

Chapter 4

Energy Prediction Mechanism with Occupant Count Using CNN-BiLSTM Hybrid Model for Smart Buildings

Building energy prediction plays a crucial role in reducing energy waste and ensuring the storage of sufficient, sustainable energy for future use [111]. However, the performance of existing building energy prediction models is very low, and they are application-specific [112]. Therefore, this chapter proposes a novel hybrid model that combines Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) to build energy prediction. The proposed hybrid model captures spatial patterns and features in the data using CNN. Furthermore, it captures temporal dependencies and long-term relationships by using BiLSTM. The simulation results show that the proposed model outperforms existing state-of-the-art approaches.

4.1 Introduction

In various domains, such as smart buildings, Industry 4.0, and smart cities, accurate energy consumption prediction is crucial for effective resource allocation, demand planning, and cost optimization [113]. The building consumes approximately one-third of society's overall energy [114]. It creates a depletion of natural energy resources and increases environmental pollution [113]. To mitigate these challenges, localized optimization is crucial within the energy market that ensures efficient resource utilization [115]. People's consumption habits are changing due to technological advancements in heat pumps and building insulation [116]. They use energy randomly without any proper requirements. Thus, optimizing energy utilization requires an accurate energy forecast model [117] [118]. In China, five terabytes of data are generated annually by smart meters, which is a huge amount of data. It provides valuable insights into energy usage patterns and enables better decision-making. Heating and refrigeration consume approximately 63% of building energy [119]. Building energy consumption contributes around 40% of total CO₂ emissions [120]. Therefore, an effective energy management strategy for a building is crucial to mitigating environmental impact and achieving sustainability goals [113]. In energy management systems, energy prediction plays a vital role in reducing wastage of energy consumption.

Existing state-of-the-art approaches, such as traditional time series forecasting methods, statistical models, and machine learning techniques, offer various solutions for energy consumption prediction. However, these approaches overlook the inclusion of occupancy count and fail to capture complex temporal and spatial dependencies in the data [113, 115, 116]. The occupancy count plays a vital role in a building's energy consumption. It also helps to enhance the generalizability of the energy prediction model across diverse building types and geographical regions. Therefore, the occupancy count is important to consider during the energy prediction for a building [117]. Furthermore, it provides a more accurate prediction.

Aaslid et al.[121] demonstrated the importance of stochastic optimization for micro-grid operation with renewable generation and storage. As per this approach, accurate energy prediction is very important for efficient renewable energy storage. Shamsi and Cuffe investigated prediction markets for probabilistic forecasting of renewable energy sources [122]. Gan et al. [123] proposed a data-driven energy a management system that utilizes Gaussian Process forecasting and Model Predictive Control (MPC) for interconnected microgrids. These approaches also indicate the importance of accurate energy prediction for renewable energy storage in the smart building domain.

The work done by Q. Qiao et al. [124] proposed a hybrid approach for building energy consumption prediction using support vector machines and seasonal autoregressive integrated moving average models. However, the performance of this approach depends on the availability and quality of training data, potentially limiting its applicability in real-world scenarios. F. Fakour et al. [125] explored applying machine learning techniques coupled with uncertainty quantification methods for predicting building energy consumption. The integration of uncertainty quantification enhances the reliability of energy consumption prediction. However, the computational complexity of these methods poses challenges for real-time prediction in large-scale buildings. Y. Ma et al. [126] focused on improving the accuracy of energy consumption prediction in green buildings using machine learning algorithms. The performance of this approach is influenced by several factors , such as the availability of comprehensive training data and selecting appropriate features. It limits the generalizability of the energy prediction model across diverse building types and geographical regions.

To address the lack of generalizability and interpretability in the papers mentioned above, this chapter proposes a novel hybrid model that integrates CNN and BiLSTM models to predict building energy consumption accurately. The CNN extracts spatial features from the input data, capturing patterns and relationships within occupancy count and energy consumption. These spatial features are then passed to the BiLSTM

model, which effectively models the temporal dependencies and predicts future energy consumption based on the learned features. Furthermore, the proposed scheme also incorporates occupancy count for energy prediction because occupancy count is not directly dependent on geographical regions or building types. It increases the generalizability of the proposed hybrid energy prediction model across diverse building types and geographical regions. Therefore, the existing model can only be applied to specific regions or types of buildings.

The major contributions of this chapter are summarized as follows:

- This chapter presents a hybrid architecture of CNN and BiLSTM for predicting energy consumption. It jointly considers both the occupancy count and the timestamp information.
- The proposed model considers spatial and temporal dependencies that significantly increase prediction accuracy. The proposed model performance is assessed using real-world data.
- Furthermore, the accuracy and effectiveness of the model are demonstrated in comparison to existing state-of-the-art methods in terms of predicted accuracy, MAE, and MSE.
- The experiment uses a novel dataset of building energy consumption and occupant count for testing the hybrid model.
- This chapter elucidates energy usage patterns and enhances decision-making for energy management.

4.2 Application Scenario

The proposed model helps to predict building, industry, and smart grid energy needs. Consider a scenario where buildings, industries, and smart grids can intelligently predict future energy needs. This capability enables precise control over when and how energy is consumed, leading to optimized resource utilization, reduced costs, and a

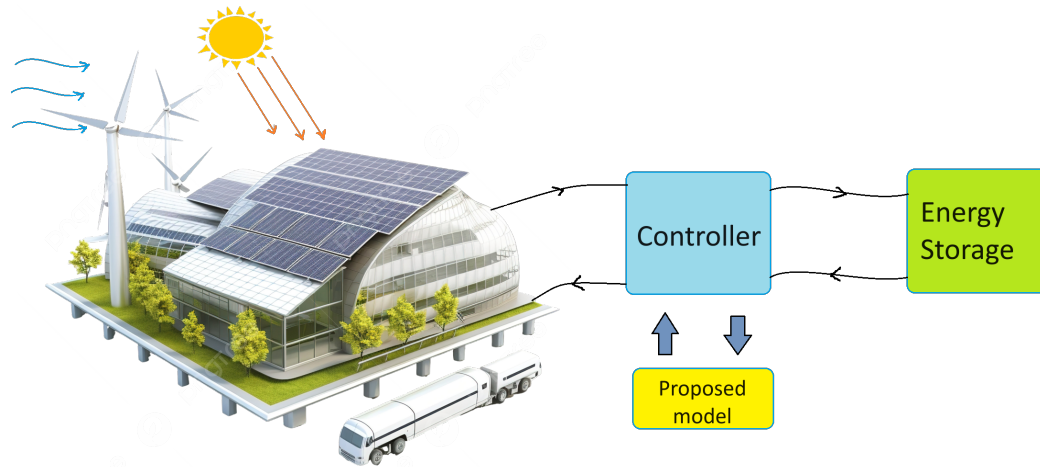


Figure 4.1: Energy storage systems for smart building.

more environment friendly approach. Furthermore, it also helps to store sufficient sustainable energy for a building’s energy utilization [127]. Figure 4.1 shows an energy storage system for a smart building where renewable energy is stored based on future energy needs. In practical terms, this research allows for smarter decisions in scheduling energy-intensive tasks, dynamically adjusting building systems based on predicted demand, and seamlessly integrating renewable energy sources.

