EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

Digital Twin-Driven Real-Time Monitoring and Control System for Building Energy Management in Smart Green Buildings

Lihua Zhang[1,*]

[1]School of Resources, Environment and Building Engineering, Chifeng University, Chifeng 024000, Inner Mongolia, China

Abstract

INTRODUCTION: Smart building energy management requires integrated monitoring, prediction, optimization, and control to improve energy efficiency while maintaining suitable indoor environmental conditions.
OBJECTIVES: This study aims to develop a Digital Twin-based real-time monitoring and control system for smart building energy management and to optimize heating, ventilation, air conditioning, lighting, and ventilation settings.
METHODS: The proposed Digital Twin-driven framework integrates Kalman Filter-based state refinement, Res-DL-PEM occupancy prediction, building-state transition modeling, energy-surrogate estimation, and DT-A-MOPSO optimization. The Kalman Filter is used to update the state of the Digital Twin; Res-DL-PEM to forecast occupancy. The state-transition model assesses the future condition of buildings and the energy surrogate predicts the energy consumption. DT-A-MOPSO derives optimal energy consumption, CO_2 concentration and thermal discomfort by choosing HVAC, light and ventilation settings. The Digital Twin is updated with the simulated control responses to make further predictions and optimizations.
RESULTS: The occupancy prediction model achieved an accuracy of 0.9776 and an area under the receiver operating characteristic curve of 0.9927. The energy surrogate achieved an R^2 of 0.9822, while state-transition prediction achieved R^2 values ranging from 0.978 to 0.995. The optimization process reduced predicted energy consumption.
CONCLUSION: The proposed Digital Twin-based framework integrates real-time monitoring, prediction, energy estimation, and multi-objective control, providing an effective approach for improving smart building energy management while considering carbon dioxide levels and thermal comfort.

Keywords: Digital Twin, smart building energy management, occupancy prediction, multi-objective optimization, thermal comfort

Received on 26 September 2026 accepted on 27 October 2026, published on 13 November 2026

Copyright © 2026 Lihua Zhang, licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/ew.15324

*Corresponding author. Email: zhanglihua1@cfu.edu.cn , zlh20260626@163.com

1. Introduction

The management of Smart Building energy has shifted from traditional monitoring and control to systems that integrate building loads, people and environment through connectivity and data. [1]. This domain is important for improving energy efficiency, supporting sustainable building operation, and reducing unnecessary energy consumption [2]. Over time, the integration of smart sensors, Internet of Things (IOT) technologies, cloud and edge computing, and Artificial Intelligence (AI) has expanded the capabilities of building energy-management systems [3]. Recent developments have increasingly incorporated ML, DL, and DT technologies to represent building conditions and support intelligent operation [4]. These technologies enable buildings to collect environmental and operational information, characterize

changing building states, predict energy-related conditions, and support adaptive management [5]. Consequently, smart building energy management has become an important component of sustainable and intelligent building infrastructure. Integrating information and communication technologies with energy systems is essential for sustainable digital infrastructure. Digital Twin-based intelligent control can coordinate building monitoring, predictive modeling, and adaptive energy management, helping reduce energy demand while maintaining indoor environmental quality.

Previous studies have demonstrated substantial progress in data-driven smart building energy management [6]. ML and DL approaches have been applied to energy-consumption and power-generation forecasting, providing predictive information for building operation [7]. IoT-based energy-management systems have supported intelligent monitoring, demand-side management, HVAC scheduling, and energy-efficient operation [8]. Occupancy prediction approaches using recurrent neural networks and optimization techniques have enabled adaptive building management based on changing occupancy conditions [9]. Multi-objective optimization has also been employed for occupancy-aware energy control by jointly considering energy consumption and indoor environmental requirements [10]. DT research has further expanded building-management capabilities through real-time analysis, continuous model calibration, intelligent monitoring, anomaly detection, and DT-enabled energy-management systems [11]. These developments demonstrate the progression toward integrated sensing, prediction, optimization, and intelligent control for smart buildings.

Despite these advances, smart building energy management remains challenging because changing occupancy, indoor environmental conditions, and energy demands require adaptive control that balances energy efficiency and occupant comfort. Past efforts have considered monitoring, occupancy and energy prediction, calibration of Digital Twins, optimization and design control of the building but all these functions either considered independently or simply not within a single closed-loop control. Digital Twin (DT)-based intelligent control is designed to fill this vacuum of advancing the building-state to assist in predictive building modelling and adapting the control decision. To address this limitation, the proposed framework integrates Res-DL-PEM for occupancy prediction, Kalman Filter-based state refinement, state-transition modeling, energy-surrogate estimation, and DT-A-MOPSO for multi-objective optimization. Control responses are simulated and the updated Digital Twin connect successive prediction/optimization cycles that optimize energy use in conjunction with CO_2 concentration and thermal comfort.

1.1. Problem Statement

Smart green building faces several energy-management problems. First, fragmented sensor monitoring provides limited representation of building conditions, restricting coordination of environmental and operational states [12]. Second, dynamic occupancy and building-state variations make short-term prediction difficult, reducing the ability to anticipate energy requirements [13]. Third, energy efficiency, indoor CO_2 concentration, and thermal comfort must be managed simultaneously, making single-objective control insufficient [14]. Fourth, approaches often treat sensing, prediction, energy modeling, optimization, and control as separate tasks, limiting integrated closed-loop management [15]. Although IoT, ML, DL, DT, and optimization methods have achieved progress, challenges remain in combining these capabilities cohesively. The proposed framework addresses these challenges by integrating synchronization, state estimation, prediction, explainability, energy modeling, multi-objective optimization, and closed-loop control within one unified workflow.

1.2. Main Contributions

- Develop a DT-driven smart green building energy-management framework using the Kaggle Room Occupancy Dataset to support building-state representation, prediction, energy modeling, and intelligent control.
- Integrate sensor-state synchronization, KF-based state estimation, Res-DL-PEM residual DT for occupancy prediction, SHAP-based explainability, state-transition modeling, and energy-surrogate regression within the proposed workflow.
- Optimize HVAC setpoint, lighting, and ventilation control using A-MOPSO to jointly manage energy consumption, indoor CO_2 concentration, and thermal discomfort.
- Evaluate the proposed framework through closed-loop simulation, assessing prediction performance, DT synchronization, state-estimation accuracy, energy prediction, optimization performance, energy savings, CO_2 conditions, thermal comfort, latency, and control responsiveness.
- The key novelty is the coordinated integration of Digital Twin synchronization, Kalman Filter-based state refinement, Res-DL-PEM occupancy prediction, SHAP-based explainability, state-transition modeling, energy-surrogate estimation, and DT-A-MOPSO within a simulation-based closed-loop workflow. This integration links predictive building-state estimation with multi-objective control of energy consumption, CO_2 concentration, and thermal discomfort.

The rest of this paper is organised as follows. In section 2, relevant literature and research gap are introduced. In section 3, methodology of the proposed approach is

explained. In section 4, the experiment and the performance evaluation are described. In section 5, discussion, conclusion, and future work are provided.

2. Literature Survey

Digital Twin monitoring and synchronization

Recent smart-building research has explored Digital Twin (DT) architectures, IoT sensing, and real-time co-simulation to improve building monitoring and operational management. Hadjidemetriou et al. [16] have designed a DT architecture which integrates monitoring, control, simulation and Hardware-in-the-Looping scenarios, for real-time and offline building analysis. Krishnan et al. [17] proposed Dynamic Semantic Behavior Data Analysis with IoT sensing to study various factors dealing with energy efficiency with the consideration of environment and user behavior. Mohseni et al. [18] explored Problembased Real-Time Co-simulation for HVAC Control using Functional Mock-up Interface; and Eneyew et al. [19] presented a Generative-model based approach to the calibration of HVAC systems with missing and noisy sensors observations. Saleem et al. [20] combined energy management, IoT and cloud computing along with energy control. These studies constitute crucial steps towards monitoring, synchronization, and DT calibration, but they lack a cohesive connection with predictive energy estimation and multi-objective closed-loop control.

Prediction and intelligent energy management

Data-driven forecasting and intelligent energy-management methods have been developed to support building operation and energy-demand estimation. Afzal et al. [21] used a Multilayer Perceptron with metaheuristic optimization for heating- and cooling-load prediction. Mounir et al. [22] combined Empirical Mode Decomposition with Bidirectional Long Short-Term Memory for load forecasting, while Selvaraj et al. [23] investigated AI-based monitoring for energy prediction and renewable-energy management. Ibrahim et al. [24] designed an energy management approach based on fuzzy logic for hybrid grid/solar photovoltaics (PV) and battery energy sources. In a PV–B battery building Liang et al. [25] made use of a forecasting process to incorporate mixed-integer programming to schedule HVAC operation. Wang et al. [26] united Differential Evolution (DE) and Random Forest Regression (RFR) to forecast renewable energy and optimize carbon. While these are methods for a better prediction and operations decisions making, monitoring and forecasting are not always integrated with the continuous update of the DT state and feedback control.

Optimization, occupant comfort, and integrated control

Recent studies have extended intelligent building management toward occupant-centric control and integrated optimization. Almadhor et al. [27] developed a DT-driven personalized thermal-comfort framework using attention-based LSTM for comfort prediction and HVAC control. Kazemi Naeini et al. [28] proposed PINN-DT, which is based on physics-informed neural networks, deep reinforcement learning, DTs and blockchain technologies with regards to energy optimization and secure smart-grid communication. Piras et al. [29] investigated DT-supported workplace monitoring through IoT integration and data-driven management, while Hosamo et al. [30] explored the integration of Bayesian network, BIM (Building Information Modeling) and machine learning for occupant comfort and HVAC predictive maintenance. The strategies expand coordination between state estimation, energy-surrogate modeling, multi-objective optimization and iterative DT feedback.

2.1. Research Gap

Previous studies have advanced building-energy prediction, Digital Twin modeling, HVAC control, and optimization, but these functions are often addressed separately or integrated to different extents. A research issue that is crucial is linking continuous DT synchronization and state estimation to occupancy prediction and building-state transition modelling, energy-surrogate estimation, and multi-objective control in an iterative feedback loop. To overcome this challenge, the proposed framework includes combining these functions with the explainability support, and incorporating predicted building conditions as a component for the control optimization during the subsequent DT update.

3. Methodology

The proposed DT-driven real-time-oriented framework integrates building-state monitoring, prediction, energy estimation, and adaptive control. Historical Room Occupancy data undergo chronological preprocessing, consistency checking, and normalization. A DT is synchronized with observations, while KF refines the estimated states. Res-DL-PEM predicts occupancy, SHAP provides feature interpretation, and a state-transition model forecasts future state. An energy surrogate estimates consumption, while A-MOPSO optimizes HVAC, lighting, and ventilation. DT-guided control enables iterative closed-loop simulation-based monitoring and evaluation, as shown in Figure 1.

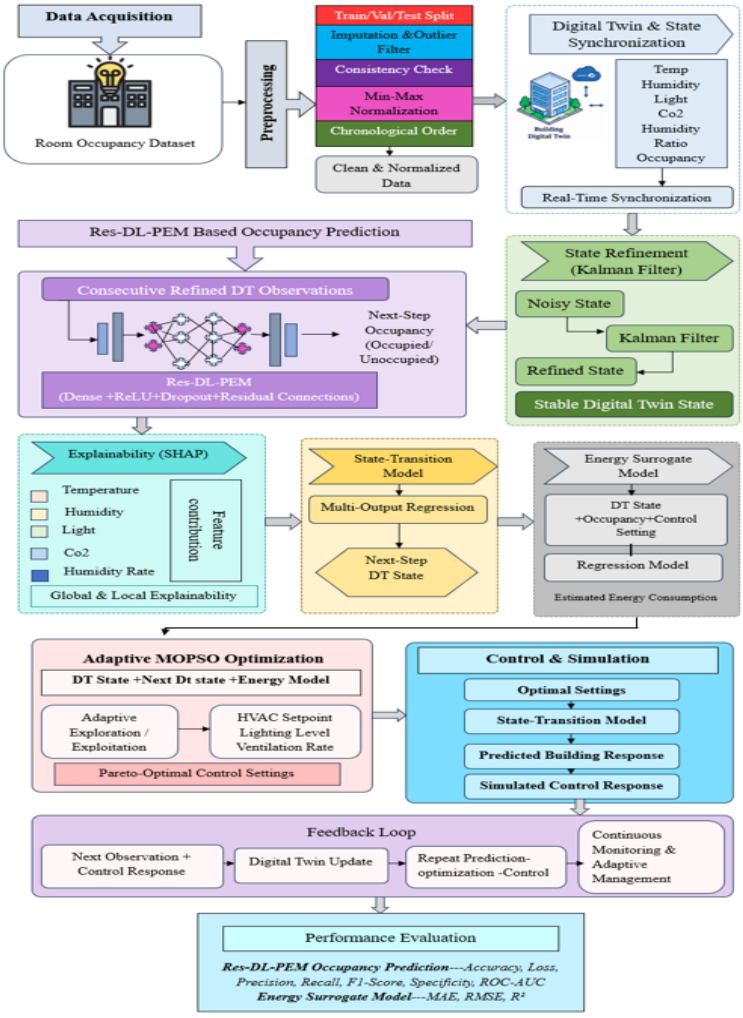


Figure 1. Overall Architecture of the Proposed DT-Driven Building Energy Management and Control Framework

3.1. Data Collection

The raw Room Occupancy Dataset [31] contains 2,665 historical timestamped observations of temperature, humidity, light, CO_2, humidity ratio, and occupancy. Control variables such as HVAC setpoints, lighting, ventilation, energy consumption, CO_2 response, and thermal discomfort are generated within the simulation-based DT environment. These simulated control-state combinations produce next-state and energy responses. 5,000 simulation-generated samples are used for energy-response learning and simulation-based validation rather than experimental building validation.

Table 1. Dataset Partitioning

Dataset	Samples	Percentage
Training	1,332	50.00%
Validation	484	18.16%
Testing	849	31.84%
Total	**2,665**	**100%**

The Room Occupancy Dataset (ROOM) data partitioning is shown in Table 1, consisting of 2,665 sequential observations. Of which 1,332 for training, 484 for validation, and 849 for independent testing. The training set is used for model training, validation for model selection and parameter tuning, and test for performance evaluation.

