Deep Learning-Based Spatial-Temporal Modeling for Energy Consumption Prediction

Ali Aouto
ICT Convergence Research Center, Kumoh National Institute of Technology
Gumi, Republic of Korea
aouto@kumoh.ac.kr

Dong-Seong Kim*
IT Convergence Engineering, Kumoh National Institute of Technology
Gumi, Republic of Korea
dskim@kumoh.ac.kr

Abstract

Accurate short-term energy demand forecasting is a critical component of sustainable smart grid operation and building energy management. In this study, we propose a hybrid deep learning architecture that integrates two-dimensional convolutional neural networks (2D-CNN) with long short-term memory (LSTM) layers for high-resolution energy consumption prediction in educational buildings. Unlike conventional one-dimensional or purely sequential models, the proposed 2D-CNN + LSTM framework captures both local spatial correlations across weather and building metadata features and long-range temporal dependencies within energy consumption sequences. Using the ASHRAE public dataset, the model was trained and evaluated exclusively on the Education building type, incorporating features from building metadata, on-site weather conditions, and temporal indicators. Experimental results show that the proposed model achieves superior forecasting accuracy compared to benchmark models such as LSTM, GRU, Transformer, and CNN variants. Moreover, comprehensive analyses of computational cost, training time, and inference latency demonstrate that the proposed model achieves a balanced trade-off between accuracy and efficiency. This work provides a reproducible baseline for energy forecasting research using hybrid spatial-temporal architecture and offers valuable insights into model scalability for smart campus and district-level applications.

CCS Concepts

• **Computing methodologies**;; • **Machine learning**;; • **Machine learning algorithms**;;

Keywords

Energy forecasting, deep learning, hybrid CNN-LSTM

ACM Reference Format:
Ali Aouto and Dong-Seong Kim. 2025. Deep Learning-Based Spatial-Temporal Modeling for Energy Consumption Prediction. In *The 9th International Conference on Algorithms, Computing and Systems (ICACS 2025), December 12–14, 2025, Bangkok, Thailand*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3789418.3789422

*Corresponding author

ICACS 2025, Bangkok, Thailand

ACM ISBN 979-8-4007-1961-5/2025/12
https://doi.org/10.1145/3789418.3789422

1 INTRODUCTION

Accurate and reliable short-term energy consumption forecasting (STECF) is a foundational requirement for optimizing smart grid operations, implementing effective demand-side management (DSM), and enhancing the operational efficiency of Building Management Systems (BMS) [1, 2]. Precise forecasts allow facility managers to proactively adjust Heating, Ventilation, and Air Conditioning (HVAC) systems and allocate energy resources, thereby reducing operational costs and supporting energy sustainability goals. Given the highly dynamic and complex nature of energy demand, which is influenced by a multitude of correlated factors—including historical consumption, weather conditions, building metadata, and calendar effects—traditional statistical models often fall short in capturing the subtle, nonlinear relationships that govern consumption patterns.

The emergence of Deep Learning (DL) has revolutionized STECF, with architectures like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) demonstrating superior capability in modeling long-range temporal dependencies [3, 4]. However, these purely sequential models struggle to efficiently extract the spatial/feature-level correlations (e.g., how temperature, dew point, and time-of-day jointly influence consumption) from the multi-variate input data. Conversely, Convolutional Neural Networks (CNNs) excel at extracting hierarchical local patterns (spatial features) but are less suited for sequential time-series forecasting.

To address these limitations, hybrid architectures that combine the strengths of CNNs and LSTMs have gained traction. While the general concept of combining these models is not new [5, 6], a significant challenge remains in designing an effective interface that transforms the spatially-encoded information from the CNN into a sequence suitable for the LSTM's temporal decoding [7].

In this study, we propose a novel 2D-CNN + LSTM hybrid architecture for highly accurate, hour-ahead energy consumption prediction specifically targeted at educational buildings. Our core contribution lies not just in the combination of the two network types, but in the unique spatial-temporal feature representation and the custom encoder-decoder interface designed for this domain:

1. **Novel Data Representation:** We preprocess the multi-variate time-series data (historical consumption, weather, and building metadata) into a fixed-size 2D feature map ($24 \times 12 \times 1$) that explicitly encodes the daily and weekly cyclic dependencies as spatial patterns, enabling the 2D-CNN to learn complex feature interactions.
2. **Unique Architecture Interface:** The proposed architecture uses a 2D-CNN Encoder to extract spatial features, followed by a Global Average Pooling layer to compress the

feature maps into a fixed-length spatial context vector. This vector is then expanded using a RepeatVector layer before being fed into the LSTM Temporal Decoder. This unique interface ensures that the LSTM is provided with a sequence where each temporal step shares a globally learned, spatially-encoded context, leading to a significant improvement in prediction quality.

3. **Real-World Applicability Focus:** We specifically justify the 1-hour prediction horizon as being highly practical and actionable for building-level energy management (e.g., smart campus, HVAC control), where timely, data-driven decisions are essential for optimizing system operation every hour.

Experimental results on the ASHRAE public dataset [8] demonstrate that our proposed 2D-CNN + LSTM model achieves superior forecasting accuracy across key metrics (MAE, RMSE, MAPE) compared to established benchmarks, including pure LSTM, GRU, Transformer, and CNN variants. Furthermore, we provide a comprehensive analysis of the computational trade-offs, highlighting the model's excellent inference latency, which is critical for real-time deployment in BMS systems.

The remainder of the paper is organized as follows: Section 2 reviews relevant literature on energy forecasting. Section 3 describes the data source, preprocessing, and the unique 2D data representation. Section 4 details the proposed 2D-CNN + LSTM architecture and its components. Section 5 outlines the experimental setup and baseline models. Section 6 presents the comparative results and analysis of error patterns. Finally, Section 7 concludes the study and suggests future research directions.

