Applied Energy 238 (2019) 1312–1326

Contents lists available at ScienceDirect

Applied Energy

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

Deep learning framework to forecast electricity demand

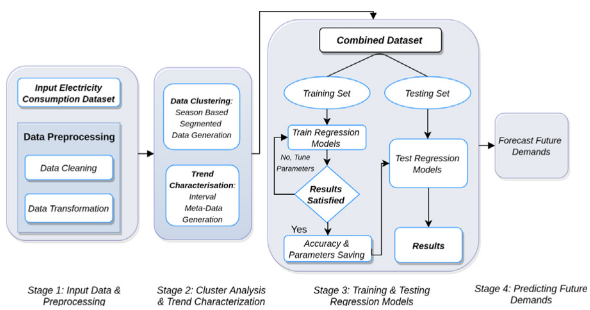
Jatin Bedi*, Durga Toshniwal

Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Uttarakhand, India

HIGHLIGHTS

- A window-based multi-input-multi-output model is proposed for demand forecasting.
- The model actively learns historical data dependencies without making any prior assumptions.
- Active learning is introduced by coupling historical data with recent demand observations.
- The proposed window-based network model outperforms other state-of-the-art prediction models.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Energy analytic
Electricity demand prediction
LSTM network
Active forecasting
Recurrent neural network

ABSTRACT

The increasing world population and availability of energy hungry smart devices are major reasons for alarmingly high electricity consumption in the current times. So far, various simulation tools, engineering and Artificial Intelligence based methods are being used to perform optimal electricity demand forecasting. While engineering methods use dynamic equations to forecast, the AI-based methods use historical data to predict future demand. However, modeling of nonlinear electricity demand patterns is still underdeveloped for robust solutions as the existing methods are useful only for handling short-term dependencies. Moreover, the existing methods are static in nature because they are purely historical data driven. In this paper, we propose a deep learning based framework to forecast electricity demand by taking care of long-term historical dependencies. Initially, the cluster analysis is performed on the electricity consumption data of all months to generate season based segmented data. Subsequently, load trend characterization is carried out to have a deeper insight of metadata falling into each of the clusters. Further, Long Short Term Memory network multi-input multi-output models are trained to forecast electricity demand based upon the season, day and interval data. In the present work, we have also incorporated the concept of moving window based active learning to improve prediction results. To demonstrate the applicability and effectiveness of the proposed approach, it is applied to the electricity consumption data of Union Territory Chandigarh, India. Performance of the proposed approach is evaluated by comparing the prediction results with Artificial Neural Network, Recurrent Neural Network and Support Vector Regression models.

1. Introduction

The growth in the population, socioeconomic and technological advancements in the past few decades has risen the demand/consumption of energy and materials to a greater extent. In particular, there is a significant increase in the total annual energy consumption of India from 418 GWh in 2005 to 874 GWh in 2013–14 and to 948 GWh in 2014–2015 [1]. It has been recorded that electricity demand has

* Corresponding author.
E-mail address: jbedi@cs.iitr.ac.in (J. Bedi).

https://doi.org/10.1016/j.apenergy.2019.01.113
Received 28 July 2018; Received in revised form 27 October 2018; Accepted 17 January 2019
Available online 29 January 2019
0306-2619/ © 2019 Elsevier Ltd. All rights reserved.

Nomenclature

x_i	input at i^{th} timestamp
y_i	output label at i^{th} timestamp
W_f, W_i	weight matrix corresponding to forget and input gate
U, V, W	weights
b	bias
ψ_n, ψ_n^*	slack variables
α_i, α_i^*, η_i, η_i^*	Lagrangian multipliers
s_t	state at timestamp t
c_t	cell state at timestamp t
h_t	hidden state at timestamp t
ECD	electricity consumption dataset
MU	million units
r	warping parameter
$TS1$, $TS2$	time series
$Input\_window$	size of the input window
Out_window	size of the output window

increased by an average of 7% every year [1]. With regard to this increased demand, sufficient amount of energy is required to satisfy the nation wide demands, while keeping care of mother nature. The utility companies are responsible for facilitating better plans and maintaining energy consumption database to improve their services continuously.

There exist various statistical [2] and traditional methods [3] that have been used to analyze trends, to characterize patterns and to forecast the energy demand. These methods can broadly be categorized into two namely [2–4], artificial intelligence and conventional methods. Conventional methods include the use of stochastic time series and regression based approaches to predict energy consumption. Stochastic time series [5] works by extending the time series patterns in future. These methods have been widely used in previous works and are capable of yielding better results while solving linear problems. With the emergence of Artificial Intelligence (AI), various learning techniques such as decision trees [6], Bayesian models [6,7], ensemble techniques [7,8] and neural network [6,8] have become very popular. These AI based methods are primarily inspired by the human brain and are very reliable due to their ability to capture non-linearity in the data. Artificial Neural Network (ANN) [6] based learning models comprise of three phases namely the learning, validation and testing phase. In the learning and validation phase, ANN is trained to generate/develop a mapping between input and output variables. Later, in the testing phase, the developed/tuned model is utilized to forecast energy demand. Due to the strong theoretical background and high generalization capabilities, Support Vector Machines (SVM) models have also been widely used in the last few decades. Several research studies have been motivated the use of SVM by comparing the prediction results of SVM with the neural network, empirical and other prediction models.

Although each of the forecasting methods (conventional and AI) have their own pros and cons, AI methods have gained the most attention due to their reliability and prediction accuracy. These AI based methods including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Ensemble methods and Multi-layer Perceptron Network (MNN) have been vastly implemented for the task of electricity load and price forecasting. MNN based techniques have good approximation capabilities, but these methods are not capable of handling historical dependencies in data. To rectify the problem associated with MNN methods, RNN and CNN based techniques have been used in the past. However, such methods fail to handle long term data dependencies. Furthermore, in recent decades, a large number of research studies were proposed to model the energy demand at region and household level [9–11], however no demand analysis and forecasting have been done to model electricity demand at the state/territory level for the country India. In this paper, we propose a deep learning based framework (D-FED) to model electricity demand at the state level while solving the issues associated with the existing methods such as handling long-term dependencies, real-time forecasting and modeling electricity demand at Union Territory (UT) level. For this purpose, we use per-day energy demand data of UT Chandigarh, India. The power requirement of UT Chandigarh is increasing at a fast pace of 52 MU (per year) [12]. Therefore, it is required to model the electricity consumption pattern and forecast demands of UT to improvise the future strategies, decision-making planning and energy performance.

The rest of this paper is organized as follows: Section 2 discusses the related work done in the field of energy analytics and Section 3 describes the research contribution of the present work. Section 4 provides a brief explanation to the theoretical models/concepts used in the present study and Section 5 explains the multi-stage methodology of the proposed D-FED framework. The application and experimental results of the D-FED framework to the dataset of UT Chandigarh are demonstrated in Section 7 concludes the paper.

2. Literature review

In the last few years, energy analytic has emerged as a significant area of research because of its prominent impact on the socioeconomic development of a country [1]. A number of research studies have been performed on consumer segmentation, profile characterization, demand pattern analysis and prediction from the data recorded by real-world sensors [10,11]. Most of the research work done in this field was on analyzing smart metering data for residential buildings. However, in this paper, we focus on characterization, active learning and long-term prediction of electricity consumption on a UT level.

Many simulation tools were developed since the year 1990s to predict energy use. These methods can be broadly classified into three namely, engineering methods [13], AI based methods [14] and hybrid methods. Engineering methods [13] were based on formulating dynamic relationship between variables. These methods are also referred as "White-box methods" as their internal logic is very clear. Yao et al. [15] introduced an approach to estimate demand patterns of the residential buildings based on several factors including space heating and appliances type. Wang et al. [16] proposed an approach based on frequency characteristics analysis to simplify the physical characteristics of the model. Engineering methods were well adapted to estimate electricity demand. However, the large computational time and high complexity of these methods make it difficult to generalize them for different type of electricity demand prediction applications.

In statistical machine learning methods, the straightforward technique is to implement linear regression [6,8] for modeling the relationship between predictors and predictant. Bauer [17] proposed a statistical approach to deal with internal and solar gains. Koksal and Ugural [18] developed a model using conditional demand analysis regression method to predict national level energy consumption. The method performed well in prediction analysis, but the poor flexibility and large input requirements of the method make this approach an unsuitable candidate for model development. Later in this year, Braun [19] proposed a framework to estimate electricity demand of a supermarket based on several climate variables such as humidity, temperature and relative humidity. Multi-linear regression model was implemented to generate a mapping between climate variables and electricity demand. From the prediction results, authors determined the variations in future demand for various categories such as gas consumption, fuels etc. Further, in order to explain the non-linear relationship between electricity demand and its influencing factors, Guo et al. [20] adopted support vector machine to forecast electricity demand. Abedinia et al. [21] introduced a novel hybrid feature selection algorithm based on maximum-relevance, minimum-redundancy, candidate interaction theory and real-coded genetic algorithm (for fine tuning parameters). The selected features were used to forecast load and price of the power systems. Statistical methods were widely utilized because of their ease of use and generalization capability. However, these methods are incapable of smoothly handling non-linearity of the data.

AI including both genetic algorithms and ANN [22] are widely used for the task of building intelligent model due to their reliability and effectiveness in handling hidden features of data [23]. Ekonomou [24] used ANN to predict the electricity requirements for a country. Multi-Layer Perceptron (MLP) model is used to forecast demand on the basis of four available factors namely: weather conditions, GDP, historical demand data and power capacity. Experimental results obtained from the data of the year 2012 and 2015 showed that results of MLP model are more accurate and promising than the linear regression model and SVM. In the year 2014, Kialashaki et al. [25] developed an ANN and Multi-Linear Regression (MLR) based approach to estimate energy demand for industrial sectors of the United States (US). Several independent factors such as GDP, energy carrier price etc. were considered for the analysis. The comparison of ANN model results with MLR showed that ANN performed better input/output mapping. In addition to this, the prediction results were validated by performing a comparative analysis with the projections of Energy Information Administration, US. Abedinia et al. [26] introduced a hybrid framework that combines Radial neural network with stochastic search to forecast short-term electricity demand. The prediction results were compared with the MLP network, wavelet transform and echo-state network to validate the approach. Furthermore, Gajowniczek et al. [27] extended the existing approaches by including the impact of resident daily activities and appliances usage to prediction results. A number of machine learning techniques such as SVM, ANN, random forest and other models (ARIMA, step-wise regression) were trained to do short-term electricity load forecasting for 24 h. The comparison of various techniques had shown that the approximation capability of ANN is very much effective at solving short-term forecasting problems.