The collected dataset is represented as:

$$D = \{(X_i, O_i, t_i)\}_{i=1}^{N} \quad (1)$$

In Equation (1) D denotes the collected Room Occupancy Dataset.$X_i$represents the environmental feature vector of the i-th observation, comprising Temperature, Humidity, Light, CO_2, and Humidity Ratio.$O_i \in \{0,1\}$denotes the corresponding binary occupancy condition (Occupied or Unoccupied).$t_i$represents the timestamp of the i-th observation.Ndenotes the total number of observations.

3.2. Data Preprocessing and Synchronization Preparation

Splitting raw room occupancy raw data into the train, validation and test sets by time-based criteria should reduce the possibility of temporal leakage in the data set and provide temporal alignment for the training of decision trees.

Chronological Data Partitioning Let the chronologically ordered building-state dataset be represented as

$$D = \{(X_i, y_i, t_i)\}_{i=1}^{N},\ t_1 \le t_2 \le \cdots \le t_N, \quad (2)$$

In Equation (2), Ddenotes the complete chronologically ordered Room Occupancy Dataset, $X_i$denotes the environmental feature vector of the i-th observation, $y_i$represents the corresponding occupancy state, $t_i$denotes the timestamp associated with the i-th observation, Nrepresents the total number of observations, and the condition $t_1 \le t_2 \le \cdots \le t_N$ensures that the observations remain in chronological order. The dataset is then divided into training, validation, and test subsets while preserving the temporal sequence as

$$D = D_{\text{tr}} \cup D_{\text{val}} \cup D_{\text{te}},$$
$$\max(t_{\text{tr}}) < \min(t_{\text{val}}) < \min(t_{\text{te}}) \quad (3)$$

In Equation (3), D_{tr}, D_{val}, and D_{te}denote the training, validation, and test datasets, respectively, t_{tr}, t_{val}, and t_{te}represent the timestamps associated with the respective subsets.

Missing-Value Imputation Missing values in the numerical building-sensor attributes are replaced using the median calculated exclusively from the training subset. The imputed value is defined as

$$X_{ij}^{*} = \begin{cases} \text{median}(X_{\text{tr},j}), & \text{if } X_{ij} \text{ is missing,} \\ X_{ij}, & \text{otherwise.} \end{cases} \quad (4)$$

In Equation (4), X_{ij}denotes the value of the j-th building-state feature in the i-th observation, X_{ij}^{*}denotes the corresponding value after imputation, $X_{\text{tr},j}$represents the values of feature $j$in the training dataset, median$(\cdot)$denotes the median operation, $i$represents the observation index, and $j$represents the feature index.

IQR-Based Outlier Filtering The interquartile range (IQR) method is employed to identify abnormal sensor observations. For each numerical feature, the IQR is calculated as

$$IQR_j = Q_{3,j} - Q_{1,j}, \quad (5)$$

In Equation (5), IQR_jdenotes the interquartile range of the j-th building-state feature, $Q_{1,j}$represents the first quartile of feature j, and $Q_{3,j}$represents the third quartile of feature j.

The lower and upper outlier boundaries are subsequently calculated as

$$LB_j = Q_{1,j} - 1.5(IQR_j),\ UB_j = Q_{3,j} + 1.5(IQR_j) \quad (6)$$

In Equation (6), LB_jdenotes the lower bound for the j-th feature, UB_jdenotes its upper bound, $Q_{1,j}$and $Q_{3,j}$represent the first and third quartiles, respectively, IQR_jdenotes the interquartile range defined in Equation (4), and 1.5is the standard IQR multiplier used to determine the outlier boundaries.

$$LB_j \le X_{ij}^{*} \le UB_j \quad (7)$$

In Equation (7), X_{ij}^{*}denotes the imputed value of the j-th feature for observation i, while $LB_j$and UB_jrepresent the lower and upper boundaries calculated for feature j, respectively. An observation satisfying the specified interval is retained for subsequent processing.

The correctness of a particular observation can probably be of the form.

$$V(X_i) = \begin{cases} 1, & \text{if } X_i \text{ satisfies all predefined consistency rules,} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

In Equation (8), $X_i$denotes the complete building-state feature vector corresponding to the i-th observation, $V(X_i)$represents the consistency status of that observation, and the value 1indicates that the observation satisfies all predefined sensor-consistency conditions, whereas 0indicates a consistency violation.

Training-Fitted Min–Max Normalization The validated numerical building-state features are normalized using Min–Max normalization. For the j-th feature, the normalized value is calculated as

$$X_{ij}^{\text{norm}} = \frac{X_{ij} - X_{\text{tr},j}^{\min}}{X_{\text{tr},j}^{\max} - X_{\text{tr},j}^{\min}} \quad (9)$$

In Equation (9), X_{ij}^{norm}denotes the normalized value of the j-th feature for observation i, X_{ij}represents the corresponding validated feature value, $X_{\text{tr,j}}^{\min}$denotes the

minimum training-set value of feature j, and $X_{tr,j}^{max}$ denotes the maximum training-set value of feature j.

3.3. DT Construction and State Synchronization

After preprocessing, normalized building observations construct the DT representation. Chronological sensor data update corresponding virtual variables, synchronizing observed conditions with their virtual representation and providing the current DT state.

$$S_t = [T_t, H_t, L_t, C_t, HR_t, O_t]^T \quad (10)$$

In Equation (10), S_t denotes the DT state vector at time t, T_t represents Temperature, H_t denotes Humidity, L_t represents Light intensity, C_t denotes CO_2 concentration, HR_t represents Humidity Ratio, O_t denotes Occupancy, and T represents the transpose operation.

At each chronological time step, the observed building variables are mapped to their corresponding virtual-state components using a consistent variable ordering. The measurements of the following parameters are then used to update the state vector of the Digital Twin: temperature, humidity, light intensity, CO_2 concentration, humidity ratio, and occupancy observations. Then, the synchronized state is compared to the corresponding observation vector, to describe the synchronization error. The mapping process preserves the correspondence between the building state observed and the building state modeled, whereas the next filtering stage, Kalman filtering, improves the state estimates, by fusing the double-sided map with the state estimate predicted.

The corresponding chronological sensor observation is represented as

$$Y_t = [T_t^{obs}, H_t^{obs}, L_t^{obs}, C_t^{obs}, HR_t^{obs}, O_t^{obs}]^T \quad (11)$$

In Equation (11), Y_t denotes the sensor observation vector at time t, T_t^{obs} represents the observed Temperature, H_t^{obs} denotes the observed Humidity, L_t^{obs} represents the observed Light intensity, C_t^{obs} denotes the observed CO_2 concentration, HR_t^{obs} represents the observed Humidity Ratio, O_t^{obs} denotes the observed Occupancy, and T represents the transpose operation.

The sensor-to-Digital-Twin mapping is performed to update the corresponding virtual building variables as

$$S_t^{DT} = Y_t \quad (12)$$

In Equation (12), S_t^{DT} denotes the synchronized DT state at time t, and Y_t denotes the corresponding chronological sensor observation.

The consistency between the sensor observation and the synchronized DT state is expressed as

$$E_t^{sync} = Y_t - S_t^{DT} \quad (13)$$

In Equation (13), E_t^{sync} denotes the DT synchronization error at time t, Y_t represents the observed building-state vector, and S_t^{DT} denotes the corresponding synchronized DT state

Algorithms 1 summarize the key computational procedures of the proposed DT framework, covering state refinement, occupancy prediction, and multi-objective control optimization.

Algorithm 1. DT State Refinement Using KF

Input: Synchronized DT state, sensor observations, system matrices, and noise covariance matrices
Output: Refined DT state
Load synchronized DT state and sensor observation.
Initialize KF state and covariance.
Predict the next DT state.
Predict state-error covariance.
Calculate Kalman gain.
Compute measurement innovation.
Update the DT state.
Update state-error covariance.
If the refined state satisfies consistency conditions, retain it.
Else, apply state-consistency correction.
Return the refined DT state.

3.4. DT State Refinement

The synchronized DT state is refined using a KF, which recursively combines predicted states with sensor observations through prediction and measurement updates while accounting for process and measurement uncertainly.

The Kalman Filter models building-state evolution using a discrete-time transition matrix, assuming that the current state estimate provides the basis for predicting the next state. Variations in occupancy and indoor environmental conditions are treated as process uncertainty, while measurement updates correct predictions using new observations. The covariance matrices are $Q = 10^{-4} I_6$ and $R = 5 \times 10^{-3} I_6$, where I_6) denotes the six-dimensional identity matrix. The filter results in a reduction of the impact of sensor readout noise and changes in sensor output due to combining the predicted state and the observed measurements by their respective uncertainties. The resulting refined state results in a more consistent representation of indoor building state in the DT, enabling the use of the DT for building occupancy prediction, state-transition modelling, optimization of building control.

The predicted building state is calculated as

$$\hat{S}_{t|t-1} = A\hat{S}_{t-1|t-1} \quad (14)$$

In Equation (14), $\hat{S}_{t|t-1}$ denotes the predicted DT state at time t based on information available up to time $t-1$, $\hat{S}_{t-1|t-1}$ represents the refined DT state from the previous observation, and A denotes the state-transition matrix describing the temporal evolution of the building state.

The predicted state covariance is obtained as

$$P_{t|t-1} = AP_{t-1|t-1}A^T + Q \quad (15)$$

In Equation (15), $P_{t|t-1}$ denotes the predicted state-error covariance matrix, $P_{t-1|t-1}$ represents the refined state-error covariance from the previous step, A denotes the state-transition matrix, A^T represents its transpose, and Q denotes the process-noise covariance matrix representing uncertainty associated with the building-state evolution.

The Kalman gain is calculated as

$$K_t = P_{t|t-1}H^T\left(HP_{t|t-1}H^T + R\right)^{-1} \quad (16)$$

In Equation (16), K_t denotes the Kalman gain at time t, $P_{t|t-1}$ represents the predicted state-error covariance, H denotes the observation matrix that relates the building state to the sensor measurements, H^T represents the transpose of the observation matrix, and R denotes the measurement-noise covariance matrix representing uncertainty in the sensor observations.

The KF updates the predicted building state and its associated covariance using current sensor observations.

$$\hat{S}_{t|t} = \hat{S}_{t|t-1} + K_t\left(Y_t - H\hat{S}_{t|t-1}\right) \quad (17)$$

In Equation (17), $\hat{S}_{t|t}$ denotes the refined DT state at time t, $\hat{S}_{t|t-1}$ represents the predicted state, K_t denotes the Kalman gain, Y_t represents the current sensor observation, H denotes the observation matrix, and $Y_t - H\hat{S}_{t|t-1}$ represents the measurement innovation or difference between the actual and predicted observations.

$$P_{t|t} = (I - K_tH)P_{t|t-1} \quad (18)$$

In Equation (18), $P_{t|t}$ denotes the updated state-error covariance matrix, I represents the identity matrix, K_t denotes the Kalman gain, H represents the observation matrix, and $P_{t|t-1}$ denotes the predicted state-error covariance.

3.5. Res-DL-PEM-Based Building-State Prediction

Refined DT states are provided to Res-DL-PEM, which uses five consecutive observations, Dense layers, ReLU activation, dropout, and residual connections to predict next-step occupancy probability.

Res-DL-PEM uses five consecutive time steps of five environmental input variables: temperature, humidity, light intensity, CO_2 concentration, and humidity ratio. Although the Digital Twin state contains six variables, occupancy is excluded from the input feature vector because it is the prediction target. Thus, there is a (5 × 5) temporal input matrix which is flattened to a 25-dimensional vector for the fully connected layers. The model takes this historical sequence of environments to forecast occupancy at the next timestep and produces the number representing the occupancy probability.

Let the five consecutive refined DT observations be represented as

$$X_t = \left[\hat{S}_{t-4|t-4}, \hat{S}_{t-3|t-3}, \hat{S}_{t-2|t-2}, \hat{S}_{t-1|t-1}, \hat{S}_{t|t}\right] \quad (19)$$

In Equation (19), X_t denotes the input sequence provided to the Res-DL-PEM at time t, $\hat{S}_{t-k|t-k}$ denotes the refined DT state at the corresponding previous time step, and k represents the temporal offset.

The output of the first Dense transformation is expressed as

$$H_1 = \mathrm{ReLU}(W_1X_t + b_1) \quad (20)$$

In Equation (20), H_1 denotes the first hidden representation, W_1 represents the trainable weight matrix of the first Dense layer, X_t denotes the five-step input sequence, b_1 represents the corresponding bias vector, and ReLU(·) denotes the Rectified Linear Unit activation function.

Dropout regularization is then applied to the hidden representation as

$$\tilde{H}_1 = \mathrm{Dropout}(H_1) \quad (21)$$

In Equation (21), $\tilde{H}_1$ denotes the regularized hidden representation, H_1 represents the output of the first Dense layer, and Dropout(·) denotes the stochastic regularization The residual transformation is formulated as

$$H_2 = \mathrm{ReLU}\left(W_2\tilde{H}_1 + b_2 + W_rX_t\right) \quad (22)$$

In Equation (22), H_2 denotes the residual hidden representation, W_2 and b_2 represent the trainable weight matrix and bias vector of the subsequent Dense transformation, $\tilde{H}_1$ denotes the dropout-regularized representation, W_r represents the trainable residual projection matrix, X_t denotes the original five-step input sequence, and ReLU(·) represents the activation function.

In the residual block, the main pathway transforms the dropout-regularized representation $\tilde{H}_1$ through the second Dense layer using W_2 and b_2. In parallel, the skip pathway connects the original 25 -dimensional input X_t to the residual block through the linear projection W_rX_t. Projected input is element-wise added to the output of the main pathway and followed by the ReLU activation. The projection W_r aligns the input and hidden-representation dimensions, thus preserving the original input information

even in the learned input-representation space to which it is projected for next-step occupancy prediction.

