Hybrid CNN-LSTM and TimeGAN Models for Enhanced Energy Consumption Prediction in Smart Homes

Budoor Ahmad Allehyani
Department of Software Engineering, College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia
balehyani@uqu.edu.sa (corresponding author)

Ghader Reda Kurdi
Department of Data Science, College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia
grkurdi@uqu.edu.sa

Received: 10 August 2025 | Revised: 31 October 2025 | Accepted: 11 November 2025

 DOI: https://doi.org/10.48084/etasr.13179

ABSTRACT

Accurate prediction of energy consumption in smart homes is essential for optimizing energy usage, reducing operational costs, and enhancing occupant comfort. Traditional statistical and machine learning models often struggle to capture the complex spatiotemporal patterns inherent in energy usage data. To address this limitation, this study proposes a hybrid Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) model for forecasting energy consumption in smart homes, combining the spatial feature extraction strengths of Convolutional Neural Networks (CNNs) with the temporal learning capabilities of Long Short-Term Memory (LSTM) networks. The model demonstrated high predictive accuracy and generalization, as reflected by the evaluation metrics: Mean Absolute Error (MAE) = 0.0325, Mean Squared Error (MSE) = 0.0031, Root Mean Squared Error (RMSE) = 0.0557, and Mean Absolute Percentage Error (MAPE) = 0.2054. However, as the model was trained on synthetic data, further validation using real-world energy consumption datasets is necessary to assess its practical applicability and robustness.

Keywords-energy consumption; smart home; machine learning; deep learning; feature engineering; sustainability

I. INTRODUCTION

With the ongoing rise in energy demand, effective energy management is growing in importance. Smart technologies and data-driven predictive models play a key role in addressing this challenge, as they allow for optimized energy use, enhanced efficiency, and reduced waste and environmental impact. In addition, they support the integration of renewable energy sources into everyday systems. Together, these advancements form the foundation for a sustainable and resilient energy future.

In this context, smart homes are a key application area for intelligent energy management and forecasting technologies. Predictive models within these environments offer significant potential for promoting sustainability through learning from occupant behavior and environmental factors to accurately forecast energy needs and optimize usage, either by providing recommendations or by automatically adjusting appliance settings to reduce unnecessary consumption. Predictive capabilities also enhance the comfort and convenience of occupants by enabling smart systems to respond in a personalized and empathetic manner.

Several machine learning models have been developed to predict energy consumption in the context of smart homes and buildings. Examples include models for forecasting appliance-level consumption within a building [1], predicting energy consumption in smart commercial buildings [2, 3], predicting and optimizing residential cooling energy [4], and using demand predictions in controlling power systems [5]. For an overview of the broader landscape of machine learning techniques in this area, readers are referred to literature reviews reported in [6-8].

Prior studies on energy consumption forecasting in buildings have employed a variety of approaches, including Artificial Neural Networks (ANNs) [1, 2, 4], Support Vector Machines (SVMs) [2], and optimization procedures [3, 4]. Despite some studies achieving high accuracy, such as ANNs predicting appliance-level consumption with errors as low as 2% on a real-world online dataset [1] and synergistic ANN–VPPSO models attaining near-perfect regression values [4], many of these studies have suffered from methodological and reporting shortcomings. These included insufficient reporting of dataset characteristics [1], missing details about training and testing procedures [1], inaccessible or incomplete datasets [1], and omission of essential performance metrics (e.g., Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE)) [9]. These missing elements make it difficult to fully assess the robustness and generalizability of the reported results and prevent objective assessment and comparison with alternative models. Other studies demonstrated trade-offs between performance and feasibility. For example, while SVMs achieved strong accuracy, they required long training durations (e.g., up to 18 h for only four tenants), limiting practicality for real-time or resource-constrained settings.

These limitations are addressed in this study by leveraging deep learning models trained on openly available datasets [10, 11], documenting the methodology in detail, and employing standardized evaluation metrics to ensure both replicability and reliable assessment. The main objective of this study is to develop a predictive framework capable of detecting energy consumption patterns, allowing dynamic adjustments to optimize energy use. Specifically, this study explores the feasibility of leveraging a Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) model to predict energy consumption in smart homes. The model combines the spatial feature extraction capabilities Of Convolutional Neural Networks (CNNs) with the sequential learning strengths of Long Short-Term Memory (LSTM) networks, addressing a key gap in capturing the complex spatial and temporal dependencies inherent in smart home environments and allowing for more accurate and context-aware energy prediction.

The primary contributions of this study are: (1) developing a CNN-LSTM model to predict smart home energy consumption, and (2) evaluating the model using a realistic synthetic dataset. The outcomes of this research support the development of more intelligent, energy-efficient home environments that promote energy savings and improve convenience. Ultimately, the findings could serve as a foundation for future advancements in smart grid integration and sustainable urban living.

II. METHODOLOGY

This section highlights the structured methodology for forecasting energy usage in smart homes using advanced deep learning techniques. The proposed methodology involves the following steps: dataset preparation, feature extraction, data preprocessing, data augmentation, model selection, and evaluation. All these steps are critical for developing accurate forecasting models, establishing credibility, and recognizing important energy consumption patterns and trends. The proposed research methodology workflow is presented in Figure 1, which illustrates the entire process, including data collection, preprocessing, model training, evaluation, and validation. This workflow demonstrates the hybrid integration of Time-series Generative Adversarial Networks (TimeGAN) and CNN-LSTM models to better capture latent representations of the temporal and sequential response characteristics of energy consumption data and support a comprehensive predictive framework.