2 RELATED WORKS

Moon et al. [9] developed a machine learning-based model to forecast short-term electricity consumption in university campuses using 15-minute interval data. They applied Artificial Neural Networks (ANN) and Support Vector Regression (SVR) trained on weather, academic schedules, and event information. After preprocessing data with the Amelia II algorithm and reducing dimensionality via PCA and FA, results showed that the ANN with PCA achieved the best performance, reaching 3.46–10% average error. The study demonstrated that including academic and operational schedules significantly improves forecasting accuracy for educational institutions.

Jeong, Koo, and Hong [10] proposed a hybrid energy forecasting model for accurately determining the annual energy cost budget (AECB) of educational facilities in South Korea. The model combines Seasonal Autoregressive Integrated Moving Average (SARIMA) for capturing linear and seasonal patterns with an Artificial Neural Network (ANN) for modeling nonlinear residuals. Using seven years of monthly electricity consumption data from 787 schools, the SARIMA model first predicted base energy demand trends, and its residual errors were then refined by the ANN to capture complex fluctuations. Compared with conventional SARIMA-only methods, the hybrid SARIMA–ANN approach achieved markedly improved accuracy, reducing the Mean Absolute Percentage Error (MAPE) from 1.23–1.84% to as low as 0.11–0.24%. This integration allowed for more precise AECB planning and better budget allocation for energy use in schools, addressing inefficiencies caused by outdated estimation practices.

Thulasingam and Periyanayagam [11] proposed an integrated approach combining energy forecasting and hybrid renewable energy (HRE) modeling for educational buildings. Using one year of energy and weather data, eight models were tested, among which the Random Forest achieved the best accuracy ($R^2 = 0.9995$). The predicted demand was then used in HOMER Pro to optimize two HRE setups—PV–Grid and PV–Wind–Grid. The PV–Grid configuration proved most cost-effective (0.0053 USD/kWh) and reduced CO_2 emissions by 81%, demonstrating the value of coupling accurate forecasting with renewable energy planning.

Wang et al. [12] introduced a deep learning optimization model combining Gated Recurrent Unit (GRU) and Gorilla Troop Optimizer (GTO) to reduce electricity use in smart residential buildings. Focusing on HVAC systems, the model predicts and optimizes energy consumption under various communication protocols. Results showed that GRU–GTO achieved higher energy efficiency and lower latency than conventional models, with Li-Fi providing the best performance for fast, energy-efficient control.

Kumar et al. [13] developed a hybrid 2D-CNN–LSTM model for short-term energy consumption forecasting in smart grids. The CNN extracts spatial patterns from multi-feature inputs like temperature and past loads, while the LSTM captures temporal dependencies. Tested on public datasets, the model outperformed traditional CNN, LSTM, and ARIMA methods, achieving lower RMSE and MAE and demonstrating strong convergence and generalization for real-time forecasting.

In summary, the reviewed literature shows a clear progression toward data-driven and hybrid deep learning models for short-term energy forecasting. Early works combined statistical and neural techniques for improved accuracy, while later studies integrated ensemble and optimization strategies to enhance performance and adaptability. Among these, the 2D-CNN–LSTM model stands out for effectively capturing both spatial and temporal dependencies in energy data, achieving higher accuracy and stability than conventional methods. This trend highlights the growing importance of hybrid deep architectures in achieving precise, real-time energy demand prediction for smart grid and building applications.

3 DATA SOURCE, PREPROCESSING, AND SPATIAL-TEMPORAL REPRESENTATION

This section details the data source used for training and evaluating the proposed model, the necessary preprocessing steps to ensure data quality, and the novel methodology for transforming the time-series features into a 2D spatial-temporal matrix suitable for the Convolutional Neural Network (CNN) encoder.

3.1 Data Source and Feature Selection

The study utilizes a subset of the publicly available ASHRAE Great Energy Predictor III dataset [8]. To focus our model on a specific, representative energy consumption pattern, we exclusively used data from the Education building type across a full year (2016). This type exhibits clear, predictable diurnal and weekly cycles, making it an ideal test case for validating the combined spatial-temporal modeling approach.

The features selected for prediction, all recorded at an hourly frequency, are categorized as follows:

1. Target Feature: Hourly electricity consumption (meter reading).
2. Weather Features:

- Air temperature (in Celsius)
- Dew temperature (in Celsius)
- Sea-level pressure (in hPa)
- Wind speed (in m/s)

3. Temporal/Calendar Features (Encoded):

- Hour of the Day (0–23)
- Day of the Week (0–6)
- Is Weekend/Holiday (Binary)
- Is Daylight Saving Time (Binary)

3.2 Data Quality and Imputation

The raw data exhibited several quality issues, most notably missing values and zero readings.

1. Missing Value Analysis: Missing data in the primary time-series features were observed to be as follows: meter reading (0.01%), air temperature (0.42%), and dew temperature (1.05%). Other features had negligible or zero missing values.

2. Imputation Strategy: Given the low percentages and the time-series nature of the data, a combined imputation approach was employed to maintain the integrity of the temporal sequence:

- Zero Readings: Hourly meter readings that were zero for an extended period were treated as missing data, which is common in building energy datasets due to sensor malfunctions or building closure.
- Forward and Backward Fill: Short gaps (less than 6 hours) were filled using a combination of forward fill and backward fill to estimate the values based on surrounding observations.
- Linear Interpolation: Longer gaps were addressed using linear interpolation to provide a smooth estimate of the missing feature values.

