Chapter 7

Intelligent Modeling and Prediction of Heat Energy Consumption for Smart Buildings

Heat energy load prediction in smart buildings is critical for saving energy and meeting energy demands [170]. As the cost of energy and demand is increasing, people must find some way to control them and save energy [171]. To overcome these issues, an intelligent energy management system must be developed to help individuals predict heat energy demand and analyze energy use patterns. This chapter introduces a data-driven short-term heat energy consumption prediction for smart buildings by developing a hybrid deep learning model considering the capabilities of Convolutional and Recurrent Neural Networks. The proposed model extracts hidden patterns from actual data collected through sensors and uses hand-crafted features to predict heat energy demand accurately by considering three types of buildings: single-family, apartment, and terraced houses. The results are promising and outperformed other state-of-the-art methods.

7.1 Introduction

Buildings account for over one-third of worldwide energy consumption, with energy being the primary resource required for efficient operation [172]. Carbon emissions from energy consumption impact the environment and system development [173]. A data-driven approach can minimize building energy use and carbon emissions [174]. The growth of energy demand greatly challenges energy generation and availability. So, energy efficiency and reliability are very important to execute optimal energy generation and demand management strategies. Hence, the electrical and heating systems are integrated into a smart grid system. Predicting demand for power and heating is crucial for energy efficiency [102]. The heat load prediction is the heat energy required to maintain an indoor room temperature that is comfortable for a person or family in a building or room. This heat load forecast considers several elements, including room size, internal temperature, exterior temperature, and occupancy. Heat load prediction models are vital in optimizing and lowering energy consumption for smart buildings.

According to [175], the building sector is the world's largest energy consumption and contributor to carbon emissions. It is critical to reduce building energy consumption and carbon emissions. Heat load prediction is crucial for HVAC systems [176, 177], energy distribution systems [178], and smart grid systems [179]. A smart building provides low-cost services such as air conditioning, ventilation, heating, and other services to its occupants while not negatively impacting the environment. The primary goal of a smart building is to provide the highest level of comfort while reducing thermal energy use and maximizing energy efficiency. It has been discovered that buildings use a lot of energy. According to the Clean Air Policy, buildings account for nearly 40% of global energy demand. At the same time, Eurostat reports that buildings account for 38.1% of energy demand in the European Union (EU), with heating demand accounting for 55% [104].

This data highlights the importance of an efficient energy management system in

saving energy [180]. Heat demand is the energy needed to make the indoor temperature comfortable enough for a person in a building. Heat demand is calculated using criteria such as space size, insulation levels, external temperature, and other variables that affect heat loss or gain. The heat demand component might change seasonally. The need for heating rises during colder months, such as winter, and declines during the summer. Monitoring and regulating the heat demand component is critical to energy conservation initiatives. Optimizing heating systems based on precise heat demand assessments can increase energy efficiency and cost savings.

The forecasting of thermal heat loads is divided into three categories: white-box physics-based models, gray-box reduced-order models, and black-box data-driven models. White-box models forecast building heat loads using software tools like EnergyPlus, Dest, and TRNSYS [181]. Gray-box models transform building thermal dynamics into lower-order Resistance and Capacity (RC) models. To reduce prediction error, these values (RC) are obtained from data rather than creating physical parameters as in white box models [182]. On the other hand, Black-box models use data to anticipate thermal load based on historical data. The amount of data used in these models is huge; therefore, it is possible to understand heat load patterns from historical data, and by using these learning patterns, predictions become easy.

Monitoring the available Heating, Ventilation, and Air Conditioning (HVAC) systems is important to reduce energy consumption [105]. Therefore, forecasting the heat energy demand in smart buildings is highly useful in improving the efficiency of building operations and the optimal energy management system. These improvements in building energy management systems can save energy from 7% to 52% [106] [107]. A heat generation profile and demand forecast are critical for accurate heat load characterization and efficient utilization of available sources to save energy and money. In buildings, heat meters are increasingly used to measure heating demand correctly and assess the home's energy use [108].

In recent times, deep learning techniques have shown their efficiency in various domains such as medical [183, 184], transportation [185], industrial [186], home automation [187, 93]. With the rise of the deep learning age, the deep neural network has become an indispensable tool for heat load prediction [188]. Apart from forecasting anything, it is required to identify which features are relevant in prediction. Time-related aspects like hour of the day, weekday and weekend projection affect occupancy prediction, heat load prediction, and other building-related usage schedules like weather factors and temperature set point, etc.

7.1.1 Motivation and Contributions

Energy demand forecasting is critical, particularly for the smooth operation of smart buildings. It has been noted that the needs for heat and electricity increase to a comparable degree during the winter [171]; however, electricity demand for cooling surges, while heat demand falls in the summer. Excess heat is generated because of the reduced heat demand for at least three months of the year, which is difficult to store. Because of these climate factors, Korea endures economic losses that are less efficient than Combined Heat and Power (CHP) operations [189]. As a result, further study into heat demand and load forecasting models is required [190].

The primary contributions of this chapter are outlined as follows.

1. An efficient hybrid deep learning model has been developed by utilizing the capabilities of CNN and RNN for predicting heat energy in smart buildings. In our approach, these CNN and RNN are merged in parallel to capture data points' spatial and temporal relationships to forecast heat energy.
2. Consider the case study of smart heat meters installed in Danish residential buildings for three years. This work defined two scenarios for analysis: 1) Short–term heat energy demand forecast for specific types of houses and 2) Short-term heat energy demand prediction for individual residences.

3. Improve the model efficiency by defining hand-crafted features such as minimum, maximum, and average consumption within a defined duration and current hour, weekday, and month.
4. Using a Danish residential building dataset, we comprehensively evaluate the proposed model over the prediction of 1-day ahead and 7-day ahead time horizons. The results show the efficiency and desirable performance of the proposed model, which outperforms other state-of-the-art methods in terms of RMSE, MAE, R2, and MAPE.

7.2 Data Collection and Data Preprocessing

This section discusses the data set utilized for the proposed study and its data preprocessing, such as data cleaning, normalization, etc.