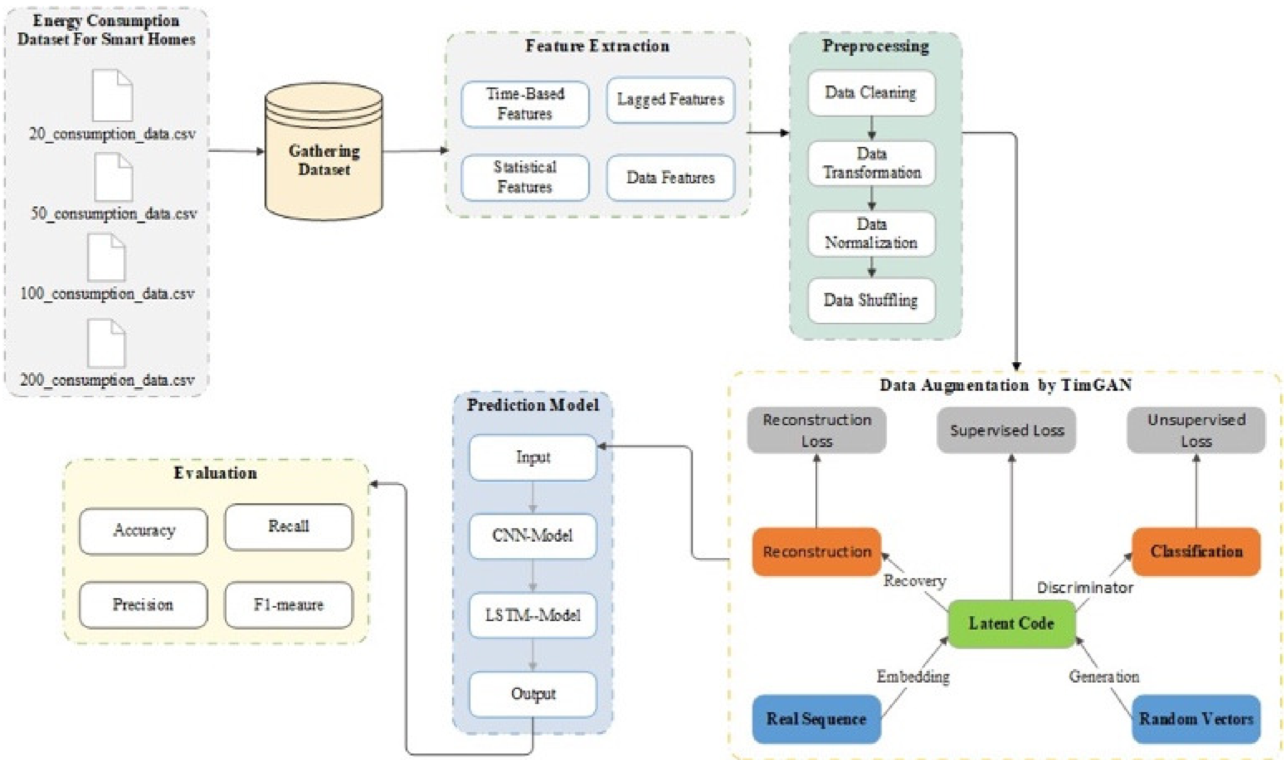


Fig. 1. Workflow of the proposed research methodology.

A. Dataset Preparation

The energy consumption dataset for smart homes [10, 11] provides a wide foundation for research geared towards energy management within smart cities. It addresses a critical absence of detailed home energy data, which are necessary for model simulations of energy consumption and production within smart city ecosystems. For researchers focusing on energy management, predictive modeling, and peer-to-peer energy transactions within the contexts of smart homes and urban environments, this dataset is highly valuable [11].

The dataset is divided into five distinct subsets, each representing different scales of smart home implementations (20, 50, 100, and 200 homes) over a full year (365 days). Each subset encompasses data on both energy consumption and production. The data were synthesized to approximate real-world variables, using sound statistical techniques to enhance their applicability and accuracy. The dataset is illustrated in Figure 2, showing the dataset structure, with each row representing a smart home, each column representing a day, and the value at their intersection indicating the overall energy consumption of a specific home on that day, measured in kilowatt-hours (kWh). Figure 3 shows a sample of the dataset after feature extraction and preprocessing. Multiple files of consumption data (20_consumption_data.csv, 50-consumption-data.csv, 100-consumption-data.csv, and 200-consumption-data.csv) were combined into one CSV file.

Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18	Day 19	Day 20
7079.168	4305.585	4977.391	4142.39	5930.726	7787.26	1799.035	5728.693	4294.974	2400.365	3423.686	2851.065	3490.599	7536.872	5085.89	7235.286	4111.294	4331.504	3014.745	3518.932
3650.472	4277.567	3632.568	4529.671	7174.067	4104.338	2953.531	5990.336	7487.407	4236.981	3279.764	3389.607	3853.771	4211.893	2301.486	9842.048	5151.486	2715.526	3338.127	3520.31
6220.812	3355.844	6891.934	2140.877	4044.4	1789.717	2674.29	2962.162	2702.915	5352.645	2655.79	4260.242	4803.679	5924.699	2076.234	2001.217	5362.05	3029.575	4947.183	830.1825
4990.971	4148.795	4053.93	3580.197	1436.27	4966.08	1772.457	2983.336	4384.423	5082.745	1201.689	4863.104	3732.092	3874.014	1380.542	3146.181	3096.687	4317.804	2669.326	4885.36
8825.135	4892.458	2330.242	3962.126	2463.955	4030.248	3532.243	5569.343	3465.704	2041.514	3737.351	1281.271	6097.559	5371.741	3785.656	4146.087	855.8791	5506.007	5112.623	5630.094
3141.386	2588.917	2368.553	3061.086	1716.356	4410.95	4039.277	1210.19	3397.159	4134.1	6588.146	3540.827	6585.867	4001.435	3271.275	4631.386	3721.155	3800.227	3286.172	5902.938
1877.095	2832.133	2408.536	8066.183	2155.422	5936.067	2833.649	3909.244	7203.453	2962.347	3401.543	4853.854	2445.107	4203.099	4734.945	1589.971	5548.083	3194.795	1955.574	5589.147
7506.576	4828.878	2371.482	4122.455	6936.92	4239.791	1765.24	3681.341	5890.038	5828.213	3904.642	3359.153	5824.429	6415.508	7163.292	2414.245	5289.868	3290.617	2072.023	4823.937
5403.649	2215.431	2436.421	1538.802	4628.602	4510.978	3165.277	4120.116	1617.842	3769.793	5978.488	5743.823	1291.55	1720.392	3901.954	4054.706	829.9548	3627.772	5641.567	3062.257
3273.379	2010.019	7017.764	3939.384	3791.158	5205.418	4341.274	3833.107	5849.595	3929.207	3693.074	4099.58	1112.597	3366.757	3910.811	4971.86	4016.727	2882.453	3515.329	3200.464
3790.031	4157.734	4584.011	8071.678	1325.474	3857.233	2911.647	3080.225	7136.35	3968.413	1664.952	2716.45	6216.046	5706.767	1963.641	5439.368	4197.396	2966.986	3519.619	5905.4
2402.778	5826.813	1093.282	3954.531	4868.521	5186.64	4331.731	5689.586	7632.509	7013.573	4933.078	2197.338	5152.631	7040.188	4723.178	4942.109	1582.263	7015.993	4611.97	2813.097
3307.874	2535.365	902.4161	3392.333	4777.86	2162.027	4240.733	5871.039	2904.797	3787.395	4649.412	6543.52	4120.358	2155.62	2421.969	5193.25	8627.236	2410.341	2525.161	8629.754
4566.897	8377.169	4214.129	3679.363	3093.628	5046.617	3258.293	8103.723	2189.26	3550.865	2058.283	4491.892	6563.193	2269.229	2468.731	2438.015	6617.529	1769.723	3689.468	3124.444
859.3351	2085.54	2374.002	1345.434	2934.198	2490.776	2837.038	3951.264	3954.744	2676.963	2979.414	3765.015	5870.152	5970.9	4292.78	3913.85	6178.505	2735.854	2521.285	1396.123
2653.991	2653.808	5407.525	4994.898	3948.936	4090.95	3636.724	6678.61	4270.494	3097.763	5137.694	1685.662	4218.447	4255.292	3426.745	4790.424	3017.627	3861.475	4475.288	5661.076
1677.506	5101.584	5185.345	6193.965	4137.799	4787.756	5722.574	6507.205	3886.107	3365.573	1895.725	1265.498	2191.91	3245.832	9073.831	7030.407	5375.294	4781.828	3778.885	6641.81
2119.743	4131.974	3742.397	5589.758	1979.713	4623.671	5050.737	987.5788	1078.05	2787.619	4431.245	5878.907	1025.445	1765.347	5634.998	4087.508	2533.079	6244.583	6630.876	1541.036
1989.667	2488.742	1343.533	6397.432	4182.257	1341.183	1649.249	8101.682	9986.015	3338.145	8153.832	4209.007	4021.198	2397.253	6799.163	7810.09	5856.291	3650.896	2220.824	4525.281
4858.519	6141.185	5128.243	3407.59	4126.897	8094.082	597.7368	4615.466	5394.193	3519.611	2451.982	4414.734	7498.88	7195.478	4058.976	6606.786	2688.326	4225.343	4053.791	4776.23