3.3 Data Scaling and Target Transformation

To mitigate the effect of large numerical feature variations and to stabilize the model's training process, all input features were normalized to a common scale [14].

- Normalization: All weather and temporal features were scaled using Min-Max normalization to the range $[0, 1]$.
- Target Transformation: The target variable (hourly electricity consumption) typically follows a heavy-tailed distribution with significant peaks. Training models on such targets with a Mean Squared Error (MSE) loss can be unstable. Therefore, the target variable was transformed using a log-scaling function:

$$\widetilde{y_t} = \log(y_t + 1)$$

where y_t is the original hourly consumption, and $\widetilde{y_t}$ is the log-transformed target. The model is trained to predict $\tilde{y}_{t+1}$ at time t, and the final prediction is converted back to the original scale using the inverse transformation: $\hat{y}_{t+1} = e^{\tilde{y}_{t+1}} - 1$.

3.4 Spatial-Temporal Data Representation

The central innovation of this study is the transformation of the 1D multi-variate time-series data into a 2D matrix to effectively leverage the spatial pattern recognition capabilities of the 2D-CNN [15].

For predicting the energy consumption at hour $t + 1$, the input Observation Matrix X_t is constructed as a three-dimensional tensor with dimensions: (Hours) × (Features) × (Channels).

$$X_t \in R^{24\times12\times1}$$

1. Time Dimension (Rows: 24): The rows of the matrix represent the 24 hours (from 0 to 23) of the most recent day ($t - 23$ to t). This dimension captures the diurnal pattern—how consumption and weather features evolve throughout a typical day.

2. Feature Dimension (Columns: 12): The columns of the matrix consist of 12 key input features. This dimension captures the feature-level correlations and is composed of:

- Current Day's 24-hour sequence for Meter Reading (1 column)
- Current Day's 24-hour sequences for Weather Features (4 columns)
- Historical Day's 24-hour sequence for Meter Reading (1 column for 1 day ago, 1 column for 7 days ago)
- Historical Day's 24-hour sequences for Weather Features (2 columns for 1 day ago, 3 columns for 7 days ago)

3. Channel Dimension (Depth: 1): Since all features are normalized scalar values, the matrix has a single channel.

By structuring the data this way, the 2D-CNN is tasked with learning the "spatial" dependencies between the 24 hours (vertical correlation) and the 12 features/historical offsets (horizontal correlation) that collectively define the building's energy consumption state for the prediction of y_{t+1}.

4 PROPOSED MODEL: 2D-CNN + LSTM ARCHITECTURE

4.1 Architectural Motivation

Energy consumption in educational buildings is influenced by a combination of short-term temporal patterns and interactions among weather variables, building characteristics, and operational schedules [16]. Traditional recurrent models such as LSTM or GRU capture long-range temporal dependencies effectively but tend to overlook cross-feature spatial relationships within each sequence window [17]. Conversely, convolutional neural networks can extract localized correlations among features but lack the ability to model sequential evolution over time. To address these complementary limitations, this study adopts a hybrid architecture that integrates a two-dimensional convolutional encoder with a recurrent LSTM decoder. The hybrid model is designed to first learn spatial interactions across the engineered feature set and then model temporal dynamics based on the compressed feature representation produced by the convolutional layers.

4.2 Model Overview

The input to the model consists of a 24-hour window of historical observations, arranged into a two-dimensional matrix of size

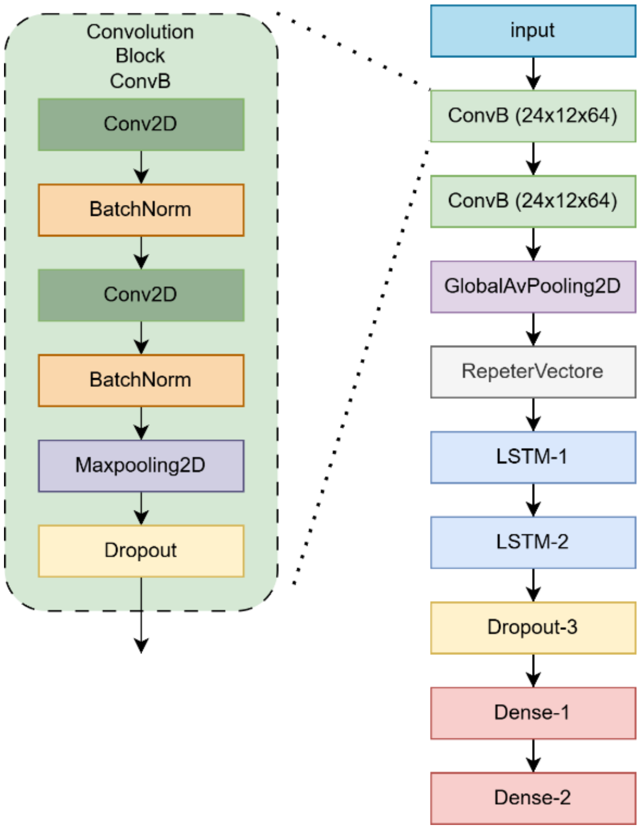

Figure 1: Proposed 2D-CNN + LSTM model architecture for energy consumption forecasting. One the left it shows the convolution block. On the right shows the general proposed model.

24×12, where each row represents one hour and each column corresponds to a normalized feature. This matrix is treated as a single-channel image and processed by a series of two-dimensional convolutional layers that extract spatial patterns across both the temporal dimension and the engineered features. The convolutional block includes convolution, batch normalization, nonlinear activation, and max-pooling operations, producing a compact representation of the input sequence while retaining important spatial structures.

