Prediction of HVAC Operational Variables Using Recurrent Neural Networks for Advanced Control Strategies

Merve KURU ERDEM [1], Osman GOKALP[2*], Gulben CALIS[3],

[1]Ph.D., Ege University, Faculty of Engineering, Department of Civil Engineering, 35040, Bornova, Izmir, Türkiye (mervekuru25@gmail.com)

[2]Assistant Professor, Izmir Institute of Technology, Department of Computer Engineering, 35430, Urla, Izmir, Türkiye (osmangokalp@iyte.edu.tr)

[3] Professor, Ege University, Faculty of Engineering, Department of Civil Engineering, 35040, Bornova, Izmir, Türkiye (gulben.calis@ege.edu.tr)

**Corresponding author*

ABSTRACT

Heating, Ventilation, and Air Conditioning (HVAC) systems are major energy consumers in buildings, making accurate prediction of their operational variables essential for improving energy efficiency and occupant comfort. This study compares three recurrent neural network (RNN) architectures, namely standard RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) for multi-input, multi-output prediction of nine HVAC variables using high-resolution data from a real office building (2018–2020). A two-stage hyperparameter optimization varying network depth, width, learning rate, and batch size was applied. Results show that a shallow architecture with a single hidden layer of 16 nodes, learning rate of 0.001, and batch size of 64 offers the best balance between accuracy and generalization. While the standard RNN showed higher errors and lower stability, both LSTM and GRU performed well. The GRU achieved the lowest average rank (4.2) and mean validation loss (0.0049) across cross-validation folds, demonstrating superior consistency and reliability, and was selected for final evaluation. On the holdout test set, GRU consistently outperformed the other models, achieving $R^2 > 0.81$ for seven of nine variables, with a peak R^2 of 0.99 for mean indoor air temperature and an overall average R^2 of 0.80. These results highlight GRU's advantages in accuracy, stability, and efficiency, suggesting its suitability for intelligent building control. The study's scope is limited to three RNN-based architectures with a constrained hyperparameter search; future work should extend comparisons to broader model families and datasets.

Keywords: HVAC operation, HVAC control strategy, recurrent neural network, LSTM, GRU.

1. INTRODUCTION

Heating, Ventilation, and Air Conditioning (HVAC) systems have broad implications for both energy efficiency and occupant comfort, making it a cornerstone of sustainable building operations [1–3]. Accurate HVAC models are capable of representing the complex, nonlinear, and time-dependent dynamics of HVAC systems that traditional regression or single-variable time series approaches often fail to capture [4,5]. Leveraging these models enables the development of predictive and adaptive control strategies that significantly reduce unnecessary energy consumption, minimize peak load demands, and improve overall system efficiency [6,7]. This not only lowers operating costs but also contributes to broader sustainability objectives such as reducing carbon emissions and aligning with net-zero energy targets [8,9]. Beyond the energy dimension, effective HVAC modeling allows for more precise prediction and regulation of indoor environmental parameters such as air temperature, relative humidity, and airflow, thereby directly improving occupant comfort [10,11]. Stable and consistent thermal conditions mitigate discomfort caused by fluctuations [12,13], while advanced multivariable models provide more personalized comfort management that addresses diverse occupant needs [14,15]. These combined benefits also extend to health and productivity, as well-regulated indoor environments have been shown to enhance cognitive performance and overall quality of life [16,17]. Taken together, these outcomes highlight that effective HVAC modeling is not merely a technical practice but a strategic necessity, shaping the future of energy-efficient, comfortable, and resilient buildings [18,19].

Subsequently, recent studies in the literature show a growing interest in developing advanced HVAC control strategies aimed at reducing building energy consumption while maintaining or improving user thermal comfort [4–9]. Achieving this goal requires accurate monitoring and modeling of HVAC operational data. Numerous studies have focused specifically on modeling energy consumption, which is one of the key operational variables of HVAC systems, using conventional statistical approaches such as regression and time series analysis [10–15]. While regression models are effective in identifying linear relationships between variables, they often fall short in capturing the inherent complexity, non-linearity, and dynamism of HVAC operations [16]. These models typically rely on fixed assumptions and cannot reflect time-varying relationships, feedback mechanisms, or the combined effects of multiple interacting