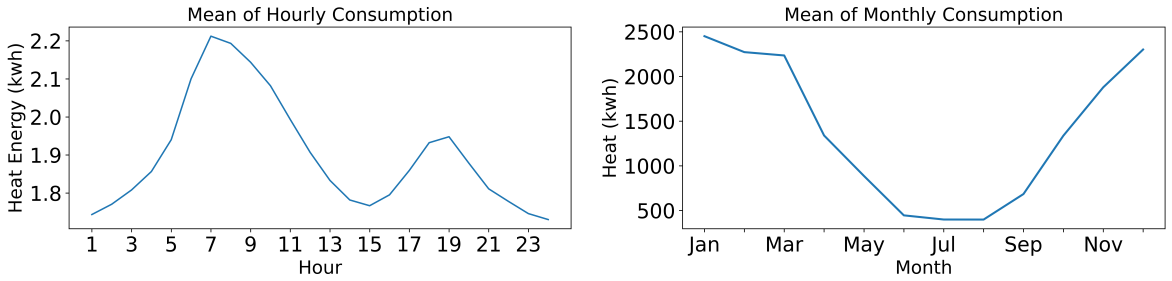
7.2.1 Data Collection

An open-source smart heat meters dataset [191] has been used to construct the proposed forecasting model. The data was gathered via smart heat meters connected to district heat networks at a 1-hour resolution. Three years of data were recorded from 03-01-2018 to 31-12-2020, considering 3021 smart heat meters installed at a Danish residential building [191]. The available dataset was preprocessed from the raw dataset using the weighted moving average imputation approach, with a window size 48. The dataset covers three types of buildings: single-family, terraced, and apartment house. Four parameters, including energy heat in kWh, volume flow in m^3, inlet flow energy in $m^{3\circ}C$, and backflow energy in $m^{3\circ}C$, were measured over the specified period. Furthermore, to calculate real consumption per hour, the data was processed for each given set of attributes by taking the sequential difference of entries in the original processed dataset. The extraction of additional data, such as the current hour, weekday, and month of the year, is also considered. We investigated two scenarios to estimate thermal energy

consumption for various types of homes.

- **Scenario 1:** The objective of this scenario is to predict the overall energy consumption for any particular kind of house. Therefore, the mean of all houses for a particular type, i.e., single-family houses, apartments, and terraced houses, has been taken. Moreover, 24-hours ahead and 168-hours ahead predictions have been considered for each kind of building. In this scenario, the houses have been separated based on their type, i.e., apartment, terraced, and single-family. Let's say the data for each building has been recorded in a file, and data points for multiple houses have been combined by taking their mean and storing them in another file. The data points in the file have been represented hourly by considering all required attributes. As data has been recorded for three years (36 months), we have split the dataset into (80-20) ratio, which means two years and five months of data for training (80% means 29 months of total data) and seven months of data for testing (20% means 7months of data).
- **Scenario 2:** The objective of this scenario is to predict the energy for an individual house/building. This scenario considers the apartment house data with predictions of 1-day- and 7-day ahead. In this scenario, the data has been prepared by appending all the data files of a particular building type. Here, data has been portioned in the ratio of (60-40), which means 60% of total files (all the houses data of a particular building type) have been considered for training and the rest 40% considered for testing. There are 88 houses of apartment types, so data of 53 houses for training and 35 for testing have been considered.

This work developed the model for short-term heat energy forecasting, considering prediction for 24 hours and prediction of 168 hours ahead, and analyzed the performance of the proposed model. Figure 7.1(a) depicts the average consumption of all households over 24 hours. Figure 7.1(b) depicts the average monthly thermal energy usage for all houses over a 12-month period. Figure 7.1(b) shows that heat energy



(a) (b)

Figure 7.1: Average of all houses heat energy consumption (a) Hourly wise and (b) Monthly wise heat energy consumption

consumption is much lower in June, July, and August for all residences (single-family houses, terraced houses, and apartment houses) compared to December, January, and February. Denmark's seasonal calendar shows that the last three months are summer, from June to August. Peak winter occurs in December, January, and February; the remaining months are spent transitioning between the two seasons.

The heat energy consumption for different buildings, such as a single-family house (Figure 7.2(a)), apartment (Figure 7.2 (b)), and terraced house (Figure 7.2 (c)) for three years, is shown in Figure 7.2. From Figure 7.2, it is observed that a surge of heat energy consumption in the early morning (6 a.m) and a little less surge in the evening (6 p.m), i.e., during sunrise and sunset time according to Denmark's time zone. All these patterns are helpful for the model. Therefore, the features such as the current hour, weekdays, and months are also included in model training. Some other hand-crafted features are also considered, i.e., minimum, maximum, and average consumption for a particular day or week.

A set of attributes/ features has been taken into consideration are described below:

- **Heat Energy:** Heat Energy in kWh (hourly) and minute wise, which is resampled per day for scenario 2.
- **Volume flow:** The volume of heat energy supplied through the pipe is measured in $m^{3\circ}C$.

