Renewable and Sustainable Energy Reviews 137 (2021) 110591

Contents lists available at ScienceDirect

Renewable and Sustainable Energy Reviews

journal homepage: http://www.elsevier.com/locate/rser

A deep learning framework for building energy consumption forecast

Nivethitha Somu [a,*], Gauthama Raman M R [b], Krithi Ramamritham [a]

[a] Smart Energy Informatics Lab (SEIL), Department of Computer Science and Engineering, Indian Institute of Technology-Bombay, Mumbai, 400076, Maharashtra, India
[b] iTrust-Centre for Research in Cyber Security, Singapore University of Technology and Design, Singapore

ARTICLE INFO

Keywords:
Energy consumption
Deep learning
Buildings
Clustering
Convolutional neural network
Long short term memory

ABSTRACT

Increasing global building energy demand, with the related economic and environmental impact, upsurges the need for the design of reliable energy demand forecast models. This work presents *k*CNN-LSTM, a deep learning framework that operates on the energy consumption data recorded at predefined intervals to provide accurate building energy consumption forecasts. *k*CNN-LSTM employs (i) ***k*** – means clustering – to perform cluster analysis to understand the energy consumption pattern/trend; (ii) Convolutional Neural Networks (CNN) – to extract complex features with non-linear interactions that affect energy consumption; and (iii) Long Short Term Memory (LSTM) neural networks – to handle long-term dependencies through modeling temporal information in the time series data. The efficiency and applicability of *k*CNN-LSTM were demonstrated using a real time building energy consumption data acquired from a four-storeyed building in IIT-Bombay, India. The performance of *k*CNN-LSTM was compared with the *k*-means variant of the state-of-the-art energy demand forecast models in terms of well-known quality metrics. It is also observed that the accurate energy demand forecast provided by *k*CNN-LSTM due to its ability to learn the spatio-temporal dependencies in the energy consumption data makes it a suitable deep learning model for energy consumption forecast problems.

1. Introduction

1.1. Energy, buildings, and environment

A nation's total energy demand can be estimated by aggregating three main economic sectors: buildings, industry, and transport [1,2]. According to World Watch Institute data, buildings as the largest energy consumer, accounts for 40% of the global annual energy consumption and 36% of the total carbon emissions, especially in urban areas [3,4]. The growth rate of building energy consumption in Organisation for Economic Co-operation and Development (OECD) and non-OECD nations for 2012 and 2040 is 1.5% and 2.1% per year, respectively [5]. Specifically, the total energy consumption of higher education institutional and commercial buildings are 45% and 30% higher than the residential buildings [6]. The proper use of energy through the implementation of appropriate energy management systems and end-user energy efficiency control strategies [7] provides lower operational costs by reducing energy use and avoiding penalties imposed by the utilities [8–10]. In this regard, several nations have augmented the implementation of energy regulations and codes for buildings, such that new energy-efficient building designs ensure reduced energy consumption, energy costs, and environmental impact (carbon emissions). According to the World Energy Council, "While overall per capita energy demand would begin to fall, demand for electricity would double by 2060", which necessitates the need for larger investments in smart infrastructures that promote energy efficiency [11]. From the smart grid perspective, buildings have become more intelligent with the integration of advanced information and communication technologies, electric vehicles, decentralized generation and storage systems, and energy management systems. Therefore, the design and implementation of smart technologies for power grids and buildings to meet the global energy demand in an effective and economically sustainable way with reduced carbon emissions have become extremely important.

1.2. Energy consumption forecast

Recently, research on energy consumption forecast in buildings has become increasingly significant as buildings are equipped with smart meters to monitor energy consumption of buildings at fine-grained intervals. Massive and high dimensional energy consumption data from the smart meters collected at different granularities helps in understanding the energy consumption patterns for its application to energy demand forecasting, demand response, heating ventilation and air

* Corresponding author.
E-mail addresses: nivethithasomu@iitb.ac.in (N. Somu), gauthamaraman_mr@live.com (G. Raman M R), krithi@cse.iitb.ac.in (K. Ramamritham).

https://doi.org/10.1016/j.rser.2020.110591
Received 4 April 2020; Received in revised form 30 October 2020; Accepted 17 November 2020
Available online 14 December 2020
1364-0321/

List of abbreviations

Terms	Explanation
k	CNN-LSTM k-means clustering based convolutional neural networks and long short term memory
CNN	Convolutional neural networks
LSTM	Long short term memory
KReSIT	Kanwal Rekhi School of Information Technology
IIT	Indian Institute of technology
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MSE	Mean squared error
RMSE	Root mean squared error
OECD	Organization for economic co-operation and development
HVAC	Heating ventilation and air conditioning
BEMS	Building energy management system
ISCOA-LSTM	Improved version of sine cosine optimization algorithm based LSTM
$E_t = \{e_t^1, e_t^2, \ldots, e_t^i, \ldots, e_t^n\}$	Energy consumed at different timestamps
MIMO	Multi-input multi-output
$Input_L$	Input window
$Output_L$	Output window
$if = (S_n - Input_L - a)/Output_L$	Total number of input and forecast windows
S_n	Total number of samples
a	Forecast interval
S_{Input}	and S_{Output} Input and the output window
$Input_L$	and $Output_L$ Size of the input and output window
SSE	Sum of squared error
LSTM	Long short term memory
RNN	Recurrent neural networks
i_t	Input gate
o_t	Output gate
f_t	Forget gate
$W_f, W_c, W_i, \text{and} W_o$	Weight matrices
$b_f, b_i, b_c, \text{and} b_o$	Bias vector
σ	Logistic sigmoid function
x_t	Input
c_t	Cell state
$\tilde{c}_t$	New candidate cell state
C_{ij}^n	Output of the convolution operations in the convolutional layer
ck_n	Convolution kernel matrix
fm_{ij}	Convolution filter matrix
n	n^{th} feature map
i	and j Number of steps of convolution filter in horizontal and vertical directions
b_n	and b_{fc} Bias
δ	Activation function
P_{ij}^n	Output of the pooling operation in the pooling layer
pfm_{ij}	Pooling filter matrix
W_{fc}	Weight matrix of the fully connected layer
FC	Output matrix of the fully connected layer
x	Input to the fully connected layer
$E = \{T_1, T_2, \ldots, T_M\}$	Energy consumption dataset
$C = \{C_1, C_2, \ldots, C_k\}$	Clusters
T_1	and T_2 Time series data
n	Length of T_1
m	Length of T_2
E_{Train}	Train dataset
E_{Val}	Validation dataset
E_{Test}	Test dataset
MQTT	Message queuing telemetry transport
x	Energy consumption value
$avg(x)$	Average value of x
$std(x)$	Standard deviation of x
$\min(x)$	and $\max(x)$ Minimum and maximum values of x
n	Total number of data points in the dataset
y_i	and $\hat{y}_i$ Actual and the forecasted energy consumption value at timestamp i

conditioning (HVAC) optimization, and fault diagnosis and detection. Accurate and reliable energy demand forecasts enable the utilities to plan resources and balance supply-demand, thereby ensuring stability and security of the power grid & reliability of service provisions [12]. Overestimation and underestimation of the energy demand lead to a severe impact on the economic and industrial developments [13]. Accurate modeling and prediction of energy demand help in efficient energy management in smart buildings, accurate demand response strategies, electricity supply management, and context aware control strategies [14]. Therefore, energy consumption forecast models have become an integral part of Building Energy Management System (BEMS) to improve the buildings' energy efficiency for a sustainable economy through a conservation-minded society, reasonable use of available energy resources, and efficient national energy strategy [15]. However, the non-linear, dynamic, and complex nature of energy consumption data, along with the presence of trend, seasonal & irregular patterns, and dependence on various exogenous factors like climatic conditions, nature of the day, socio-economic factors, etc. presents accurate and reliable energy consumption forecast as an interesting research problem.

1.3. Machine learning based energy forecasting

Energy consumption forecast models in the literature fall into three categories: (i) **Engineering methods:** Uses physical and thermodynamic laws & require complex building and environmental parameters; Difficult and time-consuming; Examples-EnergyPlus, Ecotect, etc [2]. (ii) **Statistical methods:** Correlates energy consumption with relevant factors like climate data, occupancy, etc.; Lacks accuracy and flexibility; Examples-Time series (autoregressive models) [14] and regression models (linear regression) [16,17], and (iii) **Artificial intelligence methods:** Learns the consumption patterns from the historical energy consumption data, i.e., discover the non-linear relationship between the input (historical data) and output (target consumption) [18,19]; Examples-artificial neural network [20], support vector regression [4], etc. Among these, artificial intelligence approaches have become 'active research hotspot' due to their efficiency and flexibility over engineering and statistical methods (Table 1) [21,22].

From Table 1, it is evident that ANN and its variants have been widely explored and applied for energy consumption forecast since they are non-linear, self-adaptable, and can approximate any function, given

Table 1
Related works.

Authors	Techniques	Dataset	Evaluation metrics
Muhammad Fayaz and DoHyeun Kim, 2018 [23]	• Deep ELM • Adaptive neuro-fuzzy inference system • ANN	4 residential buildings, Seoul, South Korea	MAE, RMSE, and MAPE
Sehrish Malik and DoHyeun Kim, 2018 [24]	• ANN • PSO	4 residential buildings, Seoul, South Korea	Number of particles used, prediction accuracy, and number of epochs
Jatin Bedi and Durga Toshniwal, 2018 [25]	• K-means • Empirical mode decomposition • LSTM	Electricity load data-Chandigarh, India	RMSE and MAPE
Chengdong Li et al., 2018 [26]	• DBN • Contrastive divergence	Retail store and office building, Fremont, CA, USA.	MAE, RMSE, and mean relative error
Jihoon Moon et al., 2018 [27]	• PCA • ANN • SVR	Private university in Seoul, South Korea	MAE, MAPE, and RMSE
Samir Touzani et al., 2018 [28]	Gradient boosting machine	Northern and Central California, Washington, D. C. and Seattle	MSE and RMSE
K.P. Amber et al., 2018 [6]	• Multiple regression • Genetic programming • ANN • DNN • SVM	South Bank Technopark, London, U·K	RMSE, MAE, MRE, MAPE, and normalized RMSE
Wei Wang et al., 2019 [3]	• LSTM • ANN • SVR	Southeast University, Nanjing, China	MAPE and RMSE
Aowabin Rahman et al., 2018 [29]	LSTM-based deep RNN	Public Safety Building, Salt Lake City, U.S. state of Utah and Residential buildings, Austin, Texas, U.S.A	RMSE and Pearson coefficient
Zhaoyang Ye and Moon Keun Kim, 2018 [30]	Levenberg–Marquardt BPNN	Shopping mall, Dalian, China	RMSE
Tianhao Yuanet al., 2018 [31]	• Sample data selection method- grey correlation method + entropy weight • BPNN • Multiple linear regression	Case study: Commercial and residential building, Tianjin, China	Coefficient of determination, absolute percentage error, MAPE, and RMSE
Federico Divina et al., 2018 [32]	Ensemble: Base learning model-Regression trees based on evolutionary algorithms, ANN, and Random forest; Top learning model: Gradient boosting machine	Energy consumption data, Spain	MRE, MAE, symmetric MAPE, RMSE, and R^2
Tanveer Ahmad and Huanxin Chen, 2018 [33]	• Compact decision tree • Fit k-nearest classifier • Simple and stepwise linear regression model	Office building, Beijing, China	MAE, RMSE, and MAPE
Shidrokh Goudarzi et al., 2019 [34]	• ARIMA • SVR • PSO • False nearest neighbours • K-means	Library building, Universiti Teknologi Malaysia	MAE, RMSE, and average relative error

Table 1 (*continued*)

Authors	Techniques	Dataset	Evaluation metrics
Jatin Bedi and Durga Toshniwal, 2019 [35]	• LSTM • MIMO	Electricity load data-Chandigarh, India	RMSE, correlation coefficient, and MAPE
Yunsun Kim et al., 2019 [36]	• ARIMA • ARIMA-GARCH • Holt–Winters' double multiple seasonal exponential smoothing • ANN	Institutional building, Chungang University, Seoul, Korea.	RMSE and MAPE
Hanane Dagdougui et al., 2019 [37]	• ANN • Learning algorithm: Bayesian regularization and Levenberg-Marquardt	Case study: 5 buildings (residential, commercial, and office), Urban educational district, Montreal	RMSE and MAPE
Dac-Khuong Bui et al., 2020 [2]	• Electromagnetism based firefly algorithm • ANN	Single family house-Istanbul, Turkey and twelve buildings simulated by Ecotect	Linear correlation coefficient, R^2, RMSE, MAE, and MAPE
Lulu Wen et al., 2020 [38]	Deep RNN with gated recurrent unit	Residential building, Austin, Texas, U.S.A	RMSE, MAE, Pearson correlation coefficient, and MAPE
Duc-Hoc Tran et al., 2020 [19]	Evolutionary neural machine inference model - Least squares SVR + RBF neural network	Residential building, Ho Chi Minh City, Viet Nam	Coefficient of determination, MAE, RMSE, and MAPE

the sufficient number of hidden layers and nodes in the hidden layer [39]. However, the inability to handle historical data dependencies and several factors like parameter initialization, trap at local minima, slow convergence, and scalability of network architecture provoked the researchers to actively work on these issues or explore deep learning approaches for building energy consumption forecast. Further, a considerable amount of information collected over various sensors deployed in buildings has transformed energy forecasting research into a "big data" research problem [40]. Interestingly, recent advancements in deep learning theory have resulted in efficient tools to handle massive and high dimensional energy consumption data, which can outperform traditional machine learning tools [41]. In this way, this work presents *k*CNN-LSTM, a deep learning framework which employs $k-$ means clustering, CNN, and LSTM for trend analysis, energy-related feature identification, and modeling long term dependencies in the energy consumption data, respectively.

1.4. Research gaps, novelty, and contributions

From the brief literature survey on the building energy consumption forecast models designed for residential, academic, and commercial buildings, the following *research gaps* were identified.