4.3 Problem Formulation and Proposed Work

This section describes problem formulation and the proposed hybrid model description in detail. Table 4.1 shows symbols and notations used in this chapter.

4.3.1 Problem Formulation

This work uses a dataset consisting of *DateandTime* (D), *OccupantCount* (O) at a specific time, and *EnergyConsumed* (E) at that time. The main objective of this work is to predict the energy consumed after 1 day, denoted as E_{nextday} . The dataset is represented as a collection of tuples (D_i, O_i, E_i) , where i represents the index of the data instance. The proposed hybrid model takes the following input features:

Table 4.1: Symbols and Notations

Symbol/Notation	Description
D	Date and Time
O	Occupant Count at a specific time
E	Energy Consumed at a specific time
E_{nextday}	Predicted Energy Consumed after 1 day
D_i	Date and Time of the i -th data instance
O_i	Occupant Count at the i -th data instance
E_i	Energy Consumed at the i -th data instance
n	Number of data instances
$\hat{E}_{\text{nextday}}^{(i)}$	Predicted energy consumed after 1 day for the i -th instance
$E_{\text{nextday}}^{(i)}$	Actual energy consumed after 1 day for the i -th instance
MSE	Mean Squared Error
σ	Activation function
$F(i, j)$	Value at position (i, j) in the feature map F
$X(i + m, j + n)$	Element in the input data X at spatial position $(i + m, j + n)$
$W(m, n)$	Weight of the filter at position (m, n)
b	Bias term
ReLU	Rectified Linear Unit activation function
Z_l	Resulting feature maps in the l -th convolutional layer
$D(l + 1)$	Filters in the $l + 1$ -th convolutional layer
W_l	Convolutional layer weight in the l -th feature learning block (FLB)
BiLSTM	Bidirectional Long Short-Term Memory Network
$\vec{f}_t, \overleftarrow{f}_t$	Forward and backward operations in BiLSTM
h_t	Hidden state vector in BiLSTM
look_back	Number of time steps to look back in the input data

$$\begin{bmatrix} D_{i-\text{look_back}} & E_{i-\text{look_back}} & O_{i-\text{look_back}} \\ \vdots & \vdots & \vdots \\ D_{i-1} & E_{i-1} & O_{i-1} \\ D_i & E_i & O_i \end{bmatrix}$$

where *look_back* is the number of previous values taken into consideration from the tuple for which energy needs to be predicted. Furthermore, it learns the temporal patterns and relationships between the current occupant count and energy consumed to make predictions about the E_{nextday} . The model architecture combines the CNN and BiLSTM layers. The CNN layers extract spatial features from the input, capturing

occupant count and energy consumption patterns. Furthermore, The BiLSTM layers capture temporal dependencies and learn the sequential patterns in the data. Combining CNN and BiLSTM layers enables the model to learn from spatial and temporal data effectively. The input now belongs to the real number of dimensions ($\text{look_back} \times 3$), and the output will be of dimension 1×1 . To train the model, the mean squared error (MSE) function is chosen as the loss function, which is calculated as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{E}_{\text{nextday}}^{(i)} - E_{\text{nextday}}^{(i)})^2 \quad (4.1)$$

where n is the number of data instances, $\hat{E}_{\text{nextday}}^{(i)}$ is the predicted energy consumed after 1 day for the i -th instance. $E_{\text{nextday}}^{(i)}$ is the actual energy consumed after 1 day for the i -th instance. The working flow of the proposed CNN-BiLSTM hybrid model is explained in Figure 4.2

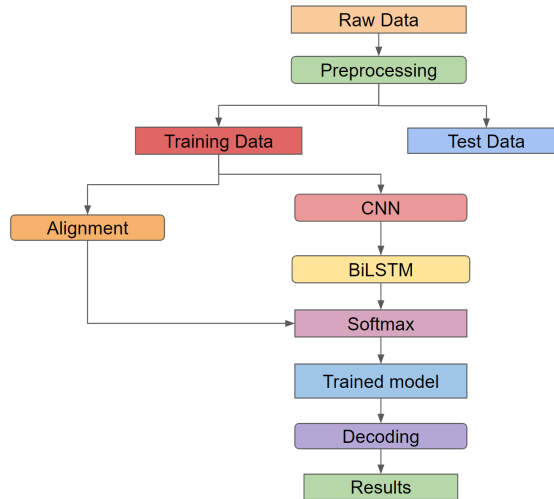


Figure 4.2: Workflow of proposed model

4.3.2 Convolutional Neural Network (CNN)

CNNs are smart tools that predict energy uses by looking at data patterns. These patterns could be anything from how many people are around to how much energy they use. CNNs use special filters to find the important details in the data. It helps to

make accurate predictions about energy consumption. It is represented as follows:

$$F(i, j) = \sigma \left(\sum_{m=1}^M \sum_{n=1}^N X(i + m, j + n) \cdot W(m, n) + b \right) \quad (4.2)$$

CNN looks at different data points and combines them to make predictions. The result is shown at a specific location by $F(i, j)$, a portion of the data is at a specific location by $X(i + m, j + n)$, a unique number for the filter is $W(m, n)$, and the activation function is $\sigma(\cdot)$. This work uses CNN to find important details about how things change over time. It uses blocks with filters and pools of numbers to do this. It is represented as follows:

$$Z_l = W_l * X_l + b_l \quad (4.3)$$

The CNN uses the activation function (ReLU) to ensure it does not miss anything important. Then, there is another step that keeps only the really useful information and gets rid of the extra information. CNN helps to understand patterns and simplifies the data. The final patterns from the CNN help another part of the system, BiLSTM, predict how much energy will be used in the future.

