 24 coordinates of $\mathbf{v}(\cdot)$ ($\mathbf{v}[0 \ldots 23]$) correspond to the $24$ h in a day. The next seven coordinates represent the days of a week, and the 31 coordinates after them represent the days of a month. The following 12 coordinates stand for the months of a year. The 74th and 75th coordinates encode public holidays.

For a given hour $h$ (a timestamp with associated calendar information), the value of $\mathbf{v}(h)$ at a coordinate is 1 if $h$ has the corresponding attribute. Fig. 3(a) depicts an example vector representation of an hour. For a cluster $C = \{h_1, h_2, \ldots, h_{|C|}\}$ of hours, $\mathbf{v}(C)$ is the distribution of hours [represented as $\mathbf{v}(\cdot)$] contained in $C$. Formally, $\mathbf{v}(C) = (1/|C|) \sum_{h \in C} \mathbf{v}(h)$.
Fig. 3(b) depicts an example vector representation of a cluster of hours (rows of \( H \)).

**D. Forecasting the Load**

Finally, we describe the mechanism to estimate the load of household \( j \) at a testing hour \( i \), i.e., the value of \( B(i,j) \). To predict \( B(i,j) \), we report an “average” of household \( j \)’s load along with a load of its top \( k \) nearest neighbors in the cluster of hours in \( A \) that are most similar to the query hour \( i \). For this, we need a measure of distance/similarity, \( d(h,C) \) between a testing hour \( h \) and a cluster of training hours \( C \). Since, the vector representations \( v(h) \) and \( v(C) \) (for hour \( h \) and cluster \( C \), respectively), encode attributes of \( h \) and \( C \) related to time, day, month, etc. We define \( d(h,C) \) to be a weighted sum of the \( l_p \)-distance between the corresponding attributes of \( h \) and \( C \). More precisely, let \( d[q] \) be the absolute difference between the \( q \)th coordinate of \( v(h) \) and \( v(C) \), i.e., \( d[q] = |v(h)[q] - v(C)[q]| \). Consider a vector “\( w \)” that contains weights of attributes \( w_1, w_2, \ldots, w_N \). Also consider another vector “\( a \)” that contains following elements of the feature vector related to calendar attributes: \( a_1 = (\sum_{q=1}^{23} d[q])^{1/p} \), \( a_2 = (\sum_{q=24}^{30} d[q])^{1/p} \), \( a_3 = (\sum_{q=31}^{61} d[q])^{1/p} \), \( a_4 = (\sum_{q=62}^{73} d[q])^{1/p} \), \( a_5 = (\sum_{q=74}^{75} d[q])^{1/p} \).

The distance \( d(h,C) \) between \( h \) and \( C \) is defined as \( d(h,C) = a^T \cdot w \). The optimum values for the weights of attributes \( w_1, w_2, \ldots, w_N \) are computed using the standard validation set approach [44]. The similarity, \( sim(h,C) \), between an hour \( h \) and a cluster \( C \) is given by \( sim(h,C) = 1 - d(h,C) \). To predict \( B(i,j) \), we find a subset \( C' \subseteq C \) of \( t \) clusters that have the highest similarity with the hour \( i \) and a subset of top \( k \) nearest neighbors of consumer \( j \) [i.e., \( N_1(j), \ldots, N_k(j) \)]. We report the similarity-weighted mean of the median consumption of the subset, which consists of loads of consumer \( j \), \( N_1(j), \ldots, N_k(j) \) in these \( t \) clusters (we empirically select \( t = 2 \) for all data sets). In other words, FMM computes a forecast \( B(i,j) \) for the load \( B(i,j) \) as follows:

\[
B(i,j) = \frac{\sum_{C \in C'}(\text{sim}(i,C) \times \text{MEDIAN}_{h \in C} \cdot [a])}{\sum_{C \in C'}\text{sim}(i,C)}
\]

\[a = \text{MEDIAN}_{x \in N(j)} A(h,x).\]