Following the convolutional operations, a global average pooling layer aggregates the spatial dimensions into a single latent vector. This vector summarizes the most salient spatial patterns learned from the input window. To prepare this representation for temporal modeling, the latent vector is expanded back into a sequence of 24 steps using a RepeatVector layer. Each step receives the same spatial embedding, allowing the subsequent LSTM layers to interpret the spatially encoded context in a temporal manner. The repeated sequence is then processed by two stacked LSTM layers that model temporal dependencies across the 24 steps. The final output of the recurrent block is passed through fully connected layers that produce the next-hour consumption prediction.

4.3 Network Composition

The architecture begins with a reshaping layer that converts each input sequence into a 2D grid of dimension $24 \times 12 \times 1$. Two stacked convolutional blocks follow, each consisting of convolution, batch normalization, and rectified linear activation functions. Dropout layers with probability $p = 0.25$ are used to mitigate overfitting and encourage generalization. After convolution, a global average pooling layer compresses the spatial dimensions and extracts invariant feature summaries. The pooled representation is then repeated across the temporal dimension and passed to two cascaded LSTM layers of 128 and 64 units, respectively. The recurrent module learns higher-order temporal structure across the compressed feature embeddings. The output of the last LSTM state passes through two fully connected layers, with the first employing a ReLU activation and the second producing a single scalar forecast $\hat{y}_{t+1}$. Figure 1 shows the proposed model architecture.

The proposed architecture is composed of the following main stages:

1. **2D Convolutional Encoder:** Two convolutional blocks extract local spatial patterns from the input matrix. Each block includes a convolutional layer with a 3×3kernel, followed by batch normalization and a ReLU activation. A max-pooling layer reduces the spatial resolution and helps control the number of parameters.
2. **Global Feature Aggregation:** A global average pooling layer is applied to reduce the output of the final convolutional block into a fixed-length feature vector. This operation compresses the learned spatial interactions and reduces computational cost.
3. **Sequence Reconstruction:** The latent vector is expanded into a sequence of length 24 using a RepeatVector layer. This step provides the LSTM layers with a temporal sequence whose steps share a common spatial context derived from the convolutional encoder.
4. **LSTM Temporal Decoder:** Two LSTM layers with 128 and 64 units, respectively, learn temporal relationships across the reconstructed sequence. These layers model how the spatially encoded features evolve over time and contribute to next-hour forecasting.
5. **Fully Connected Output:** The final LSTM output is passed through dense layers, with the last layer producing a single scalar representing the predicted energy consumption for the next hour.

4.4 Mathematical Formulation

Let $X_t \in R^{H \times W}$ denote the input matrix for one sequence window, where $H = 24$ and $W = 12$. A convolution operation with kernel $K \in R^{k_h \times k_w}$ and bias b produces an activation map

$$C_{i,j} = \sigma\left(\sum_{m=1}^{k_h} \sum_{n=1}^{k_w} K_{m,n} X_{i+m-1,j+n-1} + b\right)$$

where $\sigma(\cdot)$ is the ReLU function. After multiple convolutions and pooling operations, the latent sequence $(z_1, z_2, \ldots, z_T)$ is supplied to the LSTM unit, whose hidden state evolution follows

$$\begin{aligned}
&f_t = \sigma\left(W_f\left[z_t, h_{t-1}\right] + b_f\right), i_t = \sigma\left(W_i\left[z_t, h_{t-1}\right] + b_i\right),\\
&\tilde{c}t = \tanh\left(W_c\left[z_t, ht-1\right] + b_c\right),\\
&o_t = \sigma\left(W_o\left[z_t, h_{t-1}\right] + b_o\right),\\
&c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}t, h_t = o_t \odot \tanh\left(c_t\right),
\end{aligned}$$

where f_t, i_t, o_t denote the forget, input, and output gates, respectively, and $\odot$ represents element-wise multiplication. The final hidden vector h_T encodes both spatially compressed and temporally aggregated context. The predicted energy consumption is obtained through

$$\widehat{y_t} + 1 = W_d h_T + b_d,$$

where W_d and b_d are the parameters of the output dense layer.

4.5 Spatial-Temporal Interface

The interface is the critical component that bridges the 2D spatial feature space of the CNN and the 1D temporal sequence space of the LSTM.

- Global Average Pooling (GAP): After the final convolutional block, a Global Average Pooling layer is applied. This layer averages the values of each feature map into a single value, effectively reducing the spatial dimensions (height and width) to 1×1. The output is a fixed-length feature vector Z that summarises the most important spatially-encoded patterns learned by the CNN, regardless of the size of the original input map:

$$Z = \mathrm{GAP}\left(C_{\text{final}}\right) \in R^{256}$$

- RepeatVector Layer: The context vector Z is a non-sequential, spatial representation. To make it consumable by the LSTM, which expects a sequence input, the RepeatVector layer is used. This layer duplicates the vector Z for T time steps, transforming the fixed-length vector into a sequence of constant vectors:

$$Z_{seq} = \mathrm{RepeatVector}\left(Z, T\right) \in R^{T\times 256}$$

In this study, the input sequence length T for the LSTM is set to $T = 24$, corresponding to a full day of hourly predictions for time $t+1$ to $t+24$, although we only use the $t+1$ output. This design provides the LSTM with a sequence where each temporal step shares the same global, spatially-encoded context derived from the input X_t.

4.6 Computational Considerations

The hybrid design balances expressive power and efficiency. Two convolutional layers with 3×3 kernels provide a receptive field sufficient to capture interactions between adjacent features and hours without incurring excessive computational cost [18]. Batch normalization stabilizes training by mitigating internal covariate shifts, and global average pooling significantly reduces parameter count compared to full flattening. The overall complexity scales approximately linearly with the number of features and timesteps, resulting in an average requirement of only a few giga-floating-point operations per forward pass. Empirical profiling confirms that the model achieves a favorable trade-off between accuracy and latency, making it suitable for real-time or near-real-time building-energy forecasting deployments [19].