Fig. 2. A sample of the dataset showing smart home energy consumption data.

Day	Measurement	Energy Consumption	Month	Week	Weekday	Quarter	Weekend	Previous Day Consumption	Two Days Ago Consumption	7-Day Rolling Mean	Consumption Weekday Interaction
13	31	0.236347635	5	19	5	2	1	0.154991015	0.200863039	0.402324395	1.181738173
31	298	0.167865645	12	52	6	4	1	0.334643797	0.283156054	0.540103411	1.007193869
31	212	0.388416862	5	22	2	2	0	0.239703645	0.376978927	0.458260643	0.776833723
26	358	0.164785828	6	26	0	2	0	0.078273526	0.298249741	0.152338595	0
27	211	0.41045091	9	39	2	3	0	0.404507263	0.219467785	0.414000276	0.82090182
7	21	0.20404855	9	36	3	3	0	0.193843578	0.298108867	0.341158043	0.61214565
12	243	0.187666177	7	28	2	3	0	0.093022241	0.188026352	0.25610254	0.375332355
26	353	0.197794733	8	34	5	3	1	0.224341932	0.421758374	0.289873955	0.988973664
21	111	0.303694825	10	42	5	4	1	0.029666674	0.177835964	0.309700797	1.518474124
19	30	0.091200758	3	11	6	1	1	0.163672688	0.441627063	0.356041131	0.547204548
14	311	0.060052913	1	2	5	1	1	0.202868879	0.094556058	0.256007928	0.300264567
22	282	0.412162253	8	34	1	3	0	0.069553424	0.083790214	0.306547484	0.412162253
21	307	0.515354124	9	38	3	3	0	0.361828559	0.145348191	0.509111403	1.546062372
3	153	0.15913429	4	14	0	2	0	0.194185926	0.240943187	0.31031081	0
14	293	0.211220736	1	2	5	1	1	0.236545224	0.340854374	0.372255852	1.05610368
23	216	0.1269765	5	21	1	2	0	0.341504911	0.153830276	0.307506944	0.1269765
19	356	0.344822111	12	51	1	4	0	0.330939141	0.328772943	0.502397516	0.344822111
20	103	0.271174493	5	20	5	2	1	0.18759837	0.353671651	0.445255506	1.355872466
25	125	0.129066499	6	25	6	2	1	0.404509604	0.140515374	0.419957057	0.774398996
22	60	0.188939863	5	21	0	2	0	0.263514625	0.172979499	0.264822488	0
9	22	0.302274928	6	23	4	2	0	0.205406715	0.446123978	0.546337329	1.209099711

Fig. 3. Dataset sample after feature extraction and preprocessing.

B. Feature Extraction

Feature engineering is the process of creating new features or transforming existing features to better capture the underlying structure of the data. Several features were extracted from the data for this study, and they significantly improved the predictive quality of the model by leveraging time-based, lagged, statistical, categorical, and interaction features. The selected features encompass a broad spectrum of attributes, designed to effectively capture both temporal patterns and energy consumption behaviors, as shown in Table I. Time-based features, including specific date, month, and day, provide critical contextual information related to temporal dynamics. Lagged features, such as energy consumption on the previous day and two days prior, facilitate the representation of sequential consumption trends, thereby enhancing the model's ability to capture temporal dependencies.

TABLE I. OVERVIEW OF FEATURES EXTRACTED FROM THE DATASET

Id	Feature name	Description	Feature type
1	Month	Month corresponding to each observation (e.g., January, February)	Time-based feature
2	Day	Specific day of the month	
3	Week	ISO week number of the year	
4	Quarter	Quarter of the year	
5	Weekday	Day of the week (e.g., Sunday, Monday)	
6	Weekend	Binary feature indicating whether a day is part of the weekend	
7	Previous day consumption	Energy consumption value from the previous day	Lagged feature
8	Two days ago consumption	Energy consumption value from two days prior	
9	7-day rolling mean	Rolling mean of energy consumption over the past 7 days	Statistical feature
10	Consumption–weekday interaction	Interaction term between energy consumption and the day of the week	
11	Measurement	Encoded categorical ID referring to the observation source or sensor	Identifier feature
12	Energy consumption	Recorded energy consumption per day in kWh	Target variable

The 7-day rolling mean is a statistical feature that provides insights into the central tendency and variability of energy use over time. This feature helps the model detect patterns in the consistency and variability of consumption. Categorical features, differentiated using label encoding, treat weekdays and weekends separately. This addresses differences in consumption patterns that may occur throughout the week.

Moreover, an interaction feature, "consumption–weekday interaction," was created to explicitly model the relationship between energy use and the day of the week. This helps the model understand the impact of certain days, such as weekends or holidays, on the variations in energy use and consumer behavior, thus enhancing the model's predictive performance.

C. Data Preprocessing

Data preprocessing is a fundamental stage of a machine learning pipeline, as it entails cleaning, validating, and preparing a dataset for analysis. As part of this study, the preprocessing functions completed include the treatment of missing data, encoding categorical data, normalizing the range of numerical data for comparability, outlier removal, dataset shuffling, and exporting the completed dataset for analysis.

1) Data Cleaning

This step removed the rows where values were missing in order not to compromise the integrity of the dataset and analytical bias. This allowed for future analyses without the risk of running them on incomplete data and greatly increased the reliability of the results. This method can decrease dataset size though, which might result in losing informative characteristics. Other forms of analysis using data imputation can deal with missing data without losing full records of individuals with missing values [12, 13].