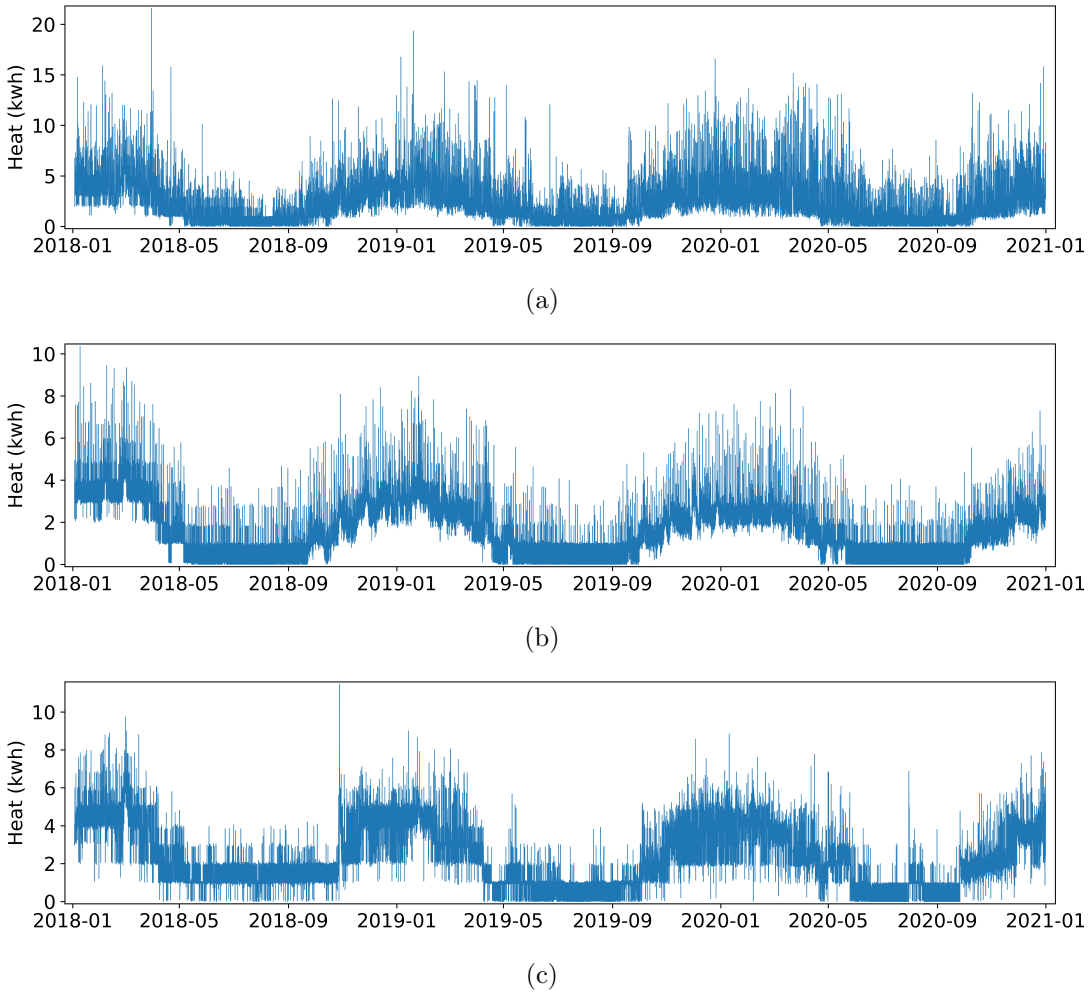


Figure 7.2: Heat energy consumption for different buildings. Heat energy is on the y-axis and represented in kWh, and the date is on the x-axis. Subfigures (a) single-family house, (b) Terraced house, (c) Apartment house

- **Inlet flow energy:** It is the product of the volume of heat energy supplied with supply temperature in $m^3{}^\circ C$.
- **Back flow energy:** This is the product of the volume of heat energy supplied with return temperature in $m^3{}^\circ C$.
- **Hour of the day:** The energy consumption varies the entire day according to usage; hence, it is an essential component used in scenario 1.
- **Week Day:** The day is also useful to predict heat energy consumption because the building can consume less energy on the weekend. So, mapping each weekday

with an integer is done by Monday as 0 to Sunday as 6.

- **Month of the day:** The other attribute is the month of the day, which is a vital attribute to know the heat energy consumption for different seasons.

7.2.2 Data Preprocessing

Data preprocessing, such as cleaning and normalizing data and selecting important features, is done in this section to improve the prediction results and the accuracy of the proposed model. Data was normalized with a z-score during preprocessing to reduce prediction errors. The z-score is formally defined as:

$$\bar{X} = \frac{x - \mu}{\sigma} \tag{7.1}$$

where x represent the original value, μ and σ denote the mean and standard deviation respectively, $\bar{X}$ denote the normalized value. Finally, some important handcraft features have been extracted to improve the accuracy of the model prediction.

7.3 Proposed Methodology

A hybrid deep learning model for efficient thermal energy prediction has been built by integrating CNN and RNN networks. CNN is a deep-learning neural network that helps extract hidden properties from data. The RNN is another type of neural network that aids in capturing temporal correlations in data points. In RNN, the input for the current state is determined by the output of the previous step. Furthermore, the hidden state is the most crucial in RNN since it aids in retaining information in sequential order. The suggested method employs CNN and RNN to forecast the next value at a given time. However, the primary issue with RNN is the vanishing gradient descent problem.

As a result, longer sequence data cannot be captured. Gating mechanisms are built into LSTM and GRU to avoid vanishing gradient concerns. The LSTM is a form of

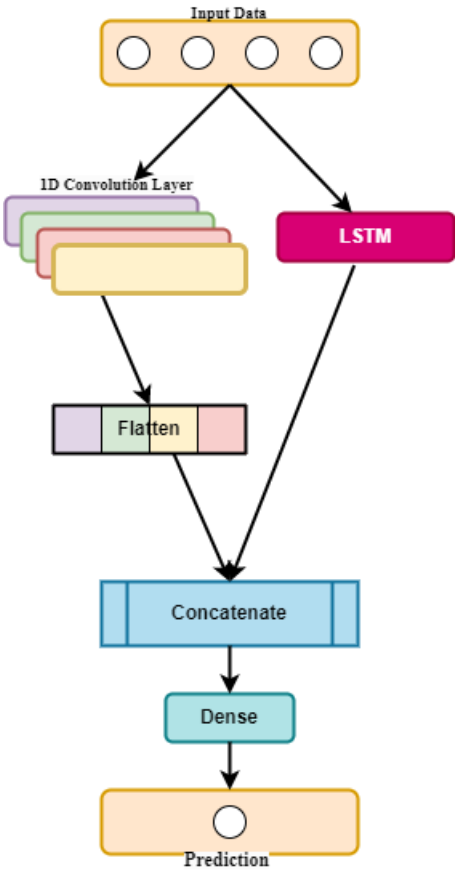


Figure 7.3: Proposed Parallel CNN-RNN Model.

RNN that processes sequential data such as time series, text, and audio. LSTM may learn long-term dependencies in sequential data, which aids in time series forecasting. The LSTM consists of a cell, an input gate, an output gate, and a forget gate. The cell state remembers the values over an arbitrary time interval. Information flows into the cell and comes out using the three gates. The input gate determines which data will be retained in the current state. The output gate indicates that information will be transferred to the output from the current state, considering both the current and prior states. The forget gate specifies which data should be overlooked or erased from the present state. The proposed work is shown in Figure 7.3.

The mathematical description of the forget gate (f_t), input gate (i_t), output gate (o_t), cell input activation vector (s_t), cell state (c_t) and hidden state (h_t) are as follows.

$$f_t = \sigma\left(w_f p_t + w_f h_{t-1} + b_f\right) \tag{7.2}$$

$$i_t = \sigma\left(w_i p_t + w_i h_{t-1} + b_i\right) \tag{7.3}$$

$$o_t = \sigma\left(w_o p_t + w_o h_{t-1} + b_o\right) \tag{7.4}$$

$$s_t = \sigma\left(w_c p_t + w_c h_{t-1} + b_c\right) \tag{7.5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot s_t \tag{7.6}$$

$$h_t = o_t \odot \sigma\left(c_t\right) \tag{7.7}$$

where σ is the sigmoid activation function, p_t is the LSTM input vector, h_{t-1} is the hidden state vector, w_f, w_i, w_o, and w_c are weight metrics corresponding to forget gate, input gate, output gate, and cell input activation vector; the b_f, b_i, b_o and b_o are bias corresponding to forget gate, input gate, output gate, and cell input activation vector and the c_{t-1} is cell input vector. The GRU has fewer gates than LSTM, and its architecture is simpler than LSTM. It consists of two gates and one state compared to the three gates and two states of LSTM.