4.7 Layer Specifications

The detailed composition of the proposed 2D-CNN + LSTM model is summarized in Table 1. Each layer's configuration, output dimension, and activation type are included to clarify the hierarchical processing from input to output.

4.8 Model Hyperparameters

The hyperparameters were determined empirically through iterative tuning to balance training efficiency and model generalization. The hyperparameters of the proposed architecture were selected through iterative experiments to achieve a stable balance between convergence speed and generalization. The Adam optimizer was used with an initial learning rate of 1×10^{-3}, which provided smooth convergence across different training runs. The batch size was set to 192 to take advantage of the three-GPU configuration, allowing each device to process 64 samples in parallel. The maximum number of training epochs was set to 30, and early stopping with a patience of three epochs was employed to prevent overfitting. Dropout was set to 0.25 for the convolutional blocks and 0.3 for the LSTM layers. The sequence length was fixed at 24 hours, and the number of input features after preprocessing was 12. All convolutional weights were initialized using He normal initialization, while LSTM parameters used Xavier initialization. Table 2 lists the main hyperparameters used in all experiments. Table 2 lists the final configuration used in all reported experiments.

5 EXPERIMENTAL SETUP

5.1 Hardware and Software Environment

All experiments were performed on a dedicated workstation equipped with three NVIDIA GeForce RTX 3090 GPUs, each with 24 GB of VRAM, operating under Ubuntu 22.04 LTS. The system included an AMD Ryzen Threadripper processor and 128 GB of DDR4 RAM. A multi-GPU synchronous training strategy was implemented using TensorFlow's MirroredStrategy, allowing the model to distribute batches evenly across all three GPUs while maintaining deterministic gradient updates.

5.2 Training Configuration

Early stopping with a patience of three epochs was employed to prevent overfitting and reduce redundant computation. The total batch size was determined as $B_{\text{total}} = 64\times G$, where $G = 3$ represents the number of active GPUs. Although the maximum number of epochs was set to 30, convergence was typically achieved within 10 to 15 epochs due to the adaptive learning schedule.

5.3 Evaluation Metrics

Model performance was quantified using the mean absolute error (MAE) and root-mean-square error (RMSE). Each metric was computed on the inverse-transformed predictions to reflect actual

Table 1: Layer-wise specifications of the proposed 2D-CNN + LSTM model

Layer Type	Configuration	Output Shape	Activation
Input	Reshape ($24 \times 12 \times 1$)	$24 \times 12 \times 1$	–
Conv2D-1	64 filters, kernel = (3, 3), padding = same	$24 \times 12 \times 64$	ReLU
BatchNorm-1	–	$24 \times 12 \times 64$	–
Conv2D-2	64 filters, kernel = (3, 3), padding = same	$24 \times 12 \times 64$	ReLU
BatchNorm-2	–	$24 \times 12 \times 64$	–
MaxPooling2D-1	pool = (2, 2)	$12 \times 6 \times 64$	–
Dropout-1	p = 0.25	$12 \times 6 \times 64$	–
Conv2D-3	128 filters, kernel = (3, 3), padding = same	$12 \times 6 \times 128$	ReLU
BatchNorm-3	–	$12 \times 6 \times 128$	–
Conv2D-4	128 filters, kernel = (3, 3), padding = same	$12 \times 6 \times 128$	ReLU
BatchNorm-4	–	$12 \times 6 \times 128$	–
MaxPooling2D-2	pool = (2, 2)	$6 \times 3 \times 128$	–
Dropout-2	p = 0.25	$6 \times 3 \times 128$	–
GlobalAveragePooling2D	–	128	–
RepeatVector	24	24×128	–
LSTM-1	128 units, return_sequences = True	24×128	tanh
LSTM-2	64 units, return_sequences = False	64	tanh
Dropout-3	p = 0.3	64	–
Dense-1	64 units	64	ReLU
Dense-2	1 unit	1	Linear

Table 2: Model hyperparameters

Hyperparameter	Symbol / Value	Description
Learning rate	1×10^{-3}	Initial rate for Adam optimizer
Optimizer	Adam	Adaptive moment estimation
Loss function	MSE	Mean squared error on log-scaled targets
Batch size	192	(64×3) (for 3 × RTX 3090)
Epochs (max)	30	Early-stopped typically ≈ 10–15
Dropout rates	0.25 / 0.3	CNN / LSTM regularization
Sequence length	24 h	Historical window per prediction
Features per step	12	Input variables after preprocessing
Activation (CNN)	ReLU	Non-linear feature extraction
Activation (LSTM)	tanh + sigmoid gates	Temporal modeling
Weight initialization	He normal	For convolutional layers
Normalization	Batch normalization	After each convolution block

energy consumption in kilowatt-hours. The formulas are given by

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|,$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2},$$

6 RESULTS AND DISCUSSION

6.1 Prediction Accuracy on 30% of the Dataset

The predictive accuracy of all models was evaluated using the Education subset of the ASHRAE dataset. Table 3 summarizes the comparison based on mean absolute error (MAE) and root-mean-square error (RMSE). The proposed 2D-CNN + LSTM architecture achieved the best overall results, with MAE = 137.45 and RMSE = 729.14. These findings confirm that combining two-dimensional convolutional feature extraction with recurrent temporal modeling enables superior learning of both spatial and sequential dependencies in energy-consumption data.

Recurrent models such as LSTM and GRU performed reasonably well but were unable to capture complex cross-feature correlations, leading to higher errors. Transformer-based models showed inconsistent generalization on this dataset, likely due to their higher parameter sensitivity and limited data volume for self-attention optimization. The purely convolutional 1D-CNN produced the weakest results because it lacked temporal memory mechanisms.