Wan et al. [28] developed a novel DeepNet architecture to forecast short-term electricity load. Convolution Neural Network (CNN) was used to extract the features from historical sequences, which further forms the basis for demand forecasting. Shi et al. [29] proposed a deep learning approach to estimate short-term electricity load. Deep RNN were implemented to estimate demand at two different levels i.e. regional aggregate level and household dis-aggregate level. Based on the experimental results, the authors stated that RNN performed better than the shallow neural network. Further, Rahman et al. [30] presented a RNN based approach to predict electricity demand of residential and commercial buildings. The accuracy comparison was carried out between RNN and ANN to determine the performance of the proposed approach. Based on the prediction results, Jesus lago [31] in his work concluded that LSTM and GRU model does better electricity prices prediction than all other neural network based models. Chen et al. [32] introduced a two-stage ensemble strategy with Monte-Carlo dropout to enhance the generalization capability of the deep neural network based prediction model. From the test cases and comparison results, the authors stated that the proposed model have high generalization capability. Kong et al. [33] introduced a DeepNet based framework to estimate short-term electricity load from various independent variables such as residential behavior, weather conditions (temperature, humidity) and weekday. The prediction results are evident that load demand is highly dependant on the residential behavior.

Hybrid methods can be defined as the combination of different type of methods to take leverage of each type of methods. Yang et al. [34] developed an approach that utilized genetic algorithms to identify various parameters related to energy savings for buildings. The approach combined HAMbase software with evolutionary algorithms to optimize wall properties, thermal resistance, thermal capacities and radiation coefficients. The overall goal was to minimize the mean absolute testing error. Zahedi [35] proposed an Adaptive Neuro-Fuzzy Inference System (ANFIS) that combines back-propagation with least square methods to project electricity demand of the territory Ontario, Canada. Several parameters such as Gross Domestic Product (GDP), population, dwelling count and weather features were used as input to estimate future electricity demand. The experimental results of the approach signify that the electricity demand is most sensitive to the employment rate. Duan et al. [36] proposed a hybrid approach that combines max-relevance min-redundancy feature selection algorithm with optimized SVM to predict buildings load. The particle swarm algorithm is implemented for the optimization purpose. From the prediction results, the authors stated that the proposed model outperforms all other prediction models and is an accurate model for predicting building loads.

3. Research contributions

The existing methods as explained in the previous section have the following limitations: Engineering methods are good at approximations, but their large computation time and low generalization capability are a big problem. In contrast to engineering methods, statistical methods have good generalization capabilities, but they lack at capturing non-linearity of data. Moreover, their high complexities and difficulty in parameter optimization is also a major issue. AI based methods (including neural networks and Genetic algorithms) have been found to be most effective while dealing with non-linearity of data. However, they lack at handling historical dependencies in the data. In the last few years, RNN have been emerged as a significant tool to deal with non-linearity and dependencies in the data. However, RNN are incapable of handling long-term dependencies in time series due to vanishing gradient problem. So, there is a need for a predictive model that should have the following characteristics:

- The model should be able to handle the nonlinear complex behavior of electricity consumption patterns.
- The existing methods are purely external features driven. So, the predictive model should be able to forecast demand from the minimum available historic data only.
- It should be able to model both medium and long-term hidden dependencies/patterns in electricity consumption data.
- The existing demand forecasting models are based on the concept of static learning as they are purely dependant on historical data only. However, the coupling of historical data with recent observations would intuitively give the better prediction result. So, the proposed model needs to be adaptive and it should support active learning.

In this paper, we propose a LSTM based deep learning framework to address the aforementioned challenges. In LSTM network, memory cells provide support for modeling nonlinear complexities, short-term and long-term data dependencies. In contrast to existing prediction models, the proposed framework is completely historical data-driven as it does not require any additional information about data characteristics. Further, the proposed model employs a novel moving window based Multi-Input-Multi-Output (MIMO) mapping approach of active learning for improved results in prediction.

4. Necessary concepts

4.1. Support vector machines

SVM [37,38] are the most popular, robust and widely used intelligence methods developed by Vapnik and Chervonekis. These methods were initially developed for the task of classification, later they were adapted to solve various other problems such as regression.

SVM is also categorized as a type of neural network due to its many similarities with ANN. However, SVM are based on different principles. ANN works on the principle of empirical risk minimization and tries to minimize estimation error over training data. In contrast, SVM tries to improve generalization error by following the structural risk minimization principle [38].

Consider we are given a training set $\{(x_1, y_1), (x_2, y_2) \ldots\ldots(x_l, y_l)\}$ where $x_i \in \mathcal{R}^n$ with n features and $y_i \in \mathcal{R}$ denotes the dependant variables associated with each input. SVM model aims to develop a mapping $f(x)$: $\mathcal{R}^n \rightarrow \mathcal{R}$, where $f(x)$ is given as: $f(x) = W^T.\, X + b$ and describes the plane that fits closest to data. The function $f(x)$ should be such that it has at most σ deviations from targets y_i (for all i). Any deviations from the solution hyperplane smaller than σ can be neglected.

Further, in order to capture non-linear nature of data, it is necessary to apply a transformation function ϕ: $\mathcal{R}^n \rightarrow \mathcal{R}^s (s > n)$ to all points in the input space. The transformation maps non-linear input space to linear feature space with higher dimensionality. The modified mapping function is then given by:

$$f(x) = W^T.\, \phi(X) + b \tag{1}$$

Formally, SVR solution to the problem can be written as a convex optimisation problem given by:

$$\begin{aligned} &minimize\ \tfrac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\psi_n + \psi_n^*)\\ &Constraints\colon\ y_i - w.\ \phi(x_i) - b \leqslant \sigma + \psi_i\\ &\qquad w.\ \phi(x_i) + b - y_i \leqslant \sigma + \psi_i^*\\ &\qquad \psi i.\ \psi_i^* \geqslant 0\\ &\qquad (for\colon\ i = 1\ldots\ldots l) \end{aligned} \tag{2}$$

The parameter C is regularisation parameter that represents trade-off between flatness and penalty. The above problem in Eq. (2) can also be expressed in its dual form given by:

$$\begin{aligned} L = \tfrac{1}{2}||w||^2 &+ C\sum_{i=1}^{l}(\psi_n + \psi_n^*) - \sum_{i=1}^{l}(\eta_i\psi_i + \eta_i^*.\ \psi_i^*)\\ &- \sum_{i=1}^{l}\alpha_i(\sigma + \psi_i + y_i + (w.\ \ \phi(x_i)) + b)\\ &- \sum_{i=1}^{l}\alpha_i^*(\sigma + \psi_i + y_i - (w.\ \ \phi(x_i)) - b) \end{aligned} \tag{3}$$

where L is Lagrangian & $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ are lagrangian multipliers such that $\alpha_i, \alpha_i^* \geqslant 0$. By using the saddle point condition, partial derivative of L w.r.t w, b, ψ_i, ψ_i^* have to vanish for optimality.

$$\partial_b(L) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{4}$$

$$\partial_w(L) = w - \sum_{i=1}^{l}(\alpha_i^* - \alpha_i).\ \ \phi(x_i) = 0 \tag{5}$$

$$\partial_{\psi_i^*}(L) = C - \alpha_i^*\eta_i^* = 0 \tag{6}$$

By putting Eqs. (4)–(6) into Eq. (3), we can get dual optimisation problem.

$$\begin{aligned} &maximize\colon\ -\tfrac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(\phi(x_i).\ \ \phi(x_j))\\ &\qquad - \sigma\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i,j=1}^{l}y_i(\alpha_i^* - \alpha_i)\\ &Constraints\colon\ \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0\ \ \&\ \ \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \tag{7}$$

Each training point is associated with a pair of Lagrange multipliers (α_i, α_i^*). Once SVM have been trained and optimal hyper-planes have been found, only the points outside σ band have one of their α_i, α_i^* multipliers distinct to zero. These points are called support vectors.

4.2. Long short term memory network

ANNs [6,7] are widely used for the task of prediction and modeling. Self-learning and self-adapting powers make ANN a suitable choice for estimating underlying data relationship. A basic ANN architecture consists of nodes (neurons) connected via direct links. Each link is labeled with a number (weight) that shows the strength of connection between neurons. These weights get updated as learning of input data proceeds. One of the most commonly used algorithms is back propagation that works on the idea of back-propagating error from output to input until error value falls below a threshold value. Although ANN can be used to model complex relationships, they are not capable of handling historical data dependencies.

Recurrent Neural Networks (RNN) [39] are a type of ANN that takes care of dependencies among data nodes. The dependencies are handled by persisting the knowledge accumulated from subsequent timestamps with the help of network loops. These network loops are used to fed network activation from previous timestamps as input to affect prediction at current timestamps. Fig. 1 shows an example of a RNN unrolled in time domain [40]. In figure, x_t denotes input at timestamp t, s_t denotes state at timestamp t, h_t denotes output at timestamp t. The current state s_t is computed on the basis of current input x_t and previous hidden state s_{t-1}. Mathematically it can be given as:

$$\left.\begin{aligned} s_t &= f_\theta(Ux_t + W_{s(t-1)})\\ h_t &= f_\alpha(Vs_t) \end{aligned}\right\} \tag{8}$$

RNNs are very successful at handling short-term dependencies, but they are incapable of handling long-term dependencies due to vanishing gradient problem [40]. This problem was solved with the introduction of LSTM networks. LSTM networks [41] (extension of RNN) have been successfully used for sequence prediction and labeling tasks. The architecture of LSTM networks replaces conventional perceptron architecture with a memory cell and gates that regulate the flow of information across the network [42]. The gating mechanism consists of mainly three gates: input, forget and output gate. Each of the memory cells has a unit "Constant Error Carousel" to support the short-term memory shortage for a large period of time. Fig. 2 shows the structure of an LSTM memory block with one cell. In Fig. 2, c_t & c_{t-1} denotes cell states at timestamps t and t − 1. The forget gate [41] takes x_t & h_{t-1} as input to determine the information to be retained in c_{t-1} using sigmoid layer. The input gate i_t uses x_t and h_{t-1} to determine value of c_t. The output gate o_t regulates the output of LSTM cell on the basis of c_t using both sigmoid layer and tanh layer. Mathematically it can be given as:

$$\left.\begin{aligned} f_t &= \sigma(W_f.\ [x_t.\ \ h_{t-1}] + b_f)\\ i_t &= \sigma(W_i.\ [h_{t-1}.\ \ x_t] + b_i)\\ c_{t'} &= tanh(W_c.\ [h_{t-1}.\ \ x_t] + b_c)\\ c_t &= f_t \odot c_{t-1} \oplus i_t \odot c_{t'}\\ o_t &= \sigma(W_o.\ [h_{t-1}.\ \ x_t] + b_0)\\ h_t &= o_t \odot tanh(c_t) \end{aligned}\right\} \tag{9}$$

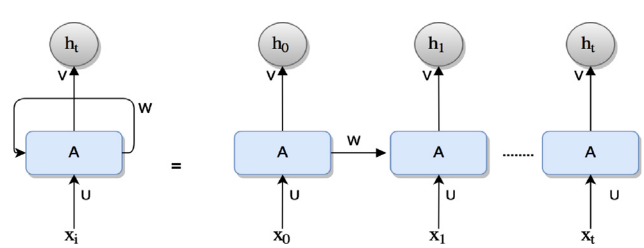


Fig. 1. Architecture of RNN network.