4.3.3 Bidirectional Long Short-Term Memory Network (BiLSTM)

Recurrent Neural Network (RNN) has effectively processed temporal data. However, conventional RNN models face two limitations: the vanishing gradient problem during back-propagation and inadequate modeling of backward dependencies. This chapter uses the LSTM network, an RNN architecture with gated regulators, to address the first issue. The proposed scheme stacks an LSTM network on top of the CNN component to learn contextual dependencies, illustrated as a blue box in Figure 4.3. A memory cell, a forget gate, an input gate, and an output gate are the components of each LSTM block. The memory cell activation vectors are f , i , o , and c for input, output, and forget gates, respectively. Nodal connections and tensor operations in a forward LSTM

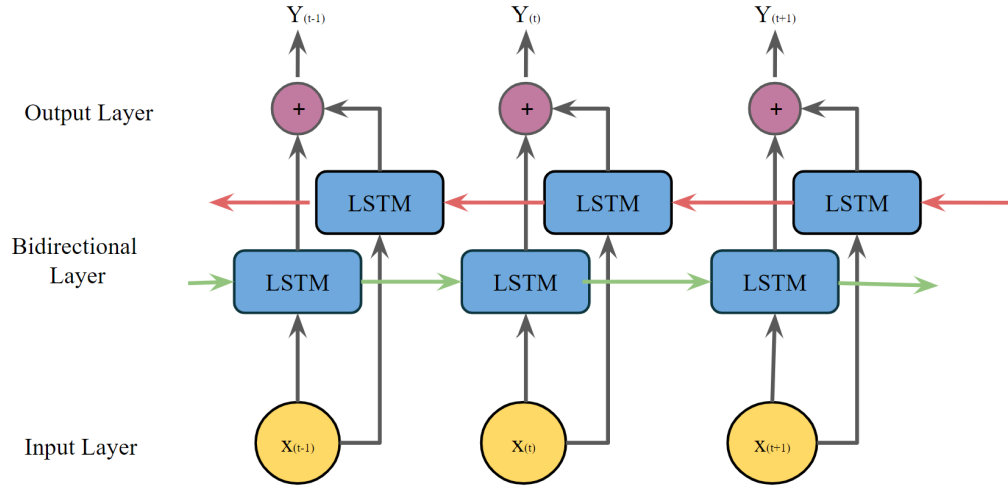


Figure 4.3: The BiLSTM architecture

hidden layer are denoted as follows:

$$f_t = \sigma(W_f \cdot h_{t-1}, X_t) + b_f \quad (4.4)$$

$$i_t = \sigma(W_i \cdot h_{t-1}, X_t) + b_i \quad (4.5)$$

$$\tilde{c}_t = \tanh(W_c \cdot h_{t-1}, X_t) + b_c \quad (4.6)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (4.7)$$

$$o_t = \sigma(W_o \cdot h_{t-1}, X_t) + b_o \quad (4.8)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (4.9)$$

where, W_f , W_i , W_c , and W_o represent the input weight matrices and b_f , b_i , b_c , and b_o are the corresponding bias vectors. The sigmoid $\sigma(\cdot)$ and hyperbolic tangent functions $\tanh(\cdot)$ serve as the activation functions. h denotes the hidden state, which is also the output of the LSTM hidden layer. \tilde{c} represents the new state candidate vector. The bracket indicates the concatenation operator. Accounting for forward and backward dependencies is necessary when projecting time series, particularly when dealing with periodic data, such as Advanced Metering Infrastructure (AMI) data. The performance of conventional recurrent models may be limited because they can only handle

unidirectional dependencies. To overcome this, the suggested method uses an LSTM configuration with a BiLSTM layer to capture forward and backward dependencies in AMI data. The node connections and tensor calculations in a BiLSTM are almost identical to those in a unidirectional LSTM, except for the processing directions:

$$\vec{f}_t = \sigma(W_{\vec{f}} \cdot \vec{h}_{t-1}, \vec{X}_t) + \vec{b}_f \quad (4.10)$$

$$\overleftarrow{f}_t = \sigma(W_{\overleftarrow{f}} \cdot \overleftarrow{h}_{t+1}, \overleftarrow{X}_t) + \overleftarrow{b}_f \quad (4.11)$$

where \vec{f}_t and \overleftarrow{f}_t represent the forward and backward operations, respectively. Two hidden state vectors, \vec{h}_t and \overleftarrow{h}_t , are generated independently and concatenate to form the final hidden state vector in the BiLSTM layer:

$$h_t = \begin{bmatrix} \vec{h}_t \\ \overleftarrow{h}_t \end{bmatrix} \quad (4.12)$$

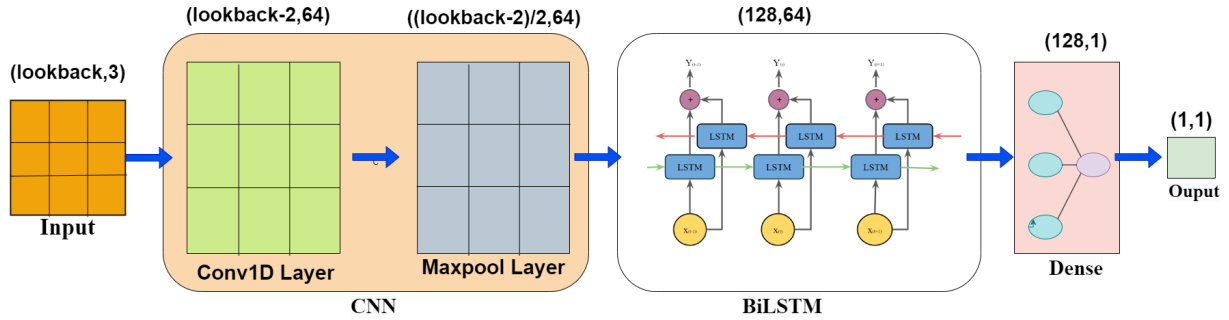


Figure 4.4: The proposed hybrid CNN BiLSTM architecture.

4.3.4 The Proposed CNN-BiLSTM Network

The proposed work consists of a two-step architecture for energy prediction, as shown in Figure 4.4. In the first step, a CNN captures spatial features from the input data. It includes the timestamp, occupant count, and energy consumption. The CNN utilizes filters to extract spatial characteristics and generate feature maps. Let X represent

the input data matrix, where each row corresponds to a specific timestamp and contains features like timestamp, occupant count, and energy consumption. The proposed scheme applies convolutional filters W with corresponding biases b to the input data. The convolution operation is denoted as $*$. Furthermore, the resulting feature maps are obtained by applying an activation function σ element-wise. It is represented as follows:

$$\text{Convolution: } F = \sigma(X * W + b) \quad (4.13)$$

The feature maps F capture spatial patterns and provide a more abstract representation of the input data. In the second step, the proposed scheme utilizes a BiLSTM network to model the temporal dependencies in the feature maps and predict the energy consumption after 1 day. LSTM is well-suited for handling sequential data and effectively captures long-term dependencies. Let H denote the hidden states of the LSTM network, and Y represent the predicted energy consumption after 1 day. The LSTM network takes the feature maps F as input and generates a sequence of hidden states H by updating the cell and hidden states based on the input and previous states. Finally, the output Y is obtained by applying a linear transformation followed by an activation function σ :

$$\text{LSTM: } H, Y = \text{LSTM}(F) \quad (4.14)$$

The LSTM network is trained using Back Propagation Through Time (BPTT) and optimized by minimizing a chosen loss function, such as Mean Squared Error (MSE). The loss function quantifies the discrepancy between the predicted energy consumption Y and the actual energy consumption after 1 day. The proposed architecture uses parallel processing by spatial and temporal information. This reduces computational overhead. The model captures the dynamic nature of building occupancy and energy

consumption patterns by incorporating occupant count and timestamp features. The BiLSTM enables the model to capture bidirectional dependencies in the temporal data. This enhances predictive accuracy. The training process involves iteratively updating the parameters of the CNN and BiLSTM components. This optimizes their performance jointly. The model architecture is adaptable and can be extended to incorporate additional features.