**E. Time Complexity of FMM**

The running time of the preprocessing is linear with respect to the size of the input. The training matrix \( A \) has dimensions \( m_1 \times n \), SVD on \( A \) that takes \( O\left(\min(m_1n^2, m_1n)\right) \) time. The next step of the algorithm is clustering training hours. The standard \( k \)-means algorithm takes time proportional to \( O(nrI) \), where \( r \) is the number of clusters and \( I \) is the number of iterations of the \( k \)-means algorithm (similar is the case for clustering the customers). Note that these \( t \)-nearest neighbors are found for each hour only, not for individual consumers, i.e., for every row of \( B \), we perform this step only once. Therefore, the total time required for all nearest neighbors computations is \( O(mryrt) \). Thus, the total running time of all these steps is \( O(mrt + mn^2/2 + m_1nr) \), where \( r, t, \) and \( I \) are user-set parameters and usually small constants. Hence, the time complexity is dominated by the SVD step. To make a forecast, we compute the medians in each nearest cluster and report an average of these medians. Thus, the worst case runtime of a forecast is \( O(m_1) \).

**IV. EXPERIMENTAL SETUP**

In this section, we first describe the three benchmark data sets that we use for evaluating FMM. We then describe the evaluation metrics used to measure the goodness of our approach. We also discuss the baseline and SOTA methods for STLF used for comparison with FMM. In the end, we show the visual representation of data to analyze the hidden patterns (if any exist in the data). Our algorithms are implemented in MATLAB and Python on a Core i7 PC with 8-GB memory. Code of FMM and the preprocessed data sets are available online\(^1\) for reproducibility.

**A. Data Set Description and Visualization**

We use real-world smart meter data of hourly consumption from different residential areas of Sweden [45], Australia [2] and Ireland [46]. Table I shows detailed statistics of these data sets.

To visually examine natural clusters in the data (if any), we embed timestamps into a 2d real vector space using \( t \)-distributed stochastic neighbor embedding \( (t\text{-SNE}) \) [47]. Recall that \( X \in \mathbb{R}^{m \times n} \) is the load matrix, where \( m \) is the number of hours, and \( n \) is the number of households. We apply \( t\text{-SNE} \) on the rows of \( X \) to get a matrix \( F \in \mathbb{R}^{m \times 2} \). We plot the rows of \( F \) with each row labeled based on the calendar attributes to observe patterns in the data visually.

Fig. 4(a) shows the \( t\text{-SNE} \) plot for the Australia data set with hours assigned month names as labels. Although we can observe some small clusters (i.e., July 2010 and June 2011 at the top center of the plot), there is no clear separation between data points based on months. Similarly, there is no clear separation between hours based on “Weekdays”

---

\(^1\)https://github.com/sarwanNasha/Load-Forecasting
Fig. 4. t-SNE plots of different labels for the Australia data set. (a) Month wise labels. (b) Weekdays/weekends. (c) Public holiday. (d) Hours of the day.

Fig. 5. t-SNE plots of different labels for the Sweden data set. (a) Month wise labels. (b) Weekdays/weekends. (c) Public holiday. (d) Hours of the day.

Fig. 6. t-SNE plots of different labels for the Ireland data set. (a) Month wise labels. (b) Weekdays/weekends. (c) Public holiday. (d) Hours of the day.

and “Weekends” or based on “Public Holiday” and “No Public Holiday” [Fig. 4(b) and (c)]. We also group hours of a day into four periods, namely, “Sleep Time” (12:00 A.M.–8:00 A.M.), “Office Hours” (8:00 A.M.–5:00 P.M.), “Commute Time” (5:00 P.M.–7:00 P.M.), and “Family Time” (7:00 P.M.–12:00 A.M.). We can observe in Fig. 4(d) that there are some patterns for sleep hours, office hours, and family time. However, there is no complete separation between different labels. Overall, this scattered behavior shows no clear pattern in the data, i.e., extracting any useful information from data without any pre-processing is not easy. A similar clustering can be seen for Sweden and Ireland data sets in Figs. 5 and 6, respectively.