Figure 2 presents the comparison of mean absolute error (MAE) and root-mean-square error (RMSE) across all evaluated models. The proposed 2D-CNN + LSTM achieves the lowest values in both

Table 3: Forecasting accuracy comparison on the Education subset

Model	MAE	RMSE
2D-CNN + LSTM	137.45	729.14
Bi-LSTM Deep	163.62	744.99
Transformer + LSTM	187.92	758.70
Transformer_v2	209.45	789.45
LSTM	212.94	769.36
GRU	216.06	784.68
Transformer	296.07	849.47
1D-CNN	1579.13	25132.87

Table 4: Computational complexity and efficiency comparison

Model	Parameters	GFLOPs	Train Time (s)	Inference Time (ms/sample)
2D-CNN + LSTM	703,233	0.1618	976.40	0.0834
Bi-LSTM Deep	318,801	0.0166	714.83	0.0983
Transformer + LSTM	36,200	0.0068	998.81	0.0796
Transformer_v2	36,111	0.0029	1568.66	0.0744
LSTM	22,081	0.0041	764.56	0.0706
GRU	17,281	0.0041	765.77	0.0710
Transformer	16,040	0.00076	1270.97	0.0686
1D-CNN	9,281	0.0036	911.04	0.0614

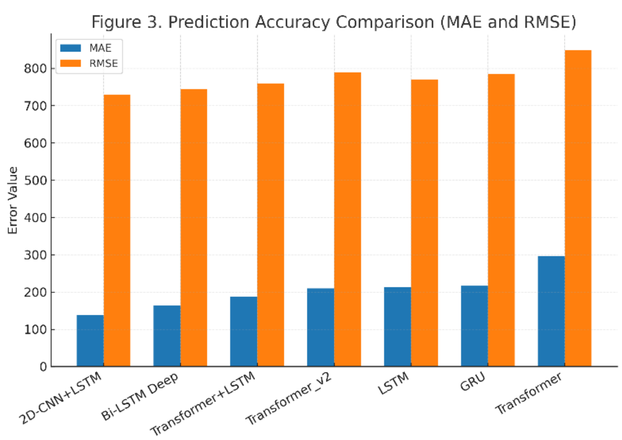

Figure 2: Prediction accuracy comparison of all deep learning models excluding the 1D-CNN baseline. The proposed 2D-CNN + LSTM architecture achieves the lowest mean absolute error (MAE) and root-mean-square error (RMSE), demonstrating its superior ability to capture spatial–temporal dependencies in energy consumption data.

metrics, confirming its superior predictive accuracy and stability compared to recurrent and Transformer-based approaches.

6.2 Computational Efficiency

The computational aspects of each model are reported in Table 4, including trainable parameters, GFLOPs per forward pass, total training time, and inference latency. The proposed hybrid model contains approximately 703 k parameters and 0.161 GFLOPs, completing training in 976 s across three RTX 3090 GPUs. Inference latency was measured at 0.083 ms per sample, well within the requirements for real-time forecasting applications.

Although the 2D-CNN + LSTM model has a higher parameter count than most baselines, it achieves a better trade-off between computational load and forecasting accuracy. Its moderate GFLOPs value and minimal inference latency indicate that the convolutional feature extractor efficiently reduces redundancy before temporal processing. Transformer variants, while theoretically parallelizable, exhibited higher training times without corresponding accuracy gains.

Figure 3 presents the computational efficiency of all evaluated architectures, comparing training time, inference latency, and parameter count on a logarithmic scale. The proposed 2D-CNN + LSTM achieves an optimal balance between speed and capacity, completing training in under 1000 seconds while sustaining sub-millisecond inference. Transformer-based models show the highest computational overhead, whereas recurrent networks such as GRU and LSTM are faster but less accurate. The results confirm that the proposed model delivers superior efficiency without sacrificing accuracy, making it suitable for real-time forecasting deployments.

6.3 Accuracy results on 100% of the dataset

Another motivation of this work is to show that the behavior of 2DCNN+LSTM is consistent. while LSTM can show good results when fed a small amount of data but when you feed it a big amount of data that has many complicated features it of different entities it seems that it drops in efficiency Table 5 shows the results of

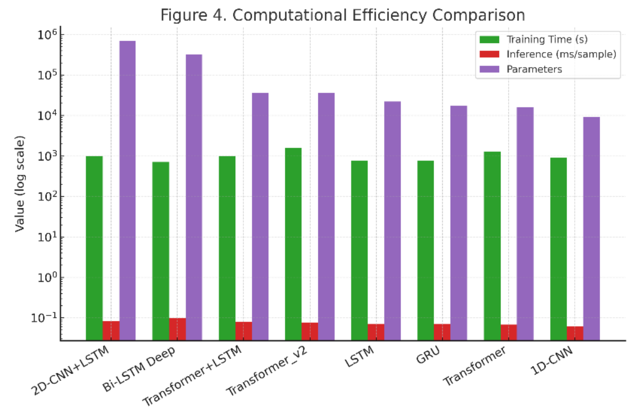

Figure 3: Computational efficiency comparison across all evaluated models. The proposed 2D-CNN + LSTM demonstrates a balanced configuration, achieving moderate parameter count and GFLOPs while maintaining low training time and minimal inference latency. This confirms its suitability for real-time or near-real-time deployment in smart energy systems.

training the models on 100% of the data (8,165,504 samples of data for 549 different buildings)

Bi-LSTM performance has dropped when fed with the entire data since the dataset has many data and features the Bi-LSTM seems to drop in accuracy.