2) Data Transformation

Date-time conversion for date columns enabled more advanced operations based on time, which facilitated the extraction of features as the day of the week or rolling means over time. It also allowed for additional time-based features, which were essential in modeling temporal patterns in energy consumption [13].

3) Categorical Encoding

This process transforms categorical variables into a numerical format so they can be fed into machine learning models. For ordinal categories like days of the week or months in a year, label encoding was used to assign a unique integer value to each category. This allows using the ordinal relationships between the categories while making the data appropriate for modeling [14]. The categorical encoding used in this study is shown in Table II.

TABLE II. ENCODING OF CATEGORICAL VARIABLES

Feature	Value	Label
Month	January	1
	February	2
	March	3
	April	4
	May	5
	June	6
	July	7
	August	8
	September	9
	October	10
	November	11
	December	12
Weekday	Monday	0
	Tuesday	1
	Wednesday	2
	Thursday	3
	Friday	4
	Saturday	5
	Sunday	6
Weekend	Yes	1
	No	0

4) Data Normalization

This step was applied to numerical features (e.g., energy consumption and rolling averages) to scale all values within 0 and 1. This step is important to ensure that all features are weighted equally in the model and no features dominate the learning process due to the larger ranges. Min-max normalization was applied for this purpose, as it is particularly suited for features with a known range. The min-max normalization formula is [15]:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

where x is the original value, x' is the normalized value, $\min(x)$ is the minimum value of the feature, and $\max(x)$ is the maximum value of the feature.

5) Data Shuffling

The order of the dataset was randomized to remove any potential ordering bias. This protects against the risk of a model learning any unintended patterns related to the order of data collection. This is a critical step, especially before dividing the data into training and testing sets [16].

D. Data Augmentation

Data augmentation is a critical step to enhance the diversity and robustness of the energy consumption dataset. Using the TimeGAN model, synthetic time-series data are generated, which closely imitates the original dataset's underlying patterns and characteristics. TimeGAN is a novel generative adversarial network designed specifically for sequential data. This enables the prediction models to generalize better throughout diverse and previously never-encountered data scenarios [17]. At a high level, the TimeGAN model works by first embedding the time-series data into a latent space, then generating synthetic sequences, and aligning those sequences with the original data distribution. For the purpose of training, this alignment is refined in an iterative manner such that the synthetic data retain high fidelity as well as diversity. TimeGAN is built upon four main components [18]:

1. Embedder (E): Converts raw time-series data into latent representation.
2. Recovery (R): Reconstructs time-series data from latent representations.
3. Generator (G): Generates synthetic latent representations.
4. Discriminator (D): Determines whether the sequence is real or synthetic.

On the other hand, the TimeGAN model is built to jointly maximize several objectives, enabling the generated synthetic data to have high fidelity, diversity, and alignment with the underlying distribution of the original dataset. These objectives are expressed through distinct loss functions, which collectively guide the model's learning process [19, 20]:

- Adversarial loss: Ensures the synthetic data are indistinguishable from real data by minimizing the classification ability of the discriminator D:

$$L_A = E\left[\log\left(D(X_{real})\right)\right] + E[\log(1 - D(G(Z)))] \quad (2)$$

where X_{real} represents real sequences and Z is random noise.

- Reconstruction loss: Promotes the accurate reconstruction of original data from its latent representations. This loss ensures the embedder E and recovery R networks preserve critical features of the input sequences:

$$LR = \left\|X_{real} - R\left(E(X_{real})\right)\right\|^2 \quad (3)$$

- Supervised loss: Facilitates the alignment of latent space representations generated by G with those produced by E, enabling the generator to produce realistic latent representations:

$$L_S = \|E(X_{real}) - G(Z)\|^2 \quad (4)$$

The total loss function combines these components, weighted by hyperparameters λ_A, λ_R, and λ_S, which balance the contributions of each objective:

$$L_{total} = \lambda_A L_A + \lambda_R L_R + \lambda_S L_S \quad (5)$$

This multi-objective optimization strategy ensures that the generated data capture the temporal dependencies, variability, and overall structure of the original sequences, making TimeGAN particularly effective for tasks requiring realistic time-series data synthesis. Furthermore, the hyperparameters of the TimeGAN model used in this study are presented in Table III. These parameters were selected through iterative experimentation to achieve a balance between model complexity and computational efficiency.

TABLE III. HYPERPARAMETERS OF TIMEGAN

Hyperparameter	Description	Value / range
Embedding dimension	The latent space dimension for embedding time-series data	10
Generator LSTM units	Number of LSTM cells in the generator network	100
Supervisor LSTM units	Number of LSTM units in the supervisor network	100
Discriminator LSTM units	Number of LSTM units in the discriminator network	100
Sequence length	Maximum length of input sub-series	30
Batch size	Number of sequences processed in the training phase per batch	30
Epochs	Total number of training iterations	8,000
Learning rate	Optimizer learning rate	0.0001
Noise dimension	Dimension of the input noise vector for the generator	20

This particular method of data augmentation using the TimeGAN model facilitates the generation of synthetic data, resulting in more effective predictions compared to using the original dataset.

E. Prediction Model

A hybrid CNN-LSTM model is applied to predict energy consumption in smart homes. This architecture effectively captures spatial and temporal correlations in the dataset, thus ultimately improving the model predictive performance, robustness, and generalizability [21].

As the first part of the architecture, the CNN portion aims to extract high-level spatial features from the input time-series data. The CNN extracts localized patterns and dependencies, such as the seasonal trends in energy consumption, through convolutional layers. These features are vital for gaining insights into energy consumption patterns and serve as the basis for temporal analysis in the next step [22]. Additionally, pooling layers are added to decrease the feature size by removing redundant information [23].

Then, the output of the CNN is passed to the LSTM to model the spatial features, because energy consumption data are sequentially dependent and long-term temporally related. LSTMs are well suited for time-series data, as they are able to remember information over long sequences, which helps with learning daily, weekly and seasonal shape patterns [24]. This capability allows the model to forecast future energy consumption well in advance, using trends from the past, which is critical in designing reliable energy management systems for smart homes.

The architecture of the proposed model is hybrid, where the CNN and LSTM networks are combined in a sequence manner to complete the prediction task. The key components include:

1. Input layer: Accepts preprocessed time-series data in three-dimensional tensors (batch size, sequence length, number of features).
2. CNN layers: Multiple convolutional layers with different filter sizes and kernel dimensions are applied to extract spatial features. Pooling layers are added after every convolutional operation to make features denser and reduce dimensionality.
3. LSTM layers: The spatial features are processed through stacked LSTM layers, which model temporal dependencies and output a sequence representation used for prediction.
4. Dense layers: Fully connected layers are utilized to combine the learned spatial and temporal features, culminating in a single output value representing the predicted energy consumption.