B. Evaluation Setup and Metrics

We use the following error metrics to test the performance of FMF: mean absolute error (MAE) [5], MAPE [10], RMSE [2], and normalized RMSE (NRMSE). In order to make a fair comparison of FMF with different methods, we report these four different measures. Since FMF is a general algorithm these different measures capture the performance of different forecasting granularities more properly. For instance, RMSE weighs bigger values more, however, at 1 h and individual household granularity, the magnitude of data is significantly small (less than 1 kWh). Thus, a better measure to evaluate the quality of forecasts for such data is percentage error (MAPE).

From all data sets, we use data from the first 18 months in our experiments. The first \( \approx 12 \) months of data \([m_1 = 8760 (h)]\) is used as training data. The succeeding \( \approx 6 \) months data \([m_2 = 4104 (h)]\) is used for testing.

C. Clustering Evaluation

We evaluate the effectiveness of clustering hours in the abstract feature space (rows of \( H = U_d \Sigma_d \)) by observing clusters representations, \( \mathbf{v}(\cdot) \). We note that hours expected to have similar consumption based on domain knowledge tend to be grouped into the same clusters.

Fig. 7 depicts three randomly chosen clusters and shows that clustering of rows of \( H \) is meaningful and successfully avoids the curse of dimensionality. The first cluster (a) contains the winter night hours (12 A.M. to 8 A.M.) of one whole week. The second cluster (b) is for the daytime summer weekends, while the last cluster (c) is for the evening hours of winter weekends.
Remark 4: Note that we perform clustering only once, and the same clusters are used to predict the entire test matrix \( B \) (i.e., \( \approx 6 \) months hourly load).

D. Comparison Algorithms

We compare the results of FMF with several baselines and SOTA methods proposed in the literature.

1) Baseline Method: We use autoregressive integrated moving average (ARIMA), a time-series model, as a baseline [5]. ARIMA treats historical loads as a time series and attempts to learn parameters for forecasting future values. Our second baseline is “PF,” which uses an average of “previous hours” loads as the forecast for the next hour.

2) State-of-the-Art Methods: The SOTA methods that we use for comparison with FMF are as follows: LSTM [10], [17], multiple linear regression (MLR) [2], RTs [2], NN [2], [17], SVR [17], and RF [17].

We also compare FMF with the methods proposed in [2] and [17]. Lusis et al. [2] reported household level STLF results on the Australia data set using four different machine learning algorithms. We apply FMF on the same data set (with the same train–test split and settings) and report forecasting accuracy (using the same error metrics). Kell et al. [17] employed four machine learning models to forecast loads for clusters of consumers for the Ireland data set. We use their method for clustering consumers into clusters and forecast their loads using FMF and compare the forecasting error with the best and recommended method by [17].

V. RESULTS AND DISCUSSION

In this section, we report forecasting results of FMF and perform a comparison with the baseline and SOTA approaches. Since FMF works at any level of user-defined granularity both in spatial and time domains, we demonstrate its effectiveness for predicting aggregated load of clusters and total load for extended periods ranging from an hour to a day.

A. Evaluation of Hourly Forecast at Household Level

In this section, we use FMF to perform STLF at household level, i.e., we forecast individual entries of the test matrix \( B \). Fig. 8 shows the boxplots of actual hourly loads (all \( m_2 \times n \) values in \( B \)) and the hourly loads forecasted using ARIMA, RF, LSTM, and FMF. Note that the loads predicted using FMF are closer to the actual loads than all other methods in all data sets.

Table II shows the comparison of different methods for hourly load prediction with FMF in terms of average error over all households and hours. In this experimental setting, we forecast the hourly load for the next six months. Note that the actual load values in all data sets are very close to 0 (see Table I) since they represent individual household loads for a short duration (an hour). Thus, a slight prediction error results in a higher percentage error. This is demonstrated by higher MAPE (> 30%) for all methods. However, RMSE, NRMSE, and MAE values in Table II are small, indicating that the predicted loads are close to actual loads (which is also evident from Fig. 8). Although the error improvement is small in magnitude, they are normalized over all the testing hours and all consumers. Thus, this error is per consumer per hour error and a small improvement in forecasting error amounts to huge cost reductions for the overall system and entails many other benefits.