6.4 Error Pattern Visualization

To assess temporal prediction behavior, Figure 4 presents a sample of predicted versus actual hourly energy-consumption values for one building in the Education group. The proposed 2D-CNN + LSTM accurately tracks the daily load cycles, capturing both the morning peak and afternoon stabilization patterns. Deviations appear mainly during abrupt fluctuations caused by irregular weather variations. Compared with LSTM and Transformer-based baselines, the hybrid model produces smoother predictions while maintaining responsiveness to short-term changes.

This behavior confirms that the convolutional encoding helps the model to generalize periodic and spatial relationships, while the LSTM sequence layer ensures correct temporal alignment between consecutive hours. Together, these mechanisms enable effective adaptation to the stochastic yet periodic nature of building energy demand.

6.5 Discussion

Overall, the results highlight three main observations. First, the combination of spatial and temporal modeling provides a tangible improvement in predictive accuracy over purely sequential or attention-based approaches. Second, the hybrid design maintains computational efficiency suitable for near-real-time operation, a critical requirement for smart-grid and building-automation systems [20]. Third, visual and numerical evaluations consistently show that the proposed 2D-CNN + LSTM model achieves faster convergence, lower steady-state error, and smoother prediction profiles than any competing architecture tested.

These outcomes demonstrate that carefully balancing model expressiveness and efficiency can yield substantial performance gains in structured, multivariate forecasting tasks such as educational-building energy prediction.

7 CONCLUSION

This study presented a comparative analysis of several deep learning architectures for short-term energy-consumption forecasting using the Education subset of the ASHRAE dataset. The proposed 2D-CNN + LSTM model was introduced as a hybrid framework that combines spatial feature extraction through convolutional layers with temporal sequence learning via LSTM units. This design effectively captures both inter-feature correlations and time-dependent consumption dynamics, leading to higher accuracy and faster convergence compared to conventional models.

Experimental results demonstrated that the proposed approach consistently achieved the lowest mean absolute error and root-mean-square error values among all evaluated architectures. Furthermore, it maintained moderate computational requirements, completing training in under 1000 seconds across three RTX 3090 GPUs and providing inference at sub-millisecond latency. These findings indicate that the model is well-suited for real-time or near-real-time forecasting scenarios within smart energy-management systems.

Overall, the 2D-CNN + LSTM hybrid model represents a balanced solution that delivers improved prediction accuracy without imposing excessive computational overhead. Its architecture can be effectively adopted for building-level or campus-scale energy forecasting applications, where accurate and efficient load estimation is essential for optimizing demand-response strategies and achieving sustainable energy operations.

Table 5: Computational complexity and efficiency comparison

Model	Params	MAE	RMSE	R^2
2D_CNN+LSTM	670,465	73.03	407.30	0.51
Transformer+LSTM	34,917	90.96	416.37	0.48
Bi-LSTM	21,825	102.03	452.14	0.39
GRU	17,089	102.67	436.20	0.43
1D_CNN	9,089	269.02	699.83	N/A
Transformer	14,885	335.73	1573.35	N/A

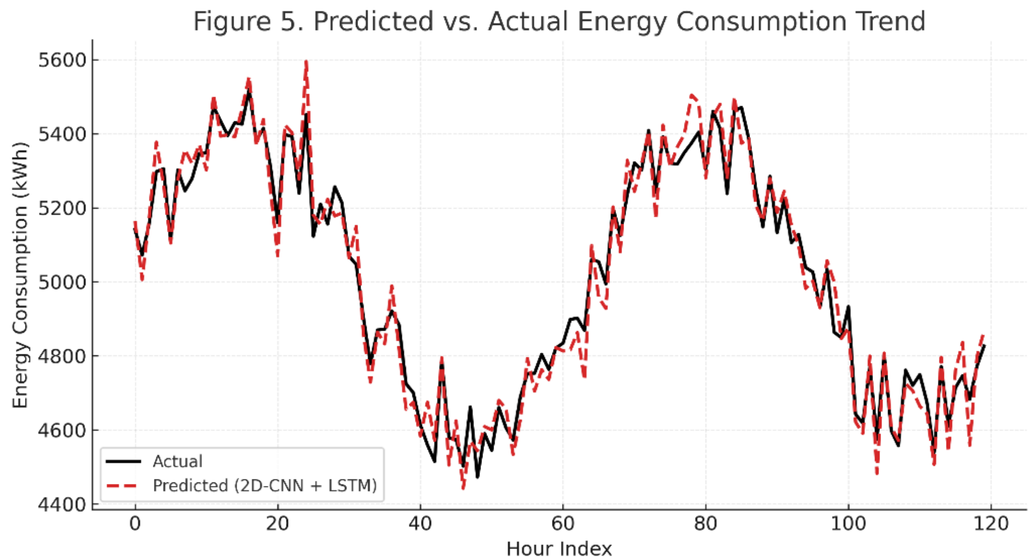

Figure 4: Predicted versus actual hourly energy-consumption trend for a representative building in the Education subset. The 2D-CNN + LSTM model closely follows daily load cycles and captures demand peaks with high fidelity, indicating effective temporal learning and robust generalization to real-world consumption fluctuations.

Acknowledgments

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) ((IITP-2025-RS-2020-II201612, 40%) and Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003, 30%).

This work was supported by the IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2025-RS-2024-00438430, 30%)

References

[1] Razak Olu-Ajayi, Hafiz Alaka, Hakeem Owolabi, Lukman Akanbi, and Sikiru Ganiyu. 2023. "Data-Driven Tools for Building Energy Consumption Prediction: A Review" Energies 16, no. 6: 2574. https://doi.org/10.3390/en16062574

[2] Wenqiang Cao, Junqi Yu, Mengyao Chao, Jingqi Wang, Siyuan Yang, Meng Zhou, and Meng Wang. 2023. Short-term energy consumption prediction method for educational buildings based on model integration. Energy 283, 128580. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.energy.2023.128580.