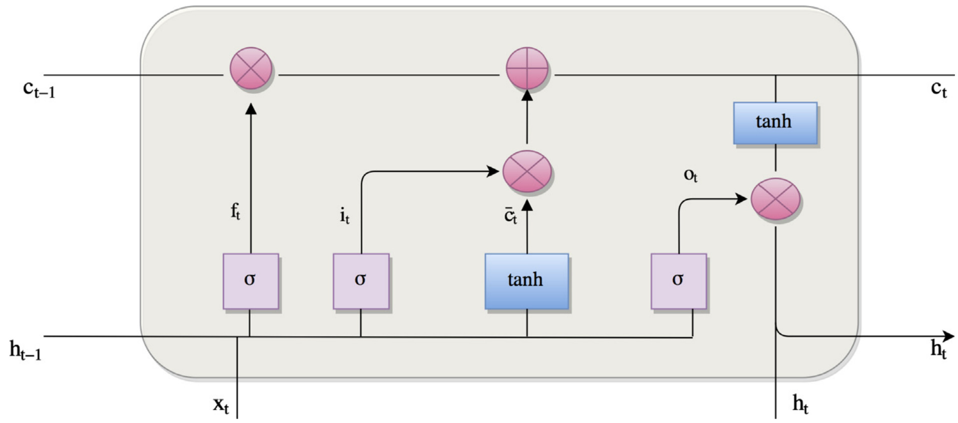


Fig. 2. Architecture of LSTM network.

5. Proposed approach

This section explains the methodology of the proposed approach for electricity demand forecasting. The multi-level architectural diagram of the proposed D-FED framework is shown in Fig. 3. The proposed approach requires historic data of electricity consumption and forecasts electricity demand for the season, day and time interval (specified by the user). Initially, the data clustering is applied on the raw data to obtain season based segmented data. Subsequently, the load trend characterization is performed to generate interval metadata for demand modeling. The moving window based MIMO mapping strategy is proposed to actively consider the recent real-time observations for predicting current timestamp demand. The proposed method is implemented to estimate load demand for the UT Chandigarh and the prediction results are compared with SVM, ANN and RNN regression models.

5.1. Data preprocessing

The dataset collected in the real world is susceptible to various discrepancies [43] including incomplete data, noise, missing values, raw format etc. These discrepancies/errors in the raw data might lead to poor data analysis. Therefore, data must be pre-processed to ensure the reliability of the process of knowledge discovery from the data. Typically, data preprocessing [43] phase involves several sub-phases which are as follows:

- *Data Cleaning*: Data Cleaning includes filling missing values, noise removal, outlier detection and resolving discrepancies within the data [7].
- *Data Transformation*: This sub-phase involves various methods such as integrating multiple files into a single usable format [7], scaling the attribute to follow specific properties etc.
- *Data Reduction*: Data reduction aims to capture most of the data properties while removing redundancies. It aims at providing a reduced representation of data either by the reducing number of attributes or by sampling.
- *Data Discretization*: Discretization involves the use of various methods such as binning (to reduce the number of values of a variable by dividing the range of attribute values into intervals), concept hierarchies to make data suitable for the task of analysis [7].

The different sub-phases can be efficiently utilized depending upon the input data formatting requirements of the approach.

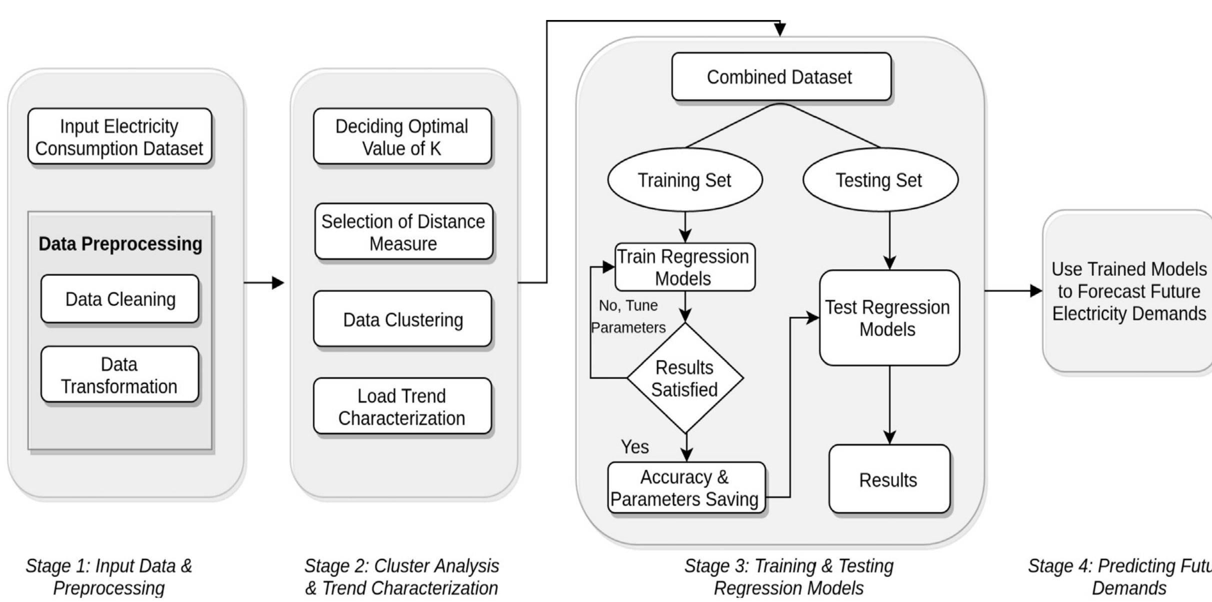


Fig. 3. Methodology of the proposed framework.

5.2. Data clustering

Cluster analysis [7,44] works by dividing data objects into groups/classes so that the objects falling to one group/cluster must be highly similar (feature based or other measure based) to all other objects in the same cluster and dissimilar to the objects in other clusters. The main aim of cluster analysis is to find similarity groups of data based on several properties. Various clustering algorithms are available for the task of cluster analysis namely, Partitioning methods [44], Density-based methods [44,7] and Hierarchical methods [7]. Each type of methods has its own way of defining groups and similarity measures. The present work implements partitioning based K-Means algorithm to assign data objects to their respective clusters.

K-Means [7] is a very simple and popular clustering algorithm. It starts by selecting k data points as initial clusters centroid, where k is a user provided parameter. Each of data object is assigned to one of the cluster centroids depending on its distance from the centroid. The data object is assigned to the cluster centroid to which its distance is minimum. Then, the centroid of each cluster is updated by computing mean of the points assigned to each individual cluster. This process goes on repeating until Sum of Squared Error (SSE) value gets minimized. The expression for SSE is defined in Eq. (10).

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} distance(C_i, x)^2 \tag{10}$$

Time series data comprises of a sequence of values measured at equal intervals of time. Due to large size, non-stationarity and high dimensionality of time series data, there are several clustering parameters such as choice of distance measure, cluster evaluation measure that needs to be decided before clustering time series data.

5.2.1. Clustering distance measures

There are various distance measures that can be used to find similarity in the time series data [7]. Some of them are as follows:

- *Euclidean Distance (ED)* [7]: ED is one of the most commonly used distance measures in the field of data clustering. However, ED is not found to be a useful measure in case of time series data because of its disability to capture distortions in the time domain. Given two series TS1 $\langle ts1_1, ts1_2, \ldots ts1_n \rangle$ and TS2 $\langle ts2_1, ts2_2, \ldots ts2_n \rangle$ each of n dimensions, ED between two is given by Eq. (11).

$$Euclidean\_distance = \sqrt{\sum_{i,j=1}^{n} (ts1_i - ts2_j)^2} \tag{11}$$

- *DTW Distance* [45]: It can be used to find an optimal non-linear alignment between two time series. Dynamic Time Warping (DTW) method works as follow: Consider two time series TS1 $\langle ts1_1, ts1_2, \ldots ts1_n \rangle$ and TS2 $\langle ts2_1, ts2_2, \ldots ts2_n \rangle$, where $ts1_i$, $ts2_i$ denotes the value of time series TS1 and TS2 at i^{th} time stamp. The first step in DTW involves constructing a n × n cost matrix whose i^{th} and j^{th} element denotes the distance between $ts1_i$ and $ts2_j$. In the second step, DTW method [46,45] tries to find an optimal path $W\langle w_1, w_2, \ldots w_k \rangle$ through the cost matrix to minimize alignment between two time series. The method goes on through recursive dynamic programming function until it founds an optimal path. DTW method has an overall complexity of O(mn) where m and n denotes the length of 1st and 2nd time series respectively.
- *LB_Keogh Distance* [47]: The high complexity of DTW distance method becomes a big problem while handling large time series data. The LB_Keogh distance measure is a modification of DTW to speed things up. LB_Keogh works by defining Lower bound and upper bound for a time series. It has an overall complexity of O(n) and it is given by:

$$LB\_Keogh(TS1, TS2, r) = = \begin{cases} TS2_i - Upper_i^2, & \text{if } TS2_i > Upper_i \\ TS2_i - Lower_i^2, & \text{if } TS2_i < Lower_i \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

$$(UpperBound): Upper_i = max(TS1_{i-r}: TS1_{i+r})$$
$$(LowerBound): Lower_i = min(TS1_{i-r}: TS1_{i+r})$$

5.2.2. Deciding the value of parameter k

K-Means is an unsupervised clustering algorithm. The goodness of clusters identified is directly related to the value of parameter k. In the past decades, numerous methods such as Elbow method [48], Silhouette Index method (SI) [49] and indices such as Gap Statistics [49], likelihood estimation were introduced to provide an accurate estimate of k. In the present work, we implement elbow curve method [48] to find an optimal k value which will produce the best clustering results. Elbow curve method works by executing iterative runs of K-Means algorithm with a range of k values and computing SSE for each k value. Finally, a plot of k-values versus SSE is generated to select optimal value of k. The basic idea of this method is to choose k value for which the error value changes abruptly in the plot.