Finally, the predicted next-step occupancy probability is obtained as

$$\hat{O}_{t+1} = \sigma(\mathrm{W}_o \mathrm{H}_2 + b_o) \tag{23}$$

In Equation (23), $\hat{O}_{t+1}$ denotes the predicted probability of occupancy at the next time step, W_o represents the output-layer weight vector, H_2 denotes the final hidden representation, b_o represents the output bias, and $\sigma(\cdot)$ denotes the sigmoid activation function that maps the output to the range $[0,1]$.

Class imbalance in occupancy prediction is addressed by training the Res-DL-PEM model using weighted binary cross-entropy (BCE) with a positive-class weight of 1.40 . The loss is defined as

$$\mathcal{L}_{\mathrm{WBCE}} = -\frac{1}{N}\sum_{i=1}^{N} [1.40 y_i \ln(p_i) + (1 - y_i)\ln(1 - p_i)] \tag{24}$$

In Equation (24), N denotes the number of training samples, $y_i \in \{0,1\}$ is the actual occupancy label, and $p_i \in [0,1]$ is the predicted occupancy probability. The weight 1.40 makes the errors of the samples that belong to the occupied class more salient for the BCE loss while reducing the errors of the unweighted samples. During training, the weighted loss is minimized in order to optimize the model parameters for use in predicting the occupancy of the next step.

Res-DL-PEM training employs ReduceLROnPlateau and early stopping to improve convergence and avoid unnecessary training. The scheduler keeps track of the validation loss and slows down the learning rate if the validation loss doesn't decrease for a predetermined number of epochs. Early stopping stops training if the loss on the validation set does not improve for a number of epochs given by its patience parameter. Those weights with minimal validation loss are used to determine occupancy. The implemented training configuration is reflected in the following settings: scheduler and early stopping.

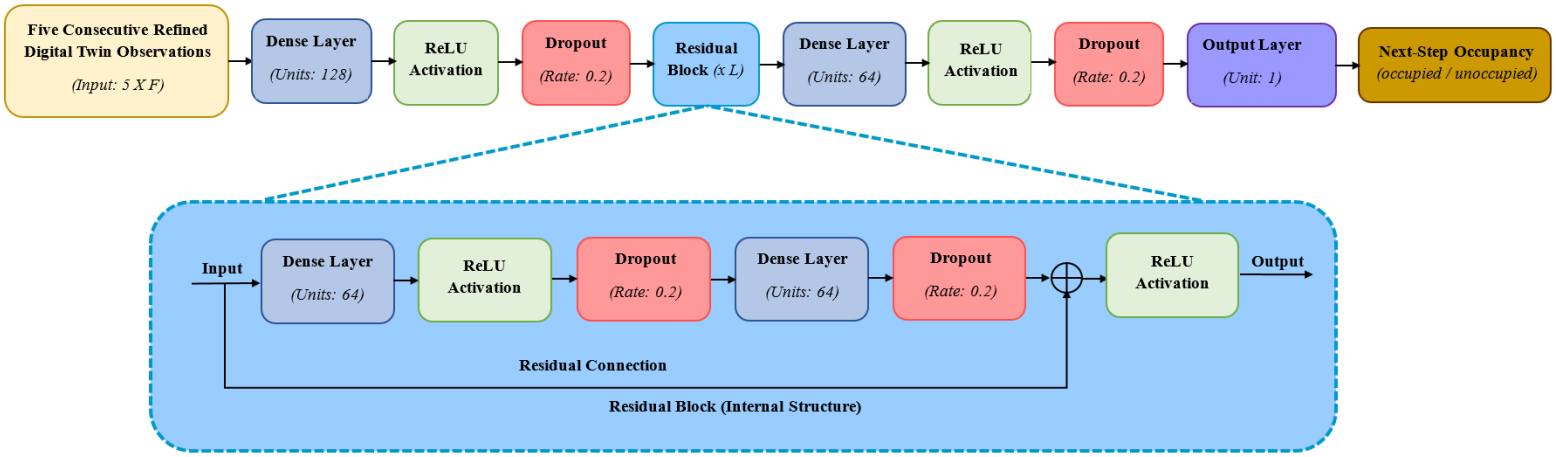


Figure 2. Architecture of the Proposed Res-DL-PEM for Next-Step Occupancy Prediction

Figure 2 shows the Res-DL-PEM receives five consecutive refined DT observations and processes them through dense layers, ReLU activation, dropout, and residual connections. The learned representation is passed through additional dense layers and an output layer to predict next-step occupancy as occupied or unoccupied.

Algorithm 2 presents the Res-DL-PEM procedure for predicting the next-step building occupancy state from consecutive refined DT observations.

Algorithm 2. Res-DL-PEM-Based Occupancy Prediction

Input: Five consecutive refined DT observations and occupancy labels
Output: Predicted next-step occupancy
Load chronologically ordered refined DT observations.
Construct five-step input sequences.
Prepare training, validation, and testing data.
Initialize Res-DL-PEM.
Process the input through Dense and ReLU layers.

Apply dropout regularization.
Generate the residual representation.
Generate occupancy probability.
If probability ≥ classification threshold, assign Occupied.
Else, assign Unoccupied.
Calculate classification loss.
Update model parameters.
Repeat training until convergence.
Return predicted occupancy.

3.6. SHAP-Based Post-hoc Explainability

SHAP is applied to the Res-DL-PEM occupancy-prediction component to interpret the model's predicted occupancy probability at the next time step, $\hat{O}_{t+1}$.The analysis takes into account the variation in temperature, humidity, light intensity, CO_2 concentration and humidity ratio for the five historical input time steps. SHAP values measure the influences of each input feature on the output of the model, compared to a reference value, called the SHAP baseline. Positive SHAP values indicate that a feature increases the predicted occupancy probability relative to the baseline, whereas negative values indicate a decrease. The absolute SHAP value indicates the strength of the feature: features with stronger SHAP values are more likely to have a larger impact than those with weaker SHAP values. The computed SHAP values for different environmental variables and historical time steps can be used to understand which conditions and time periods have the most significant influence in the probability of the occupancy prediction, enhancing the interpretability of the model's predictions in the proposed Digital Twin framework.

SHAP values are computed from validation or test observations to quantify each environmental variable's contribution to Res-DL-PEM occupancy predictions, providing post-hoc interpretation of model outputs.

$$\phi_i(f,\mathrm{x}) = \sum_{\mathcal{S}\subseteq\mathcal{N}\setminus\{i\}} \frac{|\mathcal{S}|!(M-|\mathcal{S}|-1)!}{M!}\left[f\left(\mathrm{x}_{\mathcal{S}\cup\{i\}}\right) - f(\mathrm{x}_{\mathcal{S}})\right] \quad (25)$$

In Equation (25), $\phi_i(\mathrm{f},\mathrm{x})$ denotes the SHAP value of the i-th feature for input x, $\mathrm{f}(\cdot)$ represents the trained Res-DL-PEM prediction function, $\mathcal{N}$ denotes the complete set of input features, $\mathcal{S}$ represents a subset of features excluding feature i, M denotes the total number of features, and $\mathrm{x}_{\mathcal{S}}$ and $\mathrm{x}_{\mathcal{S}\cup\{\mathrm{i}\}}$ represent the model inputs with and without feature i, respectively.

For the proposed framework, the feature set is defined as

$$\mathcal{F} = \{T, H, L, C, HR\} \quad (26)$$

In Equation (26), T, H, L, C, and HR denote Temperature, Humidity, Light, CO_2, and Humidity Ratio, respectively.$\mathcal{F}$ represents the environmental feature set used for interpretation.

The global importance of each environmental variable is obtained from the mean absolute SHAP value as

$$I_i = \frac{1}{N}\sum_{j=1}^{N} \left|\phi_i^{(j)}\right| \quad (27)$$

In Equation (27), $\mathrm{I_i}$ denotes the global importance of feature i, N represents the number of validation or test observations analyzed, and $\phi_i^{(j)}$ denotes the SHAP value of feature i for the j-th observation. Higher $\mathrm{I_i}$ values indicate greater contribution of the corresponding environmental variable to the occupancy prediction.

3.7. Data-Driven DT State-Transition Modeling

The DT state-transition model predicts subsequent building states using current states, predicted occupancy, control actions, and environmental disturbances, forming one-step transitions from chronological historical observations.

The state-transition model uses the refined Digital Twin state, predicted occupancy, control settings, and simulated environmental disturbances to estimate subsequent building conditions. The disturbances are added as extra features alongside the current state and control variables and the GradientBoostingRegressor models take account of the variability of the environment when creating their model of the next state. Addressing the variation between the different response patterns of the above-mentioned variables, separate models are trained for the different variables: temperature, humidity, light intensity, CO_2 concentration, humidity ratio, and occupancy. They take their six predictions together in a set order to create the next-state vector that will be used to update the current virtual building state for the next cycle of control. The 200 estimators, maximum depth=4, learning rate=0.05, and subsampling ratio=0.80 is used for each model.

$$\mathrm{Z}_t = \left[\hat{\mathrm{S}}_t, \hat{O}_{t+1}, \mathrm{U}_t, \mathrm{D}_t\right] \quad (28)$$

In Equation (28), Z_t denotes the state-transition input vector, $\hat{\mathrm{S}}_t$ represents the current refined DT state, $\hat{O}_{t+1}$ denotes the predicted next-step occupancy, U_t represents the control variables including HVAC setpoint, lighting level, and ventilation rate, and D_t denotes the predicted environmental disturbances.

The predicted building state connects state-transition modeling with the energy-management control process by providing estimates of indoor conditions under the selected control settings. The HVAC energy-surrogate model's energy-consumption predictions are combined with these estimates to assess the desired objectives for HVAC setpoints, lighting levels and ventilation rates. On the basis of these estimates DT-A-MOPSO can identify control settings that enable to meet a tradeoff between energy consumption, CO_2 concentration and thermal discomfort.

These are then applied to the next simulated state transition and Digital Twin Update cycle, resulting in linkage of building-state prediction with iterative energy-management decisions.

The subsequent DT state is predicted using a multi-output regression function as

$$\hat{S}_{t+1} = f_{STR}(Z_t; \theta) \quad (29)$$

In Equation (29), $\hat{S}_{t+1}$ denotes the predicted next-step DT state, $f_{STR}(\cdot)$ represents the data-driven state-transition regression model, Z_t denotes the transition input vector, and θ represents the trainable model parameters. The output $\hat{S}_{t+1}$ contains the predicted Temperature, Humidity, Light, CO_2, Humidity Ratio, and Occupancy states.

The chronological training samples are constructed by pairing each current state with its subsequent state as

$$\mathcal{D}_{ST} = \{(Z_t, S_{t+1})\}_{t-1}^{N-1} \quad (30)$$

In Equation (30), $\mathcal{D}_{ST}$ denotes the state-transition training dataset, Z_t represents the input variables at time t, S_{t+1} denotes the observed subsequent building state, and N represents the total number of chronological observations.

The multi-output regression model is trained by minimizing the mean squared error across the predicted state variables as

$$\mathcal{L}_{ST} = \frac{1}{N_s d} \sum_{t=1}^{N_s} \left\| S_{t+1} - \hat{S}_{t+1} \right\|_2^2 \quad (31)$$

In Equation (31), $\mathcal{L}_{ST}$ denotes the state-transition training loss, N_s represents the number of training transition samples, d denotes the number of DT state variables, S_{t+1} represents the observed nextstep state, $\hat{S}_{t+1}$ denotes the predicted next-step state, and $\| \cdot \|_2^2$ represents the squared Euclidean error across the multiple output states.

3.8. Data-Driven Energy Surrogate Model Development

The energy-surrogate model was evaluated using R2, NRMSE, NMAE, RMSE, and MAE, with a GradientBoostingRegressor configured with 300 estimators and a maximum depth of 5. The 5,000 simulation-generated samples were not divided into training, validation and testing sets therefore the reported values reflect reporting the evaluation and not independent test. For DT-A-MOPSO optimization, the trained surrogate models a value of the electrical energy required for candidate electric power settings of the control. Independent evaluation on held-out simulation samples is required to assess generalization to unseen data and distinguish predictive performance from optimization use.

$$\hat{E}_t = f_{\theta_E}(\mathbf{E}_t) \quad (32)$$

In Equation (32), where $\hat{E}_t$ denotes the estimated energy consumption, $f_{\theta_E}(\cdot)$ represents the trained GradientBoostingRegressor, and θ_E denotes its learned parameters. The input vector $\mathbf{E}_t$ contains the building state and control variables passed on to the energy-surrogate model such as the relevant environmental conditions, occupancy information, HVAC temperature setpoint, lighting level and ventilation rate. These inputs represent the simulated building condition and the set of candidate control variables, and the trained model can simulate energy use in the building for each of the candidate control variables that are being evaluated by DT-A-MOPSO.

Because the Room Occupancy Dataset lacks measured energy consumption, a data-driven surrogate model uses simulated energy data to estimate consumption from DT states, predicted occupancy, and controls.

$$E_t = \left[\hat{S}_{t+1}, \hat{O}_{t+1}, T_t^{\text{set}}, L_t^{ctrl}, V_t^{ctrl}\right] \quad (33)$$

In Equation (33), E_t denotes the input vector to the energy surrogate model, $\hat{S}_{t+1}$ represents the predicted next-step DT state, $\hat{O}_{t+1}$ denotes the predicted occupancy, T_t^{set} represents the HVAC temperature setpoint, L_t^{ctrl} denotes the lighting control level, and V_t^{ctrl} represents the ventilation rate.

The estimated building energy consumption is obtained using a multi-variable regression function as

$$\hat{E}_t = f_{ES}(E_t; \theta_E) \quad (34)$$

In Equation (34), $\hat{E}_t$ denotes the estimated building energy consumption, $f_{ES}(\cdot)$ represents the data-driven energy surrogate regression model, E_t denotes the model input vector, and θ_E represents the trainable parameters of the energy surrogate model.

The simulated energy dataset is used to calibrate the surrogate model by minimizing the prediction error between simulated and estimated energy consumption:

$$\mathcal{L}_E = \frac{1}{N_E} \sum_{i=1}^{N_E} \left(E_i - \hat{E}_i\right)^2 \quad (35)$$

In Equation (35), $\mathcal{L}_E$ denotes the energy-surrogate training loss, N_E represents the number of simulated energy samples, E_i denotes the simulated energy consumption for sample i, and $\hat{E}_i$ represents the corresponding energy consumption estimated by the surrogate model.