1. Most of the energy consumption data used in the literature are multi-featured (occupancy, temperature, humidity, building schedule, etc.) and data-rich (massive and high dimensional data). The research challenge in building energy demand forecasting is to achieve high forecast accuracy using the energy consumption data recorded at each timestamp.
2. The energy consumption in an academic building does not reveal the trend and seasonality during the time series analysis. It might be due

to several factors like high dynamics in schedules, occupancy related status, operations, etc.

3. Moreover, the existing energy forecasting models follow static learning, where its performance is entirely dependent only on the historical energy consumption data. However, the inclusion of recent observation with the historical data with the help of a sliding window approach would result in better forecast accuracy.
4. Further, the existing research works on LSTM based building energy consumption forecast model operates over static data (benchmark datasets) instead of real time building operation data obtained from the BEMS.

What makes *k*CNN-LSTM different from the existing building energy consumption models? The distinctive features of *k*CNN-LSTM are.

1. **Feature generation from timestamp:** The considered building energy consumption data is a *nxm* dimensional data, where *n* denotes the rows (energy consumption records) and *m* represents the columns ($m = 2$; timestamp in dd-mm-yyyy HH: MM: SS format and energy consumption). As a data preprocessing step, seven features (day of the year, season, month, day of the week, hour of the day, minute of the hour, type of the day) were generated from the timestamp, which enables the learning model to gain better insight on the trend and seasonality of the energy consumption data.
2. **Clustering algorithms:** An academic building's energy consumption is quite complex and dynamic, which does not exhibit an apparent trend and seasonality in the initial analysis. In such cases, the application of clustering algorithms to the energy consumption data before data modeling provides better insights into the trend and seasonal characterization of the data through the generation of clusters.
3. **Multi-input and multi-output sliding window:** The application of multi-input and multi-output sliding window to *k*CNN-LSTM provides robust and reliable forecasting by moving through the window of historical and recent energy consumption observations.
4. **Static or live data:** *k*CNN-LSTM has been implemented as an energy consumption forecast model in the BEMS designed for Kanwal Rekhi School of Information and Technology (KReSIT), IIT-Bombay, India.

The key *contributions* of this paper are.

1. *k*CNN-LSTM, a deep learning framework, is presented to provide reliable and accurate building energy consumption forecasts.
2. As a data preprocessing step, seven timestamp based features were generated to enrich the energy consumption data recorded at regular time intervals as parameter rich data.
3. The complex trend and seasonality in the energy consumption data are analyzed using the $k-$ means clustering algorithm that uses the LB-Keough distance metric to identify the similarity between the time series (energy consumption data) of different months in the considered annual energy consumption data.
4. The spatio-temporal dependencies in the energy consumption data are learned and modeled by the convolutional neural networks and long short term memory neural networks, respectively.
5. Multi-input and multi-input sliding window mechanism is deployed to provide accurate and reliable energy consumption forecast.
6. Further, effective modeling of higher order and non-linear dependencies in the energy consumption data enables *k*CNN-LSTM to provide accurate forecasts in real time for a long period without retraining.
7. The effectiveness of kCNN-LSTM for reliable building energy consumption forecast is validated through a case study using the real time building operational data acquired from the BEMS deployed at KReSIT, IIT-Bombay. Further, the performance of KCNN-LSTM over the $k-$ means variant of existing energy demand forecast models was assessed in terms of MAE, MSE, MAPE, and RMSE for the considered year, weekdays, and weekend.

1.5. Organization

The paper follows the following structure. Section 2 introduces the basic principles of $k-$ means, energy consumption forecasts, LSTM neural networks, and convolutional neural networks. Section 3 presents a brief description of the formulation of the energy consumption forecast problem and the proposed deep learning framework to forecast the energy consumption of the buildings. Section 4 provides a detailed analysis of the performance of *k*CNN-LSTM and state-of-the-art energy demand forecast models in terms of MAE, MSE, MAPE, and RMSE. Section 5 concludes the paper.

2. Preliminaries

This section presents a clear view of the formulation of energy consumption forecast problem, $k-$ means clustering, long short term memory, and convolutional neural networks.

2.1. Energy consumption forecast problem

A multi-featured building energy demand forecasting problem can be defined as the energy consumed by several components (e.g., plugs, lights, fans, air conditioners, servers, computers, etc.) and environmental factors (humidity, temperature, etc.) which are monitored and recorded by the sensors (e.g., smart meters, temperature sensors, etc.) installed at various levels of the considered building. The energy consumed at timestamp *t* can be represented as in Eqn. (1).

$$E_t = \{e_t^1, e_t^2, \dots, e_t^i, \dots, e_t^n\} \tag{1}$$

where, e_t^i is the energy consumption logged by the i^{th} sensor at timestamp *t*. This work uses Multi-Input and Multi-Output (MIMO) sliding window approach for energy consumption forecast to achieve better forecast accuracy. Therefore, let $\{Input_L, Output_L\} \in N$ represents the size of the input and output window. Further, the total number of input and output window can be defined as $if = (S_n - Input_L - a)/Output_L$, where S_n and a denotes the number of samples and output intervals, respectively.

Input window (S_{Input}) of size ($Input_L$) and output window (S_{Output}) of size ($Output_L$) are represented as in Eqn. (2) and Eqn. (3) (Fig. 1).

$$S_{Input} = \{E_t, E_{t+1}, \dots, E_{t+if}\} \tag{2}$$

$$S_{Output} = \left\{\tilde{E}_t, \tilde{E}_{t+1}, \dots, \tilde{E}_{t+if}\right\} \tag{3}$$

In this work, CNN-LSTM is modeled as an energy forecaster and therein an approximation function (f) which relates S_{Input} (Eqn. (2)) and S_{Output} is defined in Eqn. (3).

$$S_{Output} = f\left(S_{Input}\right), f : P^{Input_L xn} \rightarrow P^{Output_L xn} \tag{4}$$

Eqn. (4) states that given an input window ($Input_L$), the model (f) learns to forecast the energy consumption values of the output window ($Output_L$) with minimal forecast error ($Error$) as defined in Eqn. (5).

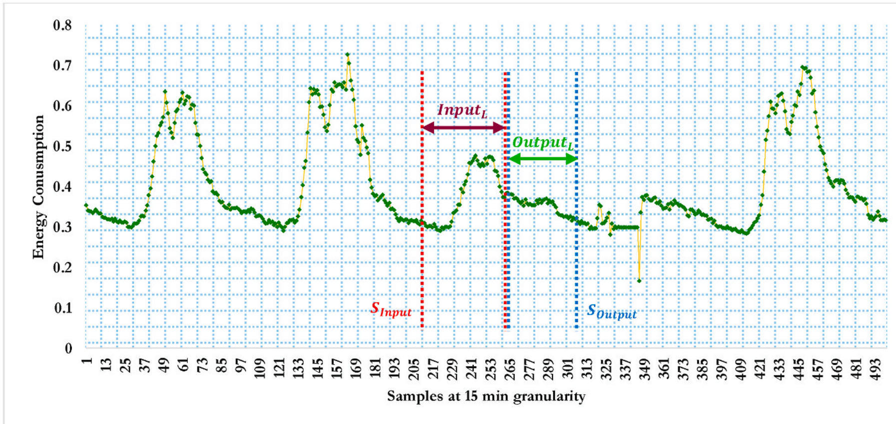


Fig. 1. Energy consumption forecast – Input and output window size.

$$Error = \frac{1}{n}\sum_{i=1}^{n}\left|e_t^i - \tilde{e}_t^i\right| \tag{5}$$

where, e_t^i and $\tilde{e}_t^t$ are the real and forecasted energy consumption value of the i^{th} sensor at t^{th} timestamp.

2.2. *k-means clustering*

$k-$ means clustering algorithm is a widely used partitioning cluster analysis method. It is an iterative hill-climbing algorithm that groups m data points into k (user-defined parameter) clusters to optimize the Sum of Squared Error (SSE) that measures the intra-cluster similarity or inter-cluster dissimilarity. Each data point is assigned to one of the cluster centroids (initialized randomly) based on its minimum distance from the centroid. Next, each cluster centroid is updated by obtaining the mean of the data points presented in the individual clusters. This procedure is repeated until SSE (Eqn. (6)) between cluster centroids and the data points is minimized.

$$SSE = \sum_{m=1}^{\mathscr{K}}\sum_{\mathscr{DP}_i \in \mathbb{C}_m}\mathscr{DP}_i - \mathbb{C}_m \tag{6}$$

The $k-$ means algorithm can be applied to massive, high-dimensional, and non-linear time-series data (sequential values measured at equal time intervals) to obtain consistent individual groups of time-series for accurate predictions. In this case, clustering parameters like distance measure, cluster evaluation measure, etc. should be decided before clustering.

2.3. *Long short term memory*

Long Short Term Memory Neural Networks (LSTM), the special and improved architecture of Recurrent Neural Networks (RNN) employs gate units and 'self-connected memory cells' to extract the underlying complex temporal dependencies in long and short time-series data, thereby addressing the 'vanishing gradient problem' of RNN [42]. LSTM consists of a memory block which is responsible for determining the addition and deletion of information through three gates, namely input gate (i_t), forget gate (f_t), and output gate (o_t) (Fig. 2) [43,44]. The memory cell in the memory block remembers temporal state information about current and previous timesteps. The workflow of LSTM at timestep t is detailed as [45,46]:

i. **Forget gate** (f_t) :Decides the information that should be discarded from the cell state represented by f_t, based on the last hidden state (h_{t-1})and the new input (x_t), i.e., the output of the previous cell state and input of the current cell state, respectively (Eqn. (7)).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{7}$$

where, W_f and b_f are the weight matrices and bias vector of the forget gate, respectively; σ is the logistic sigmoid function. The degree of information retention relies on the value of the forget gate, which lies in the range of [0,1] ('0'-forget all; '1'-remember all).

ii. **Input gate** (i_t) :Decides the information of input (x_t) that should be stored in the cell state represented by the input gate (i_t), where the information in the input gate (i_t) and the new candidate cell state ($\tilde{c}_t$) are updated (Eqn. (8) and Eqn. (9)). The new cell state (c_t) is updated by combining the previous cell state (c_{t-1}) and $\tilde{c}_t$ with the impact of forget gate (f_t) and input gate (i_t) (Eqn. (10)).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8}$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{9}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{10}$$

where, W_i and b_i are the weight matrices and bias vector of the input gate, respectively; W_c and b_c are the weight matrices and bias vector of the cell state, respectively; * is the point-wise multiplication; *tanh* is a hyperbolic tangent function with the range of [-1,1].

iii. **Output gate** (o_t) :Decides the information in the cell state (c_t) that should flow as the output of the output gate (o_t). Eqn. (11) evaluates which part of the cell state is to be exported, and Eqn. (12) computes the final output.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{11}$$

$$h_t = o_t . \tanh(c_t) \tag{12}$$

where, W_o and b_o are the weight matrices and bias vector of the output gate, respectively; The activation functions ($\sigma(.)$ and $\tanh(.)$) used to express the non-linearity of the LSTM network can be defined as in Eqn. (13) and Eqn. (14).

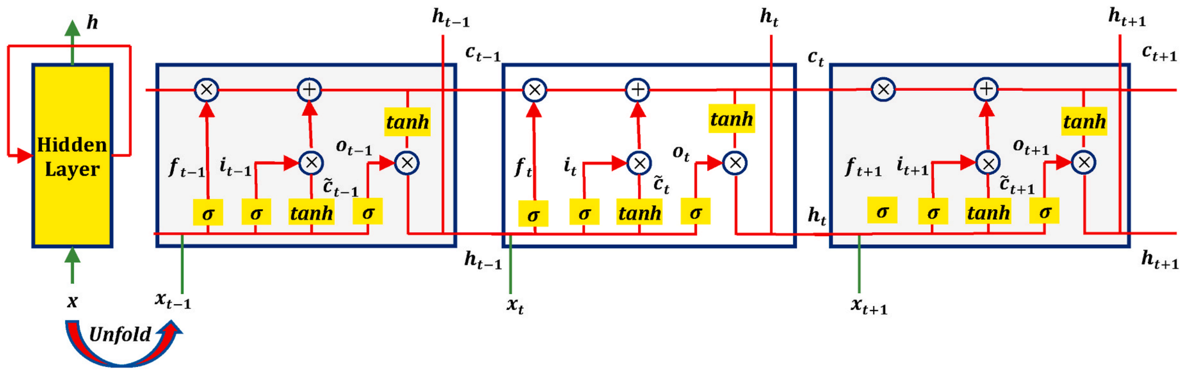


Fig. 2. Long short term memory.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{14}$$

2.4. Convolutional neural networks

Yann LeCun, et al. designed CNN or ConvNets, a class of deep feed-forward neural network which has proven its significance in extracting the spatial features in time series data, image recognition, and classification tasks [47].

The overall architecture of CNN comprises of convolutional layer, pooling layer, and fully connected layer to model complex data (Fig. 3) [48,49]. In general, CNN has several hierarchies of convolutional and pooling layers, wherein several convolution runs are done to extract the important features from the input data. In the convolutional layer, neurons of different layers of the network are locally connected through a weight sharing technique. Further, the convolutional layer forms the core of CNN, which performs convolution and activation operations on the input data to create a feature map. A sliding window of convolutional filter (matrix) of size equal to the size of the convolutional kernel moves across the horizontal and vertical directions of the $2-$ dimensional input data (Fig. 4(a)). The size of the convolutional filter should be equal to the size of the convolutional kernel. The convolution operation is completed by the kernel through the construction of a feature map, a 2D representation of the kernel generated by calculating the dot product of the convoluted kernel and the convolution filter. The convolution operation and activation operation in the convolutional layer can be defined as in Eqn. (15) and Eqn. (16).

$$C_{ij}^n = sum\left(ck_n \otimes fm_{ij}\right) + b_n \tag{15}$$

$$Y^n = \delta(C^n) \tag{16}$$

where, C_{ij}^n is the output of the convolution operations in the convolutional layer; ck_n is the convolution kernel matrix; fm_{ij} is the filter matrix; $sum(*)$ adds all elements in *; n is the n^{th} feature map; i and j are the number of steps of convolution filter in horizontal and vertical directions; b_n is the bias; $\delta()$ is the activation function.