variables. Similarly, time series models such as Autoregressive Integrated Moving Average (ARIMA) rely heavily on historical patterns to forecast future values, often using only past energy consumption, weather conditions, time of day, or prior loads [17,18]. These models are generally designed to predict a single dependent variable based solely on its historical values, which means they follow a single-input, single-output (SISO) approach. As a result, they attempt to estimate HVAC system behavior by examining its previous trends without considering interactions with other relevant system parameters. Given these complexities, there is a growing need for robust data-driven modeling approaches that can effectively predict the dynamic nature of HVAC operational data. Robust data-driven modeling approaches refer to methods that can handle large-scale, heterogeneous, and noisy building data while capturing non-linear and time-varying dependencies across multiple variables. Unlike conventional regression and SISO time-series models, robust frameworks incorporate multi-input, multi-output architectures, regularization strategies to mitigate overfitting, and in some cases probabilistic or Bayesian structures to quantify prediction uncertainty. Such characteristics make them particularly suited for real-world HVAC systems, which operate through numerous interrelated variables, including return air temperature, supply air velocity, fan speed, as well as indoor and outdoor environmental conditions. These variables exhibit time-dependent and non-linear interactions that jointly influence both energy consumption and occupant comfort.

In this context, deep learning-based Recurrent Neural Network (RNN) architectures, particularly Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, offer significant advantages over traditional time series approaches in forecasting HVAC energy use and related operational variables [19,20]. These models can learn complex temporal dependencies over extended time horizons and are capable of handling multi-input, multi-output (MIMO) prediction tasks [21]. Unlike conventional models that rely on predefined assumptions and linear relationships, LSTM and GRU can directly learn time-varying and non-linear correlations from data [22]. One of the key advantages of LSTM and GRU architectures is their ability to retain long-term contextual information, which is particularly important in operational environments where the effect of one variable may be delayed or influenced by others over time [20–23]. As a result, these models provide a more holistic and robust data-driven framework for understanding and optimizing HVAC system performance, enabling

improved control strategies and enhanced decision-making for energy management and thermal regulation.

These advantages have been empirically validated in the literature through various comparative studies. For example, a study conducted on a single-family house in the United States compared several AI models including Random Forest (RF), Support Vector Regression (SVR), XGBoost, Multi-Layer Perceptron (MLP), LSTM, and 1D Convolutional Neural Network (Conv-1D) to determine the most accurate technique for predicting building electricity load. Using 10-fold cross-validation, the LSTM model demonstrated the lowest error and variance on both validation and test datasets [24]. In another study focused on office buildings with energy storage-integrated HVAC systems, a forecasting approach was developed by combining GRU, RF, and SVR models with a Pearson correlation-based time-shifting method. The GRU model achieved the highest accuracy, with a 12% reduction in RMSE and an R^2 score of 0.850 [25]. Additionally, in study [26] LSTM outperformed CNN and Artificial Neural Network (ANN) models in predicting hourly multi-energy vectors across various building types using a large dataset generated from EnergyPlus simulations. Among all models, LSTM delivered the highest accuracy and lowest error rate, successfully predicting 50.7% of tasks with a CVRMSE below 20%, and emerged as the most effective approach for multi-energy forecasting.

As a result, the use of RNN-based models for forecasting building energy consumption during the operational phase has increased significantly in recent years. According to a study by Yin et al., between 2015 and 2023, approximately 20% of the publications in this field employed LSTM, 5.5% used GRU, 2% relied on conventional RNNs, and 1.5% utilized Elman networks [27]. In one such study [28], three different Bayesian deep neural network models based on RNN, LSTM, and GRU were developed to probabilistically forecast building electricity load. The models demonstrated that accurate day-ahead predictions could be achieved using only 10 hours of past load data, even without weather information. In another study [29], a single-layer GRU network was proposed for fast and accurate occupancy prediction in smart building automation, using PIR sensor data collected from a large academic building in California. The GRU model outperformed LSTM by achieving 1.21% lower error, 13.57% fewer parameters,

and 10% faster training time, indicating its suitability for large-scale smart building applications.

More recently, hybrid and advanced architectures have also been proposed. For instance, Liu et al. developed a GRU-based adaptive model with feature clustering, which significantly improved dynamic HVAC energy prediction accuracy across seasonal variations [30]. Similarly, a Hybrid GRU-KAN model outperformed standalone GRU and LSTM in predicting cooling system energy consumption, highlighting the potential of hybrid deep learning approaches [31]. Beyond sequence models, reinforcement learning-based strategies have been applied to optimize HVAC operations in multi-zone offices, achieving up to 37% energy savings while maintaining comfort [32]. Transformer-based decision models, such as HVAC-DPT, have also been introduced, enabling in-context learning and generalization to unseen buildings with reported energy savings of 45% [33]. Although recent advances in deep learning and reinforcement learning have shown promise [24,25,34–36,26–33], most studies lack comparative evaluations of recurrent architectures under real operational data or are still limited to single-variable forecasting such as energy consumption.