Thermal discomfort is defined as the absolute deviation between predicted indoor temperature and the desired thermal setpoint. Smaller deviations being closer to the target and larger deviations being greater thermal departure. This objective is checked for each candidate HVAC, lighting and ventilation configuration where the temperature is predicted based on the present configuration of the Digital Twin, which includes control settings, occupancy and sensor data. DT-A-MOPSO reduces the energy consumption and CO_2 concentration compared to

the specified thermal condition and reduces the thermal deviation.

$$D_T(t) = |T_{\text{pred}}(t) - T_{\text{set}}| \quad (36)$$

In Equation (36), $D_T(t)$denotes the simulated thermal discomfort at time $t$in °C, $T_{\text{pred}}(t)$is the predicted indoor temperature obtained from the state-transition model, and $T_{\text{set}}$is the desired indoor temperature setpoint.

The desired setpoint T_{set} serves as the reference temperature for evaluating the predicted indoor temperature $T_{\text{pred}}(t)$; consequently, thermal discomfort is zero when the predicted temperature equals the setpoint and increases with the magnitude of the temperature deviation.

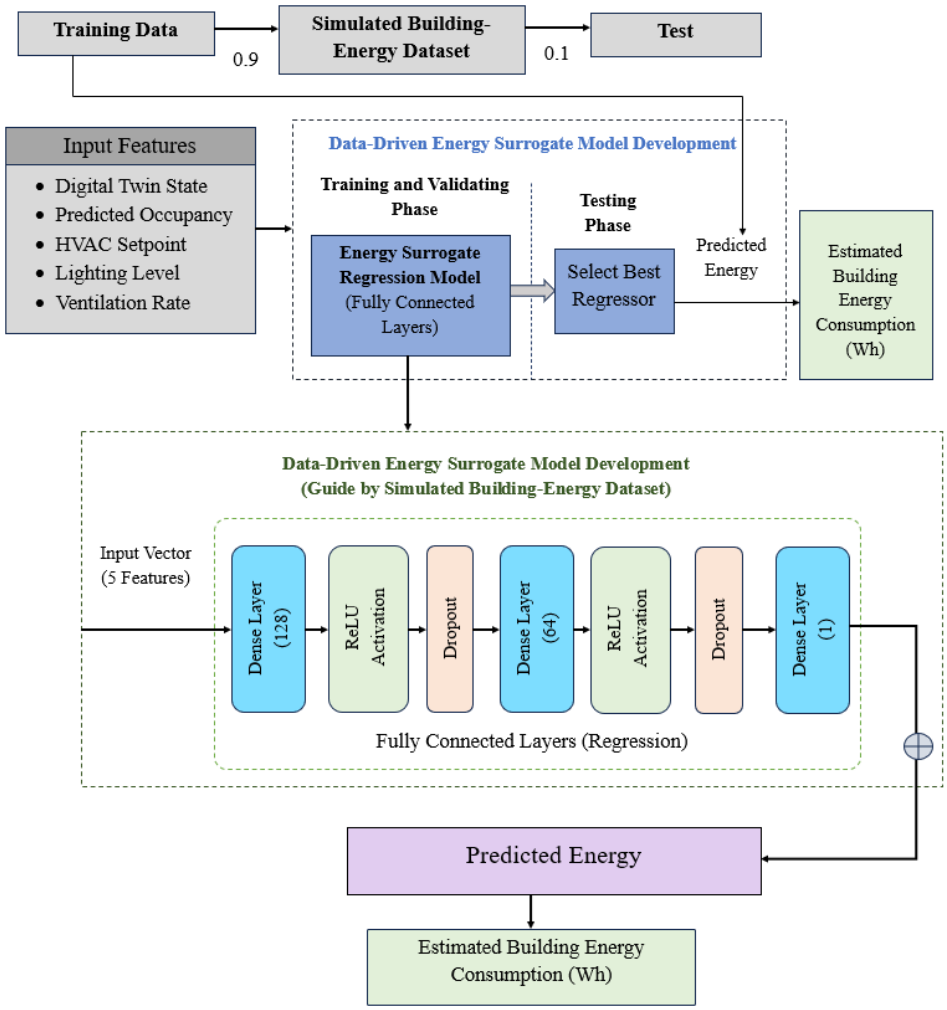


Figure 3. Proposed Energy Consumption Prediction Model for DT-Based Building

Figure 3 shows that the model uses DT state, predicted occupancy, HVAC setpoint, lighting level, and ventilation rate as inputs. Fully connected layers with ReLU activation and dropout learn energy relationships from simulated building-energy data, producing predicted energy and estimated building energy consumption in Wh.

3.9. DT-Guided A-MOPSO

DT-guided A-MOPSO determines control settings that jointly minimize estimated energy consumption, CO_2 concentration, and thermal discomfort using current and predicted DT states with the energy surrogate.

Energy consumption, CO_2 concentration, and thermal discomfort are normalized before optimization using min–max scaling:

$$\hat{f}_j = \frac{f_j - f_j^{\min}}{f_j^{\max} - f_j^{\min}} \quad (37)$$

In Equation (37), f_j is the original objective value, $f_j^{\min}$ and $f_j^{\max}$ are its lower and upper reference bounds, and $\hat{f}_j$

is its normalized value. The bounds are the lower and upper bounds in the feasible set of objective function values and are unchanged when optimizing. DT-A-MOPSO concurrently reduces NMO to find the set of non-dominated control solution(s).

The adaptive mechanism in DT-A-MOPSO adjusts the inertia weight and cognitive and social coefficients throughout optimization to balance exploration and convergence. The inertia weight and the cognitive coefficient are reduced as the iteration count goes up, and the social coefficient is raised implying that the search is gradually focused on search towards preferable solutions from the Pareto set. The parameters are updated as follows:

$$w(t) = w_{\max} - \frac{t}{T}(w_{\max} - w_{\min}) \quad (38)$$

$$c_1(t) = c_{1,\max} - \frac{t}{T}\left(c_{1,\max} - c_{1,\min}\right) \quad (39)$$

$$c_2(t) = c_{2,\min} + \frac{t}{T}\left(c_{2,\max} - c_{2,\min}\right) \quad (40)$$

In Equations (38) & (39) & (40), t is the current iteration, T is the maximum number of iterations, $w_{\max}$ and $w_{\min}$ are the inertia-weight bounds, and $c_{1,\max}, c_{1,\min}, c_{2,\max}$, and $c_{2,\min}$ define the coefficient bounds. These adaptive parameters regulate particle velocity according to

$$\mathbf{v}_i(t+1) = w(t)\mathbf{v}_i(t) + c_1(t)r_1[\mathbf{p}_i(t) - \mathbf{x}_i(t)] + c_2(t)r_2[\mathbf{g}_i(t) - \mathbf{x}_i(t)] \quad (41)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (42)$$

In Equations (41) & (42), $\mathbf{x}_i$ and $\mathbf{v}_i$ denote particle position and velocity, $\mathbf{p}_i$ is the particle's personal best, $\mathbf{g}_i$ is a selected Pareto-archive leader, and $r_1, r_2 \in [0,1]$ are random coefficients. The updated positions represent candidate HVAC, lighting, and ventilation settings.

$$\mathrm{u}_t = \left[T_t^{nt}, L_t^{\mathrm{tril}}, V_t^{\mathrm{trtr}}\right] \quad (43)$$

In Equation (43), $\mathrm{u_t}$ denotes the control decision vector, $\mathrm{T_t^{ext}}$ represents the HVAC temperature setpoint, $\mathrm{L_t^{thr\ t}}$ denotes the lighting level, and $\mathrm{V_t^{etrl}}$ represents the ventilation rate.

For each candidate control vector, the predicted next DT state and energy consumption are obtained as

$$\left(\hat{\mathrm{S}}_{t+1}, \hat{E}_t\right) = \left(f_{\mathrm{ST}}(\mathrm{Z}_t, \mathrm{u}_t), f_{\mathrm{ES}}(\mathrm{E}_t, \mathrm{u}_t)\right) \quad (44)$$

In Equation (44), $\hat{\mathrm{S}}_{t+1}$ denotes the predicted next DT state, $\hat{\mathrm{E}}_t$ represents the estimated energy consumption, $f_{\mathrm{ST}}(\cdot)$ denotes the state-transition regression model, $f_{\mathrm{ES}}(\cdot)$ denotes the energy surrogate model, $\mathrm{Z_i}$ represents the current state and predicted disturbance information, $\mathrm{E_i}$ represents the energy-model input variables, and $\mathrm{u_t}$ denotes the candidate control vector.

The multi-objective optimization is formulated as

$$\min_{\mathrm{u}_t} \mathrm{F}(\mathrm{u}_t) = \left[\hat{E}_t, \hat{C}_{t+1}, D_{t+1}\right] \quad (45)$$

In Equation (45),$\mathrm{F(u_i)}$ denotes the multi-objective fitness vector, $\hat{\mathrm{E}}_t$ represents estimated energy consumption, $\hat{\mathrm{C}}_{t+1}$ denotes the predicted $\mathrm{CO_2}$ concentration, and $\mathrm{D_{t+1}}$ represents the predicted thermal discomfort.

The particle position is iteratively updated according to the A-MOPSO mechanism as

$$\mathrm{v}_i^{k+1} = \omega^k \mathrm{v}_i^k + c_1^k r_1\left(\mathrm{p}_i^k - \mathrm{x}_i^k\right) + c_2^k r_2\left(\mathrm{g}^k - \mathrm{x}_i^k\right) \quad (46)$$

$$\mathrm{x}_i^{k+1} = \mathrm{x}_i^k + \mathrm{v}_i^{k+1} \quad (47)$$

In Equation (46) & (47), v_i^k denotes the velocity of particle i at iteration k, u^k represents the adaptive inertia weight, c_1^k and c_2^k denote the adaptive cognitive and social coefficients, r_1 and r_2 are random values in $[0,1]$, p_i^k represents the particle's personal best position, g^k denotes the selected global Pareto-guiding positio

The non-dominated candidate solutions obtained during the optimization form the Pareto set:

$$\mathcal{P}^* = \left\{\mathrm{u}_i^{(j)} \mid \mathrm{u}_t^{(j)} \text{ is non-dominated}\right\} \quad (48)$$

In Equation (48), $\mathcal{P}^*$ denotes the Pareto-optimal solution set, and $\mathrm{u_t^{(j)}}$ represents the j-th candidate control vector that is not dominated by another feasible solution across the energy, $\mathrm{CO_2}$, and thermal-discomfort objectives.

Algorithm 3 presents the DT-guided A-MOPSO procedure for optimizing HVAC, lighting, and ventilation controls under energy, $\mathrm{CO_2}$, and thermal-comfort objectives.

Algorithm 3. DT -Guided A-MOPSO

Input: Current DT state, predicted occupancy, state-transition model, energy surrogate model, and control **constraints**
Output: Pareto-optimal HVAC, lighting, and ventilation settings
Initialize particles and velocities within feasible control ranges.
Evaluate candidate HVAC, lighting, and ventilation settings.
Predict the next DT state using the state-transition model.
Estimate energy consumption using the energy surrogate.
Calculate $\mathrm{CO_2}$ concentration and thermal discomfort.
Evaluate the multi-objective fitness.
Identify non-dominated solutions.
Update particle velocity and position.

Adapt exploration and exploitation parameters.
If the stopping criterion is not satisfied, repeat the optimization.
Else, generate the Pareto front.
Select the required Pareto-optimal control solution.
Return optimized HVAC, lighting, and ventilation settings.

3.10. DT-Guided Building Control

The selected Pareto-optimal control solution is applied within the simulated building environment, where HVAC, lighting, and ventilation settings are provided to the state-transition model for next-state prediction.

$$u_t^* = \left[T_t^{set*}, L_t^{ctrl*}, V_t^{ctrl*}\right] \quad (49)$$

In Equation (49), u_t^* denotes the selected Pareto-optimal control vector, T_t^{set*} represents the optimized HVAC temperature setpoint, L_t^{ctrl*} denotes the optimized lighting level, and V_t^{ctrl*} represents the optimized ventilation rate.

$$\hat{S}_{t+1} = f_{ST}(Z_t, u_t^*) \quad (50)$$

In Equation (50), $\hat{S}_{t+1}$ denotes the predicted next-step building state, $f_{ST}(\cdot)$ represents the trained state-transition regression model, Z_t denotes the current DT state and predicted environmental conditions, and u_t^* represents the selected optimized control vector.
The resulting state is used to characterize the simulated building-control response as

$$R_{t+1} = \hat{S}_{t+1} \quad (51)$$

In Equation (51), R_{t+1} denotes the simulated building-control response, while $\hat{S}_{t+1}$ represents the predicted next-step DT state containing the resulting building environmental and occupancy conditions.

3.11. Simulated Physical-Virtual Feedback Loop

The simulated feedback loop iteratively updates the DT, refines states, predicts occupancy, optimizes controls, evaluates responses, and propagates predicted states forward.

At each control cycle, DT-A-MOPSO selects a Pareto-optimal combination of HVAC setpoint, lighting level, and ventilation rate. These control settings are then sent to state transition model to forecast the next step building state, and the energy surrogate is the calculated energy consumption for the building involving the next step. The predicted state from the Digital Twin and the simulated control response are then fed into the next chronological observation to update the Digital Twin. It keeps building on the state estimated from the updated state to pass to the next state estimation, prediction and optimisation cycle to create an iterative closed loop process.

The chronological feedback process is represented as:

$$S_t^{DT} \rightarrow \hat{S}_t \rightarrow \hat{O}_{t+1} \rightarrow \hat{S}_{t+1} \rightarrow u_t^* \rightarrow \hat{E}_t \rightarrow S_{t+1}^{DT} \quad (52)$$

In Equation (52), S_t^{DT} denotes the current DT state, $\hat{S}_t$ represents the Kalman-refined state, $\hat{O}_{t+1}$ denotes the predicted next-step occupancy, $\hat{S}_{t+1}$ represents the predicted next DT state, u_t^* denotes the selected Pareto-optimal control action, $\hat{E}_t$ represents the estimated energy consumption, and S_{t+1}^{DT} denotes the updated DT.