The mathematical description of each gate, such as update gate (z_t), reset gate (r_t), and hidden state (h_t) of GRU, is as follows.

$$z_t = \sigma(w_z p_t + w_f h_{t-1} + b_z) \tag{7.8}$$

$$r_t = \sigma(w_r p_t + w_r h_{t-1} + b_r) \tag{7.9}$$

Algorithm 7.1 Training process in **Scenario 1**

Data:F=$\{f^i; 1 \leq i \leq N\}$: *Set of data files*
$f^i \in R^{T \times A}$: $T \rightarrow$ *Total data points,* $A \rightarrow$ *Number of attributes*
$W \in \{$ 2, 4, 8, 12, 24$\}$: *Sample window size*
$Q \in \{Q1, Q2\}$: $Q1 \rightarrow$ 24 hours ahead, $Q2 \rightarrow$ 168 hours ahead
Result :Trained model **M**
/*Initialization*/
X[][] = 0, Model **M**, trainD = {}
for *i=1 to N* **do**
  f= read(file f^i)
  for *j=1 to T* **do**
    for *k=1 to A* **do**
      X[j][k] = X[j][k] + f[j][k]
    end
  end
end
for *j=1 to T* **do**
  for *k=1 to A* **do**
    X[j][k] = X[j][k]/ N
  end
end
/* Split the data set into 80-20 ratio */
$trainX$ = X[: 0.8*T][:]
$testX$ = X[0.8*T :][:]
/* Preparation of training data */
for *i=1 to T-(W+Q-1)* **do**
  x = trainX[i : (i+W-1)][:]
  y = trainX[(i+W+Q-1)][1]
  trainD = trainD $\cup$ $\{(x, y)\}$
end
M = trainModel(**M**, trainD)

$$s_t = \phi_n(w_h p_t + w_h(r_t \odot h_{t-1}) + b_h) \tag{7.10}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot s_t \tag{7.11}$$

where σ is the sigmoid activation function, p_t is the input vector, h_{t-1} is the hidden state vector, the w_z, w_r and w_h are weight metrics corresponding to the update gate, reset gate, and candidate activation vector, the b_z, b_r and b_h are bias corresponding to the update gate, reset gate, and candidate activation vector and h_t is output vector.

Scenario 1 and Scenario 2 are represented in Algorithm 1 and Algorithm 2, respectively.

Algorithm 7.2 Training process in **Scenario 2**

Data: F=$\{f^i; 1 \leq i \leq N\}$: *Set of data files*
$f^i \in R^{T \times A}$: $T \leftarrow$ *Total data points,* $A \leftarrow$ *Number of attributes*
$W \in \{$ 2, 4, 8, 12, 24$\}$: *Sample window size*
$Q \in \{Q1, Q2\}$: $Q1 \rightarrow$ 1 day ahead, $Q2 \rightarrow$ 7 day ahead
Result: Trained model **M**
/*Initialization*/
X[][] = 0, model **M**, trainD = {}
/*Re-sampled the data at 24-hour intervals*/
for *i=1 to N* **do**
  f= read(f^i)
  d=0
  while $(d + 24) \leq T$ **do**
    for *k=1 to A* **do**
      for *j=d to (d+24)* **do**
        X[j][k] = X[j][k] + f[j][k]
      end
    end
    d = d+24
  end
  write(f^i, X)
end
/* Split the data for training and testing */
trainF, testF = split(F, train=0.6)
for *i=1 to len(trainF)* **do**
  trainX= read(f^i)
  for *i=1 to T-(W+Q-1)* **do**
    x = trainX[i : (i+W-1)][:]
    y = trainX[(i+W+Q-1)][1]
    trainD = trainD $\cup$ $\{(x, y)\}$
  end
end
M = trainModel(**M**, trainD)

In this proposed work, the hybrid combination of two deep learning techniques is done in parallel. Data is concurrently provided to the one-dimensional convolution and LSTM layers. The convolution layer retrieves hidden features from the data, which are then passed to flatten the layer. Parallelly, the LSTM will discover hidden patterns in

the data. Finally, the data will be concatenated and fed into a dense layer to anticipate heat energy consumption. The proposed parallel CNN-LSTM combines the input layer, four convolution layers, one flatten layer, and one LSTM layer in parallel. Finally, both are concatenated before combining with a dense layer for prediction. The mathematical representations of the proposed parallel CNN-LSTM are shown below.

$$m_p^l = \sigma\left(\sum_{x=1}^{N_{l-1}} conv1d\left(\gamma_{xj}^l, \delta_x^{l-1}\right) + b_l^p\right) \prod LSTM \tag{7.12}$$

where γ_{xj}^l is the kernel from the x^{th} neuron at layer l to the j^{th} neuron, the δ_x^{l-1} is the output of x^{th} neuron at layer l-1, the b_l^p is the bias of the l^{th} neuron at layer p, the $conv1d$ is *1d* convolution layer without zero padding, N is a total number of data points, and the $\prod LSTM$ represents the parallel LSTM. The σ is the ReLU activation function and is defined as follows.

$$\sigma(\delta) = \begin{Bmatrix} 0 & \forall\delta < 0 \\ \delta & \forall\delta \geq 0 \end{Bmatrix} \tag{7.13}$$

The model takes input data for a particular lag window, i.e., 2, 4, 8, 12, 24. Filters represent the number of output channels, whereas kernels represent the size of the convolution filter. Filter and kernels are set to two. The activation function employed in this suggested work is ReLU, which processes the input and generates the output. A single-layer LSTM has 16 neurons and a ReLU activation function. The Adam optimizer with a learning rate 0.001 and mean absolute error is utilized as the loss function.

As discussed, we analyzed in two perspectives: **Scenario 1** and **Scenario 2**. Assume each home type in the dataset includes N files. Each file is defined as ($f \in R^{T\times A}$), where T is the number of data points in the file, and A is the set of attributes such as heat energy, energy backflow, etc. The dataset is specified as $F = f^i; i \in [1..N]$. The detailed training process for Scenario 1 and Scenario 2 is available in **Algorithm 7.1** and **Algorithm 7.2**. The first part of **Algorithm 7.1** calculates the total of all data

points for all defined timestamps. The second segment calculates the average energy use. The third component explains the data partitioning and sample preparation for training the model, while the last segment denotes the model training. A similar analogy, such as initializing variables, resampling data at 24-hour intervals, splitting the data, and training the data for prediction, is described in **Algorithm 7.2**.