Remark 5: Note that we are forecasting hourly load for the next six months. This is still called short-term load forecasting as the hourly load is being forecasted. In the case of medium-term load forecasting, the “total load” for a few months is forecasted rather than the hourly load.

Since variability in low loads is relatively much higher, algorithms tend to perform better if the load values of households are on the higher side. Table I shows the average and standard deviation of loads of three data sets. The average

![Fig. 7. Bar graphs of \( v(\cdot) \) for three clusters of hours (Ireland data set), showing similar load pattern hours clustering.](image)

![Fig. 8. Actual and forecasted loads of all hours and all households for ARIMA, RF, LSTM, and FMF.](image)

**TABLE II**

| Method         | Dataset | MAE (kWh) | RMSE (kWh) | NRMSE (%) | MAPE (%) |
|----------------|---------|-----------|------------|-----------|----------|
| ARIMA          | Sweden  | 1.55      | 1.87       | 10.79     | 98.3     |
|                | Ireland | 1.01      | 1.44       | 48.50     | 233.2    |
|                | Australia| 0.71      | 0.89       | 6.79      | 199.5    |
| RF [17]        | Sweden  | 1.03      | 1.57       | 0.014     | 51.49    |
|                | Ireland | 1.20      | 2.06       | 0.02      | 354.9    |
|                | Australia| 0.72      | 0.98       | 0.09      | 234.2    |
| LSTM [10]      | Sweden  | 0.79      | 1.35       | 0.017     | 31.45    |
|                | Ireland | 0.65      | 1.30       | 0.015     | 149.9    |
|                | Australia| 0.405     | 0.65       | 0.06      | 90.68    |
| FMF            | Sweden  | 0.78      | 1.24       | 0.012     | 37.1     |
|                | Ireland | 0.60      | 1.20       | 0.014     | 92.9     |
|                | Australia| 0.401     | 0.60       | 0.05      | 97.4     |
| FMF (%) improvement over ARIMA | Sweden| 49.67     | 33.68     | 99.9     | 62.3     |
|                | Ireland | 40.59     | 16.66     | 99.9     | 58.3     |
|                | Australia| 43.6      | 32.58     | 99.2     | 51.1     |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
load is the largest for the Sweden data set, and we achieve the minimum MAPE on it compared to other data sets (see Table II). The average load of households for the Australia data set is the smallest among the three data sets, and we got a larger MAPE for this data set. This also points to the widely accepted phenomenon that forecasting larger aggregated loads is substantially easier than smaller individual loads.

From Table II it is clear that in most cases, FMF outperforms the baseline and SOTA approaches. However, these are the mean values of errors, which are very sensitive to outliers. Recall that many actual loads are 0 or very close to it; there is a sizeable number of outliers in the errors. Therefore, we also plot all point-wise absolute percentage errors in Fig. 9 to show that for the majority of consumers FMF achieves very low percentage errors. It is clear from Fig. 9 that the median percentage error achieved by FMF is the smallest compared to other methods (on the Sweden data set it is very close to the median performance of LSTM).

1) Month-Wise Performance Comparison: To analyze the effect of seasons on forecasting, we report the results for different months. We observe that on the Sweden data set [Fig. 10(a)], FMF is better than all methods for May and June, better than ARIMA and RF for all other months, and comparable to LSTM for Jan to April. For the Ireland and Australia data set [Fig. 10(b) and (c)], FMF is better than all methods for every month, except in the Australia data set, it is comparable with LSTM.

2) Comparison With Method Proposed in [2]: Lusis et al. [2] performed STLF at even finer granularity (30 min ahead) using different machine learning methods to perform STLF. For comparison, as performed in [2], we use 28 days for testing and all remaining data for training “≈ 2.5 years” (for Australia data set). The test data consists of randomly selected weeks in September (2012), December (2012), March (2013), and June (2013). The comparison, using the same parameters, data set, and experimental settings as in [2], is given in Table III. Observe that FMF achieves up to 26.5% improvement in RMSE and 94.1% improvement in NRMSE.