The DT is updated using the next chronological observation and simulated response as

$$S_{t+1}^{DT} = \mathcal{U}(Y_{t+1}, \hat{S}_{t+1}) \quad (53)$$

In Equation (53), where $\mathcal{U}(\cdot)$ represents the DT state-update operation. S_{t+1}^{DT} denotes the updated DT state, Y_{t+1} represents the next chronological sensor observation, and $\hat{S}_{t+1}$ denotes the simulated next building state

The iterative feedback operation can be expressed as

$$\mathcal{W}_{t+1} = \mathcal{F}(Y_{t+1}, S_{t+1}^{DT}, u_t^*), t = 1, \ldots, N-1 \quad (54)$$

In Equation (54), $\mathcal{W}_{t+1}$ denotes the next simulation iteration, $\mathcal{F}(\cdot)$ represents the closed-loop feedback operation, Y_{t+1} denotes the next chronological observation, S_{t+1}^{DT} represents the updated DT state, u_t^* denotes the control action applied in the preceding iteration, and N represents the total number of chronological observations.

4. Results

The research aims to develop and evaluate a DT-driven framework for smart green building energy management; experiments are required to validate prediction, synchronization, optimization, energy efficiency, and control performance.

4.1. Experimental Configuration Setup

The experiments evaluate the proposed DT-driven framework for building-state prediction, energy estimation, multi-objective optimization, and adaptive control using the Room Occupancy Dataset. The framework is implemented on Windows 11 using Python 3.14.5, PyTorch 2.10.0+cpu, Scikit-learn 1.8.0, Pandas 2.3.3, NumPy 2.3.4, Matplotlib 3.x, and OpenPyXL. It integrates chronological preprocessing, DT synchronization, KF refinement, Res-DL-PEM prediction, SHAP explainability, state-transition modeling, energy-surrogate regression, MOPSO optimization, and closed-loop simulation, requiring 90.43 s overall. The measured 169.15 ms average latency represents the computational execution time of the simulation-based DT control loop on the

specified CPU environment, rather than end-to-end physical sensor-to-actuator or building-control latency.

4.1.1 Hyperparameter Configuration Strategy

Res-DL-PEM uses five observations, 128–64–32 Dense layers, ReLU, residual connections, 0.30 dropout, Adam, weighted BCE, while MOPSO uses 30 particles, 80 iterations, and 100 solutions.

Table 2. Hyperparameter Configuration and Experimental Settings

Component	Parameter	Configured Value
Room Occupancy Dataset	Dataset observations	2,665 historical timestamped observations
	Environmental variables	Temperature, Humidity, Light, CO_2, Humidity Ratio, Occupancy
	Data type	Historical dataset observations
	Data ordering	Chronological
	Training samples	1,332
	Validation samples	484
	Testing samples	849
	Preprocessing	Missing-value imputation
	Imputation strategy	Training median
	Outlier filtering	IQR (1.5 × IQR)
	Normalization	MinMaxScaler [0, 1]
Digital Twin (DT)	State dimension	6
	State variables	Temperature, Humidity, Light, CO_2, Humidity Ratio, Occupancy
	Synchronization	Historical observation → DT state
	Control variables	HVAC temperature setpoint, lighting level, ventilation rate (simulation-generated)
	Control-response environment	Simulation-based DT
Kalman Filter (KF)	Synchronization	Sensor observation to DT state
	Process noise covariance	$1 \times 10^{-4} \times I_6$
	Measurement noise covariance	$5 \times 10^{-3} \times I_6$
Res-DL-PEM	State dimension	6
	Input dimension	25
	Sequence length	5
	Hidden layer units	[128, 64, 32]
	Activation	ReLU
	Residual connection	Linear projection / skip connection
	Dropout rate	0.30
	Optimizer	Adam
	Learning rate	4×10^{-4}
	Weight decay	1×10^{-4}
	Learning-rate scheduler	ReduceLROnPlateau
	Scheduler factor	0.5
	Scheduler patience	7
	Minimum learning rate	1×10^{-5}
	Loss function	BCEWithLogitsLoss
	Class-imbalance weight	1.40
	Batch size	32
	Maximum epochs	100
	Early-stopping patience	20

State-Transition Model	Estimator	GradientBoostingRegressor with separate regression outputs for each DT state variable
	Number of estimators	200
	Maximum tree depth	4
	Learning rate	0.05
	Subsample ratio	0.80
	Input	Refined DT state, predicted occupancy, control variables, simulated disturbances
	Output	Six next-step simulated DT state variables
	Response type	Simulation-based control response, not measured building response
Energy Surrogate Model	Estimator	GradientBoostingRegressor
	Number of estimators	300
	Maximum tree depth	5
	Training samples	5,000 simulation-generated control-response samples
	Input	DT states, predicted occupancy, HVAC setpoint, lighting level, ventilation rate
	Output	Simulation-based estimated building energy consumption
	Data source	Simulation-based Digital Twin, not physical energy measurements
Simulation-Based Control Responses	Generated responses	Next-state DT variables, energy consumption, CO_2 concentration, thermal discomfort
	HVAC setpoint range	20.3–22.6 °C
	Lighting range	0–0.2
	Ventilation range	0.1–1.0
	Generation procedure	Control–state combinations are applied within the simulation-based DT to generate simulated next-state and building-response values
	Data source	Simulation-based Digital Twin
	Physical-data status	Simulation-generated responses rather than measured intervention outcomes
	Data partitioning	5,000 samples used directly for energy-response learning without a separate train-validation-test partition
	Validation type	Simulation-based validation, not experimental building validation
A-MOPSO	Swarm size	30
	Maximum iterations	80
	Pareto archive size	100
	Inertia weight	[0.4, 0.9]
	Cognitive coefficient (C_1)	1.5
	Social coefficient (C_2)	1.5
	Decision variables	HVAC setpoint, lighting level, ventilation rate
	Objectives	Energy, CO_2, thermal discomfort
Explainability	Method	SHAP Kernel Explainer
	Explanation level	Global and local
Execution	Platform	Windows 11
	Python	3.14.5
	PyTorch	2.10.0+cpu
	Scikit-learn	1.8.0
	Pandas	2.3.3
	NumPy	2.3.4
	Matplotlib	3.x
	OpenPyXL	Installed
	End-to-end execution time	90.43 s
	Average simulation-loop latency	169.15 ms
	Maximum simulation-loop latency	235.51 ms

Table 2 summarizes the experimental parameters for preprocessing, DT synchronization, KF refinement, Res-DL-PEM prediction, state-transition modeling, energy-surrogate estimation, A-MOPSO optimization, and SHAP explainability. The 2,665 records are historical observations, while 5,000 samples are independently simulation-generated.

4.2. Performance Evaluation

The proposed framework is evaluated in a simulation-based Digital Twin environment using computational latency, control response, energy, CO_2, and thermal discomfort, without physical sensor or actuator validation. The evaluation considers varying occupancy levels and building operating conditions to examine the behavior of the integrated control workflow under different simulated demands. Low, moderate, and high occupancy conditions are examined alongside variations in HVAC setpoints, lighting levels, and ventilation rates. In all scenarios, Res-DL-PEM estimates the occupancy, the Digital Twin maintains and updates the building state, the energy-surrogate model estimates the energy response and DT-A-MOPSO determines the control settings. Energy consumption, CO_2 concentration, thermal discomfort, and computational latency are compared across scenarios using consistent evaluation measures.

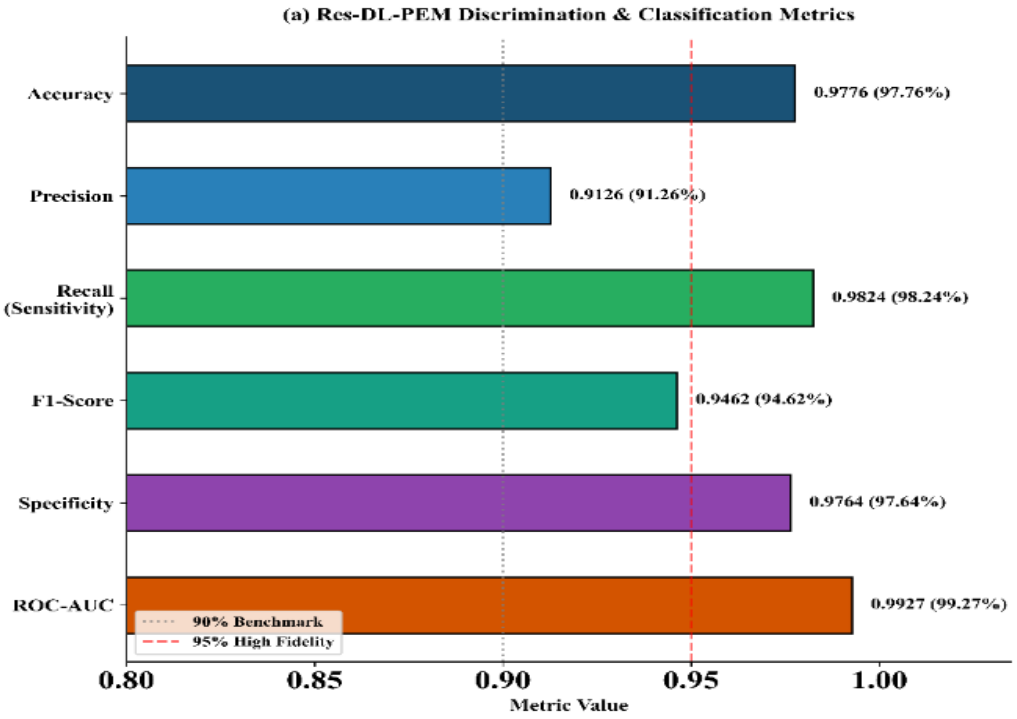


Figure 4. Performance Evaluation of the Proposed Res-DL-PEM for Next-Step Occupancy Prediction

Figure 4 evaluates Res-DL-PEM occupancy prediction using classification and loss metrics within the simulation-based Digital Twin environment. The model achieves 0.9776 accuracy, 0.9126 precision, 0.9824 recall, 0.9462 F1-score, 0.9764 specificity, 0.9927 ROC-AUC, and 0.0794 BCE loss, demonstrating reliable computational occupancy prediction.

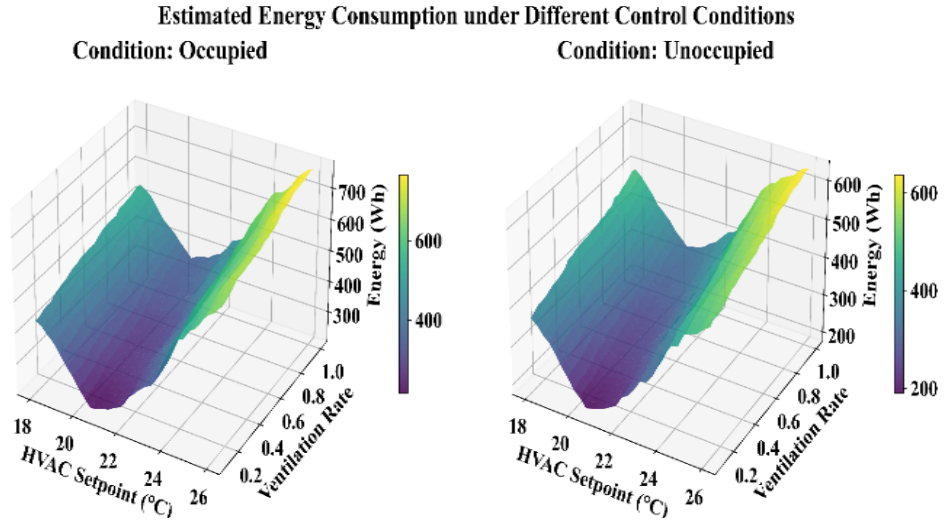


Figure 5. Estimated Building Energy Consumption under Occupied and Unoccupied Conditions

Figure 5 illustrates simulation-based estimated energy consumption under occupied and unoccupied conditions. The surfaces relate HVAC temperature setpoints and ventilation rates to estimated energy consumption, showing their variation with occupancy and supporting evaluation of candidate operating conditions within the simulation-based Digital Twin optimization process.