7.4 Experimental Setup

This section includes a full overview of several performance evaluation measures used for comparison purposes. The experiments were conducted on an Intel-Xeon Skylake supercomputer node with 2.4GHz 6148 CPU cores and 192 GB of RAM. In our experiment, various lags in hours (2h, 4h, 8h, 12h and 24h) have been considered.

7.4.1 Evaluation Criteria

Four metrics, MAE, RMSE, R^2, and MAPE are used to compare and evaluate the models, which are as follows. Let $\hat{p_i}$ and p_i signify the predicted and actual values of the i^{th} sample in the data set, and n signifies the total number of data points utilized in the experiment.

1. Mean Absolute Error (MAE): It is the absolute difference between actual values and the values predicted by the model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - \hat{p_i}| \tag{7.14}$$

2. Root Mean Square Error (RMSE): RMSE measures data deviation in which the square root of the mean of the squared errors is calculated.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(p_i - \hat{pi}\right)^2}{n}} \tag{7.15}$$

Table 7.1: Different Models and their Parameters used in the proposed work.

Model	**Parameters**
Linear Regression	Input shape: lag x 2 dimension
LSTM	LSTM layer (There are 16 units in LSTM, activation function: ReLU), Loss function: Mean Absolute Error(MAE), Optimizer: Adam, The Batch size: 128, Learning Rate: 0.001
GRU	GRU layer (There are 16 units in GRU, activation function: ReLU), Loss function: MAE, Optimizer: Adam, Batch size; 128, Learning Rate: 0.001
CNN-LSTM	CNN layer (# Filters: 2, size of each filter:2, activation function: ReLU), Flatten Layer, LSTM layer (There are 16 units in CNN-LSTM, Activation function: ReLU), Dense layer: 1, Loss function: MAE, Optimizer: Adam, Batch size: 128, Learning Rate: 0.001
CNN-GRU	CNN layer (# Filters: 2, size of each filter:2, activation function: ReLU), Flatten Layer, GRU layer (There are 16 units in CNN-GRU, activation function: Relu), Dense layer: 1, Loss function: MAE, Optimizer: Adam. Batch size: 128, Learning Rate: 0.001
Parallel CNN-LSTM	CNN layer (# Filters: 2, size of each filter:2, activation function: ReLU), Flatten Layer, LSTM layer (There are 16 units in Parallel CNN-LSTM, activation function: Relu), Dense layer: 1, Loss function: MAE, Optimizer: Adam. Batch size: 128, Learning Rate: 0.001

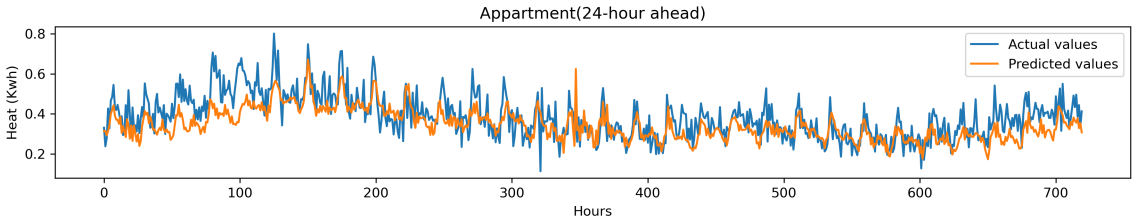
3. R^2 Score: It shows how well the data fit the regression model.

$$R^2 = 1 - \frac{\sum \left(p_i - \hat{pi}\right)^2}{\sum \left(p_i - \bar{p}\right)^2} \tag{7.16}$$

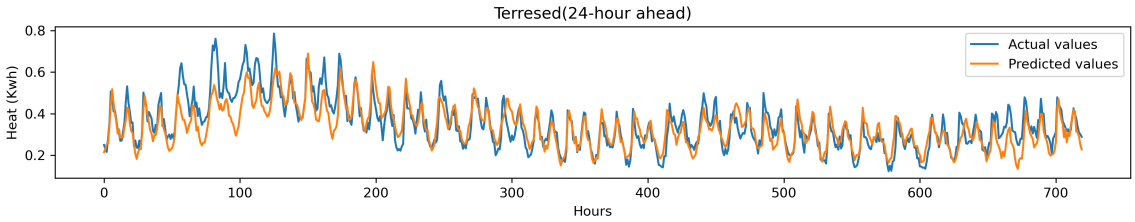
   Where $\bar{p}$ represents the mean of all predicted values.

4. Mean Absolute Percentage Error (MAPE): It is a measure of prediction accuracy of a forecasting method and usually expresses the accuracy as a ratio defined by the formula.

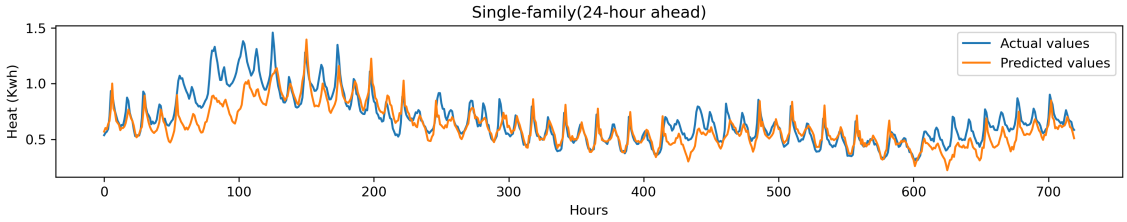
$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{|p_i - \hat{p}_i|}{p_i} \tag{7.17}$$



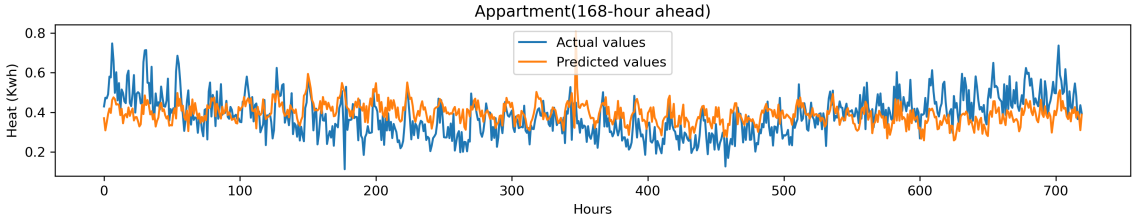
(a)