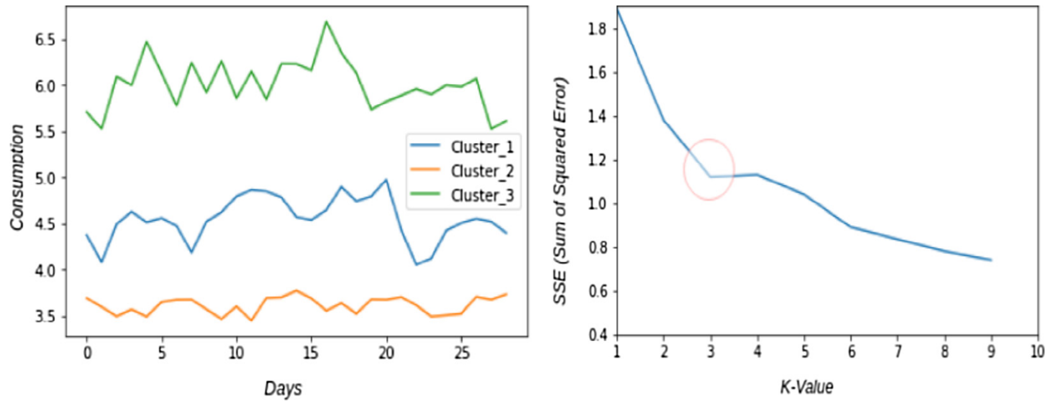
5.3. Load trend characterization

Data clustering enables us to identify the groups/months with the similar consumption patterns. Further, load trend characterization can be used to capture a more rigorous and deep understanding of the behavior of electricity consumption patterns. It enables to find sub-groups with similar demand trends which further helps in developing the machine learning based solutions to various energy patterns analytic problems.

The load trend characterization can be done at different levels of

Table 1
Distance matrix.

LB_Keogh Distance	January	February	March	April	May	June	July	August	September	October	November	December
January	0.0	1.34	1.57	3.10	3.23	2.94	1.94	1.90	2.04	1.10	0.89	0.87
February	0.97	0.0	1.34	2.20	2.33	2.42	2.97	3.1	2.89	1.37	0.92	1.02
March	1.5	0.93	0.0	2.0	2.31	2.19	2.68	2.13	2.93	1.68	1.72	1.3
April	3.10	2.09	2.03	0.0	1.28	1.04	1.61	1.90	2.14	3.18	3.59	3.29
May	3.12	2.18	2.00	0.39	0.0	0.27	0.89	1.26	1.92	1.71	1.75	2.23
June	2.14	2.11	1.99	0.93	0.89	0.0	1.83	1.90	2.19	2.93	2.63	2.58
July	2.39	2.09	2.21	1.92	1.81	1.73	0.0	0.91	1.02	1.57	1.89	2.93
August	1.82	2.05	1.94	1.14	1.69	1.06	0.33	0.0	0.61	1.99	2.87	2.37
September	1.65	1.95	2.36	1.29	1.46	1.38	0.55	0.26	0.0	1.27	1.96	2.12
October	0.93	0.95	1.03	1.79	1.94	1.62	1.37	1.40	2.60	0.0	0.70	0.77
November	0.61	0.83	1.12	1.99	2.23	2.68	2.34	3.02	3.12	1.09	0.0	0.12
December	0.23	0.97	1.11	2.73	2.64	2.13	3.08	2.79	2.0	0.93	0.16	0.0



(a) Clustering Results: Resultant Clusters Formed (b) Results of Elbow Method: To Determine Optimal K-value

Fig. 4. Clusters analysis.

Table 2
Statistical data of UT Chandigarh electricity demand.

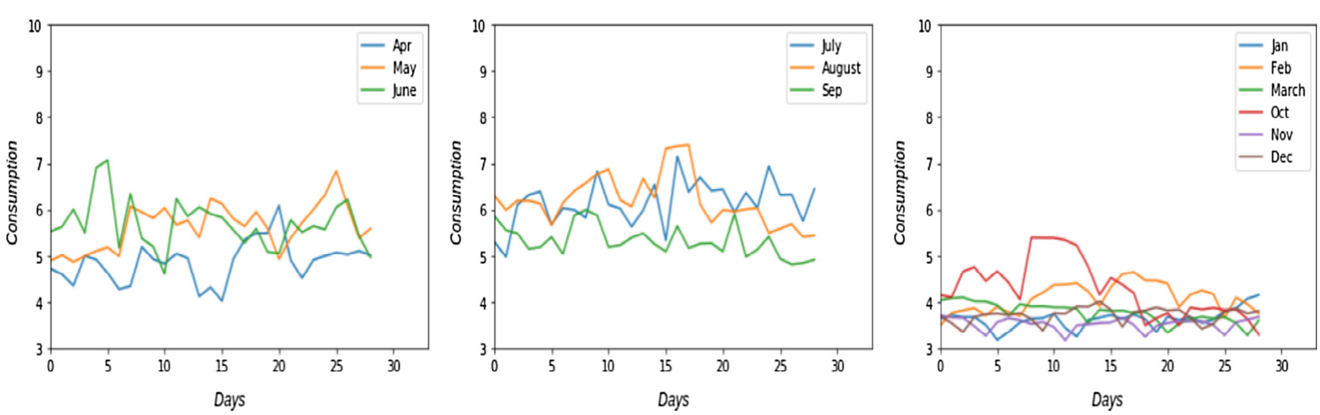
Season	Day	Average/Peak	Min demand	Max demand	Mean demand
Summer	Monday	Average	89.40	336.43	197.71
		Peak	118.72	369.68	225.09
	Wednesday	Average	101.34	318.16	219.34
		Peak	129.85	346.82	240.21
	Sunday	Average	88.25	287.17	183.76
		Peak	106.32	313.85	207.16
Rainy	Monday	Average	146.96	342.92	241.39
		Peak	166	368.76	268.88
	Wednesday	Average	143.07	338.01	242.40
		Peak	163.96	351.13	269.15
	Sunday	Average	145.70	291.96	221.46
		Peak	161.3	325.03	242.41
Winter	Monday	Average	84.36	224.84	145.17
		Peak	96.08	293.82	190.91
	Wednesday	Average	75.46	240.54	148.02
		Peak	96.66	293.33	188.77
	Sunday	Average	79.15	190.79	135.84
		Peak	91.71	252.20	180.93

granularity and it plays a significant role in system planning, maintenance, load management and marketing [2]:

- *Seasonal Analysis*: to show the variations in the electricity consumption patterns over the different seasons/clusters of a year.
- *Daily Analysis*: to demonstrate change in the patterns of power consumption over the days in a week (weekdays, weekend).
- *Periodical Analysis*: Based on the consumption patterns, the entire daytime duration can be divided into different periods (peak, off-peak, morning, Afternoon, night period) to have a better and in-depth trend analysis.
- *Total Consumption Analysis*: The overall change in electricity consumption patterns of a state over the years can be used as an indicator for future demand. It will also be helpful in making future strategies and better planning of available resources.

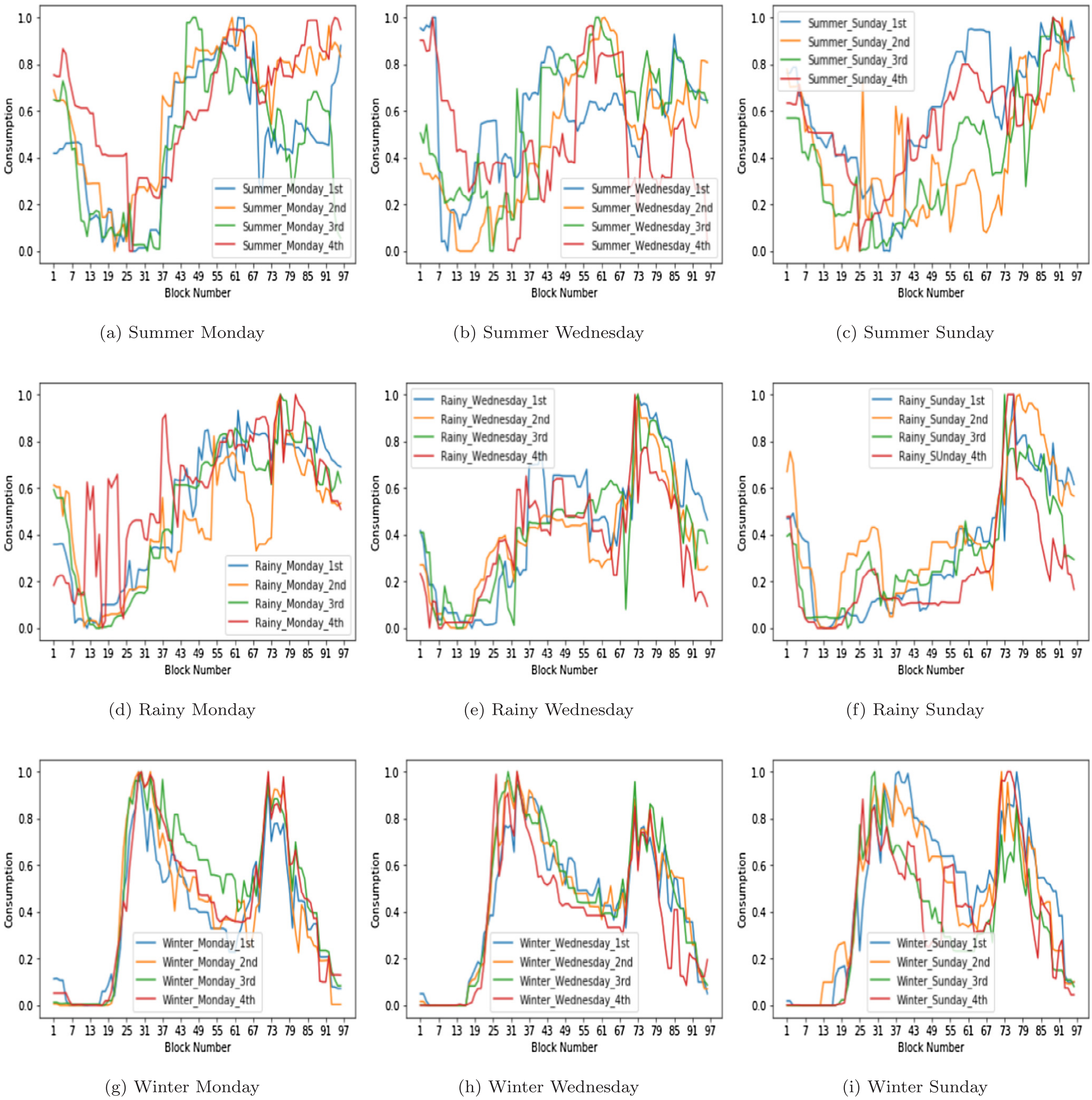
5.4. Predicting future demands using machine learning models

One of the major aims of trend analysis of consumption patterns is to make timestamp prediction of future electricity demand. After deciding on different parameters (distance measure, k, characterization level), the next step is to develop machine learning based regression models to forecast electricity demand by using historical electricity consumption data. There exist various regression models [18,19,24,27,16] that are widely used to efficiently estimate future demand due to their ability to capture complex relationship. In this study, we build four regression models namely SVM [37,38], ANN [7], RNN [39] and LSTM [41] to predict future demand of the UT Chandigarh. For the purpose of building the regression models (SVM, ANN, RNN and LSTM), electricity consumption dataset of each selected day in all three seasons is partitioned to three parts namely training, validation



(a) Cluster-1 (b) Cluster-2 (c) Cluster-3

Fig. 5. In-depth clusters analysis (months falling to each of the clusters).