To address these drawbacks, this study aims to address these research gaps through the following specific objectives:

- To comparatively evaluate the performance of three recurrent neural network architectures, namely RNN, LSTM, and GRU, for multi-input, multi-output (MIMO) prediction of HVAC operational parameters.
- To train and validate the models on a high-resolution real-world office building dataset comprising nine key HVAC variables, including indoor air temperature, fan speed, and energy load.
- To identify the optimal architecture and hyperparameter configuration that balances predictive accuracy, convergence speed, and generalization capability.
- To discuss the computational efficiency and deployment readiness of the models for integration into real-time HVAC control frameworks.
- To demonstrate the potential of lightweight architectures, particularly GRU, for enabling adaptive and energy-efficient smart building environments.

The remainder of this paper is structured as follows: Section 2 (Method) outlines the methodological framework, including data preprocessing, model architecture, raining procedures, and evaluation metrics. Section 3 (Dataset Description) introduces the building data used for training and testing the models, with detailed descriptions of key HVAC variables. Section 4 (Findings) presents the experimental setup, hyperparameter optimization results, and test performance for the RNN, LSTM, and GRU models. Section 5 (Discussion) provides a comparative interpretation of the model outcomes and discusses their implications for HVAC control strategies. Finally, Section 6 (Conclusion) summarizes the key findings, highlights the practical relevance of GRU-based forecasting models, and offers directions for future research in intelligent building operations.

2. METHOD

In this study, standard RNN, LSTM and GRU were selected as the recurrent neural networks to predict HVAC operational variables and their performances were compared. The RNN, LSTM, and GRU architectures were selected due to their differing capabilities in capturing temporal dependencies. Standard RNNs offer a simple recurrent structure but are limited by vanishing gradient issues, making them less effective for long sequences. LSTMs incorporate gating mechanisms that allow learning long-term dependencies, which is advantageous for weekly HVAC operational patterns. GRUs simplify the LSTM design by combining certain gates, providing comparable predictive performance with fewer parameters and faster training. These models were thus evaluated to identify the architecture that balances predictive accuracy, training efficiency, and generalization across the dataset. Figure 1 illustrates the basic units of these deep neural networks.

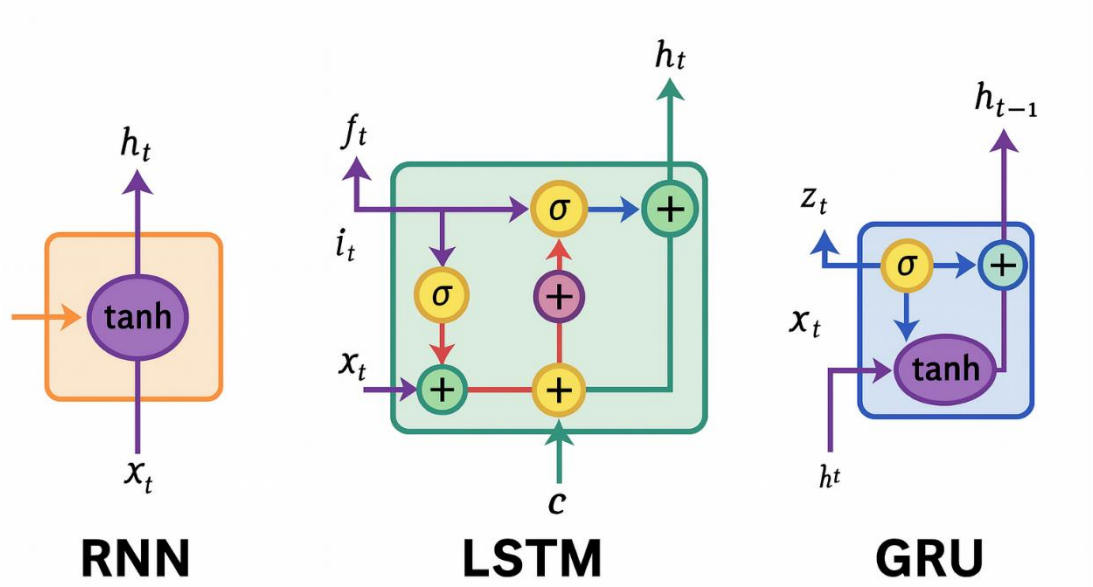

Figure 1. Basic units of standard RNN, LSTM and GRU

A Standard RNN is the simplest form of a recurrent neural network designed to process sequential data by using feedback connections. Unlike feedforward neural networks, RNNs maintain a hidden state that is updated at each time step based on both the current input and the hidden state from the previous time step. This allows RNNs to model temporal dependencies in sequential data [28,29,34].