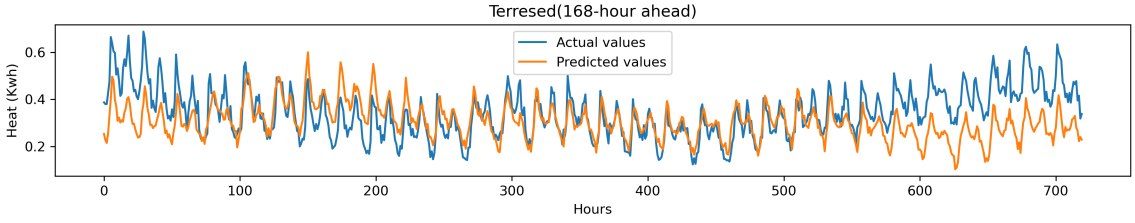
(b)



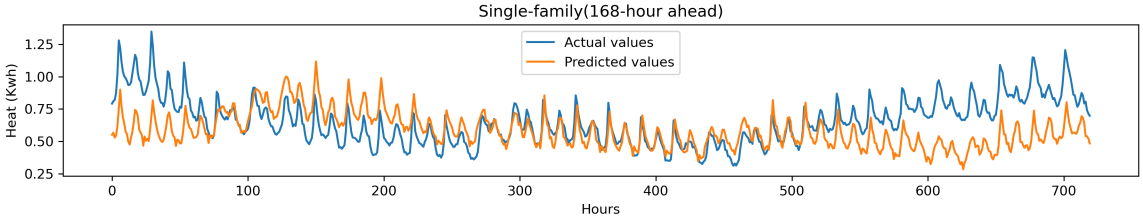
(c)



(d)



(e)



(f)

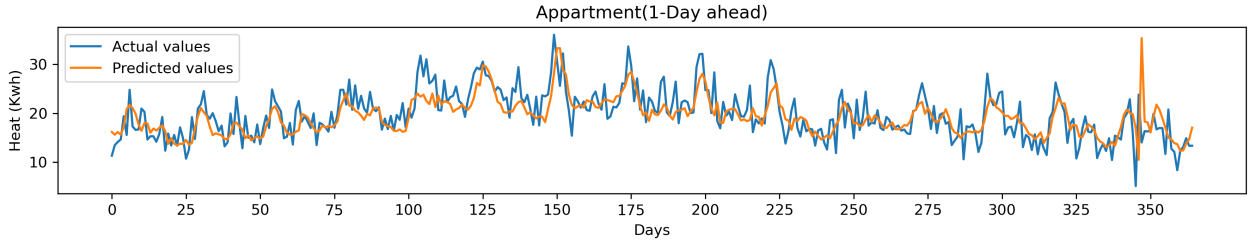
Figure 7.4: Actual v/s Predicted pf M6 model: (a)-(c) 24 hours ahead prediction for one month and (d)-(f) 168 hours ahead prediction for one month

Table 7.2: 24 hours ahead prediction for terraced house

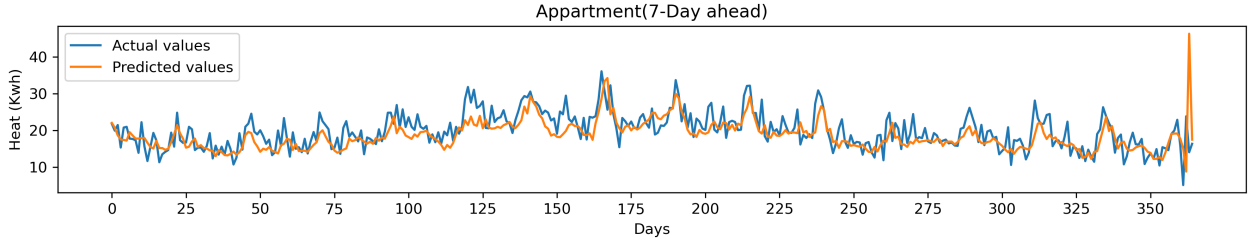
	M1			M2			M3			M4			M5			M6		
Lag	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2	0.104	0.145	0.944	0.096	0.138	0.950	0.096	0.139	0.949	0.102	0.145	0.945	0.101	0.145	0.945	0.098	0.138	0.950
4	0.103	0.144	0.945	0.096	0.138	0.950	0.094	0.136	0.951	0.103	0.148	0.942	0.104	0.149	0.941	0.097	0.138	0.950
8	0.100	0.140	0.949	0.104	0.143	0.946	0.092	0.133	0.953	0.102	0.147	0.943	0.099	0.141	0.948	0.095	0.136	0.951
12	0.100	0.140	0.949	0.093	0.134	0.953	0.105	0.144	0.945	0.103	0.144	0.945	0.099	0.142	0.947	0.103	0.145	0.945
24	0.095	0.135	0.952	0.095	0.134	0.953	0.092	0.132	0.954	0.116	0.153	0.938	0.099	0.141	0.948	0.094	0.132	0.954

Table 7.3: 168 hours ahead prediction for terraced house

	M1			M2			M3			M4			M5			M6		
Lag	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2	0.151	0.204	0.892	0.129	0.176	0.920	0.124	0.170	0.925	0.139	0.185	0.911	0.146	0.200	0.896	0.139	0.181	0.915
4	0.150	0.202	0.894	0.126	0.172	0.923	0.129	0.176	0.919	0.144	0.199	0.898	0.142	0.198	0.898	0.121	0.167	0.928
8	0.147	0.199	0.897	0.148	0.206	0.889	0.131	0.180	0.915	0.143	0.196	0.901	0.155	0.208	0.888	0.124	0.170	0.925
12	0.147	0.198	0.898	0.134	0.182	0.914	0.122	0.167	0.927	0.145	0.197	0.899	0.144	0.198	0.899	0.121	0.169	0.926
24	0.141	0.191	0.906	0.139	0.192	0.905	0.130	0.178	0.918	0.146	0.196	0.901	0.146	0.197	0.898	0.152	0.198	0.928



(a)



(b)

Figure 7.5: Actual v/s Predicted values using proposed model M6 for apartment type house (a) 1 day ahead prediction (b) 7 day ahead prediction.