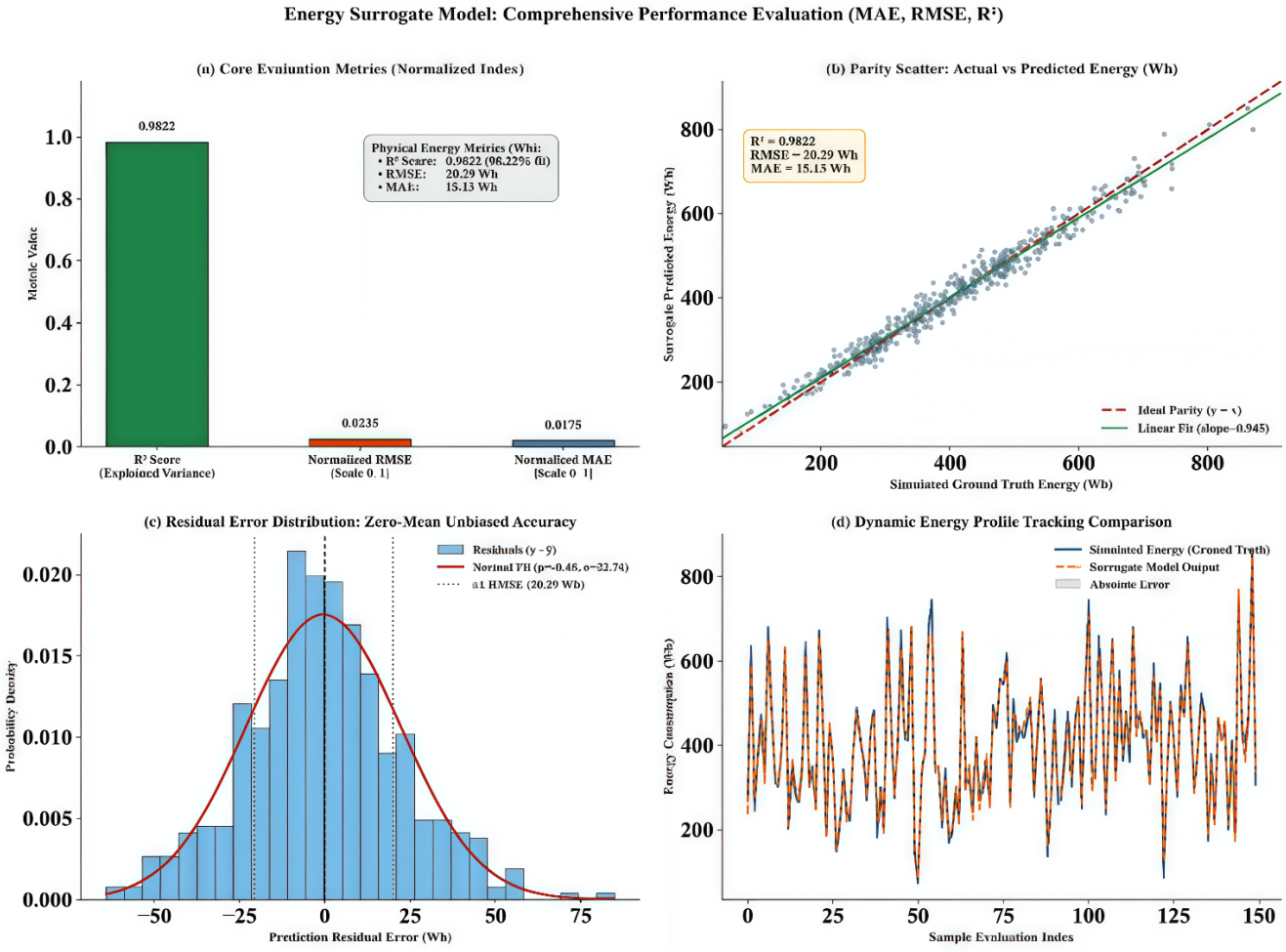


Figure 6. Comprehensive Performance Evaluation of the Proposed Data-Driven Energy Surrogate Model

Figure 6 evaluates the energy surrogate model within the simulation-based Digital Twin. (a) reports R^2 = 0.9822, normalized RMSE = 0.0235, and normalized MAE = 0.0175. (b) shows simulated and predicted energy, with RMSE = 20.29 Wh and MAE = 15.13 Wh. (c) presents residuals near zero, indicating limited bias, while (d) demonstrates close profile tracking.

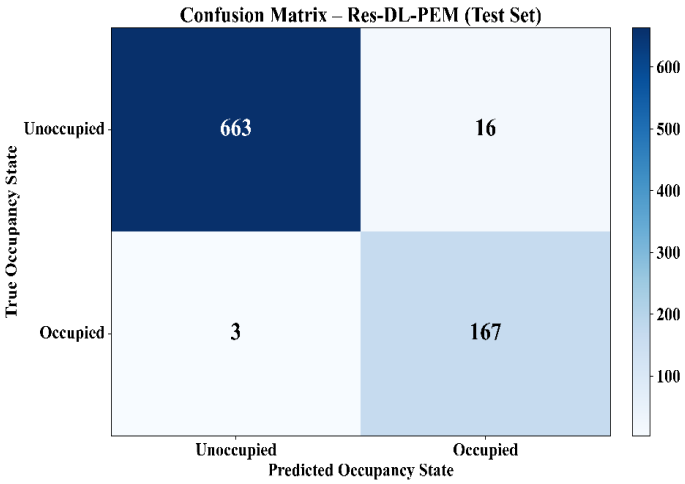


Figure 7. Confusion Matrix of the Proposed Res-DL-PEM for Next-Step Occupancy Prediction

Figure 7 presents the Res-DL-PEM confusion matrix on the independent test set. The test set contains 849 observations, comprising 663 correctly classified unoccupied samples, 167 correctly classified occupied samples, 16 unoccupied samples classified as occupied, and 3 occupied samples classified as unoccupied. The 2,665 historical observations were partitioned chronologically into training, validation, and testing subsets, with 849 observations reserved for independent testing. The remaining 1,816 observations were used for training and validation.

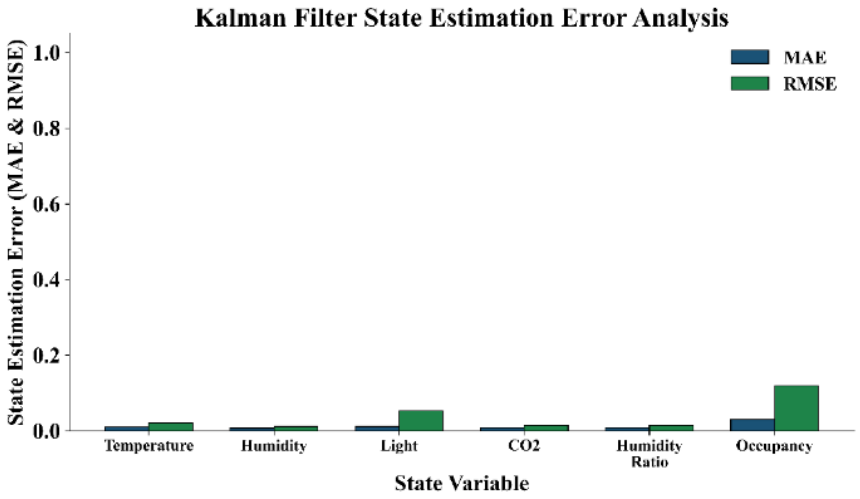


Figure 8. KF State Estimation Error Analysis for DT State Refinement

Figure 8 compares MAE and RMSE for the observed DT state variables after Kalman refinement. Temperature, Humidity, Light, CO_2, Humidity Ratio, and Occupancy achieve MAE/RMSE values of 0.01176/0.02265, 0.00833/0.01321, 0.01301/0.05552, 0.00981/0.01784, 0.00865/0.01577, and 0.03419/0.12605, respectively, indicating refined state estimates within the simulation-based DT environment.

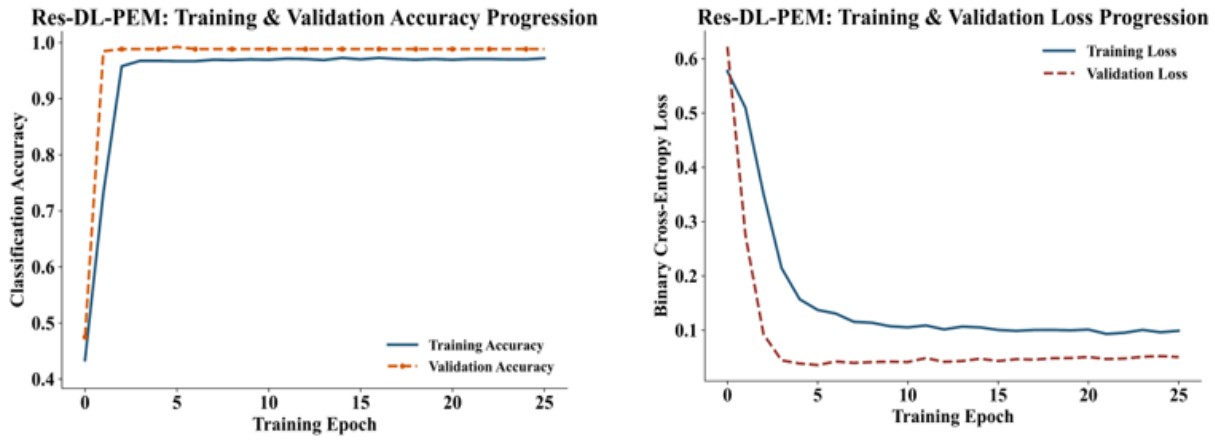


Figure 9:Training and Validation Accuracy and Loss of the Proposed Res-DL-PEM

Figure 9 presents Res-DL-PEM training and validation across 25 epochs within the simulation-based Digital Twin framework. Accuracy converges to approximately 0.97 for training and 0.99 for validation, while binary cross-entropy loss decreases to 0.10 and 0.04, respectively, indicating stable computational convergence during occupancy prediction.

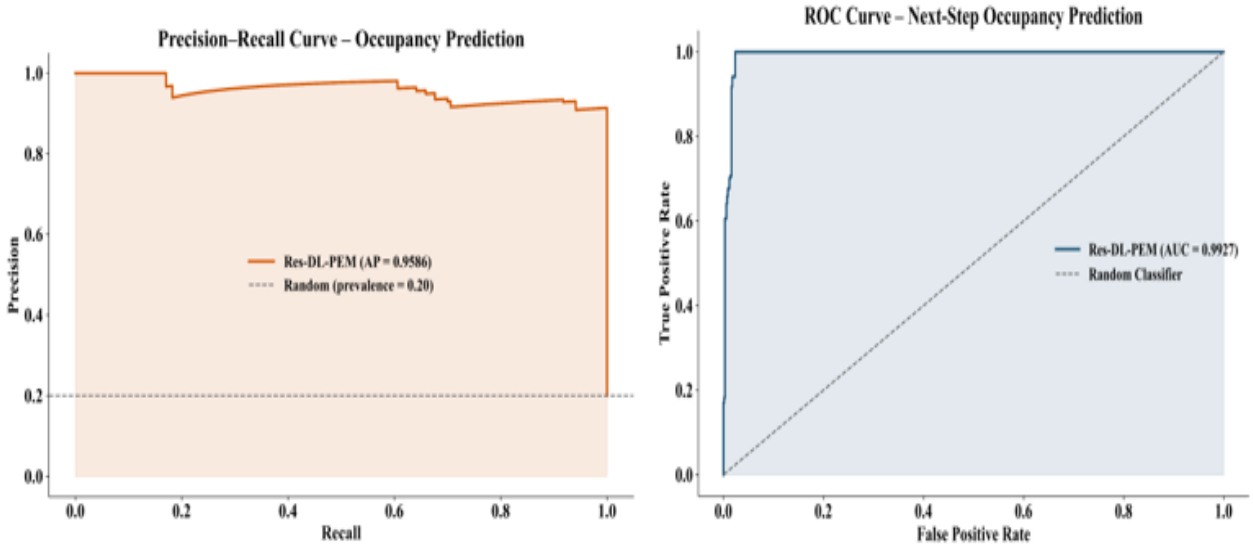


Figure 10. Precision–Recall and ROC Curves for Res-DL-PEM Occupancy Prediction

Figure 10 evaluates Res-DL-PEM occupancy discrimination within the simulation-based Digital Twin. The model achieves AP = 0.9586 against a random-prevalence baseline of 0.20 and ROC-AUC = 0.9927, indicating strong precision–recall performance and effective computational separation of occupied and unoccupied states.

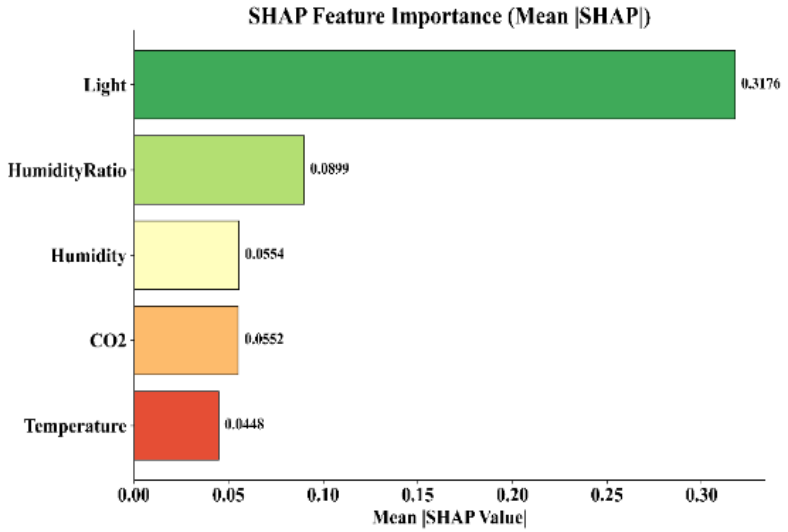


Figure 11. SHAP-Based Global Feature Importance for Res-DL-PEM Occupancy Prediction

Figure 11 presents SHAP-based global feature contributions for Res-DL-PEM occupancy prediction within the simulation-based Digital Twin. Light has the highest mean absolute SHAP value (0.3176), followed by Humidity Ratio (0.0899), Humidity (0.0554), CO_2 (0.0552), and Temperature (0.0448), indicating their relative contributions to computational occupancy prediction.