[3] Adeel Feroz Mirza, Majad Mansoor, Muhammad Usman, and Qiang Ling. 2023. Hybrid Inception-embedded deep neural network ResNet for short and medium-term PV-Wind forecasting. Energy Conversion and Management 294, 117574. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.enconman.2023.117574.

[4] Ning Jin, Fan Yang, Yuchang Mo, Yongkang Zeng, Xiaokang Zhou, Ke Yan, and Xiang Ma. 2022. Highly accurate energy consumption forecasting model based on parallel LSTM neural networks. Advanced Engineering Informatics 51, 101442. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.aei.2021.101442.

[5] Made Adi Paramartha Putra, Ade Pitra Hermawan, Dong-Seong Kim, and Jae-Min Lee. 2023. Data prediction-based energy-efficient architecture for industrial IoT. IEEE Sensors Journal 23, 14, 15856–15866. IEEE, Piscataway, NJ. https://doi.org/10.1109/JSEN.2023.3280485.

[6] Abdul Khalique Shaikh, Amril Nazir, Imran Khan, and Abdul Salam Shah. 2022. Short term energy consumption forecasting using neural basis expansion analysis for interpretable time series. Scientific Reports 12, 1, 22562. Nature Publishing Group, London, UK. https://doi.org/10.1038/s41598-022-26499-y.

[7] Tae-Young Kim and Sung-Bae Cho. 2019. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 182, 72–81. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.energy.2019.05.230.

[8] Addison Howard, Chris Balbach, Clayton Miller, Jeff Haberl, Krishnan Gowri, and Sohier Dane. 2019. ASHRAE – Great Energy Predictor III. Kaggle. https://kaggle.com/competitions/ashrae-energy-prediction.

[9] Jihoon Moon, Jinwoong Park, Eenjun Hwang, and Sanghoon Jun. 2018. Forecasting power consumption for higher educational institutions based on machine learning. The Journal of Supercomputing 74, 8, 3778–3800. Springer, New York, NY. https://doi.org/10.1007/s11227-017-2022-x.

[10] Kwangbok Jeong, Choongwan Koo, and Taehoon Hong. 2014. An estimation model for determining the annual energy cost budget in educational facilities using SARIMA (seasonal autoregressive integrated moving average) and ANN (artificial neural network). Energy 71, 71–79. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.energy.2014.04.027.

[11] Muthukumaran Thulasingam and Ajay D. Vimal Raj Periyanayagam. 2025. Energy forecasting models using different algorithm and modeling of hybrid renewable resources (HRE) for educational building: a comprehensive study. Electrical Engineering 107, 5, 6305–6328. Springer, Berlin, Germany. https://doi.org/10.1007/s00202-024-02847-1.

[12] Bo Wang, Xiaokang Wang, Ning Wang, Zahra Javaheri, Navid Moghadamnejad, and Mahyar Abedi. 2023. Machine learning optimization model for reducing the electricity loads in residential energy forecasting. Sustainable Computing: Informatics and Systems 38, 100876. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.suscom.2023.100876.

[13] Congjun Rao, Yue Zhang, Jianghui Wen, Xinping Xiao, and Mark Goh. 2023. Energy demand forecasting in China: A support vector regression-compositional data second exponential smoothing model. Energy 263, Part C, 125955. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.energy.2022.125955.

[14] Roman V. Klyuev, Irbek D. Morgoev, Angelika D. Morgoeva, Oksana A. Gavrina, Nikita V. Martyushev, Egor A. Efremenkov, and Qi Mengxu. 2022. Methods of forecasting electric energy consumption: A literature review. Energies 15, 23, 8919. MDPI, Basel, Switzerland. https://doi.org/10.3390/en15238919.

[15] Quota Alief Sias, Rahma Gantassi, and Yonghoon Choi. 2025. Multivariate bidirectional gate recurrent unit for improving accuracy of energy prediction. ICT Express 11, 1, 80–86. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.icte.2024.10.002.

[16] Md. Morshed Alam and Yeong Min Jang. 2023. Deep learning based optimal energy management framework for community energy storage system. ICT Express 9, 3, 333–340. Elsevier, Amsterdam, Netherlands. https://doi.org/10.1016/j.icte.2022.05.007.

[17] Ruming Chen, Jiashun Hu, Zhihao Song, Yixuan Wang, Xingzhao Zhou, Lin Zhao, and Bin Chen. 2023. The spatiotemporal distribution of NO_2 in China based on refined 2DCNN-LSTM model retrieval and factor interpretability analysis. Remote Sensing 15, 17, 4261. MDPI, Basel, Switzerland. https://doi.org/10.3390/rs15174261.

[18] Marcele O. K. Mendonça, Sergio L. Netto, Paulo S. R. Diniz, and Sergios Theodoridis. 2024. Machine learning: Review and trends. In Signal Processing and Machine Learning Theory, ed. Paulo S. R. Diniz, 869–959. Academic Press, Cambridge, MA. https://doi.org/10.1016/B978-0-32-391772-8.00019-3.

[19] Pramila P. Shinde and Seema Shah. 2018. A review of machine learning and deep learning applications. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, Pune, India, 1–6. https://doi.org/10.1109/ICCUBEA.2018.8697857.

[20] S. Ray. 2019. A quick review of machine learning algorithms. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). IEEE, Faridabad, India, 35–39. https://doi.org/10.1109/COMITCon.2019.8862451.