3) Comparison With Persistent Forecast and Autocorrelation Analysis: We designed two settings for the PF model and compared them with the FMF results. In the first setting, PF1, given a query hour \( t \), we take the load values for \( t - 1, t - 2 \) h of the same day, and \( t \) and \( t - 1 \) h of the previous day and take the average of all these load values as the predicted value. In the second setting, PF2, given a query hour \( t \), we take the load values for \( t - 1, t - 2 \) h of the same day, and \( t \) and \( t - 1 \) h of the previous day and take the average of all these load values as the predicted value. In the second setting, PF2, given a

![Figure 9](image-url) Absolute percentage errors of all households and all hours for ARIMA, RF, LSTM, and FMF.
Fig. 10. Monthly MAPE of ARIMA, RF, LSTM, and FMF for (a) Sweden, (b) Ireland, and (c) Australia data sets (lower value is better).

Moreover, SVD also helps reduce the computational runtime of the underlying clustering algorithm (by reducing the dimensionality of the data). We report the effect of dimensionality reduction on the running time of clustering $m$ hours. Recall that each hour is an $n$-d vector ($n = 709$, for the Ireland data set). Fig. 11 plots the runtimes clustering varying number of hours through $k$-means algorithm ($k = 80$), when the dimensionality of hours is 709 and 424 (approximately 80% energy in the singular values is preserved, see Section III-B).

**B. Evaluation of FMF at Higher Granularity**

As discussed above, load forecasting at coarser levels (either in space or time dimensions) is substantially easier. This, however, is an important problem in the power sector decision-making. FMF is most suited for STLF at the individual household level, but it is a general-purpose method that works at any level of user-defined granularity and yields significantly more accurate forecasts in most cases. In this section, we use FMF to forecast load for longer durations (up to a day), for groups (clusters) of households, and both combined.

1) Forecast for Longer Durations: In this section, we show results of FMF and other methods on forecasting loads of individual consumers for longer periods. We add consecutive hours (rows) of the original load matrix in this setting. We aggregated 2, 4, 12, and 24 consecutive hours of the original load matrix (training and testing data before transformation). Fig. 12 shows the MAPE of FMF, LSTM, RF, and ARIMA with increasing hours aggregation. Clearly, with no hour aggregation (when $x$-axis value is 1), FMF is better than RF and ARIMA on all data sets. FMF yields better results than LSTM on the Ireland, comparable on the Sweden and performs slightly worse on the Australia data set. However, the performance of LSTM degrades with increasing hours aggregation.

| Techniques | MAPE for different numbers of clusters |
|------------|---------------------------------------|
|            | 2 | 3 | 4 | 5 | 6 | 7 |
| NN         | 6 | 5.3 | 5 | 5.1 | 5 | 5.2 |
| SVR        | 6 | 5.3 | 5.1 | 5.1 | 5.2 | 5.4 |
| RF         | 6.1 | 4.6 | 4.6 | 4.7 | 4.7 |
| LSTM       | 10.8 | 9.2 | 9 | 8.5 | 8.7 | 8.6 |
| FMF        | 4.4 | 3.2 | 4.8 | 4.3 | 4.3 | 5.6 |

% improvement of FMF over RF: 22.8, 36, −4.3, 6.5, 8.5, −19.1

2) Forecast for Clusters of Households: To evaluate the performance of FMF on higher spatial granularity, we group the households into clusters and forecast the total load of each cluster for individual hours. To compare FMF with [17], we follow the same train–validation–test split setting (9 months–3 months–6 months) for the Ireland data set (as used in [17]). We cluster households, represented by our feature vectors (see Section III-B) into varying numbers of clusters (2–7). Table VI shows the MAPE of FMF and the scheme presented in [17]. FMF outperforms other methods in the majority of the cases.
We also report improvement over RF, the top performing and recommended method in [17].