The structure of a standard RNN is illustrated in Figure 1. As shown in Eq. (1), the hidden state h_t at time step t is updated by applying the tanh activation function to the weighted sum of the previous hidden state h_{t-1} and the current input x_t, where W_h and W_x are the corresponding weight matrices and bbb is the bias term. Eq. (2) then calculates the output yty_tyt at time step t by applying a linear transformation to the hidden state h_t, using the output weight matrix W_y and bias term b_y.

$$h_t = tanh(W_h \cdot h_{t-1} + W_x \cdot x_t + b) \quad \text{Eq. (1)}$$

$$y_t = (W_y \cdot h_t + b_y) \quad \text{Eq.(2)}$$

Standard RNNs do not have complex gates or multiple states and they are useful for simpler tasks involving short-term dependencies. Despite being conceptually simple, Standard RNNs suffer from significant limitations when dealing with long-term dependencies due to the vanishing gradient problem [34]. During backpropagation through time (BPTT), gradients can become exceedingly small, leading to minimal updates in earlier layers and causing the

network to "forget" important long-term dependencies. This problem makes it challenging for Standard RNNs to learn patterns that span long sequences. Moreover, because Standard RNNs use a single hidden state to store all information, they may struggle to capture complex dependencies in sequential data. This limitation motivated the development of more advanced architectures like LSTM and GRU, which include specialized gating mechanisms to manage long-term dependencies more effectively [28].

In LSTM networks, there is a special structure known as the cell state. This structure carries information throughout the network and updates it by adding or removing data as needed. LSTM consists of three gates: the forget gate, the input gate, and the output gate. These three gates collectively control the storage and deletion of data [35].

The formulations in a LSTM unit are given by Equations 3-7 [28,35]. As shown in Eq. (3), the forget gate f_t determines which part of the previous cell state c_{t-1} should be retained at time t. Here, x_t is the input vector at time t, h_{t-1} is the hidden state from the previous time step. Eq. (4) defines the input gate i_t, which decides how much of the new input x_t should be written to the cell state. Eq. (5) represents the output gate o_t, which regulates how much of the cell state will contribute to the output. Vectors. Eq. (6) updates the cell statec_t by combining the retained previous memory and the new candidate information. In all these equations, the W terms represent the corresponding weight matrices and the b terms represent the biases. Eq. (7) computes the hidden state h_t, which is also the output of the LSTM unit at time t.

$$f_t = \sigma(W_{fh} \cdot h_{t-1} + W_{fx} \cdot x_t + b_f) \quad \text{Eq. (3)}$$

$$i_t = \sigma(W_{ih} \cdot h_{t-1} + W_{ix} \cdot x_t + b_i) \quad \text{Eq. (4)}$$

$$o_t = (W_{oh} \cdot h_{t-1} + W_{ox} \cdot x_t + b_o) \quad \text{Eq. (5)}$$

$$c_t = f_t \times c_{t-1} + i_t \times \tanh(W_o \times [h_{t-1}, x_t] + b_o) \quad \text{Eq. (6)}$$

$$h_t = o_t \times \tanh(c_t) \quad \text{Eq. (7)}$$

LSTM networks are particularly well-suited for capturing long-range temporal dependencies, thanks to their internal cell states and forget gate mechanisms [36]. These components help alleviate the vanishing gradient problem during backpropagation, thereby supporting the training of deeper architectures and the modeling of longer sequences [36]. Moreover, the adaptable structure of LSTM enables it to handle complex patterns in sequential and time

series data effectively. Nevertheless, this architectural sophistication comes at a cost: the large number of parameters and the intricate design of multiple gating mechanisms significantly increase computational demands and make optimization more challenging [36].

To overcome these disadvantages, the GRU model was developed. GRU does not have a separate component called the cell state; instead, the cell state and hidden state in LSTM are combined. GRU has two gates because it merges the input gate and forget gate of LSTM into a single gate called the update gate [35].