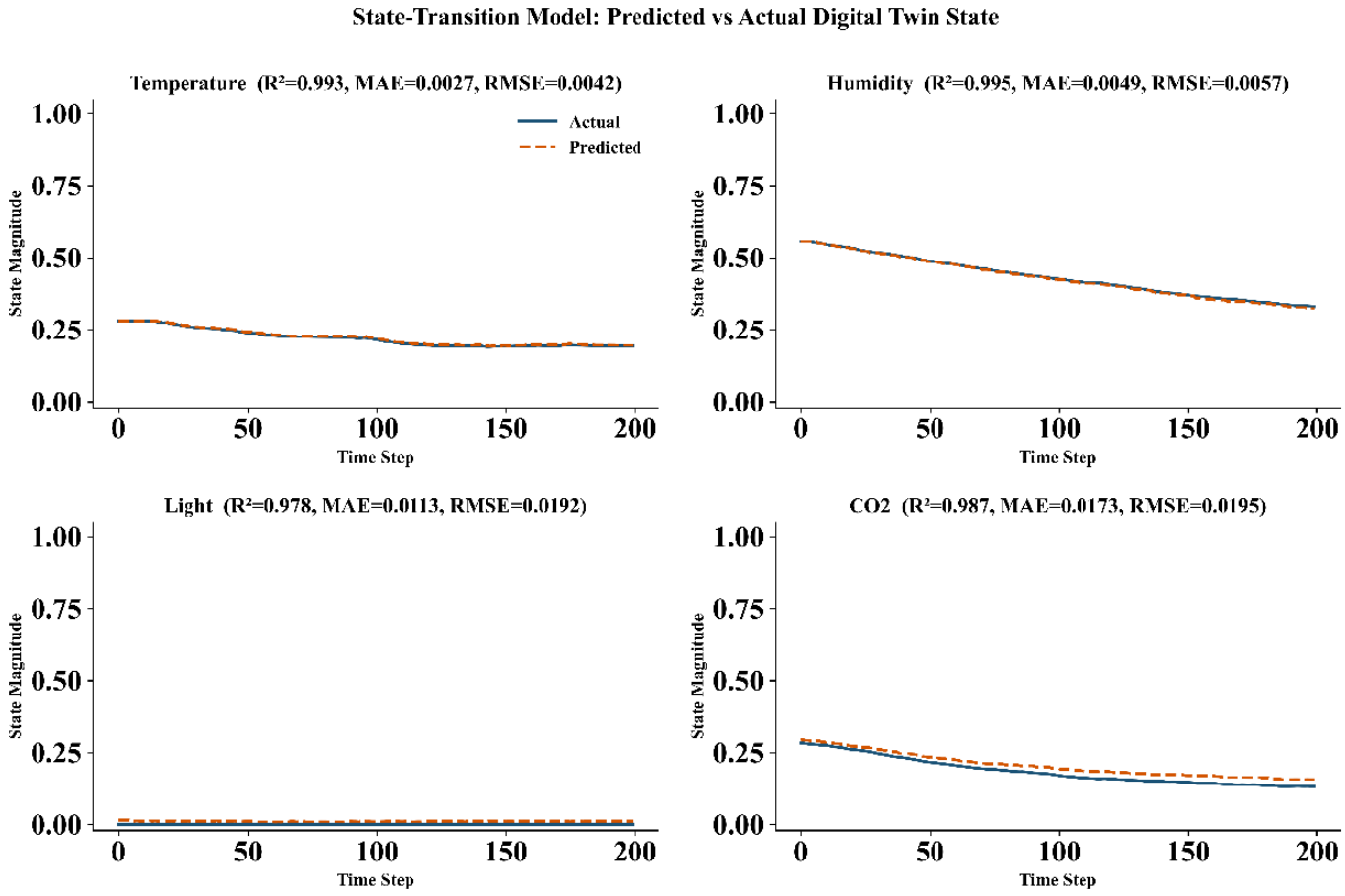


Figure 12. State-Transition Model: Predicted vs. Actual DT State

Figure 12 compares actual and predicted DT states within the simulation-based Digital Twin. Temperature achieves R^2 = 0.993, MAE = 0.0027, RMSE = 0.0042; Humidity R^2 = 0.995, MAE = 0.0049, RMSE = 0.0057; Light R^2 = 0.978, MAE = 0.0113, RMSE = 0.0192; and CO_2 R^2 = 0.987, MAE = 0.0173, RMSE = 0.0195, indicating close agreement between simulated and predicted states.

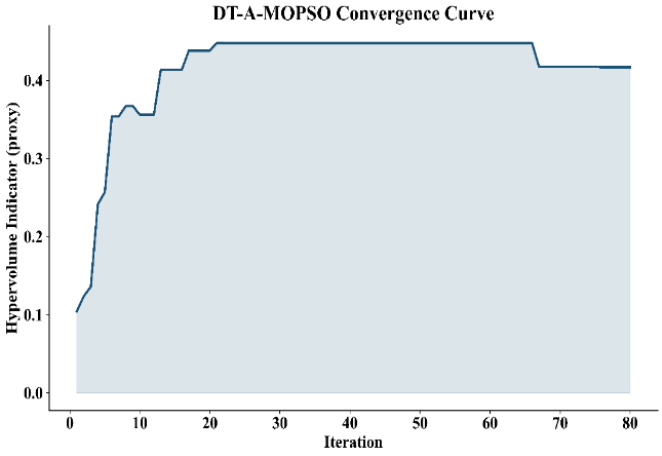


Figure 13. DT-A-MOPSO Convergence Curve

Figure 13 presents DT-A-MOPSO convergence across 80 iterations within the simulation-based Digital Twin. Hypervolume increases from 0.104 at iteration 1 to 0.35 at iteration 6, reaching 0.447 at iteration 21, remaining stable until iteration 66, and subsequently declining to 0.416, reflecting the simulated optimization trajectory.

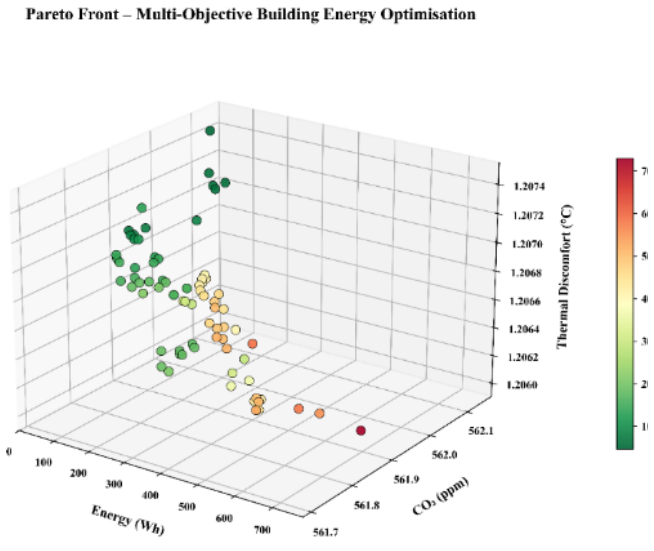


Figure 14. Pareto Front of Multi-Objective Building Energy Optimisation

Figure 14 presents Pareto-optimal solutions from DT-A-MOPSO for simultaneous energy, CO_2 concentration, and simulation-based thermal-discomfort optimization. Energy consumption spans approximately 100–730 Wh, while CO_2 concentration remains around 561.7–562.1 ppm and thermal discomfort varies between 1.2060–1.2074°C, representing predicted indoor-temperature deviation from the desired thermal setpoint.

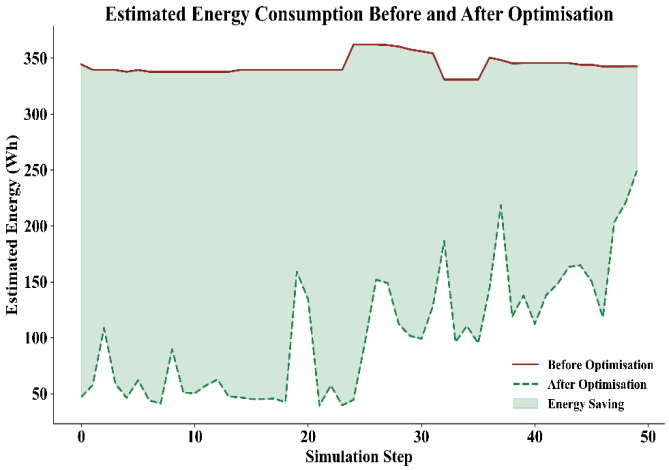


Figure 15. Estimated Energy Consumption Before and After Optimisation

Figure 15 compares simulation-based estimated building energy consumption before and after DT-A-MOPSO across 50 simulation steps. Estimated consumption decreases from approximately 330–363 Wh before optimization to 40–250 Wh after optimization, with reductions varying across simulation conditions and reaching approximately 320 Wh. These values represent simulation-based estimated reductions rather than measured physical building energy savings.

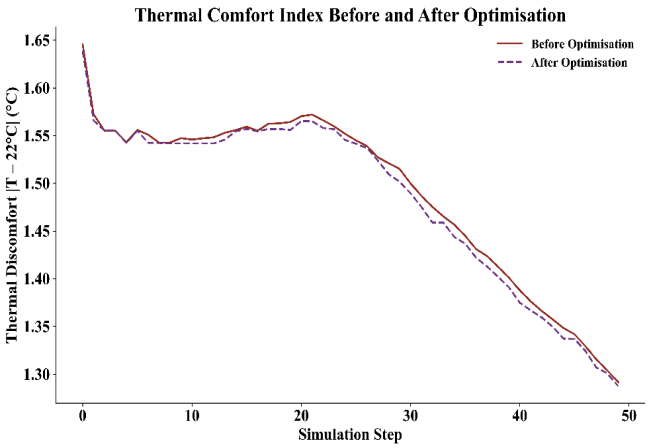


Figure 16. Thermal Discomfort Index Before and After Optimisation

Figure 16 compares simulation-based thermal discomfort before and after DT-A-MOPSO across 50 steps. Discomfort decreases from 1.65°C to 1.29°C before optimization and from 1.64°C to 1.29°C after optimization, indicating a slight reduction following optimization.

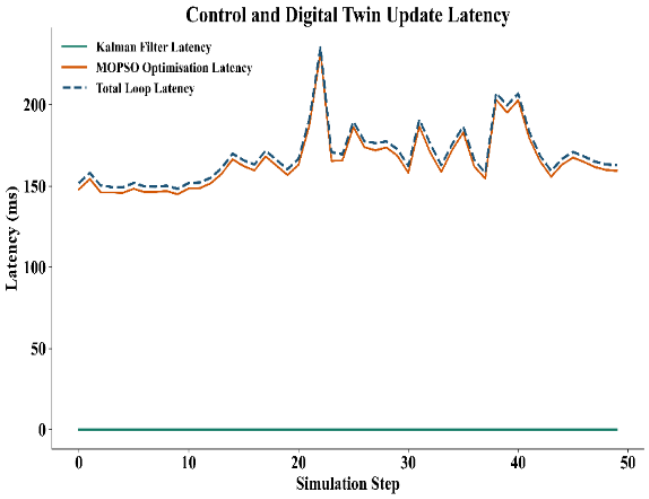


Figure 17. Control and DT Update Latency

Figure 17 reports computational latency across 50 simulation steps. KF latency remains near 0 ms, MOPSO ranges from 145–235 ms, and total loop latency from 148–238 ms, representing simulation-based execution time rather than physical sensor-to-actuator response latency.

The reported latency represents the complete simulation-based control cycle, comprising Kalman Filter state estimation, occupancy and building-state prediction, energy estimation, DT-A-MOPSO optimization, and Digital Twin updating using the simulated response and the next chronological observation. The average and maximum loop latency was 169.15ms and 235.51ms, respectively, for 50 simulation steps. The entire time to perform experiments was 90.43 s. These measurements are based on execution of the computation in the simulation realm and do not account for delays due to physical communication of the sensors, actuation, or interactive response of the building.

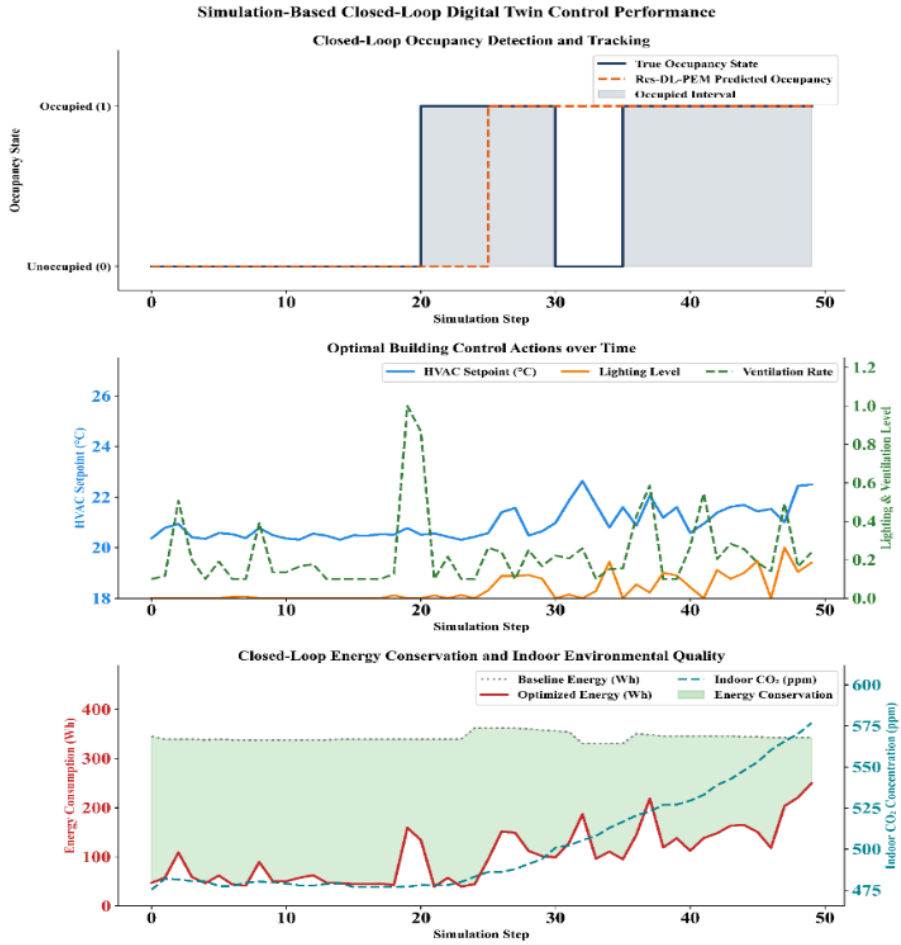


Figure 18. Simulation-Based Closed-Loop DT Control Performance

Figure 18 presents simulation-based closed-loop DT performance across occupancy, simulated controls, estimated energy, and CO_2 response. HVAC setpoints range 20.3–22.6°C, lighting 0–0.2, ventilation 0.1–1.0, energy 40–250 Wh, and CO_2 475–580 ppm.

4.3. Ablation Study

The ablation study evaluates the contribution of individual framework components by systematically removing selected modules and comparing their effects on prediction, state estimation, energy optimization, and closed-loop control performance.