The model was trained on the prepared dataset using an 80:20 split for training and validation, respectively. The Adam optimizer, selected for its adaptive learning capabilities, was used to minimize the Mean Squared Error (MSE) loss function [25]. To reduce overfitting, dropout regularization was applied between layers [26], and early stopping was implemented to optimize the training process and prevent unnecessary iterations. Table IV presents the hyperparameters of the CNN-LSTM model in detail.

The hybrid CNN-LSTM model was selected because it combines the sequential learning power of LSTMs with the spatial feature extraction capabilities of CNNs. The development of dependable and useful energy management systems for smart homes depends on the model's ability to capture the complex relationships seen in energy consumption data, which is ensured by this complementary method. The model strikes a balance between generalizability and precision by utilizing this hybrid design, which makes it especially appropriate for the complex and dynamic nature of energy consumption forecasting [27].

TABLE IV. HYPERPARAMETERS OF THE HYBRID CNN-LSTM MODEL

Hyperparameter	Description	Value
Sequence length	Number of time steps used in each input sequence	5
Filters	Number of filters in the Conv1D layer	64
Kernel size	Kernel size of convolution	2
Activation function	Activation functions in Conv1D and LSTM layers	ReLU
Pool size	Output shape of the MaxPooling1D layer	2
LSTM units	Number of units in the LSTM layer	50
Dropout rate	Dropout rate applied after the LSTM layer	0.2
Dense layer output	Number of units for prediction	1
Optimizer	Optimizer used for training the model	Adam
Loss function	Loss function used during training	MSE
Epochs	Number of training iterations	2
Batch size	Number of samples per batch	32
Validation split	Fraction of training data used for validation during training	0.2
Random state	Seed for train-test split to make it reproducible	42

F. Model Evaluation

The predictive accuracy and generalization capabilities of the hybrid CNN-LSTM model were evaluated to forecast continuous energy consumption estimates. We conducted a comprehensive assessment of the model's capabilities and dependability using a wide range of performance metrics. The indicators have been intentionally selected to cover a range of predictive performance aspects, ensuring the validity and interpretability of the results [28]. The following lists the evaluation metrics, together with their mathematical formulation and significance:

1. MAE: It measures the average of the absolute errors between the predicted and observed values. As a scale-dependent metric, MAE is easily interpretable in terms of the model's predictive performance, with a lower value implying better accuracy. The MAE is defined as [29]:

   $$\text{MAE} = \frac{1}{n}\sum_{1=1}^{n}|y_i - \hat{y_i}| \quad (6)$$

   where y_i is the actual or true value of the i-th observation, $\hat{y_i}$ is the predicted value of the i-th observation, and n is the total number of observations.

2. MSE: It measures the average of the squared differences between the actual values against the predicted values, providing insight into the variance of prediction errors. Since larger errors are squared, MSE is very sensitive to outliers. Mathematically, MSE is defined as [30]:

   $$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})2 \quad (7)$$

3. RMSE: A widely popular metric in regression analysis, calculated as the square root of MSE. Additionally, since RMSE takes the square root of the average error, it converts the error back to the original units of the target variable, making it easier to interpret and associate with the predicted values. Mathematically, the RMSE is formally defined as follows [31]:

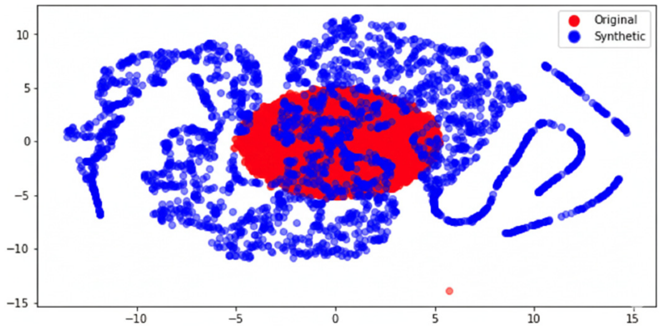
$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)} \tag{8}$$

4. Mean Absolute Percentage Error (MAPE): It evaluates the predictive accuracy of a model by quantifying the error as a percentage of the actual values. It is a scale-independent metric, offering intuitive interpretability for assessing model performance. A lower MAPE value reflects higher prediction accuracy. However, MAPE is sensitive to instances where actual values are small or zero, as this can disproportionately inflate the error, potentially skewing the evaluation. The MAPE is mathematically expressed as [32]:

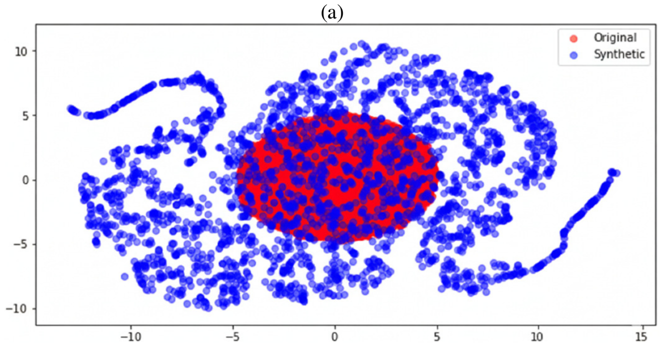
$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{9}$$

III. RESULTS AND DISCUSSION

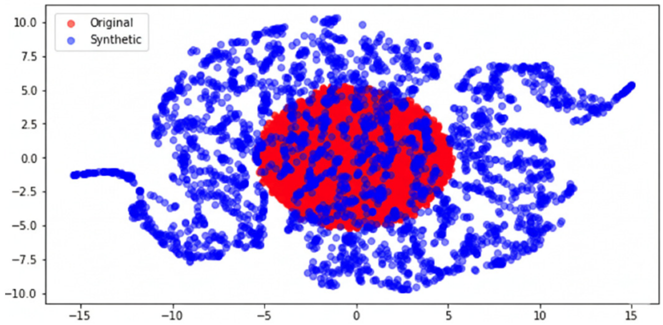
The synthetic data generated by the TimeGAN model were evaluated using t-Distributed Stochastic Neighbor Embedding (t-SNE) to visualize its alignment with the original dataset. Figure 4 illustrates the t-SNE projections for synthetic and original data at different stages of model training (epochs 100, 1,000, 4,000, and 8,000, respectively). In the visualizations, red points represent the original data, whereas blue points indicate the synthetic data generated by TimeGAN.