The formulations in a GRU unit are given by Equations 8-11 [28,35]. As shown in Eq. (8), the update gate z_t determines how much of the previous memory is retained at the current time step, whereas the reset gate r_t, as shown in Eq. (9), controls how much of the past information should be forgotten. The candidate activation h'_t, calculated in Eq. (10), is obtained by applying the reset gate to the previous hidden state. Finally, as described in Eq. (11), the hidden state h_t is computed by linearly interpolating between the previous hidden state and the candidate activation based on the update gate. In all equations, W terms represent the weight matrices and b terms represent the bias vectors.

$$z_t = \sigma(W_{zh} \cdot h_{t-1} + W_{zx} \cdot x_t + b_z) \quad \text{Eq. (8)}$$

$$r_t = \sigma(W_{rh} \cdot h_{t-1} + W_{rx} \cdot x_t + b_r) \quad \text{Eq. (9)}$$

$$h'_t = tanh\,(W_{xh}x_t + r_t W_{hh} h_{t-1}) \quad \text{Eq. (10)}$$

$$h_t = z_t h_{t-1} + (1 - z_t) h'_t \quad \text{Eq. (11)}$$

Due to their forget and update gates, GRUs can capture both short- and long-term dependencies [37]. Additionally, because they have fewer gates and, therefore, fewer parameters, they are faster and more computationally efficient compared to LSTMs. However, since they are not as flexible as LSTMs, they may sometimes be inadequate in capturing very complex dependencies [37]. Furthermore, GRUs are not as effective as LSTMs in solving the vanishing gradient problem [37]. This can lead to a decline in performance when dealing with very long sequences.

In this study, to evaluate the performance of the standard RNN, LSTM and GRU models, the Mean Absolute Error (MAE), the Mean Squared Error (MSE), the Root Mean Squared Error (RMSE), the Mean Squared Logarithmic Error (MSLE) and the Coefficient of Determination (R^2) were calculated.

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is computed as the mean of the absolute differences between the predicted values ($\widehat{y_i}$) and the actual values (y_i) , as shown in Eq. (12):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \quad \text{Eq.(12)}$$

MSE represents the average of the squares of the errors, i.e., the average squared difference between the estimated values and the actual value. It is given by Eq. (13):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \quad \text{Eq. (13)}$$

RMSE is the square root of the average of the squared differences between the predicted values and the actual values. Unlike MAE, it gives higher weight to large errors and is calculated using Eq. (14):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2} \quad \text{Eq. (14)}$$

MSLE measures the mean squared difference between the logarithms of the predicted and actual values. It penalizes under-predictions more than over-predictions and is useful when the data span several orders of magnitude. It is computed as follows in Eq. (15):

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(log_e(1 + y_i) - log_e(1 + \widehat{y_i}))^2 \quad \text{Eq. (15)}$$

R^2, also known as the coefficient of determination, measures how well the predicted values explain the variability of the actual values. An R^2 value closer to 1 indicates better predictive accuracy. It is calculated using Eq. (16):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad \text{Eq. (16)}$$

3. DATASET DESCRIPTION

In this study, an open-access dataset collected by Luo et al. [38] was used. The data span the years 2018 to 2020 and were obtained from a medium-sized office building located on the Lawrence Berkeley National Laboratory campus in Berkeley, California. The building, referred to as Building 59 or Wang Hall, is divided into north and south wings and consists of 57 thermal zones. Thermal zones with external walls and windows are classified as perimeter zones, while the others are considered interior zones.

Heating and cooling in the office spaces are provided through an Underfloor Air Distribution (UFAD) system. The system utilizes four rooftop units (RTUs) equipped with water-cooled DX coils to supply cold air to the underfloor plenums. Each RTU serves specific vertical sections on the third and fourth floors of the building. However, physical separation between service areas is not present. The control scheme of the RTU-based HVAC systems is presented in Figure 2.

More detailed information on the building and its HVAC configuration can be found in [38]. Since each RTU controls different parts of the building with varying occupancy levels, significant variability exists in the historical operational data of the RTU-based HVAC system. Therefore, separate predictive models must be developed for each RTU. In this study, data from 2018 and 2019 were used to build prediction models for the operational variables of RTU1, which serves 14 thermal zones in the north wing of the building.