Table 7.4: 24 hours ahead prediction for Apartment

	M1			M2			M3			M4			M5			M6		
Lag	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2	0.133	0.186	0.924	0.122	0.171	0.935	0.121	0.172	0.935	0.132	0.189	0.921	0.132	0.19	0.921	0.128	0.177	0.931
4	0.133	0.186	0.924	0.121	0.17	0.936	0.121	0.173	0.934	0.129	0.186	0.923	0.128	0.182	0.927	0.118	0.168	0.938
8	0.128	0.18	0.928	0.119	0.169	0.937	0.119	0.169	0.937	0.127	0.181	0.927	0.129	0.187	0.923	0.117	0.166	0.939
12	0.131	0.179	0.929	0.121	0.171	0.935	0.117	0.166	0.939	0.13	0.181	0.928	0.126	0.182	0.927	0.118	0.167	0.938
24	0.12	0.17	0.937	0.116	0.165	0.94	0.117	0.164	0.941	0.134	0.187	0.923	0.127	0.178	0.93	0.116	0.163	0.942

Table 7.5: 168 hours ahead prediction for Apartment

	M1			M2			M3			M4			M5			M6		
Lag	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2	0.184	0.250	0.864	0.161	0.218	0.897	0.155	0.212	0.902	0.170	0.237	0.877	0.174	0.245	0.870	0.121	0.172	0.935
4	0.183	0.249	0.865	0.154	0.211	0.903	0.174	0.242	0.873	0.168	0.235	0.879	0.168	0.236	0.879	0.121	0.171	0.936
8	0.178	0.242	0.872	0.147	0.205	0.909	0.152	0.208	0.905	0.168	0.236	0.879	0.181	0.245	0.870	0.139	0.190	0.920
12	0.178	0.240	0.875	0.148	0.207	0.907	0.148	0.205	0.909	0.162	0.229	0.886	0.174	0.239	0.876	0.124	0.180	0.929
24	0.168	0.227	0.888	0.182	0.256	0.858	0.144	0.202	0.912	0.155	0.216	0.899	0.158	0.218	0.897	0.116	0.165	0.940

Table 7.6: 24 hours ahead prediction for single-family house

	M1			M2			M3			M4			M5			M6		
Lag	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2	0.140	0.198	0.963	0.147	0.203	0.961	0.138	0.196	0.964	0.146	0.207	0.960	0.138	0.198	0.964	0.140	0.196	0.964
4	0.139	0.197	0.964	0.134	0.189	0.967	0.140	0.199	0.963	0.176	0.229	0.951	0.148	0.206	0.961	0.152	0.200	0.963
8	0.136	0.193	0.965	0.147	0.200	0.963	0.133	0.188	0.967	0.140	0.196	0.964	0.145	0.203	0.962	0.130	0.185	0.968
12	0.137	0.192	0.966	0.137	0.192	0.966	0.132	0.188	0.967	0.144	0.202	0.962	0.144	0.204	0.961	0.158	0.218	0.956
24	0.135	0.190	0.966	0.146	0.198	0.963	0.136	0.191	0.966	0.145	0.206	0.960	0.141	0.200	0.963	0.137	0.191	0.966

Table 7.7: 168 hours ahead prediction for single-family house

	M1			M2			M3			M4			M5			M6		
Lag	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2	0.238	0.309	0.912	0.213	0.281	0.928	0.224	0.288	0.924	0.211	0.276	0.930	0.220	0.285	0.925	0.219	0.286	0.925
4	0.237	0.307	0.913	0.211	0.273	0.932	0.220	0.292	0.922	0.228	0.303	0.916	0.237	0.310	0.912	0.220	0.290	0.923
8	0.234	0.304	0.915	0.205	0.267	0.935	0.205	0.268	0.934	0.235	0.312	0.911	0.223	0.298	0.918	0.216	0.289	0.923
12	0.234	0.302	0.916	0.199	0.265	0.936	0.207	0.273	0.932	0.210	0.277	0.929	0.207	0.274	0.931	0.220	0.289	0.923
24	0.227	0.295	0.921	0.203	0.265	0.935	0.199	0.262	0.937	0.241	0.312	0.911	0.214	0.280	0.928	0.212	0.273	0.932

Table 7.8: 1-Day ahead prediction for apartment house

	M1			M2			M3			M4			M5			M6		
Lag	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2	6.253	11.255	0.917	6.122	11.367	0.915	6.181	11.428	0.914	6.153	11.324	0.915	6.162	11.330	0.915	6.156	11.479	0.913
4	6.141	11.088	0.919	6.029	11.175	0.918	5.999	11.140	0.918	6.060	11.203	0.917	6.150	11.406	0.914	6.037	11.157	0.918
8	6.087	10.959	0.921	6.016	11.260	0.916	6.027	11.251	0.917	6.040	11.061	0.919	6.026	11.114	0.919	5.934	11.074	0.919
12	6.071	10.905	0.922	5.896	10.988	0.920	5.935	11.011	0.920	6.114	11.178	0.918	5.992	11.083	0.919	5.946	11.036	0.920
24	6.046	10.813	0.923	6.753	13.555	0.879	5.839	10.686	0.925	5.970	11.008	0.920	5.954	10.965	0.921	5.877	10.930	0.921

Table 7.9: 7-Day ahead prediction for apartment house

	M1			M2			M3			M4			M5			M6		
Lag	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2	MAE	RMSE	R2
2	8.721	15.712	0.837	8.619	16.384	0.823	8.416	15.640	0.839	8.606	15.862	0.834	8.605	15.810	0.835	8.412	15.581	0.840
4	8.527	15.418	0.843	8.179	15.321	0.845	8.120	15.131	0.849	8.665	16.252	0.826	8.421	15.634	0.839	8.246	15.565	0.840
8	8.370	15.130	0.849	7.998	14.977	0.852	7.925	14.695	0.858	8.430	15.416	0.843	8.431	15.996	0.831	8.032	15.076	0.850
12	8.320	15.016	0.851	8.092	15.050	0.851	8.110	15.034	0.851	9.363	18.648	0.771	8.065	14.924	0.853	8.103	15.198	0.848
24	8.259	14.864	0.855	7.767	14.378	0.864	7.875	14.537	0.861	8.475	15.982	0.832	8.489	16.166	0.828	7.826	14.488	0.862

Table 7.10: MAPE results of the proposed model considering all types of houses with 1 hour, 24 hours, and 168 hours ahead predictions

	Apartment			Terreced			Single-family		
Lag	1h-ahead	24h-ahead	168h-ahead	1h-ahead	24h-ahead	168h-ahead	1h-ahead	24h-ahead	168h-ahead
2	0.102	0.154	0.210	0.07	0.131	0.191	0.058	0.127	0.174
4	0.095	0.150	0.199	0.065	0.151	0.200	0.050	0.144	0.168
8	0.095	0.148	0.207	0.072	0.141	0.190	0.053	0.111	0.162
12	0.091	0.147	0.180	0.061	0.126	0.191	0.049	0.101	0.161
24	0.093	0.146	0.190	0.061	0.129	0.173	0.047	0.107	0.162