3) Forecast of Clusters of Households for Longer Durations: In this section, we first aggregate hours (add consecutive hours) and then households (cluster the households and aggregate their total loads) and then forecast the total load using FMF. We report the errors in clusters’ total forecasted loads for different periods (consecutive hours sum shown on the horizontal axis) for varying numbers of clusters. We can see from Fig. 13 that forecasting the aggregated load (both row “hours” and column “households” wise) helps to reduce the error in most cases. The error values show some variations, but overall the error reduces as we aggregate the hours and households. The spikes in some cases are due to randomness.

VI. CONCLUSION AND FUTURE WORK

In this work, we proposed a matrix factorization-based method called FMF for STLF at the individual consumer level. FMF is computationally less expensive and can be applied to any locality because it does not use any consumers’ demographic or activity pattern data. FMF forecast the hourly load of individual households with high accuracy compared to SOTA machine learning-based methods. We have observed up to 49% improvement in MAE, up to 33% in RMSE, up to 99% in NRMSE, and up to 62% improvement in MAPE over other techniques for STLF on different data sets. FMF also produce promising results on forecasting loads of clusters of consumers for longer durations. This illustrates the general ability of FMF to apply in any temporal and spatial granularity. Potential future work is to combine FMF with existing machine learning methods in an ensemble-based approach. Another future direction is to extend the approach of FMF to the problems of wind speed and solar intensity forecasting.

REFERENCES

[1] S. Chen et al., “Internet of Things based smart grids supported by intelligent edge computing,” IEEE Access, vol. 7, pp. 74089–74102, 2019.
[2] P. Luis, K. Khalipour, 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.
[3] K. Gajowniczek and T. Zabkowski, “Electricity forecasting on the individual household level enhanced based on activity patterns,” PLoS One, vol. 12, no. 4, 2017, Art. no. e0174098.
[4] T. Hong and S. Fan, “Probabilistic electric load forecasting: A tutorial review,” Int. J. Forecast., vol. 32, no. 3, pp. 914–938, 2016.
[5] 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.
[6] P. Jiang, F. Liu, and Y. Song, “A hybrid forecasting model based on date-framework strategy and improved feature selection technology for short-term load forecasting,” Energy, vol. 119, pp. 694–709, Jan. 2017.
[7] N. Amjad and F. Keynia, “Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm,” Energy, vol. 34, no. 1, pp. 46–57, 2009.
[8] Y. Wang, Y. Shen, S. Mao, X. Chen, and H. Zou, “LASSO and LSTM integrated temporal model for short-term solar intensity forecasting,” IEEE Internet Things J., vol. 6, no. 2, pp. 2933–2944, Apr. 2019.
[9] M. Malekizadeh, H. Karami, M. Karimi, A. Moshari, and M. J. Sanjari, “Short-term load forecasting using ensemble neuro-fuzzy model,” Energy, vol. 166, Apr. 2020, Art. no. 111727.
[10] W. Kong, Z. Dong, Y. Xia, D. 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.
[11] X. Qiu, P. N. Suganthan, and G. A. J. Amarathunga, “Ensemble incremental learning random vector functional link network for short-term electric load forecasting,” Knowl.-Based Syst., vol. 145, pp. 182–196, Apr. 2018.
[12] M. Felice and X. Yao, “Short-term load forecasting with neural network ensembles: A comparative study [application notes],” IEEE Comput. Intell. Mag., vol. 6, no. 4, pp. 47–56, Aug. 2015.
[13] T. Aslam and N. Arshad, “Soft load shedding: An efficient approach to manage electricity demand in a renewable rich distribution system,” in Proc. Int. Conf. Smart Cities Green Energy, 2018, pp. 101–107.
[14] S. Ali, H. Mansoor, I. Khan, and N. Arshad, “Fair allocation based soft load shedding,” in Proc. Intell. Syst. Conf., 2020, pp. 1–7.
[15] U. Amato, A. Antoniadis, I. De Feis, Y. Goude, and A. Lagache, “Forecasting high resolution electricity demand data with additive models including smooth and fragmented components,” Int. J. Forecast., vol. 37, no. 1, pp. 171–185, 2021.
[16] 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.
[17] A. Kell, A. McGough, and M. Forsahw, “Segmenting residential smart Meter data for short-term load forecasting,” in Proc. Int. Conf. Future Energy Syst., 2018, pp. 99–106.
[18] 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.
[19] H. Li, Y. Zhao, Z. Zhang, 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. Energy, 2015, pp. 1–6.
[20] R. Zhang, X. Yu, 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, 2016, pp. 1–5.
[21] 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. 779–786, Mar. 2015.
[22] Y. Chakchoukh, 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.
[23] 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.
[24] 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.
[25] U. Tayab, A. Zia, F. Yang, J. Lu, and M. Kashif, “Short-term load forecasting for microgrid energy management system using hybrid HHO-FNN model with best-base stationary wavelet packet transform,” Energy, vol. 203, Jul. 2020, Art. no. 117857.
[26] A. Yang, W. Li, and X. Yang, “Short-term electricity load forecasting based on feature selection and least squares support vector machines,” Knowl.-Based Syst., vol. 163, pp. 159–173, Jan. 2019.
[27] Z. Liao, H. Pan, X. Fan, Y. Zhang, and L. Kuang, “Multiple wavelet convolutional neural network for short-term load forecasting,” IEEE Internet Things J., vol. 8, no. 12, pp. 9730–9739, Jun. 2020.
[28] M. A. Zamee, D. Han, and D. Won, “Online hour ahead load forecasting using approximate time-delay neural network based on multiple correlation–multicollinearity analysis in IoT energy network,” IEEE Internet Things J., vol. 9, no. 14, pp. 12041–12055, Jul. 2022.
[29] 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.
[30] S. Ryu, J. Noh, and H. Kim, “Deep neural network based demand side management for short term load forecasting,” Energies, vol. 10, no. 1, p. 3, 2017.
[31] X. Dong, L. Qian, and L. Huang, “Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach,” in Proc. Int. Conf. Big Data Smart Comput., 2017, pp. 119–125.
[32] 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.
[33] 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 Trans. Smart Grid, vol. 5, no. 1, pp. 402–410, Jan. 2014.
[34] K. Gajowniczek and T. Zabkowski, “Short term electricity forecasting using individual smart meter data,” Procedia Comput. Sci., vol. 35, pp. 589–597, Sep. 2014.