With the completion of convolution, there are n feature maps that serve as input to the pooling layer. Pooling reduces the dimension of feature maps by reducing the redundant features without losing important information (Fig. 4(b)). The pooling operations reduce the computational burden of the model by compressing the input feature map. The average, max, and sum pooling can be computed using Eqn. (17) – Eqn. (19), respectively.

$$P_{ij}^n = Average_P\left(pfm_{ij}\right) \tag{17}$$

$$P_{ij}^n = Maximum_P\left(pfm_{ij}\right) \tag{18}$$

$$P_{ij}^n = Sum_P\left(pfm_{ij}\right) \tag{19}$$

where, P_{ij}^n is the output of the pooling operation in the pooling layer; i and j are the number of steps of pooling filter in horizontal and vertical directions; pfm_{ij} is the pooling filter matrix; $Average_P(*)$ provides the average of all the elements in *; $Maximum_P(*)$ provides the maximum element in *; $Sum_P(*)$ provides the sum of all the elements in *. Finally, the role of the flatten layer is to flatten the 2D data into a 1D vector representation, which is fed as input to the fully connected layer (Fig. 4

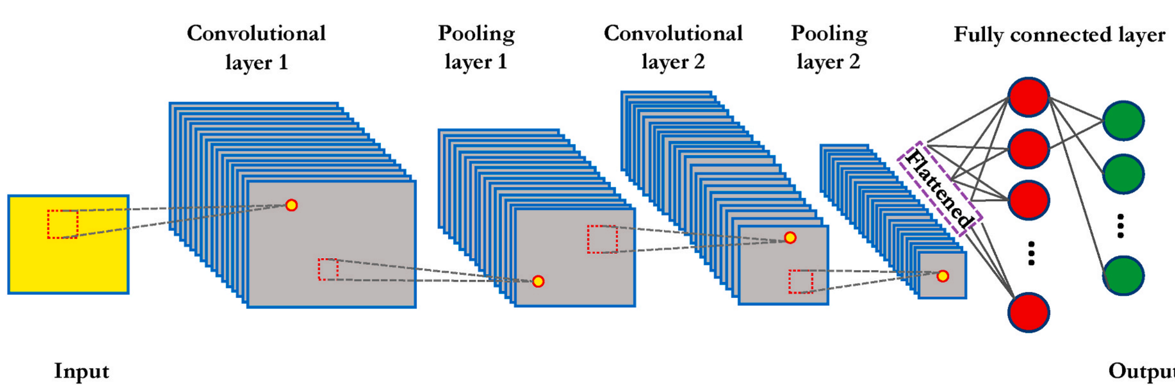
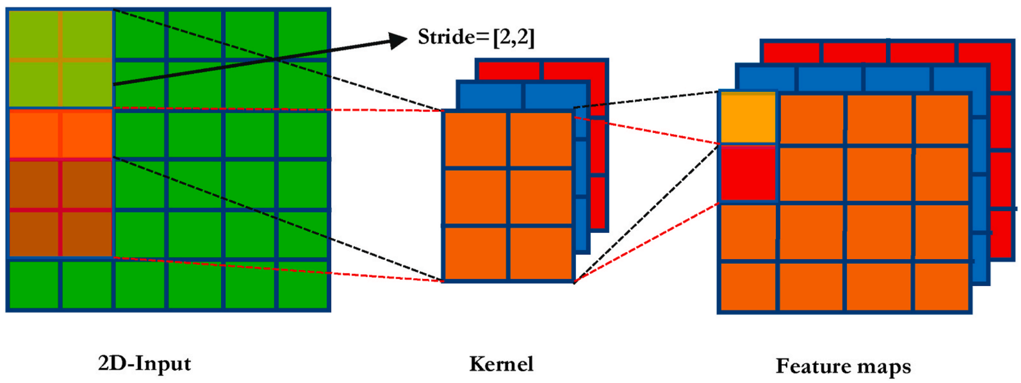
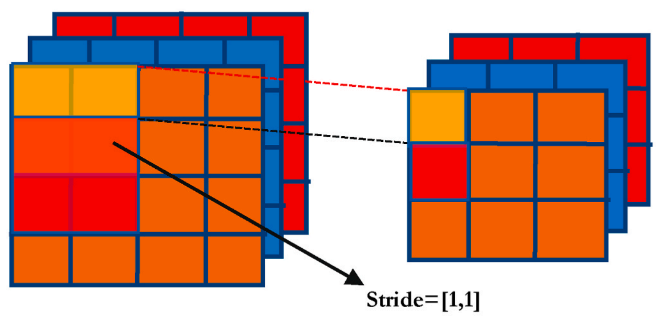


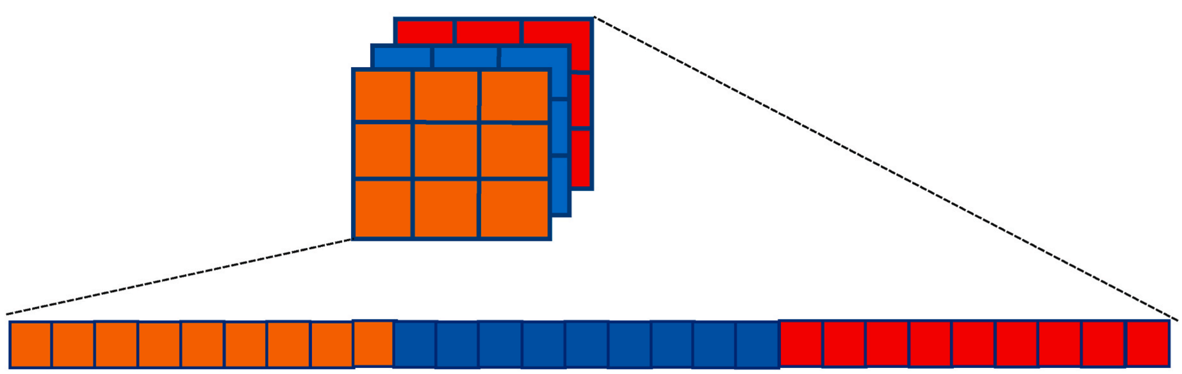
Fig. 3. Convolutional neural network architecture.





(a)

(b)



(c)

Fig. 4. (a) 2D convolution operation; (b) Pooling operation with pooling size = [2,2]; (c) Flatten operation [48].

(c)). The main idea of the fully connected layer is to connect the adjacent layers to integrate the features in providing the linear output for regression problems [50]. The calculation in the fully connected layer is defined in Eqn. (20).

$$FC = \delta\left(W_{fc}x + b_{fc}\right) \tag{20}$$

where, W_{fc} is the weight matrix; FC is the output matrix of the fully connected layer; b_{fc} is the bias; x is the input to the fully connected layer.

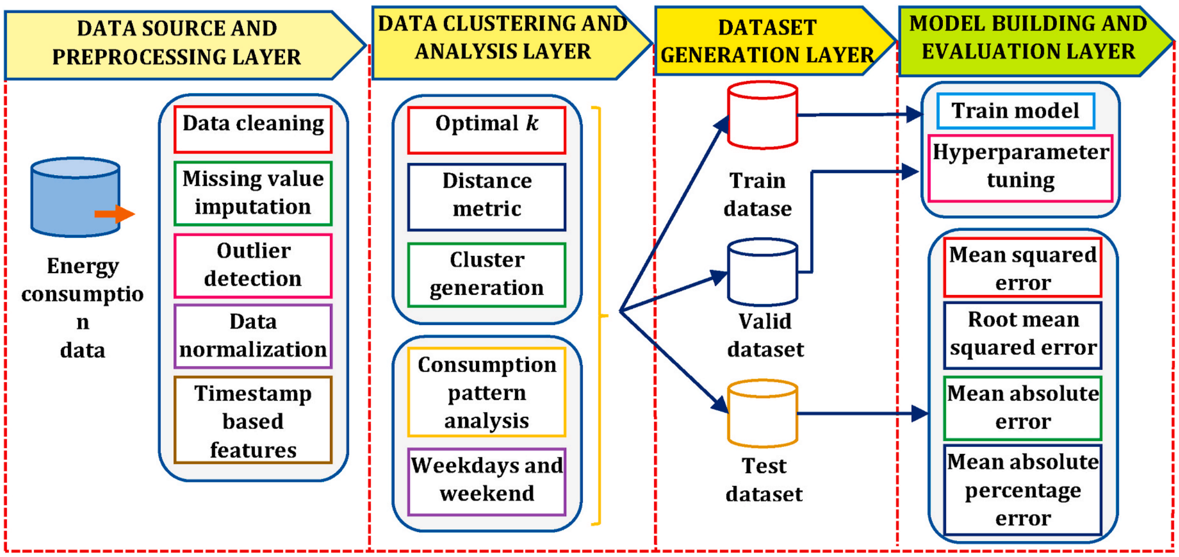


Fig. 5. kCNN-LSTM building energy consumption forecast model-Workflow.

Table 2
Preprocessed energy consumption dataset.

Year	Day of the year	Season	Month	Day of the week	Hour of the day	Minute of the hour	Type of the day	Energy Consumed
2017	1	3	1	1	0	0	1	256
2017	1	3	1	1	0	15	1	258
…	…	…	…	…	…	…	…	…
2019	334	3	11	49	23	45	0	426

3. CNN-LSTM: proposed deep learning framework for building energy consumption forecast

Fig. 5 presents the architecture of *k*CNN-LSTM to forecast the energy consumption of buildings. The overall workflow of *k*CNN-LSTM consists of (i) **Data source and preprocessing layer:** Transform the raw energy consumption data into a compatible format, (ii) **Data clustering and analytics layer:** cluster the data into different groups for trend analysis, (iii) **Dataset generation layer:** generate datasets for train, valid, and test, and (iv) **Model building and evaluation layer:** create models to forecast energy consumption and evaluate their performance using various error metrics. *k*CNN-LSTM energy consumption forecast model is then integrated with the BEMS to forecast building energy consumption for the user-specified intervals (time slot, day, week, month, season, etc.).

3.1. Data source and preprocessing layer

The energy consumption data of the KReSIT building at IIT, Bombay, contains more than 30 electrical related features. Out of these, this study considers timestamp (dd-mm-yyyy HH: MM: SS) and energy consumption columns for processing, as it is available across different hierarchies of buildings. Since the yearly energy consumption data of 2018 is considered for analysis, the available per second granularity data is aggregated to 15 min granularity. Totally, there are 96 energy consumption values for each day in the selected year (Block0 - 00:00 to 00:15, Block1 - 00:15 to 00:30, …, Block 95–23:45 to 00:00). For techniques employed to handle missing values, outliers, and high magnitude values, refer Section 4.2. Further, a simple Python code is designed to generate seven features (day of the year (0–365), season (0-Summer; 1-Monsoon; 2-Winter), month, day of the week (0–52), hour of the day (0–24), minute of the hour (0–60), type of the day (0-Holiday; 1-Working day)) from the available timestamp. Table 2 shows the structure of the energy consumption data used for experimentation.

3.2. Data clustering and analysis layer

Given an energy consumption dataset E with M objects, $E=\{T_1,T_2,\dots,T_M\}$, where T_i is a time series. Time series clustering of E into clusters $C=\{C_1,C_2,\dots,C_k\}$ is done by grouping the data points based on the similarity in the consumption trends ($E=U_{i=1}^{k}C_i$, where $i\neq j$ and $C_i\cap C_j=\varnothing$) [51]. The procedure for the selection of the cluster parameters like distance measure, evaluation measures, etc. is detailed below.

3.2.1. Distance measure

Euclidean distance and Dynamic Time Warping (DTW) form the most widely used distance metric for time series clustering due to the efficiency of Euclidean distance (simple, fast, and parameter-free) and the effectiveness of DTW.

(i) **Euclidean distance measure:** It provides one-to-one matching between the timestamps of the time series data $T_1=\{t1_1,t1_2,\dots,t1_n\}$ and $T_2=\{t2_1,t2_2,\dots,t2_n\}$ as defined in Eqn. (21).

$$Euclidean_{(T1,T2)}=\sqrt{\sum_{i,j=1}^{n}\left(t1_i-t2_j\right)^2} \tag{21}$$

However, Euclidean distance is not the best choice for time series data since it depends on the domain & time series characteristics of the data and disability to capture the distortions in the time domain, i.e., sensitive to shifts in the time axis.

(ii) **DTW distance measure** [52]**:** DTW finds the optimal non-linear alignment between two time series by calculating their matching error as follows:

a. Consider two time series of the same length n, $T_1=\{t1_1,t1_2,\dots,t1_n\}$ and $T_2=\{t2_1,t2_2,\dots,t2_n\}$, where $t1_i$ and $t2_i$ are the energy consumption value of time series T_1 and T_2 at i^{th} timestamp.

b. Construct a cost matrix of n x n dimension, whose i^{th} and j^{th} element represents the Euclidean distance between $t1_i$ and $t2_j$.

c. Find the path (P), i.e., optimal alignment between T_1 and T_2 through the constructed cost matrix that minimizes the cumulative distance (Eqn. (22)).

$$P^* = argmin_P\left(\sqrt{\sum_{k=1}^{K} p_k}\right) = argmin_P\left(\sqrt{\sum_{i,j=1}^{n} \left(t1_i - t2_j\right)^2}\right) \tag{22}$$

where, $P = (p_1, p_2, \ldots, p_K)$, each element P denotes the distance between $i^{th}$and j^{th} data point in T_1 and T_2.

d. Find the optimal path using a recursive dynamic programming function.

DTW has a complexity of $O(nm)$, where n and m are the length of first and second time series, respectively.