Table 3. Ablation Study of the Proposed DT Framework

Configuration	Mean Energy (Wh)	Energy Saving (%)	Mean CO_2 (ppm)	Thermal Discomfort (°C)	DT Sync RMSE	Mean Latency (ms)
Full DT Framework	102.72	70.09	501.53	1.485	0.0118	173.73
Without Kalman Filter	130.00	62.14	515.08	1.447	0.0384	169.09
Without Res-DL-PEM	134.52	60.83	501.53	1.485	0.0118	165.61
Without State-Transition Model	182.64	46.82	501.56	1.484	0.0118	33.69
Without Energy Surrogate	389.33	−13.37	501.50	1.485	0.0118	138.42
Without DT-A-MOPSO	416.46	−21.27	501.73	1.485	0.0118	5.48

Table 3 shows that removing key modules increased energy consumption: 130.00 Wh without KF, 134.52 Wh without Res-DL-PEM, and 182.64 Wh without the State-Transition Model, compared with 102.72 Wh for the complete framework.

4.4. Comparative Analysis

This section compares the proposed framework with existing approaches to assess differences in prediction accuracy, energy efficiency, environmental conditions, thermal comfort, and control performance under comparable evaluation settings.

Table 4. Baseline Comparison of Next-Step Occupancy Prediction Models

Model	Accuracy	Loss	Precision	Recall	F1-Score	Specificity	AUC-ROC
Proposed Res-DL-PEM	0.9776	0.0794	0.9126	0.9824	0.9462	0.9764	0.9927
Logistic Regression	0.9694	0.1320	0.9037	0.9751	0.9380	0.9740	0.9868
Random Forest	0.9670	0.3703	0.9080	0.9294	0.9186	0.9764	0.9765
KNN (k=5)	0.9764	0.7084	0.9121	0.9765	0.9432	0.9764	0.9817
MLP	0.9717	0.3004	0.9061	0.9740	0.9388	0.9752	0.9908

Table 4 compares Res-DL-PEM with four baseline models using occupancy prediction metrics. The models achieve comparable classification performance, while Res-DL-PEM records the lowest loss and highest AUC-ROC, indicating strong predictive performance.

Table 5. Performance Evaluation Comparison with Existing Occupancy Prediction Models

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Res-DL-PEM (Proposed)	**0.9776**	**0.9126**	**0.9824**	**0.9462**	**0.9927**
CNN [32]	0.9342	0.9296	0.9342	0.9222	0.9614

LSTM[33]	0.9100	0.9000	0.9200	0.9100	0.9500
XGBoost [34]	0.9580	0.9510	0.9560	0.9530	0.9780

Table 5 compares the proposed Res-DL-PEM with established deep-learning occupancy models reported in existing studies, covering accuracy, precision, recall, F1-score, and AUC-ROC to contextualize its predictive performance.

4.5. Discussion

A simulation-based closed-loop framework integrated DT monitoring, prediction, optimization, and control. Res-DL-PEM achieved 0.9776 accuracy and 0.9927 ROC-AUC, while the energy surrogate achieved R^2=0.9822. DT-A-MOPSO reached 0.447 hypervolume, reducing estimated energy from 330–363 Wh to 40–250 Wh. Ablation confirmed increased consumption across module removals. The simulation results highlight the potential of integrating Digital Twins, predictive analytics, and optimization to improve building energy management. This ICT-enabled approach supports energy-efficient operation and illustrates the relevance of intelligent control to sustainable Energy Web ecosystems.

Limitations

The framework was evaluated using historical data and simulation-based building responses rather than physical deployment. Thus, real-world disturbances, sensor failures, equipment constraints, occupant variability, and communication delays were not fully captured. Generalization across buildings and climates requires further validation. Physical real-time deployment with sensors, actuators, HVAC, lighting, and ventilation remains future work.

5. Conclusion & Future Work

The proposed Digital Twin (DT)-driven framework integrates Kalman Filter-based state estimation, Res-DL-PEM occupancy prediction, SHAP-based explainability, building-state transition modeling, energy-surrogate estimation, and DT-A-MOPSO optimization within a closed-loop simulation workflow. The proposed framework demonstrates the potential of integrating Digital Twin technology, predictive intelligence, and multi-objective optimization for energy-efficient building control. The simulation results push forward the physical validation to sustainable, connected and energy-aware digital infrastructure. Occupancy prediction and state-transition prediction are used to guide the control decision, SHAP enhances the interpretability of occupancy predictions, and multi-objective optimization involves balancing energy use, CO_2 concentration, and thermal discomfort. The results presented here are simulated, and they show the benefit from the interaction of these parts in the prospect of predictive and explainable building energy management. But the reported results are simulation based and need to be physically validated for their practical use. Future work will incorporate real-time sensor streams and physical DT implementations to evaluate control performance under operational conditions. Further investigations will explore transferability by building type, climate, and occupancy patterns, and further optimize to include renewable energy availability, equipment constraints, dynamic pricing, and carbon intensity. The deployment and HIL testing of the edge will also be discussed to evaluate computational latency, robustness, reliability, and adaptability.

Declarations

Data Availability

https://www.kaggle.com/datasets/sachinsharma1123/room-occupancy

Author Contribution

Lihua Zhang: Conceptualization, methodology, data analysis, investigation, writing—original draft, writing—review and editing, and project administration. The author read and approved the final manuscript.

Acknowledgement

The author would like to thank the School of Resources, Environment and Building Engineering, Chifeng University, for its academic support during this research.

References

[1] Abid MT, Abu-shaikha M, Al-Bdour H, Alkhadrawi S. Designing adaptive smart buildings: an RNN and Migrating Birds Optimization approach for occupancy prediction. Asian J Civ Eng. 2023;25(3):2653-2663. doi:10.1007/s42107-023-00935-4.

[2] Hakawati B, Mousa A, Draidi F. Smart energy management in residential buildings: the impact of knowledge and behavior. Sci Rep. 2024;14(1):1702. doi:10.1038/s41598-024-51638-y.

[3] Hodavand F, Ramaji I, Sadeghi N, Zandi Goharrizi S. Digital Twin-enabled framework for intelligent monitoring and anomaly detection in multi-zone building systems. Buildings. 2025;15(22):4030. doi:10.3390/buildings15224030.

[4] Hwang J, Kim J, Yoon S. DT-BEMS: digital twin-enabled building energy management system for information fusion and energy efficiency. Energy. 2025;326:136162. doi:10.1016/j.energy.2025.136162.

[5] Khan SU, Khan N, Ullah FUM, Kim MJ, Lee MY, Baik SW. Towards intelligent building energy management: AI-based framework for power consumption and generation forecasting. Energy Build. 2023;279:112705. doi:10.1016/j.enbuild.2022.112705.
[6] Natarajan Y, SPKR, Wadhwa G, Choi Y, Chen Z, Lee DE, et al. Enhancing building energy efficiency with IoT-driven hybrid deep learning models for accurate energy consumption prediction. Sustainability. 2024;16(5):1925. doi:10.3390/su16051925.
[7] Poosapadi D, Devi GM, Padmapriya G, Srinivasan K, Logeshwaran T, Rani JJA, et al. Energy efficient smart control strategies: applying machine learning to enhance efficiency in eco-friendly building management for renewable energy systems. Sustain Comput Inform Syst. 2026;50:101315. doi:10.1016/j.suscom.2026.101315.
[8] Saeed MA, Eladl AA, Alhasnawi BN, Motahhir S, Nayyar A, Shah MA, et al. Energy management system in smart buildings-based coalition game theory with fog platform and smart meter infrastructure. Sci Rep. 2023;13(1):2023. doi:10.1038/s41598-023-29209-4.
[9] Mekala A, Baishya BK, Vigneswara Rao KT, Vidhate DA, Drave VA, Prasanth PV. Big data and machine learning framework for temperature forecasting. EAI Endorsed Trans Energy Web. 2023;10:1-6. doi:10.4108/eai.4195.
[10] Sheela MS, Gopalakrishnan S, Begum IP, Hephzipah JJ, Gopianand M, Harika D. Enhancing energy efficiency with smart building energy management system using machine learning and IoT. Babylon J Mach Learn. 2024:80-88. doi:10.58496/BJML/2024/008.
[11] Zhang S, Kong C. Deep learning and multi-objective optimization for real-time occupancy-based energy control in smart buildings. Sci Rep. 2025;15(1):39466. doi:10.1038/s41598-025-23134-4.
[12] Ajitha A, Akhilesh G, Rajkumar T, Radhika S, Goel S. Design and implementation of smart home energy management system for Indian residential sector. Energy Convers Manag. 2024;307:118369. doi:10.1016/j.enconman.2024.118369.
[13] Alijoyo FA. AI-powered deep learning for sustainable Industry 4.0 and Internet of Things: enhancing energy management in smart buildings. Alexandria Eng J. 2024;104:409-422. doi:10.1016/j.aej.2024.07.110.
[14] Mousavi S, Gheibi M, Wacławek S, Behzadian K. A novel smart framework for optimal design of green roofs in buildings conforming with energy conservation and thermal comfort. Energy Build. 2023;291:113111. doi:10.1016/j.enbuild.2023.113111.
[15] Sagar, Arya Y, Singhal P. Optimized building energy management system of a residential green building. IETE J Res. 2025;71(10):3520-3537. doi:10.1080/03772063.2025.2512905.
[16] Hadjidemetriou L, Stylianidis N, Englezos D, Papadopoulos P, Eliades D, Timotheou S, et al. A digital twin architecture for real-time and offline high granularity analysis in smart buildings. Sustain Cities Soc. 2023;98:104795. doi:10.1016/j.scs.2023.104795.
[17] Krishnan P, Prabu AV, Loganathan S, Routray S, Ghosh U, AL-Numay M. Analyzing and managing various energy-related environmental factors for providing personalized IoT services for smart buildings in smart environment. Sustainability. 2023;15(8):6548. doi:10.3390/su15086548.
[18] Mohseni SR, Zeitouni MJ, Parvaresh A, Abrazeh S, Gheisarnejad M, Khooban MH. FMI real-time co-simulation-based machine deep learning control of HVAC systems in smart buildings: digital-twins technology. Trans Inst Meas Control. 2023;45(4):661-673. doi:10.1177/01423312221119635.
[19] Eneyew DD, Capretz MAM, Bitsuamlak GT. Continuous model calibration framework for smart-building digital twin: a generative model-based approach. Appl Energy. 2024;375:124080. doi:10.1016/j.apenergy.2024.124080.
[20] Saleem MU, Shakir M, Usman MR, Bajwa MHT, Shabbir N, Shams Ghahfarokhi P, et al. Integrating smart energy management system with Internet of Things and cloud computing for efficient demand side management in smart grids. Energies. 2023;16(12):4835. doi:10.3390/en16124835.
[21] Afzal S, Ziapour BM, Shokri A, Shakibi H, Sobhani B. Building energy consumption prediction using multilayer perceptron neural network-assisted models: comparison of different optimization algorithms. Energy. 2023;282:128446. doi:10.1016/j.energy.2023.128446.
[22] Mounir N, Ouadi H, Jrhilifa I. Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system. Energy Build. 2023;288:113022. doi:10.1016/j.enbuild.2023.113022.
[23] Selvaraj R, Kuthadi VM, Baskar S. Smart building energy management and monitoring system based on artificial intelligence in smart city. Sustain Energy Technol Assess. 2023;56:103090. doi:10.1016/j.seta.2023.103090.
[24] Ibrahim O, Bakare MS, Amosa TI, Otuoze AO, Owonikoko WO, Ali EM, et al. Development of fuzzy logic-based demand-side energy management system for hybrid energy sources. Energy Convers Manag X. 2023;18:100354. doi:10.1016/j.ecmx.2023.100354.
[25] Liang X, Chen K, Chen S, Zhu X, Jin X, Du Z. IoT-based intelligent energy management system for optimal planning of HVAC devices in net-zero emissions PV-battery building considering demand compliance. Energy Convers Manag. 2023;292:117369. doi:10.1016/j.enconman.2023.117369.
[26] Wang Q, Yin Y, Chen Y, Liu Y. Carbon peak management strategies for achieving net-zero emissions in smart buildings: advances and modeling in digital twin. Sustain Energy Technol Assess. 2024;64:103661. doi:10.1016/j.seta.2024.103661.
[27] Almadhor A, Ghazouani N, Bouallegue B, Kryvinska N, Alsubai S, Krichen M, et al. Digital twin-based deep learning framework for personalized thermal comfort prediction and energy efficient operation in smart buildings. Sci Rep. 2025;15(1):24654. doi:10.1038/s41598-025-10086-y.
[28] Kazemi Naeini H, Shomali R, Pishahang A, Hasanzadeh H, Asadi S, Gholizadeh Lonbar A. PINN-DT: optimizing energy consumption in smart building using hybrid physics-informed neural networks and digital twin framework with blockchain security. Sensors. 2025;25(19):6242. doi:10.3390/s25196242.
[29] Piras G, Agostinelli S, Muzi F. Smart buildings and digital twin to monitoring the efficiency and wellness of working environments: a case study on IoT integration and data-driven management. Appl Sci. 2025;15(9):4939. doi:10.3390/app15094939.
[30] Hosamo HH, Nielsen HK, Kraniotis D, Svennevig PR, Svidt K. Improving building occupant comfort through a digital twin approach: a Bayesian network model and predictive maintenance method. Energy Build. 2023;288:112992. doi:10.1016/j.enbuild.2023.112992.
[31] Room occupancy [Internet]. Kaggle; 2026 [cited 2026 Sep 22]. Available from: https://www.kaggle.com/datasets/sachinsharma1123/room-occupancy.

[32] Sayed AN, Bensaali F, Himeur Y, Dimitrakopoulos G, Varlamis I. Enhancing building sustainability: a digital twin approach to energy efficiency and occupancy monitoring. Energy Build. 2025;328:115151. doi:10.1016/j.enbuild.2024.115151.
[33] Almadhor A, Alsubai S, Kryvinska N, Ghazouani N, Bouallegue B, Al Hejaili A, et al. A synergistic approach using digital twins and statistical machine learning for intelligent residential energy modelling. Sci Rep. 2025;15(1):26088. doi:10.1038/s41598-025-09760-y.
[34] Hindasi YA, Ilbeigi M, Rezaei M, Zare Mohammadians F, Jahanshahi E, Barjani A, et al. Toward explainable digital twins for indoor environmental risk prediction using open IoT sensor data. Results Eng. 2026;32:112313. doi:10.1016/j.rineng.2026.112313.