(a) Summer Monday (b) Summer Wednesday (c) Summer Sunday

(d) Rainy Monday (e) Rainy Wednesday (f) Rainy Sunday

(g) Winter Monday (h) Winter Wednesday (i) Winter Sunday

Fig. 6. Trend analysis of electricity consumption patterns over three days in different seasons.

and testing set. During the training phase, all training patterns are fed into regression models to generate a mapping from input to output efficiently. The validation data is used to provide an unbiased evaluation of a model fit on training data while tuning hyper-parameters. After training and validation evaluation of regression models (SVM, ANN, RNN and LSTM), the performance of the trained model is evaluated on the testing set. There are few things that need to be carefully considered while building a prediction model:

- *Hyper-parameters selection:* There are several hyperparameters (number of input neurons (i_n), number of hidden layers (i_h), number of output neurons (i_o), window_size (W), stateful, memory_between_batches, batch_size, the trade-off between flatness & tolerance and width of ϵ) related to the regression models used in this study. These parameters should be determined accurately for building effective prediction models. The parameters i_h, i_n and i_o are determined on the basis of input data dimensions and time horizon of prediction. Several other parameters namely C, η, number of epochs, batch_size are selected by performing iterative runs of training and validation.
- *Optimization Technique and Loss Function:* Stochastic Gradient Descent (SGD) [50] is an effective and efficient optimization technique that has been widely used in the field of science and engineering. However, determining an optimal value for the step size parameter is a big difficulty while using SGD for the optimization task. The problem gets resolved with the introduction of a new optimization technique i.e. Adam (Adaptive moment estimation) algorithm [50]. Adam is the best stochastic optimizer technique for deep learning models and inherits the benefits of two popular optimization algorithms namely AdaGrad (works well with sparse

Table 3
Prediction results for Cluster-1 (summer season, RMS error).

Day	Regression model	Un-Normalized(UN)/Normalized(N)	Average demand prediction results		Peak demand prediction results	
			RMS train error	RMS test error	RMS train error	RMS test error
Mondays	SVM	N	0.13	0.177	0.128	0.163
		UN(%Error)	26.97	35.29	28.02	38.33
	ANN	N	0.123	0.17	0.13	0.17
		UN(%Error)	24.43	33.46	31.93	39.54
	RNN	N	0.12	0.16	0.13	0.15
		UN(%Error)	24.82	34.09	28.73	34.95
	LSTM	N	**0.12**	**0.14**	**0.11**	**0.13**
		UN(%Error)	**21.11 (10.67%)**	**26.02 (13.16%)**	**21.19 (9.41%)**	**26.07 (11.58%)**
Wednesdays	SVM	N	0.12	0.14	0.13	0.14
		UN(%Error)	26.10	29.41	28.73	30.08
	ANN	N	0.14	0.15	0.14	0.18
		UN(%Error)	30.70	32.93	30.84	34.88
	RNN	N	0.12	0.13	0.12	0.14
		UN(%Error)	25.13	28.98	26.87	29.47
	LSTM	N	**0.07**	**0.10**	**0.09**	**0.12**
		UN(%Error)	**14.74 (6.72%)**	**19.98 (9.10%)**	**19.57 (8.13%)**	**24.88 (10.35%)**
Sundays	SVM	N	0.141	0.19	0.138	0.189
		UN(%Error)	25.07	32.49	27.94	38.32
	ANN	N	0.13	0.16	0.12	0.16
		UN(%Error)	25.27	30.97	25.90	36.69
	RNN	N	0.12	0.17	0.11	0.15
		UN(%Error)	23.58	30.37	22.11	30.09
	LSTM	N	**0.08**	**0.11**	**0.10**	**0.11**
		UN(%Error)	**13.47 (7.33%)**	**20.19 (11.04%)**	**18.33 (8.84%)**	**23.19 (11.19%)**
		Average % Prediction Error Using LSTM	**8.24%**	**11.10%**	**8.79%**	**11.04%**

Bold values represent the performance of the proposed approach (best results).

Table 4
Prediction results for Cluster-2 (rainy season, RMS error).

Day	Regression model	Un-Normalized(UN)/Normalized(N)	Average demand prediction results		Peak demand prediction results	
			RMS train error	RMS test error	RMS train error	RMS test error
Mondays	SVM	N	0.14	0.17	0.18	0.22
		UN(%Error)	19.58	24.59	24.12	27.35
	ANN	N	0.15	0.17	0.18	0.24
		UN(%Error)	22.90	26.76	23.91	30.77
	RNN	N	0.13	0.15	0.15	0.19
		UN(%Error)	18.72	21.27	23.10	25.37
	LSTM	N	**0.10**	**0.12**	**0.10**	**0.12**
		UN(%Error)	**15.28 (6.33%)**	**17.02 (7.05%)**	**17.21 (6.40%)**	**19.16 (7.12%)**
Wednesdays	SVM	N	0.14	0.15	0.13	0.14
		UN(%Error)	25.96	27.84	27.38	28.66
	ANN	N	0.16	0.16	0.16	0.17
		UN(%Error)	29.86	30.24	30.00	31.47
	RNN	N	0.12	0.14	0.13	0.14
		UN(%Error)	23.33	25.49	26.95	27.55
	LSTM	N	**0.09**	**0.10**	**0.10**	**0.11**
		UN(%Error)	**15.90 (6.55%)**	**17.18 (7.08%)**	**17.30 (6.42%)**	**19.51 (7.24%)**
Sundays	SVM	N	0.15	0.16	0.153	0.18
		UN(%Error)	19.14	20.17	18.35	21.72
	ANN	N	0.15	0.16	0.19	0.21
		UN(%Error)	20.39	24.16	22.58	27.97
	RNN	N	0.13	0.14	0.15	0.17
		UN(%Error)	18.16	19.02	18.75	21.23
	LSTM	N	**0.09**	**0.11**	**0.15**	**0.16**
		UN(%Error)	**12.10 (5.46%)**	**15.89 (7.17%)**	**17.24 (7.54%)**	**19.77 (8.09%)**
		Average % Prediction Error Using LSTM	**6.11%**	**7.10%**	**6.78%**	**7.48%**

Bold values represent the performance of the proposed approach (best results).

Table 5
Prediction results for Cluster-3 (winter season, RMS error).

Day	Regression model	Un-Normalized(UN)/Normalized(N)	Average demand Prediction results		Peak demand Prediction results	
			RMS train error	RMS test error	RMS train error	RMS test error
Mondays	SVM	N	0.13	0.168	0.21	0.25
		UN(%Error)	14.82	18.17	21.79	23.69
	ANN	N	0.13	0.17	0.25	0.27
		UN(%Error)	15.23	18.69	24.11	28.76
	RNN	N	0.09	0.12	0.15	0.19
		UN(%Error)	11.87	13.78	19.26	20.22
	LSTM	N	**0.07**	**0.08**	**0.11**	**0.14**
		UN(%Error)	**8.55 (5.88%)**	**10.09 (6.95%)**	**14.78 (7.74%)**	**16.23 (8.50%)**
Wednesdays	SVM	N	0.11	0.13	0.10	0.13
		UN(%Error)	17.60	20.72	20.50	24.86
	ANN	N	0.13	0.14	0.12	0.15
		UN(%Error)	21.69	22.63	24.17	28.22
	RNN	N	0.09	0.10	0.09	0.12
		UN(%Error)	14.51	15.64	16.89	22.57
	LSTM	N	**0.07**	**0.09**	**0.08**	**0.10**
		UN(%Error)	**11.82 (7.98%)**	**12.45 (8.41%)**	**13.29 (7.04%)**	**19.17 (10.15%)**
Sundays	SVM	N	0.10	0.189	0.13	0.16
		UN(%Error)	10.42	19.08	17.86	24.28
	ANN	N	0.12	0.14	0.13	0.18
		UN(%Error)	13.90	16.79	19.43	28.25
	RNN	N	0.09	0.11	0.11	0.18
		UN(%Error)	10.28	12.00	14.85	26.92
	LSTM	N	**0.07**	**0.08**	**0.09**	**0.13**
		UN(%Error)	**7.98 (5.87%)**	**9.02 (6.64%)**	**10.72 (6.92%)**	**16.78 (9.27%)**
		Average % Prediction Error Using LSTM	**6.57%**	**7.33%**	**7.23%**	**9.30%**

Bold values represent the performance of the proposed approach (best results).

gradient) and RMSProp (works well for non-stationary settings). There are several benefits of using Adam for the optimization task [50]:

- The algorithm performs adaptive parameters learning from the first and second moments estimate of the gradients.
- The algorithm have a fast convergence rate as compared to various other optimization techniques such as SGD and momentum.
- The algorithm is computationally very efficient and have less memory requirements.

Furthermore, in the present work, we have used mean squared error loss function to evaluate the network performance on training and validation dataset.

- *Evaluation Measures*: There exist various performance measures that can be utilized to assess the prediction accuracy of the learning/regression models. The evaluation measures used in this study are as follows:
  - Root Mean Squared Error (RMSE) [7]: It is defined as standard deviation of the differences between predicted values and actual values. It is given as:

$$RMSE = \sqrt{\frac{1}{2n}\sum_{t=1}^{2}\sum_{j=1}^{n}(y_{j,t} - \bar{y}_{j,t})^2} \tag{13}$$

where $\bar{y}_{j,t}$, $y_{j,t}$ denotes the predicted and actual values at timestamp t respectively.