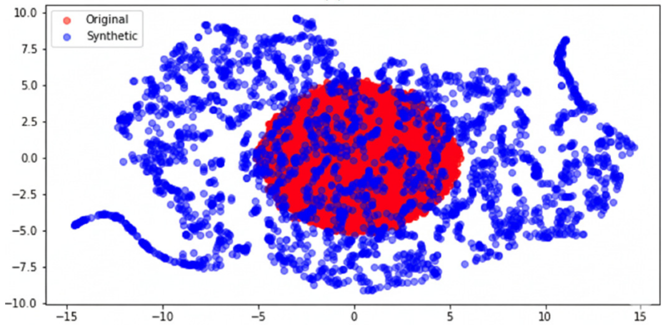
(a)



(b)



(c)



(d)

Fig. 4. t-SNE visualizations showing the alignment of synthetic (blue) and original (red) data during TimeGAN training: (a) epoch 100, (b) epoch 1,000, (c) epoch 4,000, (d) epoch 8,000.

At the initial training stage (epoch 100), the synthetic data points were widely scattered, forming distinct clusters around the central distribution of the original data. This suggests that, while the TimeGAN model partially captured aspects of the latent space, it had not yet achieved sufficient alignment with the true data distribution. Despite the diversity observed in the synthetic data, there was minimal overlap with the original dataset, reflecting the model's early learning phase and limited understanding of the temporal and sequential patterns.

At epoch 1,000, the t-SNE visualization indicates significant progress in the model's learning. The synthetic data distribution demonstrated an increased overlap with the original data, with synthetic points converging toward the central region. This improvement illustrates the model's growing ability to learn the underlying structure and features of the original data, balancing fidelity with diversity. While some synthetic points remained in the periphery, the overall alignment showed marked improvement compared to earlier stages.

At epoch 4,000, the TimeGAN model achieved the highest degree of alignment between the synthetic and original data. The t-SNE projection reveals a dense overlap, where most synthetic points closely matched the original data distribution. This indicates that the model effectively captured the temporal dynamics and sequential relationships present in the original dataset. While a few outliers were still observed, they were negligible, and the synthetic data displayed a high degree of accuracy and diversity, making this stage the most optimal for generating synthetic data.

At epoch 8,000, slight divergence emerged in the synthetic data, with some points forming noticeable outliers away from the central cluster. These outliers suggest that the model encountered difficulties in replicating certain rare or complex features of the original data. Nevertheless, the majority of synthetic data points maintained an alignment with the core cluster of the original data. This indicates that while the model was still effective overall, it faced challenges in fully generalizing the nuances of the original dataset during extended training.

To analyze the performance quantitatively and qualitatively, Table V summarizes the evaluation of the synthetic data at each training epoch. The metrics include the distribution overlap, which indicates the percentage of synthetic data points aligned with the original dataset; the pattern consistency, which reflects the extent to which synthetic data capture temporal patterns of the original data; and the outlier ratio, which represents the percentage of synthetic data points deviating significantly from the main cluster.

TABLE V. PERFORMANCE METRICS FOR SYNTHETIC DATA

Epoch	Distribution overlap (%)	Pattern consistency	Outlier ratio (%)
100	45	Low	28
1,000	72	Moderate	15
4,000	89	High	7
8,000	83	Moderate-high	12

In summary, the t-SNE visualizations in Figure 4 and the summarized metrics in Table V demonstrate the progressive improvement in the quality and fidelity of the synthetic data with increased training epochs. Epoch 4,000 emerged as the most optimal stage, achieving the best balance between alignment, pattern consistency, and minimal outliers, making it the ideal choice for generating high-quality synthetic data.

Moreover, the performance of the hybrid CNN-LSTM model in predicting energy consumption was systematically evaluated across different training epochs (10, 50, 100, 500, and 1,000). To assess the model's predictive accuracy and robustness, multiple evaluation metrics were utilized, including MAE, MSE, RMSE, and MAPE. The detailed results for each epoch are presented in Table IV.

TABLE VI. PERFORMANCE OF THE CNN-LSTM MODEL.

Epoch	MAE	MSE	RMSE	MAPE
10	0.0938	0.0136	0.1167	0.1561
50	0.0721	0.0102	0.1010	0.4523
100	0.0618	0.0078	0.0883	0.3927
500	0.0325	0.0031	0.0557	0.2054
1,000	0.0981	0.0141	0.1221	0.572

The results in Table VI demonstrate a clear trend of improvement in predictive performance as the number of training epochs increased. At the initial stages of training (epoch 10), the model exhibited relatively high errors, with an MAE of 0.0938 and RMSE of 0.1167. These values indicate limited learning of the temporal features inherent in the data. By epoch 50, the model showed significant improvement, with MAE and RMSE reduced to 0.0721 and 0.1010, respectively. However, the MAPE value (0.4523) suggests that the model still struggled to capture relative deviations in the data effectively.

At epoch 500, the model achieved its best performance, with MAE, MSE, RMSE, and MAPE values of 0.0325, 0.0031, 0.0557, and 0.2054, respectively. These metrics indicate that the CNN-LSTM model successfully learned the sequential patterns in the dataset, achieving a balance between accuracy and generalization. However, extending the training to epoch 1,000 resulted in a decline in performance, with a noticeable increase in errors across all metrics. This degradation suggests potential overfitting, as the model began to memorize the training data instead of generalizing well to unseen data.

To contextualize the hybrid CNN-LSTM model's performance, a comparison with standalone CNN and LSTM models at epoch 500 was conducted. Table VII summarizes this comparison, highlighting the superior performance of the CNN-LSTM model.

TABLE VII. PERFORMANCE COMPARISON OF CNN, LSTM, AND CNN-LSTM MODELS AT EPOCH 500

Model	MAE	MSE	RMSE	MAPE
CNN	0.0453	0.0045	0.0671	0.3256
LSTM	0.0382	0.0038	0.0616	0.2874
CNN-LSTM	0.0325	0.0031	0.0557	0.2054

The hybrid CNN-LSTM model outperformed both CNN and LSTM models across all evaluation metrics, demonstrating its superior ability to capture temporal dependencies and spatial features simultaneously. Specifically, the CNN-LSTM achieved the lowest MAE, MSE, RMSE, and MAPE values, underscoring its effectiveness in energy consumption prediction tasks.

To further evaluate the model's performance, Figure 5 provides a comparison of the actual and predicted energy consumption values for the first 100 samples, as a function of the number of training iterations:

- After epoch 10, the actual and predicted values start to deviate significantly from each other, implying that the model does not capture the data well.
- At epoch 100, a slight improvement in alignment of predicted and actual values is observed, although some disparity remains.
- By epoch 500, the predicted values align with the actual values, showing the model's capability of capturing underlying temporal patterns in a highly accurate manner.
- At epoch 1,000, there is a noticeable drop in alignment between actual and predicted values, with the predicted line appearing relatively flat and less responsive to the fluctuations in actual energy consumption. This indicates the onset of overfitting, consistent with the observed quantitative results. The model at this stage fails to generalize well, leading to a decline in prediction accuracy compared to earlier epochs, such as 500.