Different machine learning models' time complexities for energy consumption prediction vary based on the algorithms and architectures employed. It is discussed in Table 4.2. In the context of time series forecasting, the ARIMA model exhibits a time complexity of $O(n)$, where n represents the size of the input data. ANN includes basic feedforward architectures, and they have a time complexity of $O(n * m)$, where n denotes the number of input features. m represents the number of neurons in the hidden layer. LSTM networks are designed to capture long-term dependencies in sequential data. They have a time complexity of $O(n * m^2)$. It reflects the additional computational overhead of recurrent connections and memory cells. Bidirectional LSTM models process input sequences in both forward and backward directions. They have a time complexity of $O(2 * n * m^2)$. The CNN-BiLSTM architecture combines spatial and temporal processing. This proposed model exhibits a time complexity of $O(n * m^2)$, where n represents the input data size, and m denotes the number of neurons in the LSTM layer.

Table 4.2: Time Complexities of Machine Learning Models

Model	Time Complexity
ARIMA	$O(n)$
ANN	$O(n * m)$
LSTM	$O(n * m^2)$
BiLSTM	$O(2 * n * m^2)$
CNN-BiLSTM	$O(n * m^2)$

4.4 Experimental Setup

This section discusses the detailed data set description, deep learning training, and simulation results. Table 4.3 shows parameter values used in the simulation.

Table 4.3: Parameters used in the model

Model Parameters	
Layer	Parameters
Input Shape	(look_back, 3)
Conv1D	Filters: 64, Kernel Size: 3, Activation: ReLU
MaxPooling1D	Pool Size: 2
Bidirectional LSTM	Units: 64, Activation: ReLU
Dense	Units: 1
Loss Function	Mean Squared Error (MSE)
Optimizer	Adam

4.4.1 Dataset Description

The dataset [128] contains information on occupant count, energy consumption, and corresponding timestamps. The dataset covered the period from May 22, 2018, to February 21, 2019. The occupant count data was available at a resolution of 1 minute, while the energy consumption data was recorded every 15 minutes. We combined the timestamps that overlapped to align the data appropriately, and the end result was a consolidated dataset with a 15-minute frequency. The dataset encompasses data about a building that consists of both a south block and a north block. However, this chapter focuses exclusively on the south block. It is crucial to address the issue of gaps in the dataset, as they could adversely affect the training of the predictive model. This work aims to mitigate the impact of these data gaps on the overall training process by identifying and implementing suitable strategies.

Handling the gaps in the dataset is necessary to ensure the accuracy and reliability of the proposed model. The dataset contains information on the occupants. The dataset has been split into monthly subsets from June 2018 to February 2019. In ex-

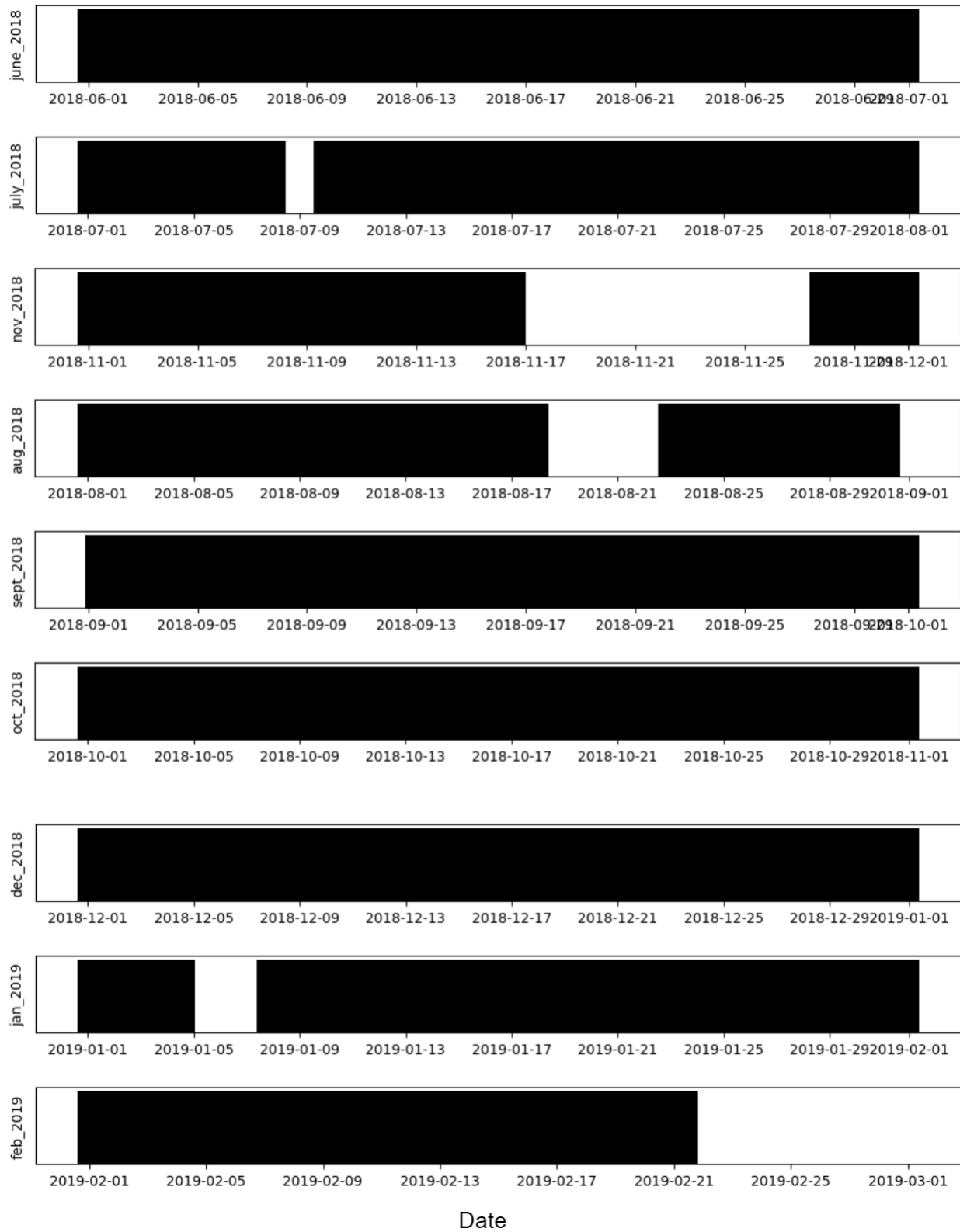


Figure 4.5: Month-wise missing data.