  - *Correlation Coefficient* [7]: It is a measure that estimates the strength of relationship between the predicted and real time observations of a feature. It is given as:

$$\rho = Correl(y, \bar{y}) = \frac{Covariance(y, \bar{y})}{\sigma_y.\ \sigma_{\bar{y}}} \tag{14}$$

  - *Mean Absolute Percentage Error (% Error)*: It measures the amount of deviation of the forecasted values from the actual values.

$$\%\ Error = \frac{1}{2n}\sum_{t=1}^{2}\sum_{j=1}^{n}\left|\frac{(\bar{y}_{j,t} - y_{j,t})}{y_{j,t}}\right| \tag{15}$$

6. Case study

In the present work, we use per-day (timestamp) energy demand dataset of UT Chandigarh. It is a well-planned city located near Shivalik hills range of the Himalayas in northwest India and is also the capital state of Punjab and Haryana. The city covers a land region of around 114×10^6 m^2. UT Chandigarh have no power generation units of its own and draws power from BBMB, PSTCL and PGCIL. The electricity requirement of Chandigarh is increasing at fast pace of 0.52×10^8 units per year [12]. We have the electricity demand data of UT sampled at a regular interval of 15 min for a lustrum starting from January 2013. Table 2 lists the statistical detail of various parameters (such as average, maximum, minimum and peak electricity demand for the selected day in every season) for the Electricity consumption dataset of UT. Each of the 15 min intervals is tagged with a Block number starting from Block 1 for 00:00 am to 00:15 am to Block 96 for 11:45 pm to 00:00 am.

6.1. Data preprocessing

In real life scenarios, time series data recorded by smart meters or sensors are subjected to noise and outliers. These problems in the data are rectified by data preprocessing.

- *Missing Values*: The set of missing values (for a particular interval) in the ECD of UT are filled by the average demand values in the previous two and next two intervals.
- *Data Aggregation*: The consumption data file format recorded by real-world sensors is not suitable for analysis. Therefore, data

Table 6
Prediction results (in terms of correlation coefficient (ρ)).

Season_Day	Regression model	Average demand prediction		Peak demand prediction	
		Correlation_Coefficient_Training ($\rho_{training}$)	Correlation_Coefficient_Training ($\rho_{testing}$)	Correlation_Coefficient_Training ($\rho_{training}$)	Correlation_Coefficient_Testing ($\rho_{testing}$)
Summer Mondays	SVM	0.864	0.807	0.883	0.854
	ANN	0.757	0.710	0.721	0.687
	RNN	0.868	0.856	0.858	0.849
	LSTM	**0.919**	**0.914**	**0.903**	**0.872**
Summer Wednesdays	SVM	0.832	0.739	0.767	0.723
	ANN	0.748	0.680	0.755	0.697
	RNN	0.820	0.773	0.773	0.744
	LSTM	**0.906**	**0.869**	**0.892**	**0.843**
Summer Sundays	SVM	0.816	0.762	0.789	0.762
	ANN	0.803	0.812	0.817	0.769
	RNN	0.853	0.840	0.796	0.793
	LSTM	**0.899**	**0.871**	**0.856**	**0.853**
Rainy Mondays	SVM	0.858	0.829	0.804	0.766
	ANN	0.807	0.760	0.768	0.721
	RNN	0.886	0.840	0.789	0.749
	LSTM	**0.906**	**0.876**	**0.848**	**0.792**
Rainy Wednesdays	SVM	0.818	0.786	0.790	0.745
	ANN	0.778	0.653	0.763	0.683
	RNN	0.872	0.790	0.826	0.790
	LSTM	**0.877**	**0.843**	**0.862**	**0.807**
Rainy Sundays	SVM	0.829	0.784	0.803	0.724
	ANN	0.740	0.721	0.772	0.701
	RNN	0.839	0.788	0.831	0.781
	LSTM	**0.850**	**0.848**	**0.874**	**0.793**
Winter Mondays	SVM	0.865	0.852	0.741	0.708
	ANN	0.798	0.831	0.726	0.677
	RNN	0.890	0.854	0.806	0.765
	LSTM	**0.960**	**0.941**	**0.877**	**0.848**
Winter Wednesdays	SVM	0.845	0.807	0.831	0.761
	ANN	0.791	0.792	0.798	0.705
	RNN	0.871	0.873	0.841	0.785
	LSTM	**0.912**	**0.900**	**0.864**	**0.812**
Winter Sundays	SVM	0.937	0.868	0.873	0.766
	ANN	0.871	0.852	0.809	0.729
	RNN	0.948	0.900	0.897	0.791
	LSTM	**0.960**	**0.904**	**0.928**	**0.875**

Bold values represent the performance of the proposed approach (best results).

aggregation is carried out to convert data in a format suitable for analysis and forecasting by regression models.

- *Data Transformation*: Data Transformation concerns scaling of differences between time series. The problem is overcome by applying a linear transformation to data. First, mean value is calculated for each time series. Subsequently, the mean value (window) is subtracted from each timestamp point of time series. This is also known as zero mean/unit variance transformation.

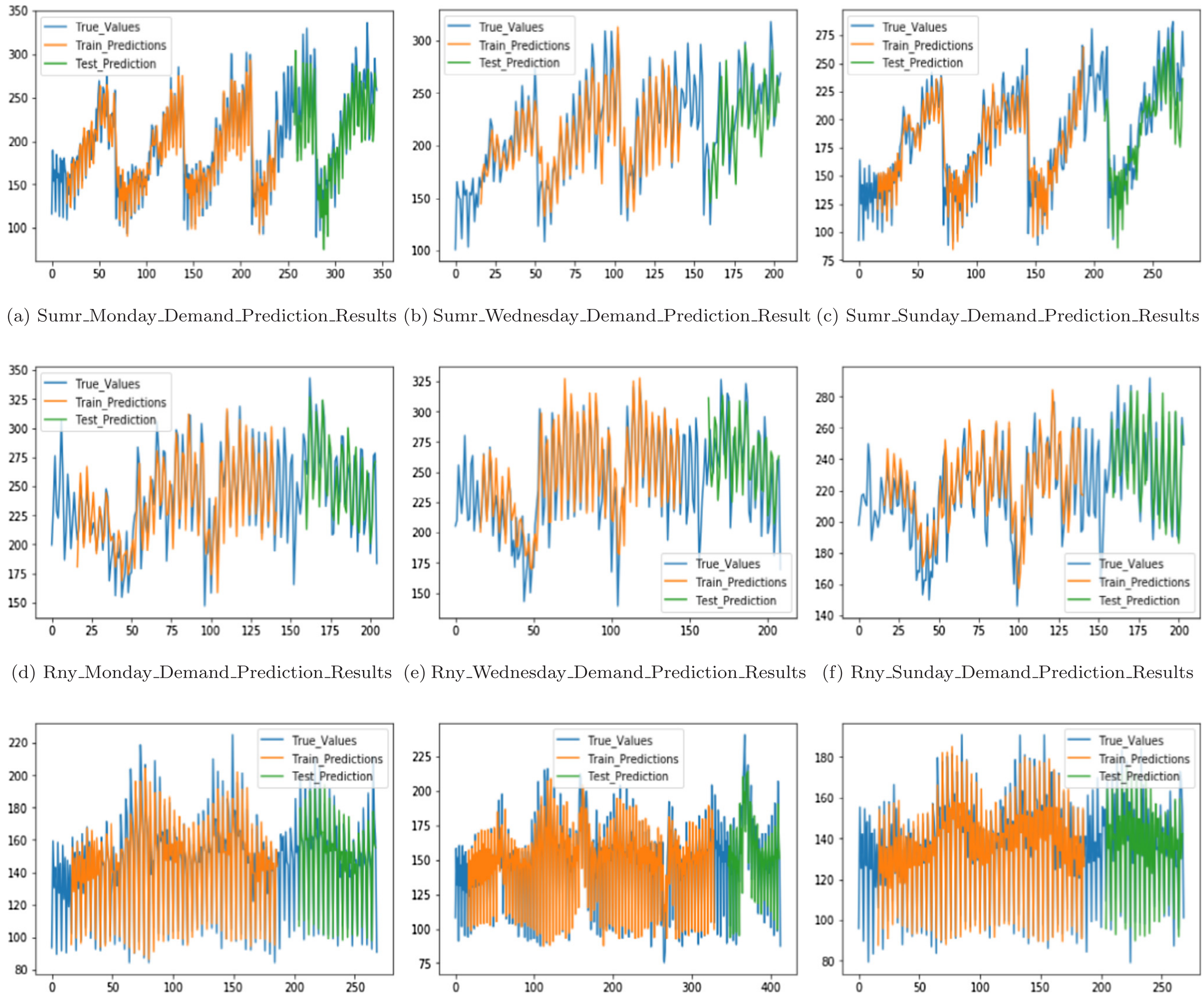
6.2. Data clustering

Cluster analysis helps to identify the natural groups present in the data. K-Means clustering algorithm is implemented on timestamp ECD of UT Chandigarh to identify the groups (consisting of months) that have similar consumption patterns. There are some parameters such as the value of k, similarity measure that need to be pre-decided in the K-Means clustering algorithm. Due to reliability and low run time complexity, the LB_Keogh [47] measure is utilized to estimate the degree of similarity among time series of all months. Table 1 shows the value of the LB_Keogh measure for time series data of different months. Further, another big challenge in implementing partitioning based K-Means for clustering is deciding the correct value of parameter k. The Elbow curve method [48] is used to find the optimal k value. The resultant curve generated from the Elbow method is shown in Fig. 4b. From Fig. 4b, it is clear that k = 3 corresponds to the optimal value of k.

The resultant clusters formed by the K-Means algorithm are shown in Fig. 4a. On the basis of clustering results, it is found that the entire consumption patterns can be divided into 3 major groups/clusters. Furthermore, Fig. 5, provides an in-depth analysis of the clustering results by listing months of the year belonging to each resultant cluster. The output three clusters basically relate to three seasons of the year namely summer, winter and rainy season.

6.3. Load trend analysis

In order to characterize the variations in the UT load patterns at a more deeper level, it is required to further analyze the patterns within the clusters. This will be of great help in efficiently and accurately forecasting demand of UT Chandigarh. In this study, load trend analysis is done at two different levels of granularity:



(a) Sumr_Monday_Demand_Prediction_Results (b) Sumr_Wednesday_Demand_Prediction_Result (c) Sumr_Sunday_Demand_Prediction_Results

(d) Rny_Monday_Demand_Prediction_Results (e) Rny_Wednesday_Demand_Prediction_Results (f) Rny_Sunday_Demand_Prediction_Results

(g) Winr_Monday_Demand_Prediction_Results (h) Winr_Wednesday_Demand_Prediction_Results (i) Winr_Sunday_Demand_Prediction_Results

Fig. 7. Average electricity consumption forecasting results (of UT Chandigarh) for three days in all seasons (*Winr: Winter, Rny: Rainy, Sumr: Summer).