(iii) **Lower bound Keough distance measure** [53,54]**:** Due to high complexity, the recursive application of DTW on a long time series data is expensive. Imposing locality constraint using a threshold determined by the window size and Lower Bound Keough (LB Keough) distance metric provides a way to speed up DTW. LB Keough is based on the fact that DTW uses global path constraints when comparing two time series. LB Keough for two time series $T_1 = \{t1_1, t1_2, \ldots, t1_n\}$ and $T_2 = \{t2_1, t2_2, \ldots, t2_n\}$ is defined as in Eqn. (23).

$$LB - Keough_{(T_1, T_2)} = \sqrt{\sum_{i=1}^{N} \begin{cases} T_{2_i} - U_i^2 & if\ T2_i > U_i \\ T_{2_i} - L_i^2 & if\ T2_i < L_i \\ 0 & Otherwise \end{cases}} \tag{23}$$

where, U_i and L_i are the lower and upper bound of time series T_1, which can be defined as $U_i = \max(T_{1_{i-r}} : T_{1_{i+r}})$ and $L_i = \min(T_{1_{i-r}} : T_{1_{i+r}})$; r depends on the type of path constraint used (e.g., Itakura parallelogram and Sakoe-Chiba band). The complexity of LB Keough is $O(m)$.

This work uses $k-$ means clustering algorithm and LB-Keough distance metric to cluster the energy consumption patterns for trend analysis.

3.2.2. Optimal value of k

The most primary and essential step in any unsupervised algorithm is to find the optimal number of clusters to which the data points are added. Several methods like Elbow, average Silhouette, gap statistics, etc. have been proposed in the literature to identify the optimal number of clusters (k). This work employs the Elbow method to identify the optimal value of k for precise clustering results. The elbow method plots various SSE values for different k values through iterative runs of $k-$ means clustering algorithm. The basic idea is that as the value of k increases, the average distortion decreases with fewer elements in the cluster. Therefore, k value at which the distortion decreases the most is the elbow point, which is chosen as the optimal k for the considered dataset.

3.3. Dataset generation layer

The energy consumption data of each cluster is partitioned into training (E_{Train}), validation (E_{Val}), and testing (E_{Test}) in the ratio of 60:20:20 for overall, weekdays, weekend, and day analysis.

3.4. Model building and evaluation layer

*k*CNN-LSTM, a building energy consumption forecast model is designed to model the spatial correlation between the time stamp based generated variables and temporal information in the irregular consumption patterns for better forecast accuracy. The model architecture of *k*CNN-LSTM consists of convolutional layer, pooling layer, LSTM layer, and fully connected layer. The convolutional and pooling layers of CNN extracts the spatial characteristics of the multivariate data and pass on the identified features as input to the LSTM. The LSTM models the irregular trends in the energy consumption data based on the spatial features provided by the CNN. Later then, a fully connected layer receives and decodes the output of LSTM to provide the forecasted energy consumption data. The multi-input and multi-output sliding window mechanism is employed in such a way that *k*CNN-LSTM learns the input data for every block (15 min).

The input of size 60×7 is fed to the convolutional layers with filters, kernels, padding, and activation of 64, 2×1, 'same', and ReLU, respectively. The output of the convolutional layer is passed to an LSTM layer with 64 units and 'tanh' activation. With the series of fully connected layer, the number of units in the dense layers are 32 and 60 to produce the forecast of the next 60 min energy demand. The hyperparameters of each layer in *k*CNN-LSTM were fine-tuned using ISCOA, an enhanced variant of SCOA to improve the learning speed and performance of the learning model. ISCOA uses Haar wavelet based mutation operator to enhance the divergence nature of the algorithm towards the global optimum. For more details related to hyperparameter tuning, refer [55].

4. Case study

This section presents a detailed description of the considered energy consumption data, data preprocessing techniques, and evaluation metrics. Further, a detailed analysis of the performance of *k*CNN-LSTM over the $k-$ means variant of the state-of-the-art building energy consumption forecast model.

4.1. Dataset

This work uses the energy consumption data of KReSIT, IIT-Bombay, India. The flow of electricity consumption data from sensing to sharing comprises several layers, physical and software components (Fig. 6). The functionalities of each layer are detailed below.

(i) **Physical layer:** It corresponds to the actual building environment for which electricity consumption is measured. KReSIT, IIT-B is a four-storeyed academic building with three wings (A, B, and C) for each floor, which consists of smart classrooms, office rooms, auditoriums, lecture halls, research laboratories, and server rooms. The location of the considered academic building is in Mumbai. The average temperature of Mumbai falls within $17 - 34^oC$ for three seasons, namely summer (April–June), monsoon (July–September), and winter (October–March). The energy and temperature profiles of KReSIT are monitored and controlled by its own BEMS built for "Minimal power consumption in an occupied room" and 'Zero consumption during zero occupancy".

Since the major objective of this research is to forecast the overall energy consumption of the considered buildings, the consumption data provided by the smart meter installed at the MAINS is used for experimentations.

For experimental purposes, the energy consumption data from January 1, 2018 to December 31, 2018 at 15 min granularity was considered. For better visualization, each of the 15 min interval is mapped to a block number, i.e., Block 0: 00:00 to 00:15, Block 2: 00:15 to 00:30, …, Block 95: 23:45 to 0:00. Fig. 8 provides the statistical detail

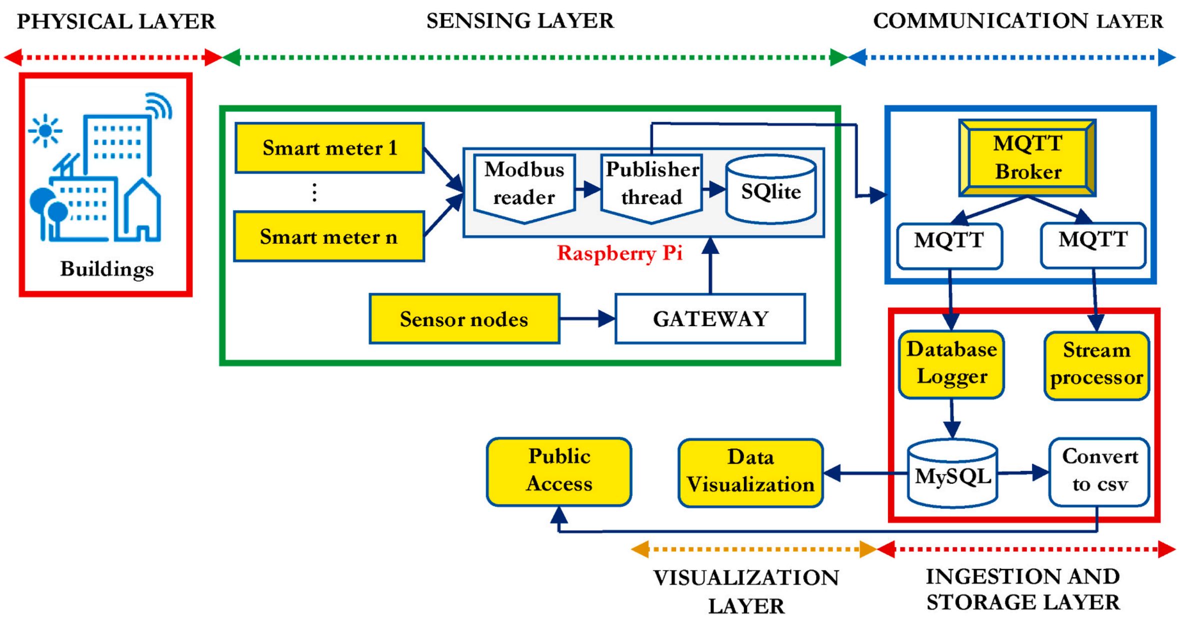


Fig. 6. BEMS, KReSIT, IIT-B: Architecture.

(i) **Sensing layer:** A three-phase smart meters deployed at various facilities of KReSIT records the various electrical parameters through a MODBUS protocol over RS485 cable. This study uses the energy consumption variable, which provides energy consumed by air conditioners, plugs, lights, and fans at per-second granularity in the considered physical space. The energy consumption data is queried from the smart meter by a Raspberry Pi at predefined intervals in a round robin fashion. A python script is executed in Raspberry Pi to collect the smart meter data and transmit to the central server using Message Queuing Telemetry Transport (MQTT) protocol. The sensor nodes communicate via. serial or low-range RF. Further, a gateway (NodeMCU and RPi) is used to send the sensor data to the central server for storage and processing.

(ii) **Communication layer:** MQTT, a fast and minimal overhead transmission protocol, is used to publish the data from the smart meters to the remote ingestion servers. It uses 'publish-subscribe' concept, which enables multiple subscribers to receive the selected data stream by registering with the central broker.

(iii) **Ingestion and storage layer:** The data stream of the smart meters are published on the topic of structure <channel>/<data type>/<sensor identifier>, where different channels used to publish data are specified by the topic, data type is used by the ingestion engine to find the schema of the published data stream, and sensor identifier corresponds to the global unique ID assigned to each smart meter. The smart meter data is stored in MySQL database and CSV format for visualization and archival.

(iv) **Visualization layer:** Grafana, an open source data visualization engine, is used to display the real-time electricity consumption of various appliances (ACs, plugs, lights, and fans), which enables the consumers to visualize their energy consumption. The academic building dataset can be accessed from Ref. [56] for research purposes. The live power consumption data can be visualized, as shown in Fig. 7.

Fig. 7. Data visualization of KReSIT overall power consumption.

of the energy consumption data for the months of 2018.

4.2. Data preprocessing

In a real world scenario, energy consumption data is susceptible to several discrepancies like missing data, incomplete data, noise, etc. which might lead to poor data analysis. The techniques used to pre-process the energy consumption data are detailed below.

(i) **Missing value:** Data interpolation method is employed to impute the missing values in the energy consumption dataset (Eqn. (24)).

$$f(x_i) = \begin{cases} \dfrac{x_{i-1} + x_{i+1}}{2} & x_i \in NaN; x_{i-1}, x_{i+1} \notin NaN \\ \dfrac{x_{i-1} + x_{i-2}}{2} & x_i \in NaN; x_{i-1}, x_{i-2} \notin NaN; x_{i+1} \in NaN \\ x_i & x_i \notin NaN \end{cases} \quad (24)$$

where, x_i represents the energy consumption value at i^{th} timestamp.

(ii) **Outlier detection:** "Three sigma rule of thumb" is used to recover the erroneous value in the energy consumption dataset (Eqn. (25)).

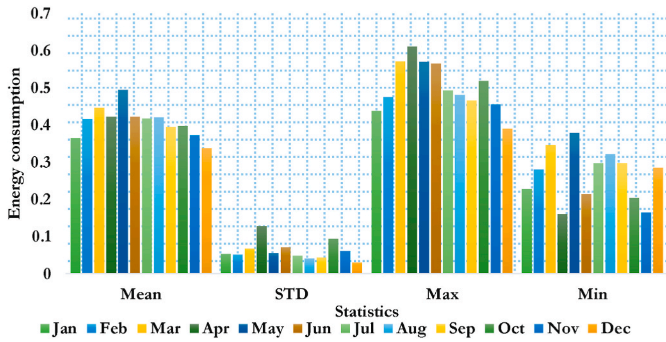


Fig. 8. Statistics of KReSIT energy consumption data for 2018.

$$f(x_i) = \begin{cases} avg(x) + 2.std(x) & \text{if } x_i > avg(x) + 2.std(x) \\ x_i & \text{otherwise} \end{cases} \tag{25}$$

where, x is a vector that consists of x_i; $avg(x)$ is the average value of x; $std(x)$ is the standard deviation of x.

(iii) **Normalization:** Min-max normalization technique is used to regularize the energy consumption data to avoid inaccurate forecasts due to the high magnitude in the energy consumption data (Eqn. (26)).

$$f(x_i) = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{26}$$

where, $\min(x)$ and $\max(x)$ are the minimum and maximum values of x, respectively.

4.3. Performance metrics

There exist several statistical measures to evaluate the performance of the learning model based on the difference between the actual and predicted value. The performance evaluation metrics used in this paper are defined below.

(i) **Mean squared error:** It is defined as the average squared difference between the actual and the forecasted value (Eqn. (27)).

$$MSE = \sum_{i=1}^{n} \frac{\left(\widehat{y}_i - y_i\right)^2}{n} \tag{27}$$

where, n is the number of data points in the dataset; y_i and $\widehat{y}_i$ are the actual and the forecasted energy consumption value at timestamp i, respectively.

(ii) **Root mean squared error:** It is defined as the standard deviation of the differences between actual and the forecasted value (Eqn. (28)).

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\left(\widehat{y}_i - y_i\right)^2}{n}} \tag{28}$$

(iii) **Mean absolute error:** It measures the absolute difference between the actual and the forecasted value (Eqn. (29)).

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| \widehat{y}_i - y_i \right| \tag{29}$$

(iv) **Mean absolute percentage error:** It is the measure of the amount of deviation of the forecasted value from the actual value (Eqn. (30)).

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\widehat{y}_i - y_i}{y_i} \right| *100 \tag{30}$$

5. Results and discussions

The implementation of kCNN-LSTM and the experimental analysis

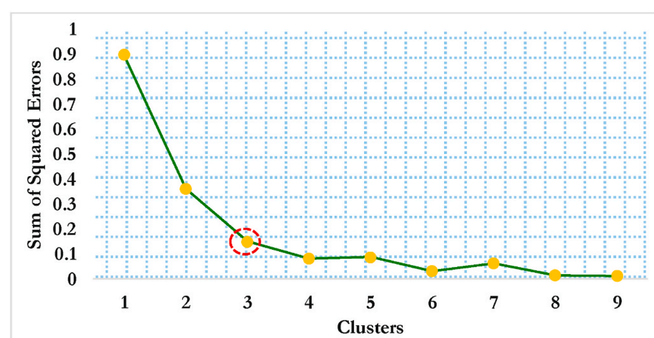


Fig. 9. Optimal number of cluster-Elbow method.