ploring this temporal data, we visualized the occurrences of missing data points within each month. This analysis allowed us to quantify the percentage of missing dates for each month. It provides insights into the completeness and reliability of the dataset. We employed visualization techniques to highlight and examine these gaps, as shown in Figure 4.5. It enables a comprehensive understanding of the temporal distribution of missing values. Furthermore, this chapter conducted a correlation analysis to investi-

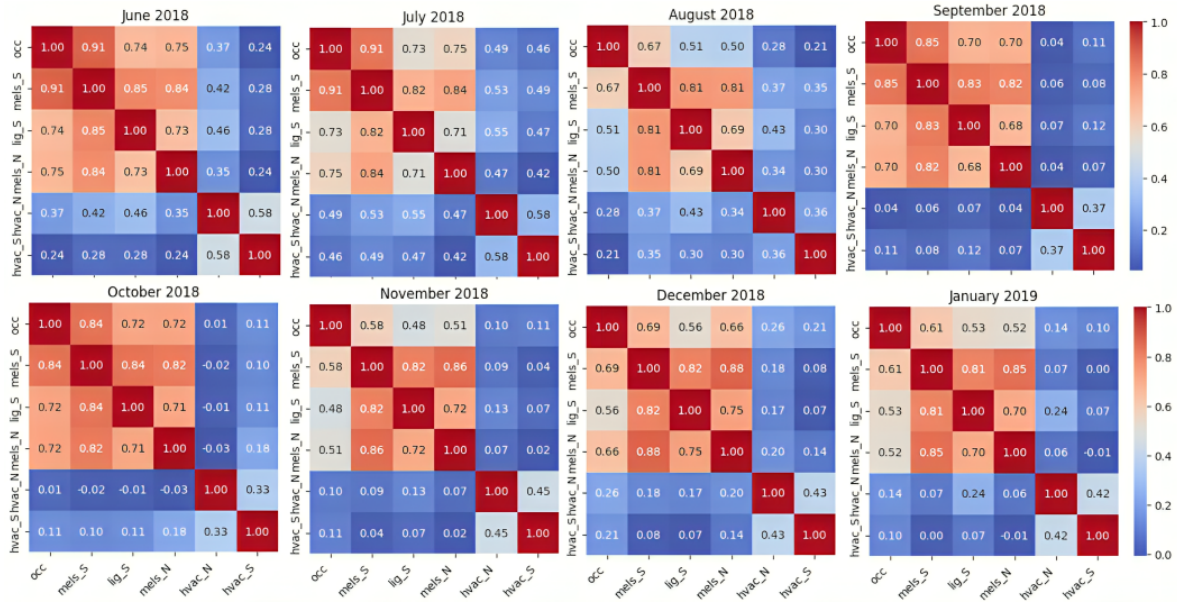


Figure 4.6: Month-wise correlation matrix between all six parameters of the data.

gate the relationships between occupant behavior and energy consumption month-wise. Figure 4.6 shows the month-wise correlation factor between all six data. It also analyzes the degree of association between these two variables in a time-dependent manner by constructing correlation matrices for each month. This approach unveiled nuanced patterns and trends, allowing us to identify potential dependencies or divergences in the interactions between occupancy patterns and energy usage over the specified time period.

Incorporating month-wise energy consumption data, alongside the corresponding energy levels after one day and occupant counts, introduces a temporal dimension to the analysis. Notably, variations in data gaps across different months add an additional

Table 4.4: Month-wise missing values and correlation values with occupant count.

Month	percent of missing values	correlation value
June 18	0	0.91
July 18	6.3	0.91
Aug 18	22.21	0.67
Sep 18	0.9	0.85
Oct 18	3.3	0.84
Nov 18	59.4	0.58
Dec 18	0	0.69
Jan 19	11.25	0.61
Feb 19	37.07	0.81

layer of complexity. The exploration of the correlation matrix on a month-wise basis reveals intriguing patterns. It exhibits higher correlations of certain months compared to others, as shown in Table 4.4. This nuanced approach recognizes the dynamic nature of energy consumption. It also indicates that temporal factors might influence the relationships between variables differently across various time frames.

As we delve into the correlation analysis between training and testing data, the impact on predictive accuracy becomes a focal point. Understanding how the strength and nature of correlations vary in different subsets of the data aids in gauging the robustness of the predictive model. This investigation is essential for identifying potential challenges and opportunities that guide the development of a model. It effectively generalizes across diverse temporal scenarios and data gaps. The month-wise exploration of correlations thus helps to optimize the accuracy and reliability of the energy consumption prediction model.

4.4.2 Deep Learning Training

This work uses data from July for training and testing the predictive model due to the absence of breaks or gaps this month. The September month dataset is divided into two subsets used for training and testing. The training dataset was allocated 70% of the data, while the remaining 30% was assigned to the testing dataset. A two-step process

utilizes CNN and BiLSTM models to train the model. In the first step, the data from the training dataset was fed into the CNN model, which is responsible for extracting spatial features or patterns from the input data. The output from the CNN model was passed to the BiLSTM model, which captures spatial features and temporal dependencies from the data and makes predictions based on the learned features. Training the hybrid CNN and BiLSTM model using the training dataset optimizes the model's ability to learn and predict energy consumption patterns. Furthermore, the testing dataset is used to evaluate the performance and accuracy of the trained model, which is kept separate from the training dataset. Through this step-by-step procedure, the proposed hybrid CNN and BiLSTM models leverage the spatial and temporal information in the data to predict energy consumption effectively.

4.5 Result and Discussion

The simulations were conducted on a workstation with an 11th Gen Intel (R) Core™ i5-11300H @ 3.10GHz with 16GB RAM. The CNN-BiLSTM model uses the Keras library with Tensorflow backend and the sci-kit-learn library in Python version 3.11. The experiments are conducted 10 times using the same random number generator seed to enhance reproducibility and maintain consistency. This approach accounts for the inherent randomness involved in the experiments, such as sampling and stochastic gradient descent (SDG).

4.5.1 Evaluation Metrics

In predictive modeling, it is essential to evaluate the performance and accuracy of the models. Three common evaluation metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2), are used to check the performance and accuracy of the proposed hybrid model. MAE provides a measure of the average absolute difference between the predicted values and the actual values. It is calculated

as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.15)$$

The lower MAE value indicates better performance of the model. MAE is robust to outliers as it does not consider the direction of errors. Furthermore, MSE measures the average variance between predicted and actual values. It is computed as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.16)$$

MSE is more sensitive to outliers due to the squaring operation. Similar to MAE, lower MSE values indicate better model performance. R^2 is also known as the coefficient of determination, and it quantifies the proportion of the variance in the dependent variable that can be predicted from the independent variables. It is calculated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4.17)$$

R^2 ranges from 0 to 1, where higher values indicate better model performance.