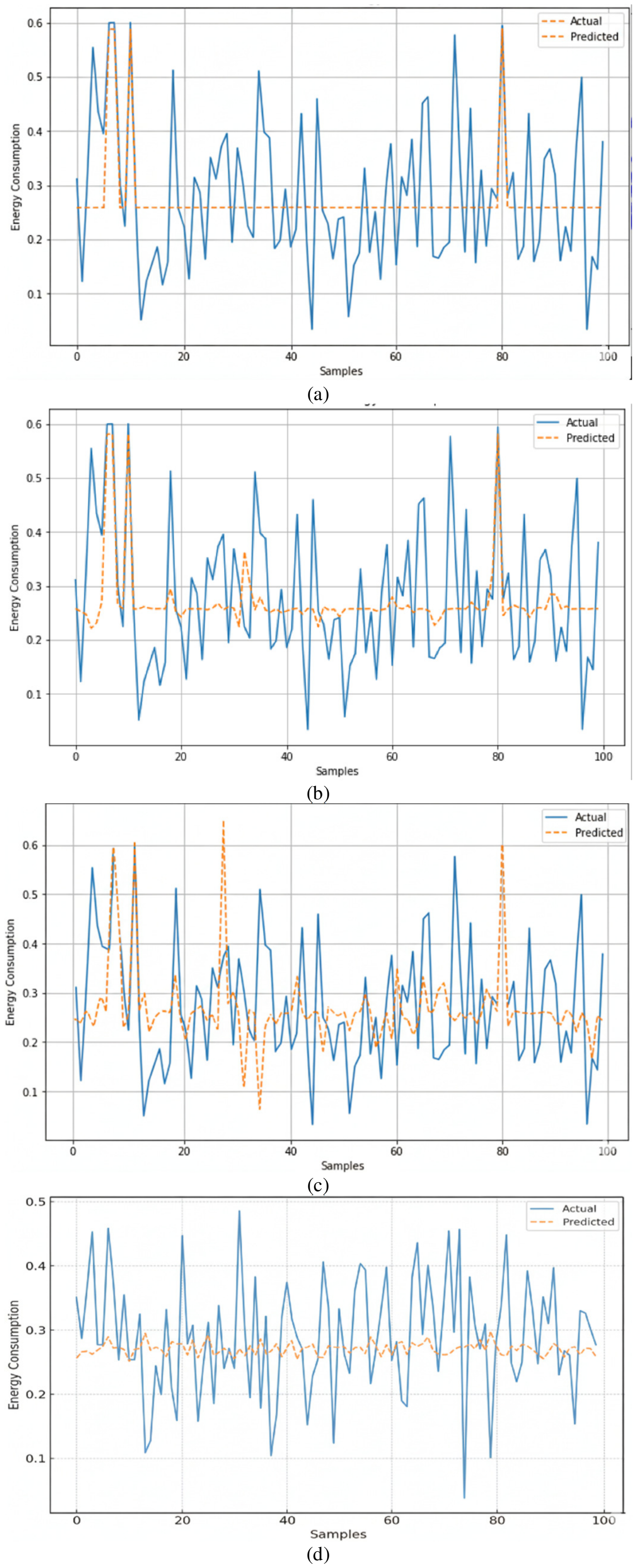


Fig. 5. Actual vs. predicted energy consumption values for the first 100 samples: (a) epoch 10, (b) epoch 100, (c) epoch 500, (d) epoch 1,000.

The evaluation results demonstrate the efficiency of the hybrid CNN-LSTM architecture in learning the temporal and sequential patterns of the dataset. The hybrid approach outperformed individual CNN and LSTM architectures trained separately by capitalizing on the benefits of both architectures. This reveals the power of the well-chosen training parameters; even with only 500 epochs, the result was very good. The drop in performance after epoch 1,000 represents overfitting, which highlights the necessity of applying early-stopping and regularization techniques in future studies.

IV. CONCLUSION AND FUTURE WORK

This study introduced a combined Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network for predicting energy consumption and forecasting energy consumption patterns in smart homes. The results demonstrate that the hybrid model achieves an outstanding balance between predictive accuracy and generalization. Optimal performance was observed after training for 500 epochs.

However, the model was trained on artificial synthetic datasets, which may limit its applicability and accuracy in real-world applications. Therefore, future work should focus on validating the model on real-world datasets of varying and diverse energy consumption to better assess the model's accuracy and robustness. Additionally, the model features mainly consist of energy consumption data. Incorporating external variables, such as weather conditions, socioeconomics factors, and local events, could provide a more comprehensive understanding of the factors influencing energy usage. Integrating these diverse data sources may improve the model's predictions and clarify how external factors affect energy consumption.

FUNDING STATEMENT

This research work was funded by Umm Al-Qura University, Saudi Arabia, under grant number: 25UQU4331451GSSR03.

ACKNOWLEDGMENT

The authors extend their appreciation to Umm Al-Qura University, Saudi Arabia, for funding this research work through grant number: 25UQU4331451GSSR03.

REFERENCES

[1] S. Bourhnane, M. R. Abid, R. Lghoul, K. Zine-Dine, N. Elkamoun, and D. Benhaddou, "Machine learning for energy consumption prediction and scheduling in smart buildings," *SN Applied Sciences*, vol. 2, no. 2, Jan. 2020, Art. no. 297, https://doi.org/10.1007/s42452-020-2024-9.

[2] M. K. M. Shapi, N. A. Ramli, and L. J. Awalin, "Energy consumption prediction by using machine learning for smart building: Case study in Malaysia," *Developments in the Built Environment*, vol. 5, Mar. 2021, Art. no. 100037, https://doi.org/10.1016/j.dibe.2020.100037.

[3] R. Dulyala, S. Audomsi, and W. Sa-Ngiamvibool, "Artificial Neural Networks Based on Optimization Technique for Short-Term Electricity Demand Forecasting: Uttaradit Rajabhat University Data Analysis," *International Journal of Engineering Trends and Technology*, vol. 73, no. 6, pp. 183–190, June 2025, https://doi.org/10.14445/22315381/IJETT-V73I6P116.

[4] N. E. Assri, M. A. Jallal, S. E. E. Aoud, S. Chabaa, and A. Zeroual, "Synergistic Neural Network and Velocity Pausing Particle Swarm

Optimization for Enhanced Residential Building Energy Efficiency: A Case Study in Kuwait," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 17507–17516, Oct. 2024, https://doi.org/10.48084/etasr.8278.