- *Seasonal Characterization*: The resultant clusters formed by the partitioning based K-Means algorithm clearly demonstrate the variations in the electricity demand over three seasons.
- *Daily Characterization/Periodical Characterization*: This level of characterization aims to perform a better and deeper analysis of timestamp demand in three seasons. To do so, we select three days (*week start day* (Monday), *weekday with highest demand variations* (Wednesday), *weekend day* (Sunday)) from a week. From ECD of last five years, we randomly choose a set of four entries for each of the selected days (Monday, Wednesday, Sunday) in all three seasons (Summer, Rainy, Winter). Fig. 6 demonstrates the fluctuations in the demand of UT for each chosen set (of 4 entries) corresponding to an individual day in all three seasons. Further, for each day we divide the electricity demand into four intervals, namely, interval_1 (00:00 am to 06:00 am), interval_2 (06:00 am to 12:00 pm), interval_3 (12:00 pm to 06:00 pm) and interval_4 (06:00 pm to 00:00 am) to generate interval metadata.

The overall goal of interval/daily characterization is to enable the regression models to forecast demand for the user specified season, day and time intervals of the day.

6.4. Prediction results and discussion

The next step after deciding characterization level is to estimate future electricity demand from the available historical dependencies. In order to develop models to predict electricity demand for the time interval defined by the user, we need to apply some transformations to the data. For the given input data $X_{SD} = (x_1, x_2, ... x_n)$ of a particular season S and day D, where x_t denotes the demand at timestamp t, we substitute each point x_i with a vector X_i given by Eq. (16):

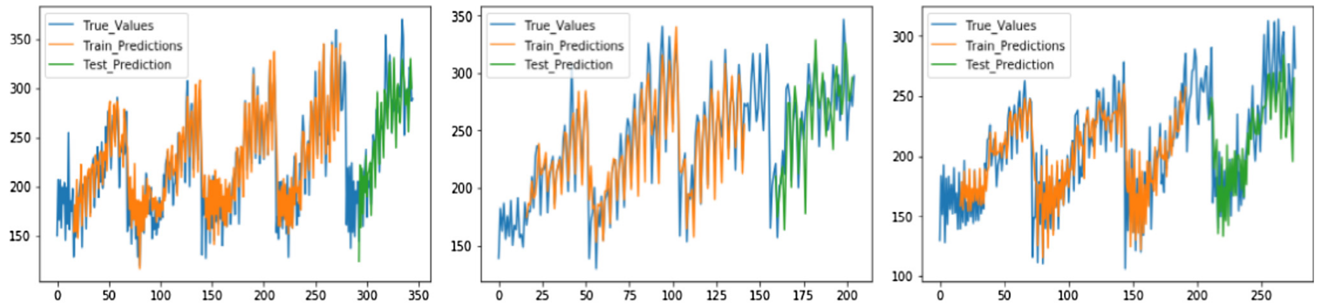
$$X_i = \langle x_{i-l}, x_{i-l-1}, ...x_{i-1}\rangle \; (for\ i = lag\_parameter\_val(l)\ to\ n) \tag{16}$$

Now, the goal is to define a function f to estimate demand at current timestamp t on the basis of previous timestamps demand. Formally, it can be given as:

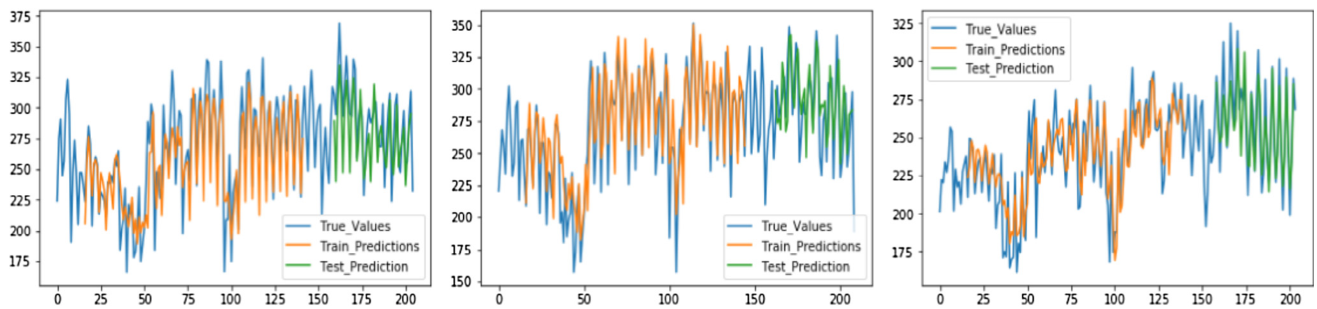
$$x_t = f(X_t) = f(x_{t-l}, x_{t-l-1}, ..., x_{t-1}) \tag{17}$$

There are three seasons (clusters identified) and we have chosen three days for a week, a total of nine data segments are available. Hence, we build 9 LSTM MIMO regression models (one for a day in all three seasons) to effectively forecast future demand from the available historical data.

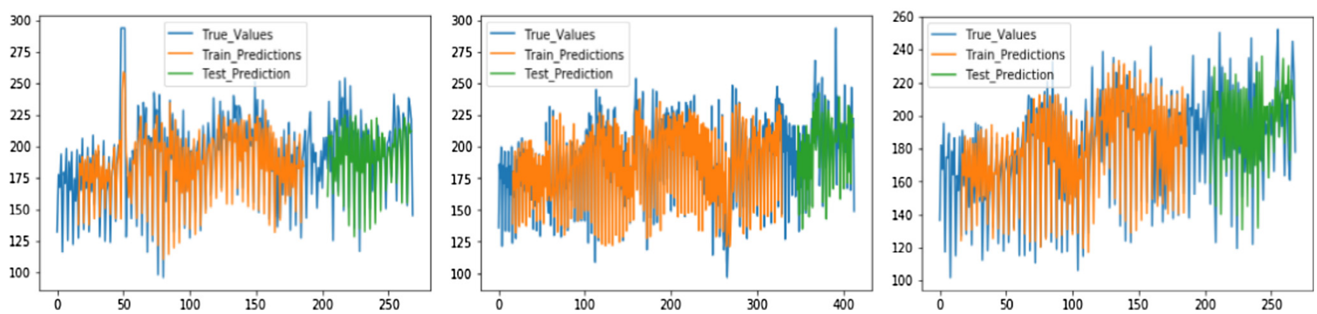
Primarily, the LSTM network expects 3-Dimensional input vectors of



(a) Sumr_Monday_Demand_Prediction_Results (b) Sumr_Wednesday_Demand_Prediction_Result (c) Sumr_Sunday_Demand_Prediction_Results



(d) Rny_Monday_Demand_Prediction_Results (e) Rny_Wednesday_Demand_Prediction_Result (f) Rny_Sunday_Demand_Prediction_Results



(g) Winr_Monday_Demand_Prediction_Results (h) Winr_Wednesday_Demand_Prediction_Result (i) Winr_Sunday_Demand_Prediction_Results

Fig. 8. Peak electricity consumption forecasting results (of UT Chandigarh) for three days in all seasons (*Winr: Winter, Rny: Rainy, Sumr: Summer).

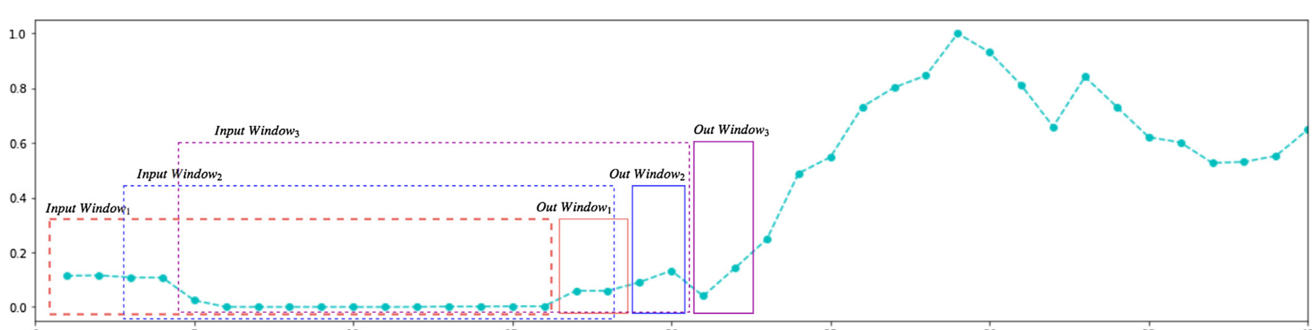


Fig. 9. An example of moving window method in LSTM network testing phase.

the form (input samples, timestamps, features), where input samples relate to one row of data, timestamp denotes different time step value of a given variable for a given observation and features refers to other measures observed at the time of observation. One of the simplest way to predict in time series using LSTM is to fed model with one features vector $X_i\ \langle x_0, x_1, \ldots, x_{i-1}\rangle$ to predict next one component. This method assumes that each timestamp value is independent of all other values in the dataset. Since this assumption is not true in the present case, we need some more efficient method to take care of data dependencies. An efficient way is to implement a moving window based approach [42]. The approach is based on a forecasting strategy that models multiple inputs and output mapping while preserving stochastic dependencies among time series events. It consists of predicting next k values $(x_{t+1}, x_{t+2} \ldots, x_{t+k})$ of a time series $(x_1, x_2 \ldots, x_n)$ composed of n observations. Formally it can be given as:

$$[x_{t+k}, \ldots, x_{t+1}] = F(x_t, \ldots, x_{t-d+1}) \tag{18}$$

where d refers to number of past values used for predicting k future events and $t \in \{d, \ldots n-k\}$. In this paper, we train LSTM network models to predict next two events (out_window) for the given current point (at timestamp t) and lagged values (d = 16: intra-correlation dependant in our case). The moving window approach works by fragmenting a time series of length n into patches of length ($Input\_window + out\_window$). So, in total there are ($n - Input\_window + out\_window$) patches. During the training process, the method goes on iterating over windows until the last window is reached. The testing of training model is done on the forecasting results of the last window. In the testing phase, we use the first window from testing data as initiation window to make prediction for next two timestamps. After each timestamp prediction, the network pops out the oldest two entries from the rear of window and appends real-time observed values to the front of the window. Fig. 9 shows the concept of sliding window MIMO mapping approach in LSTM network model. This process of sliding window goes on iterating until it reaches the end of data. In this way, LSTM network models provide support for robust and precise active learning.