The accuracy and goodness of fit of the proposed hybrid model are evaluated by the above-proposed metrics. MAE and MSE provide insights into the magnitude and variance of errors, respectively, while R^2 indicates the proportion of variance explained by the model. These metrics aid in model comparison, hyperparameter tuning, and selecting the best model for the given prediction task. These evaluation metrics are crucial for effective model evaluation and interpretation in various domains, including regression analysis and machine learning.

4.5.2 Basic Results

The model training phase uses the data from the first three weeks of July to train the model. Based on the information gathered from the initial three weeks, the trained model was then used to predict July's fourth week. The actual and predicted values of the energy are shown in Figure 4.7. The evaluation of the model's performance yielded the following metrics: MSE of 0.0473, MAE of 0.1374, and an R-squared (R^2) value of 0.97003760. These metrics are obtained after the model underwent 50 training epochs, indicating a substantial learning process. The MSE value of 0.0473 suggests that the squared difference between the predicted and actual values is relatively low. The MAE value of 0.1374 represents the average magnitude of the errors, which provides insight into the absolute difference between predicted and actual values. The predicted and actual energy values after 1 day are shown in Figure 4.7. The R^2 value of 0.97003760 indicates that the model explains approximately 97% of the target variable's variability. It demonstrates a strong fit to the data.

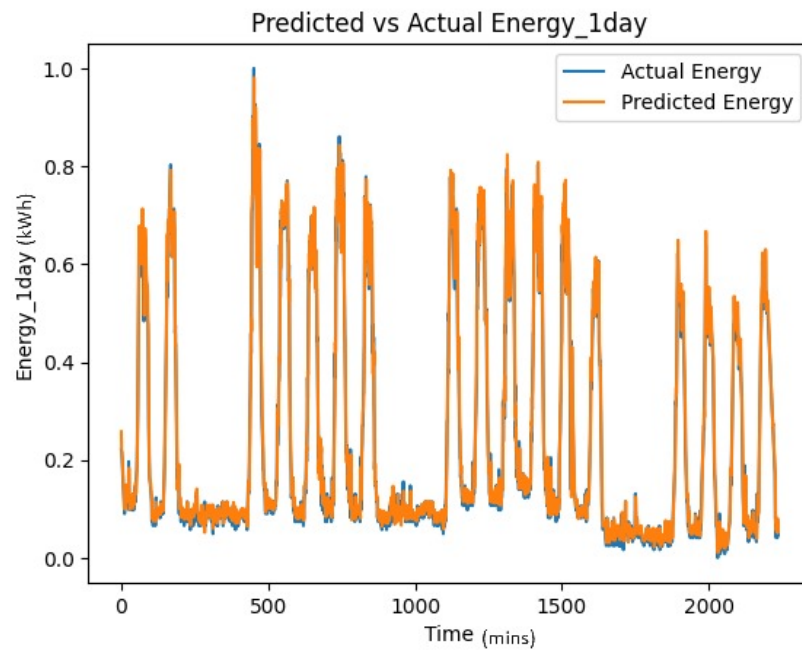


Figure 4.7: Month in Year: predicted vs. actual energy for the 1st day of August, based on a model trained with data from the first 3 weeks of July.

To assess the model's generalization ability, we conducted testing in an August month using the model trained on data exclusively from the month of July. Figure 4.8 shows the energy's actual and predicted values. The evaluation of the model's performance on the August data revealed the following metrics: MSE of 0.2077, MAE of 0.31112, and R-squared (R^2) value of 0.9723. The MSE value of 0.2077 indicates that the squared difference between the predicted and actual values in August is relatively moderate. The MAE value of 0.31112 represents the average magnitude of the errors in August, providing insight into the absolute difference between the predicted and actual values. The R^2 value of 0.9723 suggests that the model explains approximately 97% of the variability in the target variable for the month of August. It also indicates a strong fit for the data in August. These results indicate that the proposed hybrid model was trained completely based on July data. It also demonstrates promising performance when it is applied to the month of August. The high accuracy is indicated by the low MSE and MAE values and the high R^2 . It suggests that the model's predictions are reliable and accurate for predicting energy-related values in any month.

The coefficient of determination (R^2) is a statistical measure that indicates the proportion of variance in the dependent variable explained by the independent variables in a regression model. In the context of energy consumption prediction, R^2 is an important evaluation metric to assess the performance of different papers and their models.

4.5.3 Performance comparison

The performance of the proposed model is compared with other building energy prediction methods that use Multi Layer Perceptron (MLP), Artificial Neural Networks (ANN), and Deep Learning (DL) models. Figure 4.9 shows the R^2 value of the different methods. The simulation result implies that the proposed model increases the R^2 value by 35% compared to LPEENCKFDML, 11% compared to MLAIECPM, 21% to EC-

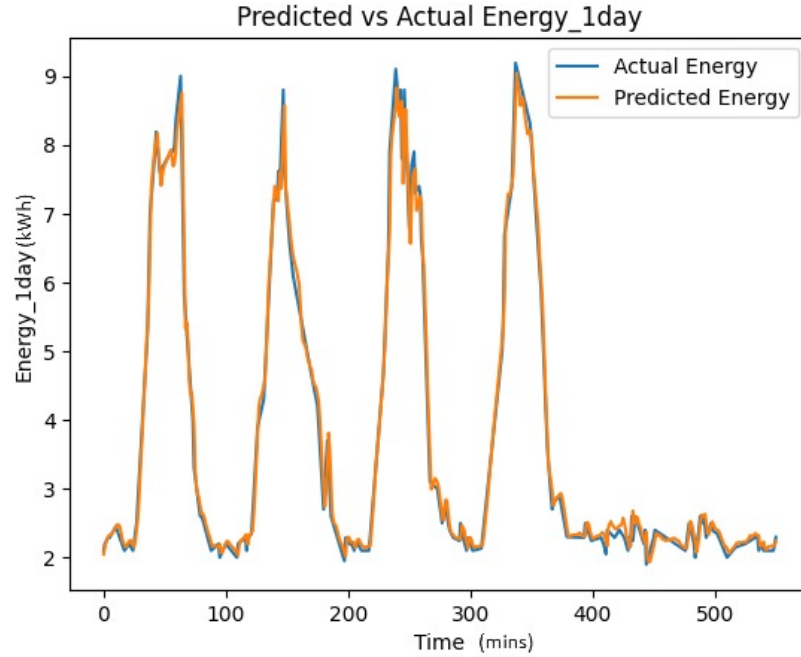


Figure 4.8: Day in Month: predicted vs. actual energy for the 1st day of the 4th week of July, based on a model trained with data from the first 3 weeks of July.

PLEBMAIE, and 12% to ESEGRPUML. This is because of the proposed CNN-BiLSTM hybrid model used occupancy count for energy prediction, which helped increase R^2 value.