[35] M. Ghofrani, M. Hassanzadeh, M. Etezadi, and M. Fadali, “Smart meter based short-term load forecasting for residential customers,” in Proc. North Amer. Power Symp., 2011, pp. 1–5.

[36] P. Anthakorala 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.

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

[38] D. Ignatiadis, G. Henri, and R. Rajagopal, “Forecasting residential monthly electricity consumption using smart meter data,” in Proc. North Amer. Power Symp., 2019, pp. 1–6.

[39] X. Luo et al., “Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy,” IEEE Trans. Ind. Informat., vol. 14, no. 11, pp. 4963–4971, Nov. 2018.

[40] E. Moscato, 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.

[41] B. Yildiz, J. I. Bilbao, J. Dore, and A. Sproul, “Household electricity load forecasting using historical smart meter data with clustering and classification techniques,” in Proc. IEEE Innov. Smart Grid Technol. Asia, 2018, pp. 873–879.

[42] G. Strang, Linear Algebra and Its Applications. Belmont, CA, USA: Thomson, Brooks/Cole, 2006.

[43] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” in Proc. Symp. Discr. Algorithms, 2007, pp. 1027–1035.

[44] P. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach. London, U.K.: Prentice-Hall, 1982.

[45] 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.

[46] “Commission for energy regulation smart metering project—Electricity customer behaviour trial, 2009–2010,” 2012. [Online]. Available: www.ucd.ie/issda/data/commissionforenergyregulationcer

[47] L. Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. 86, pp. 2579–2605, 2008.