Table 3
LB-Keough distance matrix.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Jan	0	0.2573	0.4000	0.5253	0.7467	0.1313	0.3930	0.5967	0.2966	0.7075	0.3134	0.4040
Feb	0.9143	0	1.1917	0.2607	1.2914	0.2951	1.1596	0.9738	0.1183	0.5721	0.2396	1.0829
Mar	0.6175	0.2488	0	0.3578	0.2298	0.1580	0.2730	0.4172	0.3766	0.3179	0.5215	1.0556
Apr	1.2903	1.0702	1.3483	0	1.5387	1.0032	1.3159	1.1997	1.1093	0.8165	1.1244	1.2989
May	1.0323	0.4832	0.1398	0.4926	0	0.2995	0.5324	0.5502	0.6560	0.4026	0.7946	1.4301
Jun	0.7177	0.3172	0.7714	0.3668	0.8568	0	0.7803	0.6995	0.3958	0.5197	0.4220	0.9394
Jul	0.3790	0.0414	0.1022	0.3927	0.4171	0.1514	0	0.2753	0.1328	0.3566	0.2659	0.8025
Aug	0.4490	0.0510	0.1279	0.0746	0.2597	0.0475	0.1072	0	0.1659	0.0817	0.2027	0.6976
Sep	0.5486	0.1005	0.7276	0.3096	0.8489	0.1709	0.6779	0.5866	0	0.4288	0.1016	0.7053
Oct	0.7388	0.2619	0.7102	0.0632	0.9346	0.3679	0.6912	0.4949	0.4222	0	0.3892	0.8452
Nov	0.6150	0.0536	0.9079	0.2452	1.0596	0.2228	0.8744	0.7127	0.0116	0.4468	0	0.6693
Dec	0.0934	0.1955	0.4630	0.3906	0.8102	0.2701	0.4003	0.3970	0.1838	0.4857	0.1660	0

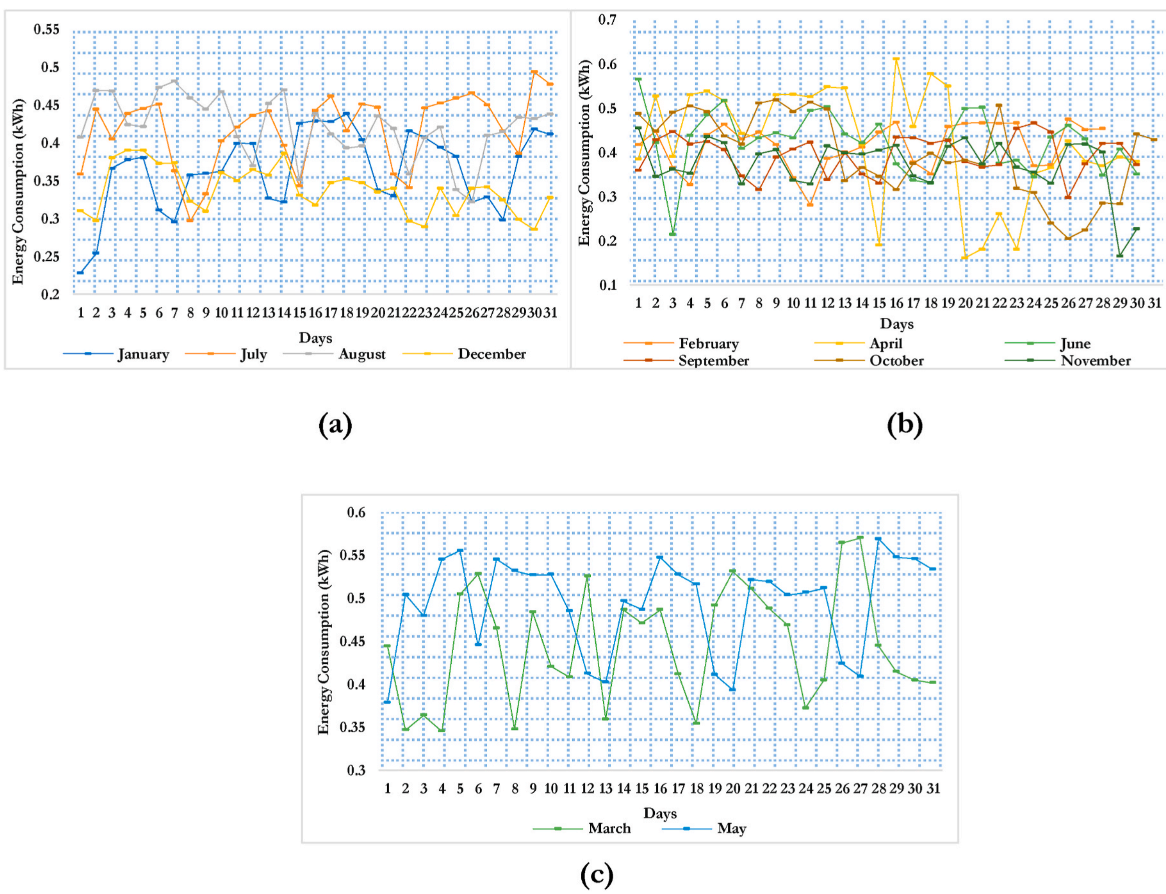


Fig. 10. Cluster analysis (a) Cluster 1 (b) Cluster 2 (c) Cluster 3.

(ii) Phase 2: Energy consumption trend analysis.

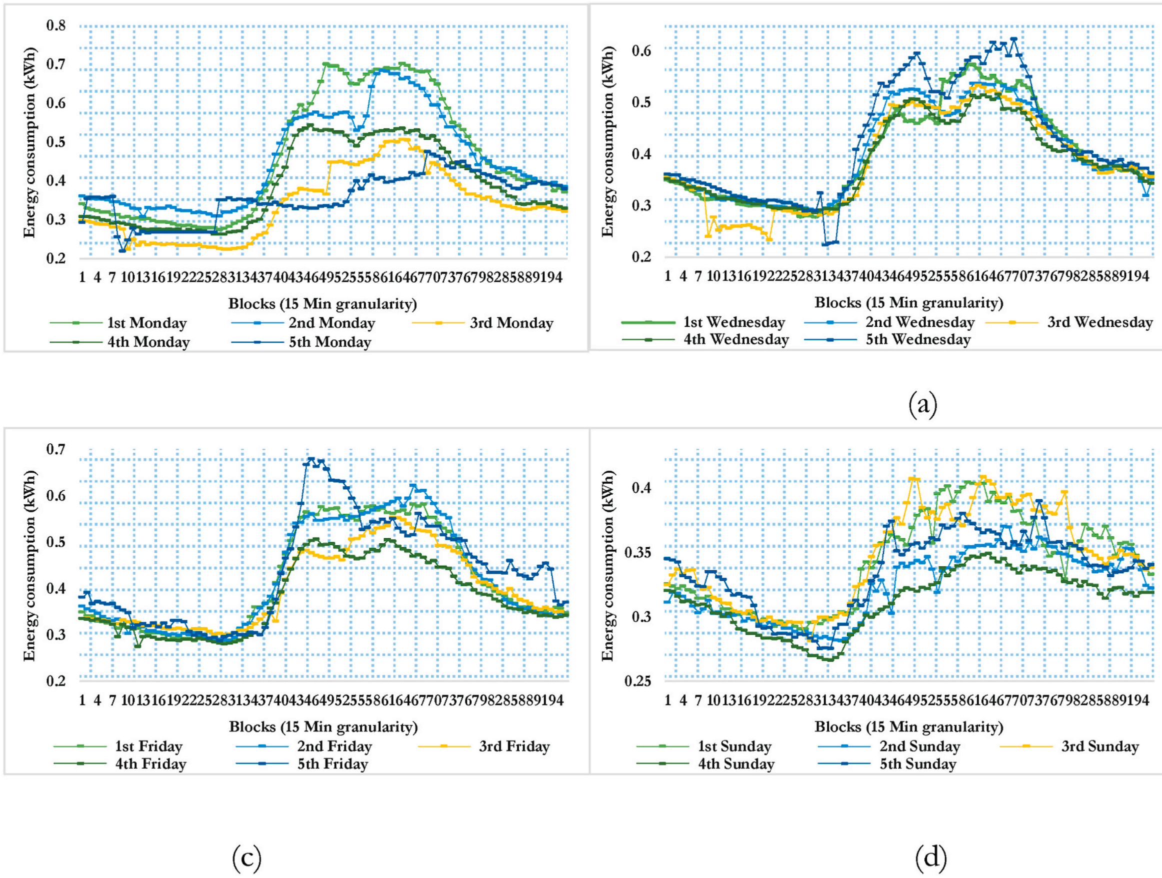


Fig. 11. Energy consumption patterns in Cluster 1 for different days (a) Monday (b) Wednesday (c) Friday (d) Sunday.

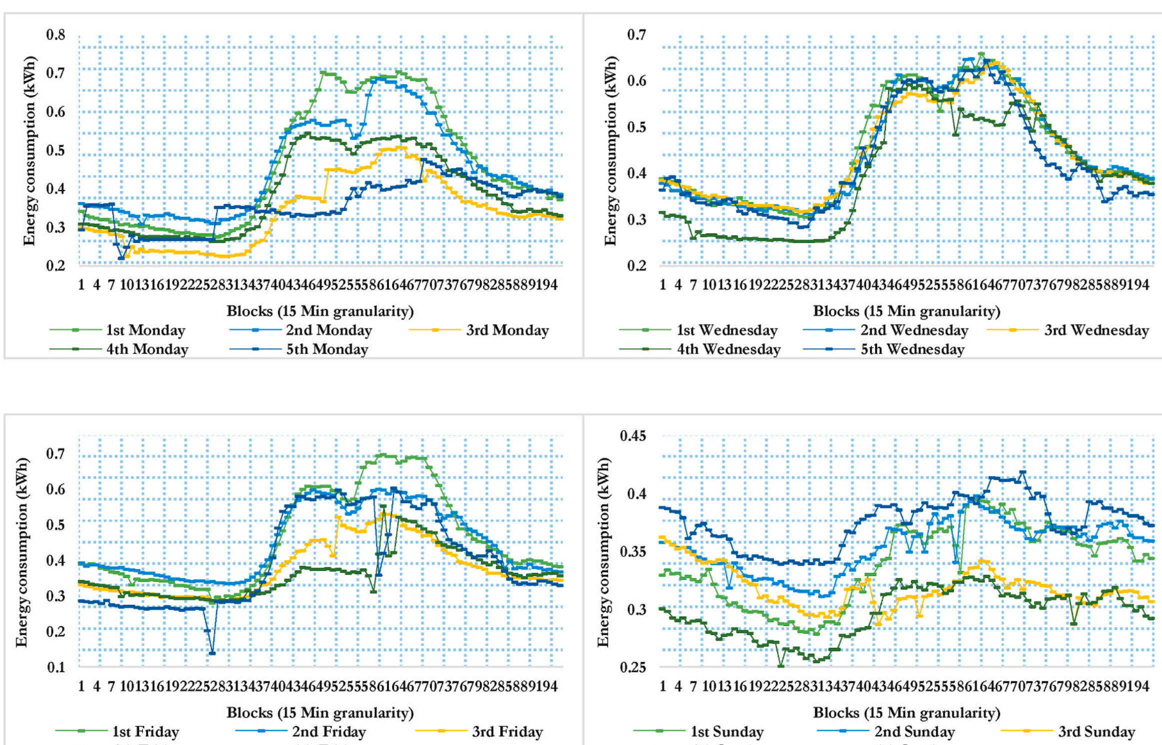


Fig. 12. Energy consumption patterns in Cluster 2 for different days (a) Monday (b) Wednesday (c) Friday (d) Sunday.

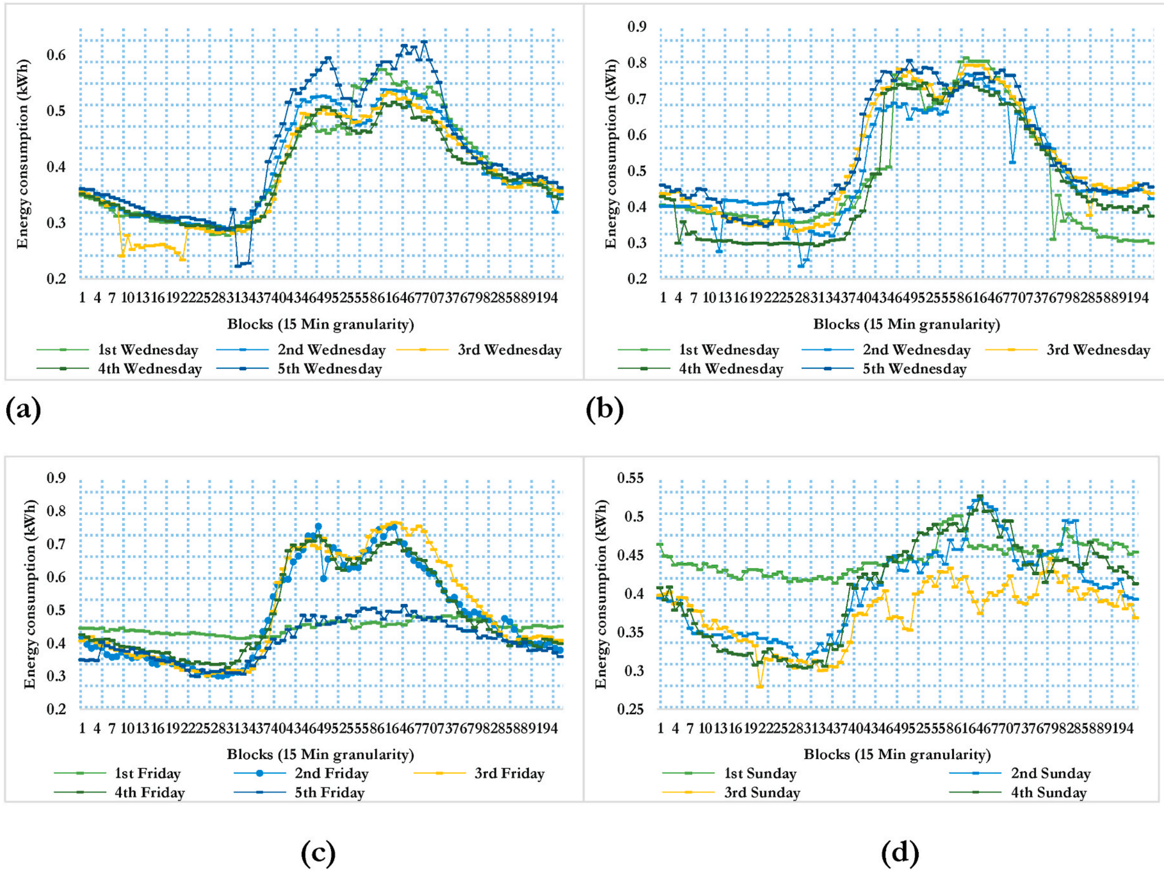


Fig. 13. Energy consumption patterns in Cluster 3 for different days (a) Monday (b) Wednesday (c) Friday (d) Sunday.