Figure 4.10 shows the comparison plot of MAE values for the different energy prediction methods. Figure 4.10 depicts that the proposed scheme reduces MAE by 15% than MLAECDP, 32% than CAMLMECPRB, 35% than ECPLEBMAIE, and 16% than ESEGRPUML. This is because of the hybridization where CNN-BiLSTMs are jointly considered to predict building energy.

Figure 4.11 shows the comparison plot of MSE values for the different energy prediction methods. Figure 4.11 depicts that the proposed scheme reduces MSE by 35% than CAMLMECPRB, 22% than ECPLEBMAIE, 27% than ESEGRPUML, and 12% than SSAMLMECP. This is because of the hybridization where CNN-BiLSTMs are jointly considered to predict building energy. This implies that the proposed energy consumption prediction model outperforms the models presented in the other papers

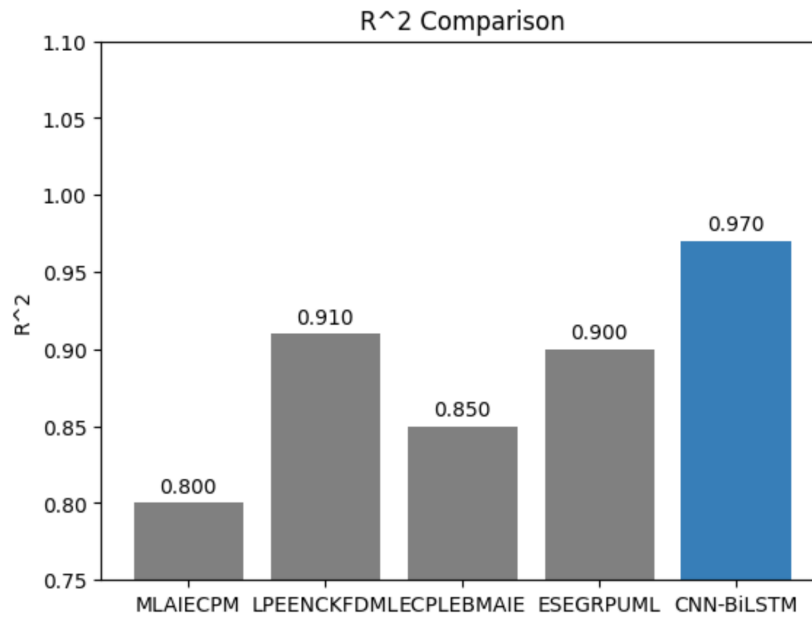


Figure 4.9: Comparing R^2 with other papers

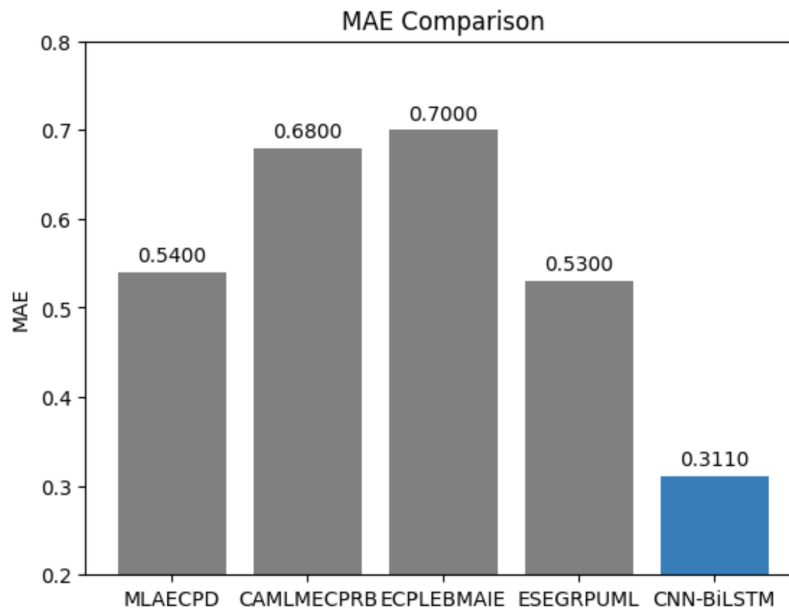


Figure 4.10: Comparing MAE with other papers.

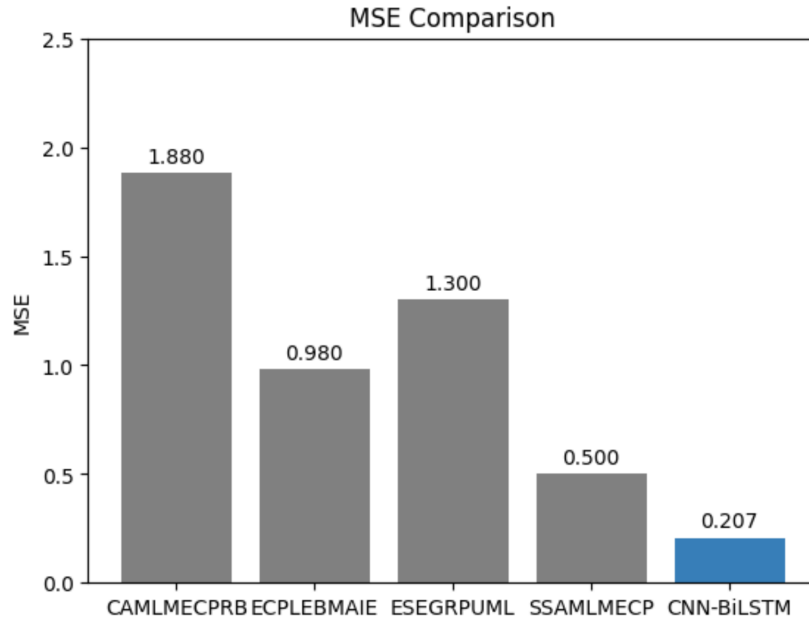


Figure 4.11: Comparing MSE with other papers.

regarding accuracy and precision. The lower MAE value suggests that the proposed model's predictions deviate less from the actual energy consumption values on average, indicating higher accuracy and reliability.