In addition to MIMO model, SVM, ANN and RNN regression models are also developed to estimate future average and peak electricity demand. These models also utilize the concept of historical dependencies (as given in Eqs. (16) and (17)) to forecast current electricity demand. Tables 3–6 shows the prediction results of RNN, SVM, ANN and LSTM regression models (for the summer, rainy and winter seasons) in terms of evaluation measures described in Section 5.4. From the prediction results listed in Tables 3–6, it is evident that the LSTM network model does more reliable and accurate prediction than SVM, ANN and RNN regression models. The average percentage prediction error in LSTM network models varies from 7% to 10%. Furthermore, the visual representation of average and peak electricity demand forecasting results of LSTM-MIMO model is given in Figs. 7 and 8. The x-axis and y-axis in Fig. 7 and 8 represents the number of samples or data points and total electricity consumption of UT Chandigarh respectively. The blue[1] line represents the real experimental data while orange and green lines represent the forecasting results of the proposed LSTM model for training and testing data. The initial gap (absence) of orange and green lines in Fig. 7 and 8 signifies the use of previous lag values Eq. (16) to estimate electricity demand. Based on the prediction results demonstrated in Figs. 7 and 8, it can be concluded that the LSTM network MIMO mapping strategy is capable of handling non-linearity and fluctuations in energy consumption data. LSTM network has a number of advantages over SVM. Firstly, SVM is trained by taking several successive inputs from a suitably chosen time window while LSTM network can access inputs from current timestamps also. Secondly, LSTM provides support for active learning by incorporating real time demand observations for next events prediction while SVM regression model supports static learning. The working environment of the proposed approach is as follows:

- *Hardware Configuration*: 64 GB RAM, Xeon Processor with 48 cores, 2 TB Hard Disk.
- *Software Configuration*: Operating System-Ubuntu 14.04.

The inferencing/testing time of the approach on a given sample is: 0.39 s.

7. Conclusion

Reliable and accurate prediction of electricity demand is of great significance but has gained little interest as compared to other fields. However, in the past few years, significant advancements have been done for developing efficient and accurate electricity forecast models. Artificial Intelligence based methods have performed well and gained a lot of importance due to their effectiveness in handling non-linear problems. In this paper, we proposed a framework (D-FED) to forecast electricity demand. The approach forms its basis on Long Short Term Memory network moving window-based technique and can be used to estimate demand for the time interval specified by the user. Further, performance comparison is carried out among proposed approach (D-FED), Recurrent Neural Network, Support Vector Machines and Artificial Neural Network regression models. Based on the prediction results of Tables 3–6, it can be stated that the proposed approach (D-FED) outperforms Support Vector Machines, Artificial Neural Network and Recurrent Neural Network regression models and can be used to efficiently predict electricity demand. There are several advantages of using proposed approach for demand forecasting as compared to other existing approaches:

- The proposed framework can effectively handle nonlinear complexities, short-term and long-term dependencies of the electricity consumption time series data.
- The simulation results indicate that the proposed D-FED framework has minimum prediction errors. So, it can be used to accurately estimate demand for the season, day and time interval specified by the user.
- The proposed model is completely adaptive and provides support for active learning i.e. the moving window based MIMO strategy couples the historical data with recent real-time demand observations to predict future electricity demand.
- The developed framework can be easily generalized to estimate demand for other demographic locations as it is purely dependant on historical data only.

For future work, various nonlinear exogenous features such as climate conditions, economic variables can be investigated for trend analysis of electricity consumption patterns. Further, various optimization techniques can be designed to improve the prediction accuracy of learning models.

[1] For interpretation of color in Figs. 7 and 8, the reader is referred to the web version of this article.

References

[1] Transition in Indian Electricity Sector the energy & resource institute. http://www.teriin.org/files/transitionreport/files/downloads/Transitions-in-Indian-Electricity-Sector_Report.pdf [accessed: 2017-12-30].
[2] Tso GK, Yau KK. Predicting electricity energy consumption: a comparison of regression analysis, decision tree and neural networks. Energy 2007;32(9):1761–8.
[3] Alfares HK, Nazeeruddin M. Electric load forecasting: literature survey and classification of methods. Int J Syst Sci 2002;33(1):23–34.
[4] Taylor JW. Short-term electricity demand forecasting using double seasonal exponential smoothing. J Oper Res Soc 2003;54(8):799–805.
[5] Harvey AC. Forecasting, structural time series models and the Kalman filter. Cambridge University Press; 1990.
[6] Nasrabadi NM. Pattern recognition and machine learning. J Electron Imaging 2007;16(4):049901.
[7] Han J, Pei J, Kamber. Data mining: concepts and techniques. Elsevier; 2011.
[8] Michalski RS, Carbonell JG, Mitchell TM. Machine learning: an artificial intelligence approach. Springer Science & Business Media; 2013.
[9] Jain RK, Smith KM, Culligan PJ, Taylor JE. Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of temporal and spatial monitoring granularity on performance accuracy. Appl Energy 2014;123:168–78.
[10] Kwac J, Flora J, Rajagopal R. Household energy consumption segmentation using hourly data. IEEE Trans Smart Grid 2014;5(1):420–30.
[11] Ghofrani M, Hassanzadeh M, Etezadi-Amoli M, Fadali MS. Smart meter based short-term load forecasting for residential customers. North American power symposium (NAPS), 2011. IEEE; 2011. p. 1–5.
[12] UT Chandigarh Action Plan ministry of environment & forest. http://www.moef.gov.in/sites/default/files/Chandigarh [accessed: 2017-12-30].
[13] Fumo N. A review on the basics of building energy estimation. Renew Sustain Energy Rev 2014;31:53–60.
[14] Kumar R, Aggarwal. Energy analysis of a building using artificial neural network: a review. Energy Build 2013;65:352–8.
[15] Yao R, Steemers K. A method of formulating energy load profile for domestic

buildings in the UK. Energy Build 2005;37(6):663–71.
[16] Wang S, Xu X. Simplified building model for transient thermal performance estimation using ga-based parameter identification. Int J Therm Sci 2006;45(4):419–32.
[17] Bauer M, Scartezzini J-L. A simplified correlation method accounting for heating and cooling loads in energy-efficient buildings. Energy Build 1998;27(2):147–54.
[18] Swan LG, Ugursal VI. Modeling of end-use energy consumption in the residential sector: a review of modeling techniques. Renew Sustain Energy Rev 2009;13(8):1819–35.
[19] Braun M, Altan H, Beck S. Using regression analysis to predict the future energy consumption of a supermarket in the UK. Appl Energy 2014;130:305–13.
[20] Guo Q, Feng Y, Sun X, Zhang L. Power demand forecasting and application based on svr. Procedia Comput Sci 2017;122:269–75.
[21] Abedinia O, Amjady N, Zareipour H. A new feature selection technique for load and price forecast of electrical power systems. IEEE Trans Power Syst 2017;32(1):62–74.
[22] Zhao H-x, Magoulès F. A review on the prediction of building energy consumption. Renew Sustain Energy Rev 2012;16(6):3586–92.
[23] Hagan MT, Demuth HB, Beale MH. Neural network design vol. 20. Boston: Pws Pub.; 1996.
[24] Ekonomou L. Greek long-term energy consumption prediction using artificial neural networks. Energy 2010;35(2):512–7.
[25] Kialashaki A, Reisel JR. Development and validation of artificial neural network models of the energy demand in the industrial sector of the United States. Energy 2014;76:749–60.
[26] Abedinia O, Amjady N. Short-term load forecast of electrical power system by radial basis function neural network and new stochastic search algorithm. Int Trans Electr Energy Syst 2016;26(7):1511–25.
[27] Gajowniczek K. Electricity forecasting on the individual household level enhanced based on activity patterns. PloS One 2017;12(4):e0174098.
[28] He W. Load forecasting via deep neural networks. Procedia Comput Sci 2017;122:308–14.
[29] Shi H, Xu M, Ma Q, Zhang C, Li R, Li F. A whole system assessment of novel deep learning approach on short-term load forecasting. Energy Procedia 2017;142:2791–6.
[30] Rahman A, Srikumar V, Smith AD. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Appl Energy 2018;212:372–85.
[31] Lago J, De Ridder F, De Schutter B. Forecasting spot electricity prices: deep learning approaches and empirical comparison of traditional algorithms. Appl Energy 2018;221:386–405.
[32] Chen K, Chen K, Wang Q, He Z, Hu J, He J. Short-term load forecasting with deep residual networks. IEEE Trans Smart Grid 2018. https://doi.org/10.1109/tsg.2018.2844307. p. 1–1.
[33] Kong W, Dong ZY, Hill DJ, Luo F, Xu Y. Short-term residential load forecasting based on resident behaviour learning. IEEE Trans Power Syst 2018;33(1):1087–8.
[34] De Wit M. Heat air and moisture model for building and systems evaluation. Eindhoven, The Netherlands: Technische Universiteit; 2006. p. 1–109.
[35] Zahedi G, Azizi S, Bahadori A, Elkamel A, Alwi SRW. Electricity demand estimation using an adaptive neuro-fuzzy network: a case study from the Ontario Province–Canada. Energy 2013;49:323–8.
[36] Duan M, Darvishan A, Mohammaditab R, Wakil K, Abedinia O. A novel hybrid prediction model for aggregated loads of buildings by considering the electric vehicles. Sustain Cities Soc 2018;41:205–19.
[37] Basak D, Pal S, Patranabis DC. Support vector regression. Neural Inform Process-Lett Rev 2007;11(10):203–24.
[38] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14(3):199–222.
[39] Mandic DP, Chambers JA. Recurrent neural networks for prediction: learning algorithms, architectures and stability. Wiley Online Library; 2001.
[40] Bandara K, Bergmeir C, Smyl S. Forecasting across time series databases using long short-term memory networks on groups of similar series. arXiv preprint arXiv:1710.03222.
[41] Gers F. Long short-term memory in recurrent neural networks [Unpublished PhD dissertation]. Lausanne, Switzerland: Ecole Polytechnique Fédérale de Lausanne.
[42] JAKOB AUNGIERS. http://www.jakob-aungiers.com/articles/a/LSTM-Neural-Network-for-Time-Series-Prediction [accessed: 2018-01-15].
[43] Garcia S, Luengo J, Herrera F. Data preprocessing in data mining. Springer; 2016.
[44] Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, et al. Top 10 algorithms in data mining. Knowl Inform Syst 2008;14(1):1–37.
[45] Keogh E. Efficiently finding arbitrarily scaled patterns in massive time series databases. European conference on principles of data mining and knowledge discovery. Springer; 2003. p. 253–65.
[46] Müller M. Dynamic time warping. Inform Retrieval Music Motion 2007:69–84.
[47] Keogh E, Wei L, Xi X, Lee S-H, Vlachos M. Lb_keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. Proceedings of the 32nd international conference on very large data bases. VLDB Endowment; 2006. p. 882–93.
[48] Robert Gove Block. https://bl.ocks.org/rpgove/0060ff3b656618e9136b [accessed: 2016-01-15].
[49] Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. J Roy Stat Soc: Ser B (Stat Methodol) 2001;63(2):411–23.
[50] Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980.