Table 4
Parameter setting.

S. No.	Model	Description	Range
1	ISCOA	Number of populations	5
		Number of iterations	20
2	CNN-LSTM	Epoch	100
		Batch size	1
		Optimizer	Adam
		Convolution and pooling layers	[1,3]
		LSTM layers	[1,3]
		Filters	[64,128]
		Kernel	(2,1)
		Units in fully connected layer	[5100]
		Hidden units in LSTM	[5100]
		Activation function	ReLu (CNN) and tanh (LSTM)

was carried out in Operating system - Windows 10; System configuration - i7 processor and 64 GB RAM; and Framework - Python 3.6. The experiments carried to analyze the efficiency of *k*CNN-LSTM for the energy consumption forecast problem can be divided into three phases, namely (i) Data clustering, (ii) Energy consumption trend analysis, and (iii) Energy demand forecast analysis.

(i) Phase1: Clustering

In general, the application of clustering to any data results in the identification of groups or clusters with similar characteristics. In this work, *k*-means clustering, a partitioning based clustering algorithm is applied to group the data points (time series data - energy consumption value recorded at each timestamp) into clusters based on the similarity in the energy consumption patterns. Due to the simplicity and minimal complexity, LB Keough distance metric is used to compute the similarity between the time series data (months). Table 3 provides the distance matrix computed using LB Keough measure for the time series data of all months in 2018.

Further, the Elbow method is used to identify the optimal value of *k*. For the same, SSE is used as an evaluation metric, which when plotted against the number of clusters, helps in the identification of the optimal *k* (Fig. 9). For the considered energy consumption data of 2018, the optimal *k* as 3.

From the clustering results, it is obvious that the energy consumption patterns of the year 2018 can be divided into three clusters, namely Cluster 1, Cluster 2, and Cluster 3. An in-depth analysis of the obtained clusters (Cluster 1, Cluster 2, and Cluster 3), provides detailed insight into the energy consumption patterns in KReSIT building across different months in 2018 (Fig. 10(a)-(c)).

In this study, a set of contextual features were generated based on the timestamp in the considered building energy consumption data < timestamp, energy>. However, to gain deeper insight on the variations in the energy consumption trend, it is vital to analyze the consumption patterns in each cluster. For the same, daily analysis was performed on

Table 5
Cluster 1 – Performance analysis of the state-of-the-art machine learning and deep learning energy demand forecast models.

Type	Metrics	Models					
		ARIMA	DBN	MLP	LSTM	CNN	CNN-LSTM
Overall	**MSE**	0.0102	0.0183	0.0189	0.6226	0.3313	**0.0095**
	RMSE	0.0982	0.1354	0.1376	0.7890	0.5756	**0.0974**
	MAE	0.1098	0.1063	0.0998	0.7310	0.5429	**0.0711**
	MAPE	0.8128	0.3550	0.3161	2.0425	1.7206	**0.2697**
Weekdays	**MSE**	0.0356	0.0966	0.0805	0.0267	0.0778	**0.0168**
	RMSE	0.1887	0.3107	0.2837	0.1633	0.2790	**0.1297**
	MAE	0.1527	0.2842	0.2547	0.1240	0.2594	**0.1113**
	MAPE	0.2701	0.8009	1.9979	0.4433	0.7682	**0.3872**
Weekend	**MSE**	0.0293	0.0040	0.0045	0.0048	0.0048	**0.0034**
	RMSE	0.1712	0.0635	0.0667	0.0693	0.0693	**0.0580**
	MAE	0.1471	0.0547	0.0537	0.0570	0.0602	**0.0481**
	MAPE	0.1447	0.1737	0.1580	0.1725	0.1826	**0.1425**
Monday	**MSE**	0.0151	0.0286	0.0073	0.0091	0.0030	**0.0013**
	RMSE	0.1232	0.1690	0.0853	0.0953	0.0551	**0.0364**
	MAE	0.1031	0.1585	0.0580	0.0805	0.0404	**0.026**
	MAPE	0.3119	0.5046	0.1320	0.2486	0.1166	**0.0728**
Wednesday	**MSE**	0.0116	0.0039	0.0023	0.0066	0.0021	**0.00100**
	RMSE	0.1077	0.0624	0.0477	0.0812	0.0455	**0.0324**
	MAE	0.0925	0.0509	0.0398	0.0635	0.0387	**0.0267**
	MAPE	0.2506	0.1635	0.1102	0.1851	0.1128	**0.0765**
Friday	**MSE**	0.3271	0.0180	0.0095	0.0038	0.0042	**0.0024**
	RMSE	0.5720	0.1343	0.0975	0.0618	0.0650	**0.0491**
	MAE	0.5024	0.1229	0.0825	0.0511	0.0551	**0.0368**
	MAPE	0.2444	0.3860	0.1860	0.1455	0.1601	**0.1033**
Sunday	**MSE**	0.0064	0.0022	0.0011	0.0010	0.0009	**0.0008**
	RMSE	0.0803	0.0474	0.0336	0.0324	0.0306	**0.0285**
	MAE	0.0679	0.0419	0.0273	0.0273	0.0252	**0.0232**
	MAPE	0.0966	0.1502	0.0867	0.0924	0.0856	**0.0783**

Table 6
Cluster 2 – Performance analysis of the state-of-the-art machine learning and deep learning energy demand forecast models.

Type	Metrics	Models					
		ARIMA	DBN	MLP	LSTM	CNN	CNN-LSTM
Overall	**MSE**	0.1006	0.1303	0.1116	0.0618	0.0639	**0.0212**
	RMSE	0.1625	0.3609	0.3341	0.2485	0.2528	**0.1456**
	MAE	0.1138	0.3215	0.2931	0.2133	0.2278	**0.0997**
	MAPE	0.2062	0.2549	0.2375	0.5785	0.2358	**0.2054**
Weekdays	**MSE**	0.1007	0.1007	0.0384	0.0504	0.0687	**0.0251**
	RMSE	0.2066	0.3174	0.1958	0.2246	0.2621	**0.1586**
	MAE	0.1456	0.2718	0.1438	0.1911	0.2206	**0.1397**
	MAPE	0.1918	0.3015	0.4496	0.1855	0.1926	**0.1653**
Weekend	**MSE**	0.0101	0.0416	0.0156	0.0057	0.0334	**0.0038**
	RMSE	0.2134	0.2039	0.1251	0.0757	0.1828	**0.062**
	MAE	0.1329	0.1892	0.1052	0.0579	0.1683	**0.0433**
	MAPE	0.8885	0.5277	0.2774	0.1704	0.4541	**0.1425**
Monday	**MSE**	0.0113	0.0272	0.0230	0.0122	0.0147	**0.0110**
	RMSE	0.1190	0.1648	0.1516	0.1106	0.1211	**0.1050**
	MAE	0.0829	0.1391	0.1230	0.0771	0.0943	**0.0721**
	MAPE	0.2252	0.4564	0.2893	0.2287	0.2838	**0.2169**
Wednesday	**MSE**	0.0094	0.0229	0.0095	0.0099	0.0126	**0.0002**
	RMSE	0.0971	0.1512	0.0975	0.0995	0.1124	**0.0164**
	MAE	0.1040	0.1125	0.0714	0.0708	0.0865	**0.0707**
	MAPE	0.2576	0.4807	0.1733	0.1817	0.2370	**0.1551**
Friday	**MSE**	0.0218	0.1466	0.1293	0.1218	0.1222	**0.0013**
	RMSE	0.1372	0.1478	0.0969	0.1118	0.0967	**0.0871**
	MAE	0.1162	1.0291	0.2317	0.8481	0.9251	**0.1137**
	MAPE	0.2259	0.3521	0.2725	0.2838	0.2915	**0.2346**
Sunday	**MSE**	0.9105	0.0653	0.0167	0.0148	0.0149	**0.0068**
	RMSE	0.0203	0.0048	0.0041	0.0039	0.0046	**0.0036**
	MAE	0.1182	0.0695	0.0640	0.0626	0.0679	**0.0600**
	MAPE	0.3103	0.0542	0.0470	0.0448	0.0512	**0.0418**

Table 7
Cluster 3 – Performance analysis of the state-of-the-art machine learning and deep learning energy demand forecast models.

Type	Metrics	Models					
		ARIMA	DBN	MLP	LSTM	CNN	CNN-LSTM
Overall	**MSE**	0.0019	0.0708	0.0380	0.0113	0.0124	**0.0010**
	RMSE	0.1047	0.2660	0.1950	0.1049	0.1114	**0.0303**
	MAE	0.0812	0.2079	0.1450	0.0880	0.0893	**0.0165**
	MAPE	0.2086	0.3882	0.3107	0.1998	0.2101	**0.1670**
Weekdays	**MSE**	0.1006	0.0552	0.4857	0.2612	0.1323	**0.0142**
	RMSE	0.3264	0.2350	0.2204	0.1616	0.1193	**0.1150**
	MAE	0.2180	0.1892	0.1678	0.1348	0.9167	**0.0922**
	MAPE	3.3808	0.3675	0.3686	0.3208	0.2199	**0.2134**
Weekend	**MSE**	0.0024	0.0019	0.0015	0.0016	0.0013	**0.0013**
	RMSE	0.1207	0.0433	0.0389	0.0397	0.0366	**0.0362**
	MAE	0.1138	0.0357	0.0308	0.0304	0.0283	**0.0275**
	MAPE	0.8961	0.0847	0.0702	0.0700	0.0677	**0.0646**
Monday	**MSE**	0.1006	0.0482	0.0417	0.0290	0.0292	**0.0256**
	RMSE	0.3250	0.2196	0.2042	0.1702	0.1708	**0.1601**
	MAE	0.2177	0.1641	0.1566	0.1347	0.1379	**0.1333**
	MAPE	0.2332	0.2704	0.4117	0.2303	0.2337	**0.2257**
Wednesday	**MSE**	0.1006	0.0662	0.0463	0.0314	0.0218	**0.0153**
	RMSE	0.1253	0.2572	0.2151	0.1771	0.1478	**0.1235**
	MAE	0.2169	0.2010	0.1623	0.1443	0.1186	**0.1045**
	MAPE	0.5244	0.3756	0.4868	0.2569	0.2460	**0.1947**
Friday	**MSE**	0.1317	0.0507	0.0578	0.0280	0.0192	**0.0101**
	RMSE	0.3630	0.2251	0.2403	0.1673	0.1386	**0.1005**
	MAE	0.3126	0.1769	0.1885	0.1339	0.1180	**0.0778**
	MAPE	0.4956	0.3336	0.6999	0.2457	0.2384	**0.1473**
Sunday	**MSE**	0.0251	0.0351	0.0201	0.0109	0.0104	**0.0075**
	RMSE	0.1586	0.1873	0.1419	0.1045	0.1018	**0.0867**
	MAE	0.1397	0.1604	0.1144	0.0835	0.0796	**0.0611**
	MAPE	0.3280	0.3643	0.3987	0.1942	0.1929	**0.1404**

all days of weekdays and weekend. The experimental design and validation are presented with the daily analysis on the selected days of weekdays (Monday, Wednesday, and Friday) and weekends (Sunday) of the identified clusters. The analysis and results of the other days of the weekdays and weekends are provided in the Appendix (Fig. A1, A2 and A3; Table A1, A2 and A3). To understand the fluctuations in the energy consumption demand of KReSIT and to provide an accurate demand forecast, the average energy consumption over the selected days across the months in each cluster was analyzed (Figs. 11, 12 and 13(a)-(d)).

The daily analysis enables a better understanding of the energy consumption patterns and aids the deep learning model to provide better forecast of the energy demands for the user-specified day, month, and interval.

(iii) Phase 3: Future energy demand forecast analysis

Finally, the performance of *k*CNN-LSTM for energy consumption forecast problem was validated through a comparative analysis with *k*-means clustering variant of ARIMA, Deep Belief Network (DBN), Multi-Layer Perceptron (MLP), CNN and LSTM. The hyperparameters of *k*CNN-LSTM (momentum, dropout, weight decay, learning rate, strides, filters, etc.) were fine-tuned through the enhanced variant of sine cosine optimization algorithm to achieve better forecast accuracy (Table 4). The hyperparameters of the contrast models were fine-tuned using a sequential grid search algorithm.

The above stated architecture of *k*CNN-LSTM was implemented using Keras framework with Tensorflow backend. The average error values obtained from 30 independent runs were taken for the analysis of *k*CNN-LSTM over *k*-means variant of the state-of-the-art energy demand forecast models. Tables 5–7 provides average error values of the considered machine learning and deep learning models for cluster wise, weekdays, weekends, and selected days of the week in the identified clusters.

From Tables 5–7, it is evident that *k*CNN-LSTM outperforms *k*-means variants of the contrast models. The main reason behind this is that, apart from learning the load trend characterization, *k*CNN-LSTM combines the benefit of the contextual features generated from the timestamp and the temporal information in the historical energy consumption data to achieve better forecast accuracy. The computation time of *k*CNN-LSTM and the considered models reveal that the proposed energy demand forecast model provides the best computational efficiency (Table 8).

Table 8
Computation time analysis.