4.5.4 Month-wise training and testing

After training on individual monthly datasets and testing against each other each month, relatively consistent accuracy values are observed. Table 4.5 presents R^2 values obtained by training each month and testing against every other month. Surprisingly, the gaps in the data exhibited minimal impact on the performance and accuracy of the trained models. Notably, testing a model in the same month it was trained consistently yielded higher accuracy than expected. It indicates the robustness of the models in capturing and generalizing patterns within each specific month. Each month's R^2 values are higher when tested against itself. It emphasizes the models' proficiency in capturing intra-month patterns. The Training Loss and MAE are plotted over epochs to examine the learning rate across different monthly datasets. Figure 4.12 shows that

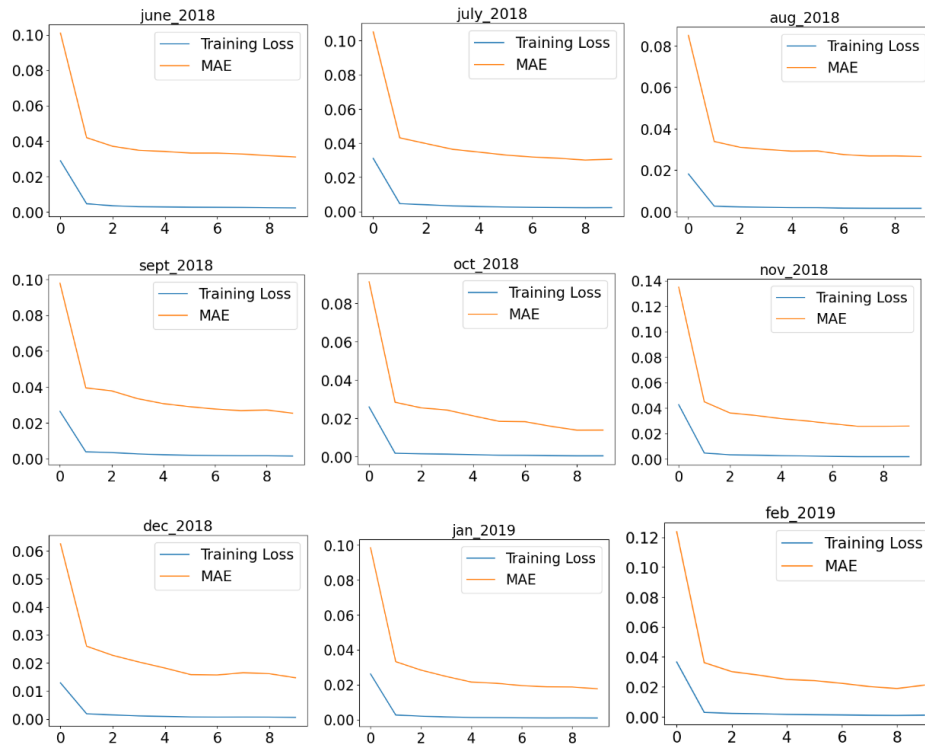


Figure 4.12: MSE and MAE drop during each month’s training.

varying gaps in the dataset have minimal impact on the learning rate. November 2018 is a month with a higher proportion of missing data points. This month exhibited a slight delay in reaching the minimum achievable accuracy. This observation suggests that the learning rate is relatively resilient to gaps in the dataset. It emphasizes the models’ ability to adapt and learn effectively, even in the face of missing information.

In Table 4.5, the R^2 values indicate that missing and correlation values have no substantial impact on the model’s performance. October 2018 has 3.3% missing values and a correlation value of 0.84. However, it performs better than June 2018 regarding accuracy, where June 2018 has 0% missing values and a correlation value of 0.91. This counterintuitive result suggests that factors beyond missing values and correlation influence the model’s performance. Moreover, October performs well when tested against itself and when trained and tested against other months. Furthermore, it highlights October’s robust performance across various scenarios. The model has also been tested

Table 4.5: Results of training one month and testing all months with the model.

Training/ Testing	June 18	July 18	Aug 18	Sep 18	Oct 18	Nov 18	Dec 18	Jan 19	Feb 19
June 18	97.47	97.55	97.43	97.97	98.99	97.47	98.51	98.37	98.95
July 18	97.27	97.86	97.30	98.11	98.70	97.62	98.66	98.33	98.86
Aug 18	97.22	97.60	97.75	98.26	99.24	97.83	98.68	98.44	98.99
Sep 18	97.19	97.64	97.44	98.46	99.34	98.12	98.90	98.67	99.14
Oct 18	96.89	97.38	97.18	98.22	99.54	98.02	98.93	98.67	99.25
Nov 18	96.84	97.44	97.09	97.99	99.21	98.14	98.77	98.54	99.06
Dec 18	96.87	97.39	97.15	98.13	99.24	97.97	98.99	98.59	99.16
Jan 19	96.88	97.42	97.19	98.23	99.44	97.97	98.89	98.87	99.19
Feb 19	96.76	97.24	97.09	98.08	99.19	97.89	98.81	98.58	99.28

on a different dataset, which yielded an R^2 accuracy of 93%, proving the generalizability and robustness of the proposed model [129]. The limitations of the proposed model are the availability and quality of the data. Real-world datasets often contain missing or noisy data. This makes it difficult to integrate seamlessly with existing systems and processes. It is essential for maximizing the model’s impact on energy management and optimization efforts.

4.5.5 Comparison with existing works

Current methods for predicting energy consumption, such as traditional forecasting, statistical models, and machine learning, offer various solutions. However, they often ignore the number of people in a building (occupancy count) and fail to capture the complex patterns in time and space within the data. Occupancy count is crucial for predicting a building’s energy use and makes the model more reliable across different building types and locations. Including this factor leads to more accurate predictions [117].

Aaslid and colleagues highlighted the importance of accurate energy predictions for managing renewable energy storage in microgrids. Similarly, Shamsi and Cuffe studied prediction markets to improve forecasts of renewable energy sources. Gan and others developed a data-driven system that uses advanced forecasting methods to manage

interconnected microgrids, also emphasizing the need for precise energy predictions in smart buildings [123].

Qiao and colleagues proposed a hybrid model for predicting building energy consumption, but its performance depends on the quality of the training data, which limits its practical use. Fakour and team explored machine learning combined with uncertainty quantification to improve prediction reliability, but these methods are too complex for real-time use in large buildings. Ma and colleagues aimed to improve energy prediction in green buildings using machine learning, but their approach also struggles with limited data and the challenge of applying the model to different types of buildings and locations [126].

4.6 Conclusion

This work proposes a novel energy prediction mechanism with occupancy count using the CNN-BiLSTM hybrid model for smart buildings. It combines CNN and BiLSTM architectures to address the challenges associated with accurate energy prediction. Furthermore, occupancy count is considered an additional feature for accurate energy prediction. The proposed model successfully captures the spatial characteristics of the data using CNN layers that help extract meaningful features related to energy consumption. The subsequent BiLSTM layers effectively model the temporal dependencies and capture long-term patterns in the data. The proposed hybrid model effectively predicts energy consumption after 1 day with high accuracy. The proposed scheme significantly improves energy prediction accuracy in terms of MAE, MSE, and R-squared (R^2). The simulation results indicate that the proposed scheme outperforms the existing state-of-the-art algorithm. The accurate predictions obtained from the model contribute to reducing energy waste, increasing energy efficiency, and ultimately reducing costs and environmental impact. The proposed hybrid model is applicable in various areas, such as industry building automation, smart city development, administrative building

automation, and residential building automation.

Due to different electrical equipment, residential and academic buildings utilize different amounts of energy when it comes to occupancy. As a result, the occupancy-based energy consumption dataset for academic buildings is considered and proposed in the next chapter.