[5] S. Audomsi, S. Wattana, N. Uthathip, and W. Sa-Ngiamvibool, "Development and Design of an Optimal Fuzzy Logic Two Degrees of Freedom-Proportional Integral Derivative Controller for a Two-Area Power System Using the Bee Algorithm," *Energies*, vol. 18, no. 4, Feb. 2025, Art. no. 915, https://doi.org/10.3390/en18040915.

[6] C. Lu, S. Li, and Z. Lu, "Building energy prediction using artificial neural networks: A literature survey," *Energy and Buildings*, vol. 262, May 2022, Art. no. 111718, https://doi.org/10.1016/j.enbuild.2021.111718.

[7] U. ur Rehman, P. Faria, L. Gomes, and Z. Vale, "Future of Energy Management Models in Smart Homes: A Systematic Literature Review of Research Trends, Gaps, and Future Directions," *Process Integration and Optimization for Sustainability*, vol. 9, no. 4, pp. 1169–1198, Sept. 2025, https://doi.org/10.1007/s41660-025-00506-x.

[8] R. V. Klyuev *et al.*, "Methods of Forecasting Electric Energy Consumption: A Literature Review," *Energies*, vol. 15, no. 23, Dec. 2022, Art. no. 8919, https://doi.org/10.3390/en15238919.

[9] E. J. Al-Reshidi, R. A. Ramadan, B. W. Aboshosha, M. Salem, and A. M. Alayba, "Real-Time Home Energy Management with IoT and Blockchain: Balancing Consumption and Peer-to-Peer Trading," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14014–14021, June 2024, https://doi.org/10.48084/etasr.7188.

[10] R. Ramadan, "Energy Consumption Dataset For Smart Homes." Mendeley Data, Dec. 18, 2023, https://doi.org/10.17632/5h86vnxw68.1.

[11] R. A. Ramadan, "Internet of things dataset for home renewable energy management," *Data in Brief*, vol. 53, Apr. 2024, Art. no. 110166, https://doi.org/10.1016/j.dib.2024.110166.

[12] J. Kaiser, "Dealing with Missing Values in Data," *Journal of Systems Integration*, vol. 5, no. 1, pp. 42–51, Jan. 2014, https://doi.org/10.20470/jsi.v5i1.178.

[13] S. Gorard, "Handling missing data in numeric analyses," *International Journal of Social Research Methodology*, vol. 23, no. 6, pp. 651–660, Nov. 2020, https://doi.org/10.1080/13645579.2020.1729974.

[14] K. Potdar, T. S. Pardawala, and C. D. Pai, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers," *International Journal of Computer Applications*, vol. 175, no. 4, pp. 7–9, Oct. 2017, https://doi.org/10.5120/ijca2017915495.

[15] H. Henderi, T. Wahyuningsih, and E. Rahwanto, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *International Journal of Informatics and Information Systems*, vol. 4, no. 1, pp. 13–20, Mar. 2021, https://doi.org/10.47738/ijiis.v4i1.73.

[16] W. Yu, Y. Wang, X. Que, and C. Xu, "Virtual Shuffling for Efficient Data Movement in MapReduce," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 556–568, Feb. 2015, https://doi.org/10.1109/TC.2013.216.

[17] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series Generative Adversarial Networks," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019, pp. 5508–5518.

[18] M. EskandariNasab, S. M. Hamdi, and S. F. Boubrahimi, "ChronoGAN: Supervised and Embedded Generative Adversarial Networks for Time Series Generation," in *2024 International Conference on Machine Learning and Applications*, Miami, FL, USA, 2024, pp. 567–574, https://doi.org/10.1109/ICMLA61862.2024.00083.

[19] M. Fochesato, F. Khayatian, D. F. Lima, and Z. Nagy, "On the use of conditional TimeGAN to enhance the robustness of a reinforcement learning agent in the building domain," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, Boston, MA, USA, 2022, pp. 208–217, https://doi.org/10.1145/3563357.3564080.

[20] Y. Zhang, Z. Zhou, J. Liu, and J. Yuan, "Data augmentation for improving heating load prediction of heating substation based on TimeGAN," *Energy*, vol. 260, Dec. 2022, Art. no. 124919, https://doi.org/10.1016/j.energy.2022.124919.

[21] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, https://doi.org/10.1162/neco.1997.9.8.1735.

[22] S. M. Hasanat *et al.*, "Enhancing Load Forecasting Accuracy in Smart Grids: A Novel Parallel Multichannel Network Approach Using 1D CNN and Bi-LSTM Models," *International Journal of Energy Research*, vol. 2024, no. 1, July 2024, Art. no. 2403847, https://doi.org/10.1155/2024/2403847.

[23] A. Zafar *et al.*, "A Comparison of Pooling Methods for Convolutional Neural Networks," *Applied Sciences*, vol. 12, no. 17, Sept. 2022, Art. no. 8643, https://doi.org/10.3390/app12178643.

[24] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in *2019 IEEE International Conference on Big Data*, Los Angeles, CA, USA, 2019, pp. 3285–3292, https://doi.org/10.1109/BigData47090.2019.9005997.

[25] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." arXiv, Jan. 30, 2017, https://doi.org/10.48550/arXiv.1412.6980.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[27] J. Jiang and A. Feofilova, "Research on Urban Road Traffic Flow Prediction Based on Hybrid CNN-LSTM Model," *Applied and Computational Engineering*, vol. 129, no. 1, pp. 77–83, Jan. 2025, https://doi.org/10.54254/2755-2721/2025.20467.

[28] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," *Energy*, vol. 182, pp. 72–81, Sept. 2019, https://doi.org/10.1016/j.energy.2019.05.230.

[29] T. Le, M. T. Vo, B. Vo, E. Hwang, S. Rho, and S. W. Baik, "Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM," *Applied Sciences*, vol. 9, no. 20, Oct. 2019, Art. no. 4237, https://doi.org/10.3390/app9204237.

[30] P. P. Phyo and Y.-C. Byun, "Hybrid Ensemble Deep Learning-Based Approach for Time Series Energy Prediction," *Symmetry*, vol. 13, no. 10, Oct. 2021, Art. no. 1942, https://doi.org/10.3390/sym13101942.

[31] A. Agga, A. Abbou, M. Labbadi, and Y. el Houm, "Short-Term Load Forecasting: Based on Hybrid CNN-LSTM Neural Network," in *2021 6th International Conference on Power and Renewable Energy*, Shanghai, China, 2021, pp. 886–891, https://doi.org/10.1109/ICPRE52634.2021.9635488.

[32] Z. A. Khan, T. Hussain, A. Ullah, S. Rho, M. Lee, and S. W. Baik, "Towards Efficient Electricity Forecasting in Residential and Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework," *Sensors*, vol. 20, no. 5, Mar. 2020, Art. no. 1399, https://doi.org/10.3390/s20051399.