Models	Computation time (Secs)		
	Cluster 1	Cluster 2	Cluster 3
ARIMA	79	85	34
DBN	97	102	62
MLP	84	95	51
LSTM	88	110	45
CNN	81	105	43
CNN-LSTM	**75**	**90**	**40**

6. Conclusions

This work presented *k*CNN-LSTM, a deep learning framework for

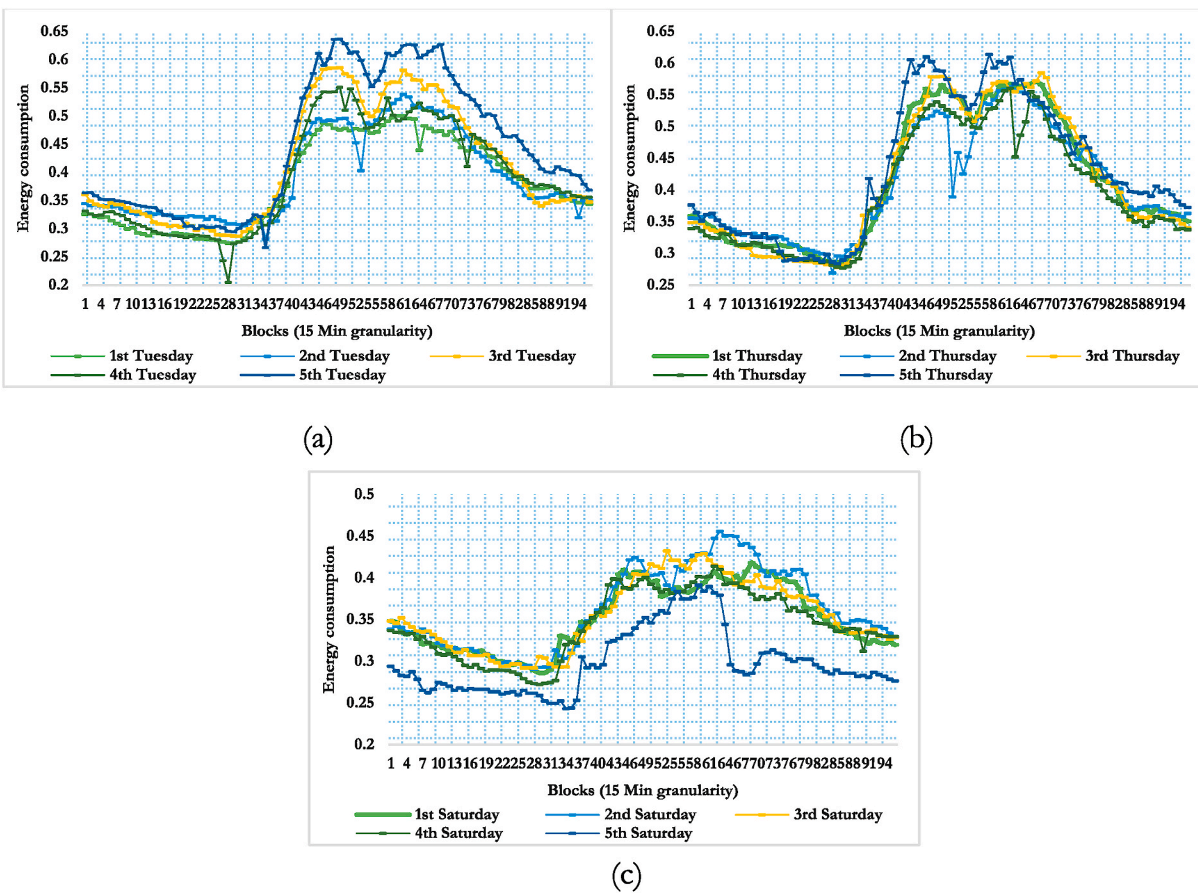
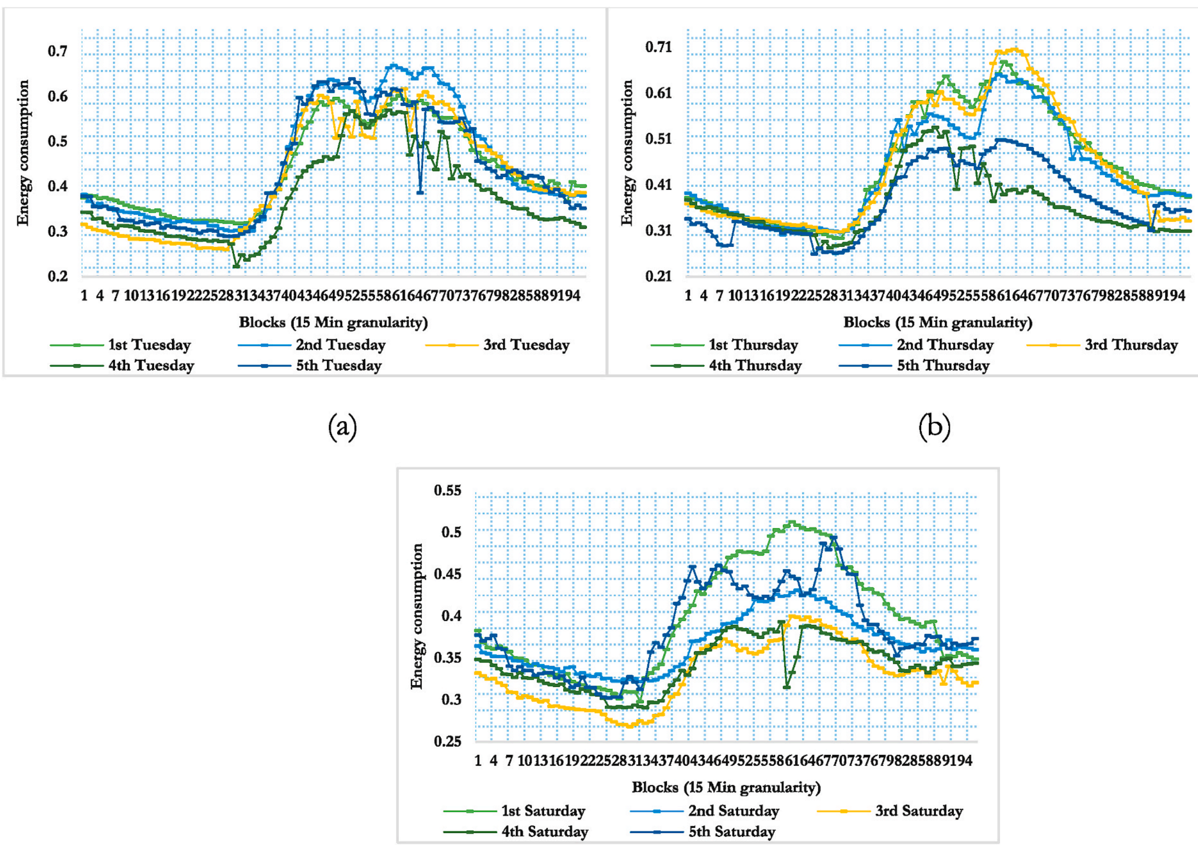


Fig. A1. Energy consumption patterns in Cluster 1 for different days (a) Tuesday (b) Thursday (c) Saturday



(a) (b)

(c)

Fig. A2. Energy consumption patterns in Cluster 2 for different days (a) Tuesday (b) Thursday (c) Saturday

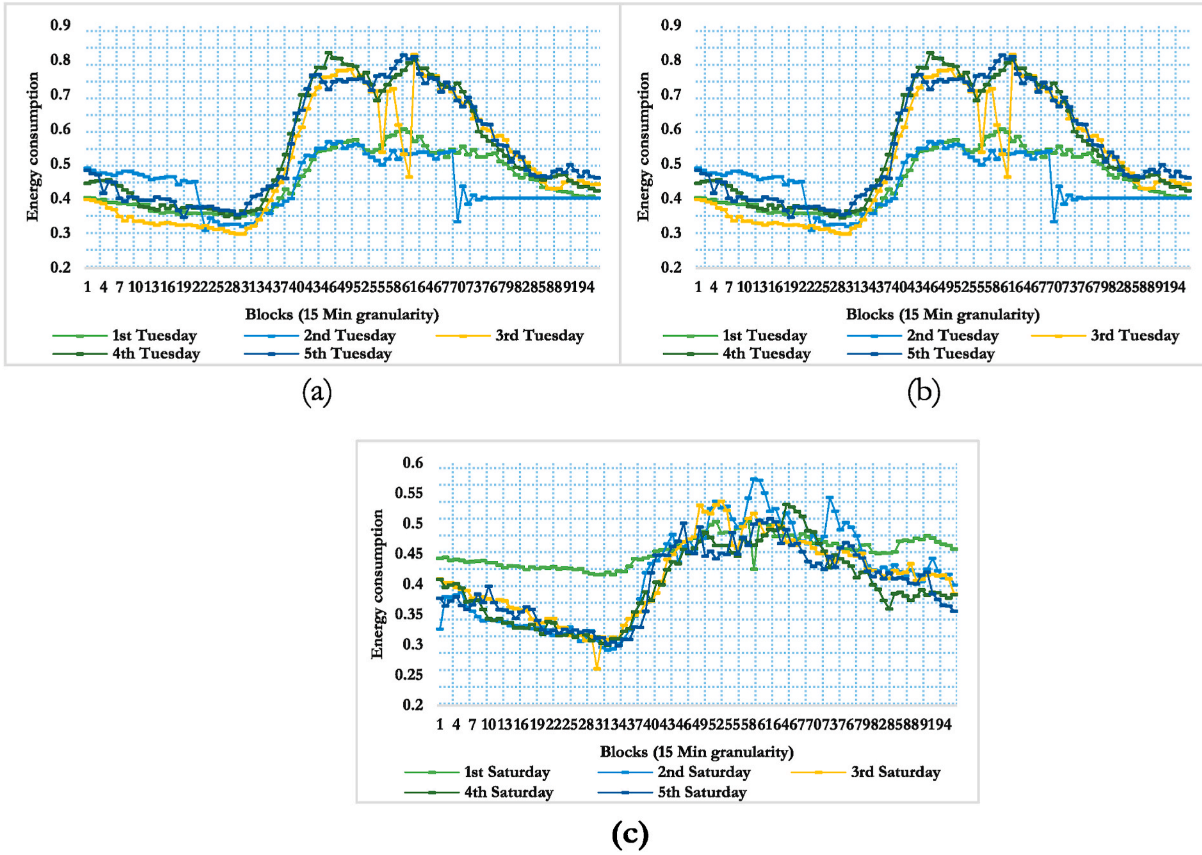


Fig. A3. Energy consumption patterns in Cluster 3 for different days (a) Tuesday (b) Thursday (c) Saturday

robust and reliable building energy consumption forecast. *k*CNN-LSTM employed *k* – means clustering, CNN, and LSTM for trend characterization, energy-related feature identification, and to model temporal information in the energy consumption data. The hyperparameters of *k*CNN-LSTM model were optimized using ISCOA, which uses Haar wavelet based mutation operator to update position, thereby avoiding premature convergence. A case study using the real time building energy consumption data acquired from the BEMS deployed at KReSIT building, IIT-Bombay was presented. The performance of *k*CNN-LSTM was compared with the *k* means variant of the state-of-the-art energy demand forecast models in terms of MSE, RMSE, MAE, and MAPE. The experimental results demonstrate the efficiency of *k*CNN-LSTM model over the existing demand forecast models in providing accurate energy consumption demand forecasting. Therefore, the implementation of *k*CNN-LSTM at the electricity network and user level can aid in decision making, demand management programs, and energy efficiency aspects. The future directions of this research include (i) Application of *k*CNN-LSTM to residential buildings and (iii) Analyze the performance of various optimization algorithms for hyperparameter tuning of *k*CNN-LSTM.

Credit author statement

Nivethitha Somu: Conceptualization, Methodology, Investigation, Data curation, Writing (Original, Review and Editing), and Visualization, Gauthama Raman: Conceptualization, Software, Writing (Review and Editing), Krithi Ramamritham: Conceptualization, Validation, Resources, Writing (Writing – review & editing), Supervision, and Funding acquisition.

Table A1
Cluster 1 – Performance analysis of the state-of-the-art machine learning and deep learning energy demand forecast models (Tuesday, Thursday, and Saturday)

Type	Metrics	Models					
		ARIMA	DBN	MLP	LSTM	CNN	CNN-LSTM
Tuesday	**MSE**	0.0158	0.0113	0.0051	0.0076	0.0065	**0.0045**
	RMSE	0.1257	0.1065	0.0715	0.0870	0.0804	**0.0668**
	MAE	0.1066	0.0778	0.0573	0.0663	0.0563	**0.0442**
	MAPE	0.2674	0.2401	0.1429	0.1886	0.1620	**0.1291**
Thursday	**MSE**	0.0144	0.0178	0.0051	0.0008	0.0016	**0.0005**
	RMSE	0.1202	0.1335	0.0716	0.0277	0.0397	**0.0218**
	MAE	0.0960	0.1239	0.0540	0.0212	0.0333	**0.0174**
	MAPE	0.4005	0.3963	0.1297	0.0619	0.0967	**0.0514**
Saturday	**MSE**	0.0251	0.0122	0.0016	0.0008	0.0005	**0.0005**
	RMSE	0.1586	0.1104	0.0400	0.0276	0.0220	**0.0213**
	MAE	0.1397	0.0872	0.0349	0.0214	0.0170	**0.0156**
	MAPE	0.3280	0.2460	0.1034	0.0708	0.0556	**0.0513**

Table A2
Cluster 2 – Performance analysis of the state-of-the-art machine learning and deep learning energy demand forecast models (Tuesday, Thursday, and Saturday)

Type	Metrics	Models					
		ARIMA	DBN	MLP	LSTM	CNN	CNN-LSTM
Tuesday	**MSE**	0.0087	0.0166	0.0175	0.0089	0.0095	**0.0003**
	RMSE	0.0933	0.1287	0.1323	0.0944	0.0972	**0.0192**
	MAE	0.0566	0.1000	0.0904	0.0585	0.0600	**0.0119**
	MAPE	0.5184	0.4124	0.2506	0.2466	0.2558	**0.1877**
Thursday	**MSE**	0.1003	0.0237	0.0209	0.021	0.0197	**0.0189**
	RMSE	0.1376	0.1538	0.1447	0.1451	0.1405	**0.0192**

(*continued on next page*)

Table A2 (*continued*)