7.5 Results and Discussion

Our case study considers the smart heat meter data installed in Danish residential buildings. The dataset has three types of buildings, apartment houses, terraced houses, and single-family houses, and recorded three years of data. The dataset was analyzed from two perspectives (Scenario 1 and Scenario 2), as described in Section 7.2.1. A parallel CNN-RNN model was constructed and tested against existing statistics and deep learning models. The models Linear Regression (LR), LSTM, GRU, CNN-LSTM, CNN-GRU, and parallel CNN-LSTM were compared and labeled as M1, M2, M3, M4, M5, and M6, respectively. Table 7.1 includes brief parameter data for these models. Tables 7.2 and 7.3 demonstrate the results for scenario 1 in which the results of 24-hours ahead and 168-hours ahead prediction for the terraced house have been shown with an R^2 accuracy of 0.954 and 0.928, respectively. The MAE and RMSE of 24 hours and 168 hours ahead prediction of terraced houses are 0.094,0.152, 0.132, and 0.198, respectively. Table 7.4 and Table 7.5 display the results for the Apartment type house (Scenario 1) by considering the various lags for all the models and represent the results for 24 hours and 168 hours ahead prediction for Apartment house, respectively. It is observed that all models performed equally, and no significant difference is found in the case of 24 hours ahead, as shown in Table 7.4. However, the proposed model (M6) outperformed all other models in the second case (168 hours ahead), as shown in Table 7.5. The proposed model (M6) has the maximum R^2 value of 0.94 for lag 24. Other models, such as M1, have a R^2 accuracy of 0.88, M4 has an accuracy of 0.89, and M5 has an accuracy of 0.89. M2 had the lowest R^2 value of 0.858, while M3 had the second highest at 0.912. Similar trends have been seen for buildings such as single-family and terraced homes. The maximum R^2 values obtained by the proposed model (M6) for 168 hours ahead are 0.928 and 0.940, respectively, for Terraced house and Apartment houses as shown in Table 7.3 and 7.5. Tables 7.6 and 7.7 demonstrate that the suggested M6 models have MAE and RMSE values of 0.137 and 0.191 for 24 hours ahead prediction

of the single-family house, and 0.212 and 0.273 for 168 hours ahead prediction. Tables 7.6 and 7.7 indicate that the suggested model (M6) achieves maximum R^2 values of 0.96 and 0.93 for single-family residences 24 and 168 hours ahead, respectively, when all scenarios and lags are taken into account. Tables 7.3 and 7.8 demonstrate that the proposed Parallel CNN-LSTM (M6) outperforms in terms of R^2, MAE, and RMSE.

The actual versus the predicted values for scenario 1 are as shown in Figure 7.4. The 24-hour ahead prediction and 168-hour ahead prediction for one month of the proposed M6 model is shown in Figure 7.4(a) and Figure 7.4(f), respectively. Figure 7.4(a) - Figure 7.4(c) shows the 24-hour ahead prediction for a complete apartment, terraced, and single-family houses data. Figure 7.4(d) - Figure 7.4(f) shows the 168-hour ahead prediction for apartment, terraced, and single-family houses data. It is observed that the proposed model predicts the heat energy with high accuracy 24 hours ahead as compared to 128 hours ahead. Table 7.8 and 7.9 shows the result obtained for scenario 2. It shows the value for different lags of 2, 4, 8, 12, and 24 hours for 1-day- and 7-day ahead prediction of an apartment house. The 1 Day ahead prediction for apartment houses by the proposed models performs better in terms of R^2 with 0.913 accuracies for a 2-hour lag, 0.918 R^2 for a 4-hour lag, 0.919 R^2 for lag 8 hours, 0.920 R^2 for 12 hours and 0.921 R^2 for 24 hours lag. The 7 Day ahead prediction for the apartment house by the proposed model is 0.840 R^2 for lag 2 and 4, 0.850 R^2 for lag 8, 0.848 R^2 for lag 12, and 0.862 R^2 for lag 24 as shown in Table 7.9. The MAE and RMSE of the proposed M6 model are 7.826 and 14.448. The Better MAE and RMSE have been found in Tables 7.8 and 7.9 for 1 Day ahead and 7 Day ahead prediction for apartment houses by the proposed model. Both Tables show that the proposed hybrid parallel CNN-LSTM performs better regarding MAE, RMSE, and R^2 for different lags. Figure 7.5 depicts the actual versus projected values of scenario 2 with heat energy in kWh on the y-axis and hours on the x-axis. Table 7.10 demonstrates the complete performance of our proposed model by considering the MAPE evaluation metric. It

is evident from the obtained results that the proposed model achieves good MAPE performance. Moreover, the proposed model expressed excellent performance with lag 12. Figures 7.5(a) & (b) illustrate the actual values of the suggested M6 for 1 day and 7 days ahead prediction of individual apartment buildings. Compared to other models, high accuracy and low error metrics are observed.

7.5.1 Comparsion with Existing works

Carbon emissions from energy use harm the environment and affect system development. A data-driven approach can help reduce both energy use and emissions. With growing energy demand, efficiency and reliability are essential for managing energy generation and usage. That's why electrical and heating systems are integrated into smart grids [173]. Predicting power and heating demand is key for energy efficiency. Heat load prediction helps maintain comfortable indoor temperatures by considering factors like room size, temperature, and occupancy. It's crucial to optimize energy use in smart buildings, which are responsible for a large share of global energy consumption and emissions. Buildings account for nearly 40% of global energy demand, with heating making up 55% of that in the EU. Smart buildings aim to provide comfort while reducing energy use and environmental impact[107].

7.6 Conclusion and Future Work

Heat load energy demand is an essential consideration in energy storage management systems. Heat load prediction for smart buildings is in significant demand today, but it faces numerous problems, including temporal-spatial correlations in multivariate information. This research presents revolutionary intelligent modeling and predicting heat energy consumption in smart buildings using smart heat meter data. The proposed hybrid deep learning model combines parallel CNN and LSTM. The CNN extracts high-level features from the given data, while LSTM extracts the temporal link be-

tween local characteristics. Aside from heat energy, other factors such as volume flow, inlet flow energy, backflow energy, hour, day, and month are incorporated while developing multivariate models. The exhaustive experimental findings were examined over 24 hours, 168 hours, one day, and seven days, respectively. The model generates promising R_2 results with lower error measures such as MAE and RMSE.

In the future, explainable AI can be used to improve the proposed model and make better decisions. The parallel CNN-LSTM outperformed other models, including LR, LSTM, GRU, CNN-LSTM, and CNN-GRU. The proposed work's limitation applies only to short-term energy forecasting. The proposed approach can be expanded for long-term prediction in the future by using a hybrid deep learning model. This model can also be used for wind and traffic flow forecasting.

The next chapter concludes the thesis work with future research directions.