Type	Metrics	Models					
		ARIMA	DBN	MLP	LSTM	CNN	CNN-LSTM
	MAE	0.1119	0.1176	0.1100	0.1105	0.0968	**0.0046**
	MAPE	0.4184	0.4390	0.3958	0.3715	0.4048	**0.2877**
Saturday	**MSE**	0.0048	0.0067	0.0045	0.0064	0.0045	**0.0035**
	RMSE	0.0695	0.0816	0.0672	0.0802	0.0673	**0.0141**
	MAE	0.0489	0.0611	0.0471	0.0602	0.048	**0.0093**
	MAPE	0.1766	0.1787	0.1299	0.1738	0.1484	**0.0604**

Table A3
Cluster 3 – Performance analysis of the state-of-the-art machine learning and deep learning energy demand forecast models (Tuesday, Thursday, and Saturday)

Type	Metrics	Models					
		ARIMA	DBN	MLP	LSTM	CNN	CNN-LSTM
Tuesday	**MSE**	0.1040	0.0463	0.0333	0.0298	0.2187	**0.0222**
	RMSE	0.2640	0.2151	0.1824	0.1726	0.1568	**0.1489**
	MAE	0.3250	0.1636	0.1300	0.1353	0.2152	**0.1218**
	MAPE	0.4844	0.3006	0.3458	0.2503	0.2404	**0.2314**
Thursday	**MSE**	0.2029	0.0429	0.0532	0.0234	0.1154	**0.0126**
	RMSE	0.1546	0.2070	0.2306	0.1531	0.1741	**0.1123**
	MAE	0.1212	0.1608	0.1671	0.1160	0.8047	**0.0883**
	MAPE	0.3397	0.3125	0.5654	0.2409	0.8758	**0.2138**
Saturday	**MSE**	0.0108	0.0209	0.0257	0.0130	0.1077	**0.0081**
	RMSE	0.1285	0.1445	0.1603	0.1139	0.1880	**0.0900**
	MAE	0.2159	0.1155	0.1328	0.0906	0.0681	**0.0662**
	MAPE	0.2617	0.2544	0.4789	0.1918	0.1574	**0.1480**

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank The Indian Institute of Technology-Bombay (Institute Postdoctoral Fellowship-AO/Admin-1/Rect/33/2019); The Ministry of Human Resource and Development, India – Impacting Research Innovation and Technology (IMPRINT - 16MOPIMP002), New Delhi, India; Prof. Kannan Krithivasan, Dean, School of Education, SASTRA Deemed University, Tamil Nadu, India (TATA Realty—SASTRA Srinivasa Ramanujan Research Cell, India) and SEIL members.

References

[1] Yang Y, Chen Y, Wang Y, Li C, Li L. Modelling a combined method based on ANFIS and neural network improved by DE algorithm: a case study for short-term electricity demand forecasting. Appl Soft Comput J 2016;49:663–75. https://doi.org/10.1016/j.asoc.2016.07.053.
[2] Bui DK, Nguyen TN, Ngo TD, Nguyen-Xuan H. An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings. Energy 2019:116370. https://doi.org/10.1016/j.energy.2019.116370.
[3] Wang W, Hong T, Xu X, Chen J, Liu Z, Xu N. Forecasting district-scale energy dynamics through integrating building network and long short-term memory learning algorithm. Appl Energy 2019;248:217–30. https://doi.org/10.1016/j.apenergy.2019.04.085.
[4] 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. https://doi.org/10.1016/j.apenergy.2014.02.057.
[5] Conti J, Holtberg P, Diefenderfer J, LaRose A, Turnure JT, Westfall L. International energy outlook 2016 with projections to 2040. 2016. https://doi.org/10.2172/1296780.
[6] Amber KP, Ahmad R, Aslam MW, Kousar A, Usman M, Khan MS. Intelligent techniques for forecasting electricity consumption of buildings. Energy 2018;157:886–93. https://doi.org/10.1016/j.energy.2018.05.155.
[7] Chou JS, Tran DS. Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. Energy 2018;165:709–26. https://doi.org/10.1016/j.energy.2018.09.144.
[8] Ocampo Batlle EA, Escobar Palacio JC, Silva Lora EE, Martínez Reyes AM, Melian Moreno M, Morejón MB. A methodology to estimate baseline energy use and quantify savings in electrical energy consumption in higher education institution buildings: case study, Federal University of Itajubá (UNIFEI). J Clean Prod 2019. https://doi.org/10.1016/j.jclepro.2019.118551.
[9] Neto AH, Fiorelli FAS. Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. Energy Build 2008;40:2169–76. https://doi.org/10.1016/j.enbuild.2008.06.013.
[10] Afzal Muhammad, Huang Qi, Amin Waqas, Umer Khalid, Raza Asif, Blockchain MN. Enabled distributed demand side management in community energy system with smart homes. IEEE Access 2020;8:37428–39.
[11] Global demand for energy will peak in 2030, says World Energy Council | Business | The Guardian n.d.
[12] Hassan S, Khosravi A, Jaafar J, Khanesar MA. A systematic design of interval type-2 fuzzy logic system using extreme learning machine for electricity load demand forecasting. Int J Electr Power Energy Syst 2016;82:1–10. https://doi.org/10.1016/j.ijepes.2016.03.001.
[13] Yaslan Y, Bican B. Empirical mode decomposition based denoising method with support vector regression for time series prediction: a case study for electricity load forecasting. Meas J Int Meas Confed 2017;103:52–61. https://doi.org/10.1016/j.measurement.2017.02.007.
[14] Deb C, Zhang F, Yang J, Lee SE, Shah KW. A review on time series forecasting techniques for building energy consumption. Renew Sustain Energy Rev 2017;74:902–24. https://doi.org/10.1016/j.rser.2017.02.085.
[15] Xiao J, Li Y, Xie L, Liu D, Huang J. A hybrid model based on selective ensemble for energy consumption forecasting in China. Energy 2018;159:534–46. https://doi.org/10.1016/j.energy.2018.06.161.
[16] Wei N, Li C, Peng X, Zeng F, Lu X. Conventional models and artificial intelligence-based models for energy consumption forecasting: a review. J Petrol Sci Eng 2019;181:106187. https://doi.org/10.1016/j.petrol.2019.106187.
[17] Ciulla G, D'Amico A. Building energy performance forecasting: a multiple linear regression approach. Appl Energy 2019;253:113500. https://doi.org/10.1016/j.apenergy.2019.113500.
[18] Liu T, Tan Z, Xu C, Chen H, Li Z. Study on deep reinforcement learning techniques for building energy consumption forecasting. Energy Build 2020;208:109675. https://doi.org/10.1016/j.enbuild.2019.109675.
[19] Tran D, Luong D, Chou J. Nature-inspired metaheuristic ensemble model for forecasting energy consumption in residential buildings. Energy 2019:116552. https://doi.org/10.1016/j.energy.2019.116552.
[20] Raza MQ, Khosravi A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renew Sustain Energy Rev 2015;50:1352–72. https://doi.org/10.1016/j.rser.2015.04.065.
[21] Mat Daut MA, Hassan MY, Abdullah H, Rahman HA, Abdullah MP, Hussin F. Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: a review. Renew Sustain Energy Rev 2017;70:1108–18. https://doi.org/10.1016/j.rser.2016.12.015.
[22] Amasyali K, El-gohary NM. A review of data-driven building energy consumption prediction studies. Renew Sustain Energy Rev 2018;81:1192–205. https://doi.org/10.1016/j.rser.2017.04.095.
[23] Fayaz M, Kim D. A prediction methodology of energy consumption based on deep extreme learning machine and comparative analysis in residential buildings. Electron 2018;7. https://doi.org/10.3390/electronics7100222.
[24] Malik S, Kim DH. Prediction-learning algorithm for efficient energy consumption in smart buildings based on particle regeneration and velocity boost in particle swarm optimization neural networks. Energies 2018;11. https://doi.org/10.3390/en11051289.
[25] Bedi J, Toshniwal D. Empirical mode decomposition based deep learning for electricity demand forecasting. IEEE Access 2018;6:49144–56. https://doi.org/10.1109/ACCESS.2018.2867681.
[26] Li C, Ding Z, Yi J, Lv Y, Zhang G. Deep belief network based hybrid model for building energy consumption prediction. Energies 2018;11:1–26. https://doi.org/10.3390/en11010242.
[27] Moon J, Park J, Hwang E, Jun S. Forecasting power consumption for higher educational institutions based on machine learning. J Supercomput 2018;74:3778–800. https://doi.org/10.1007/s11227-017-2022-x.
[28] Touzani S, Granderson J, Fernandes S. Gradient boosting machine for modeling the energy consumption of commercial buildings. Energy Build 2018;158:1533–43. https://doi.org/10.1016/j.enbuild.2017.11.039.
[29] 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. https://doi.org/10.1016/j.apenergy.2017.12.051.
[30] Ye Z, Kim MK. Predicting electricity consumption in a building using an optimized back-propagation and Levenberg–Marquardt back-propagation neural network: case study of a shopping mall in China. Sustain Cities Soc 2018;42:176–83. https://doi.org/10.1016/j.scs.2018.05.050.
[31] Yuan T, Zhu N, Shi Y, Chang C, Yang K, Ding Y. Sample data selection method for improving the prediction accuracy of the heating energy consumption. Energy Build 2018;158:234–43. https://doi.org/10.1016/j.enbuild.2017.10.006.
[32] Divina F, Gilson A, Goméz-Vela F, Torres MG, Torres JF. Stacking ensemble learning for short-term electricity consumption forecasting. Energies 2018;11:1–31. https://doi.org/10.3390/en11040949.

[33] Ahmad T, Chen H. Utility companies strategy for short-term energy demand forecasting using machine learning based models. Sustain Cities Soc 2018;39:401–17. https://doi.org/10.1016/j.scs.2018.03.002.
[34] Goudarzi S, Anisi MH, Kama N, Doctor F, Soleymani SA, Sangaiah AK. Predictive modelling of building energy consumption based on a hybrid nature-inspired optimization algorithm. Energy Build 2019;196:83–93. https://doi.org/10.1016/j.enbuild.2019.05.031.
[35] Bedi J, Toshniwal D. Deep learning framework to forecast electricity demand. Appl Energy 2019;238:1312–26. https://doi.org/10.1016/j.apenergy.2019.01.113.
[36] Kim Y, Son H gu, Kim S. Short term electricity load forecasting for institutional buildings. Energy Rep 2019;5:1270–80. https://doi.org/10.1016/j.egyr.2019.08.086.
[37] Dagdougui H, Bagheri F, Le H, Dessaint L. Neural network model for short-term and very-short-term load forecasting in district buildings. Energy Build 2019;203. https://doi.org/10.1016/j.enbuild.2019.109408.
[38] Wen L, Zhou K, Yang S. Load demand forecasting of residential buildings using a deep learning model. Elec Power Syst Res 2020;179:106073. https://doi.org/10.1016/j.epsr.2019.106073.
[39] Wei Y, Zhang X, Shi Y, Xia L, Pan S, Wu J, et al. A review of data-driven approaches for prediction and classification of building energy consumption. Renew Sustain Energy Rev 2018;82:1027–47. https://doi.org/10.1016/j.rser.2017.09.108.
[40] Molina-Solana M, Ros M, Ruiz MD, Gómez-Romero J, Martin-Bautista MJ. Data science for building energy management: a review. Renew Sustain Energy Rev 2017;70:598–609. https://doi.org/10.1016/j.rser.2016.11.132.
[41] Kim TY, Cho SB. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 2019;182:72–81. https://doi.org/10.1016/j.energy.2019.05.230.
[42] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
[43] Ghimire S, Deo RC, Raj N, Mi J. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. Appl Energy 2019;253:113451–70. https://doi.org/10.1016/j.apenergy.2019.113541.
[44] Chen Y, Zhang S, Zhang W, Peng J, Cai Y. Multifactor spatio-temporal correlation model based on a combination of convolutional neural network and long short-term memory neural network for wind speed forecasting. Energy Convers Manag 2019;185:783–99. https://doi.org/10.1016/j.enconman.2019.02.018.
[45] Song X, Liu Y, Xue L, Wang J, Zhang J, Wang J, et al. Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model. J Petrol Sci Eng 2019;186:106682. https://doi.org/10.1016/j.petrol.2019.106682.
[46] Bai Y, Zeng B, Li C, Zhang J. An ensemble long short-term memory neural network for hourly PM2.5 concentration forecasting. Chemosphere 2019;222:286–94. https://doi.org/10.1016/j.chemosphere.2019.01.121.
[47] Yann LeCun YBGH. Deep learning 2015:436–44. https://doi.org/10.1038/nature14539.
[48] Li J, Li X, He D. A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction. IEEE Access 2019;7:75464–75. https://doi.org/10.1109/ACCESS.2019.2919566.
[49] Lu X, Lin P, Cheng S, Lin Y, Chen Z, Wu L, et al. Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph. Energy Convers Manag 2019;196:950–65. https://doi.org/10.1016/j.enconman.2019.06.062.
[50] Sadaei HJ, de Lima e Silva PC, Guimarães FG, Lee MH. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. Energy 2019;175:365–77. https://doi.org/10.1016/j.energy.2019.03.081.
[51] Aghabozorgi S, Ying Wah T, Herawan T, Jalab HA, Shaygan MA, Jalali A. A hybrid algorithm for clustering of time series data based on affinity search technique. Sci World J 2014;2014. https://doi.org/10.1155/2014/562194.
[52] Minnaar A. Time series classification and clustering with Python | alex minnaar's blog. 2014.
[53] Rath TM, Manmatha R. Lower-bounding of dynamic time warping distances for multivariate time series 2 . Multivariate time series extension. 2002.
[54] Karydis I, Nanopoulos A, Papadopoulos AN, Manolopoulos Y. Evaluation of similarity searching methods for music data in P2P networks. Int J Bus Intell Data Min 2005;1:210–28. https://doi.org/10.1504/IJBIDM.2005.008363.
[55] Somu N, GR MR, Ramamritham K. A hybrid model for building energy consumption forecasting using long short term memory networks. Appl Energy 2020;261:114131. https://doi.org/10.1016/j.apenergy.2019.114131.
[56] Laboratory SEI. Academic building dataset. n.d. http://seil.cse.iitb.ac.in/datasets/. [Accessed 4 December 2019].