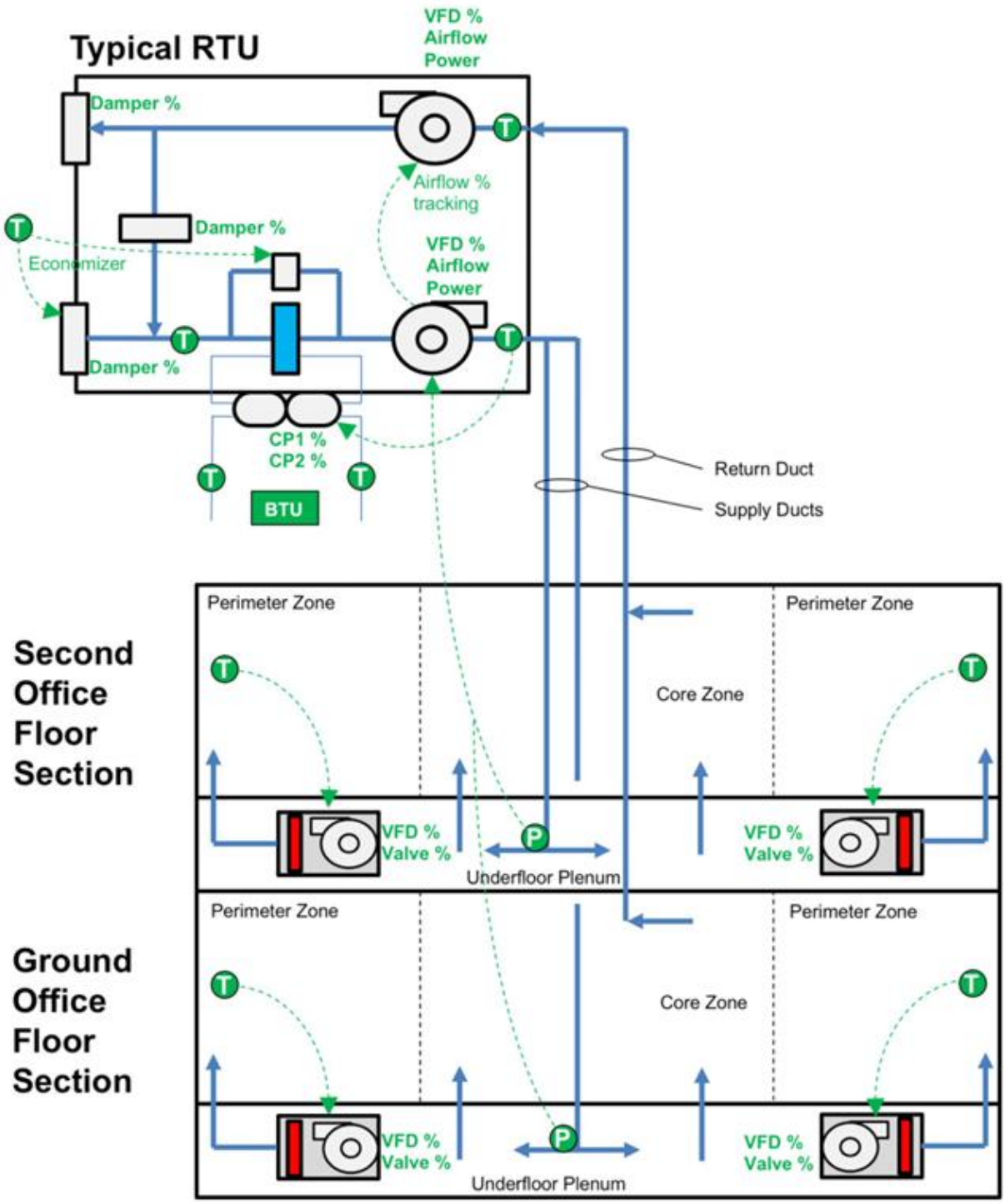

Figure 2. Control schematic for the RTU HVAC systems. Important temperature (T) and pressure (P) sensors and associated control points available through the ALC BMS interface are labelled [38].

For this study, all variables in the original dataset were converted to the International System of Units (SI), and the data were resampled into 15-minute intervals. As an initial step in the data preprocessing phase, both missing values and outliers were identified. Outliers were

detected using the standard deviation method, with a z-score threshold of three—a widely adopted approach for outlier detection in time series data [39,40]. Subsequently, the identified missing and outlier values, which occurred at the daily level, were imputed using the spline interpolation method. This method generates a smooth curve between the existing data points and places the missing points on this curve; in other words, it fills the gaps in the data using a smooth and continuous function. Thus, it is well-suited for estimating gaps in sensor-based time series data [41–43]. Upon completion of the interpolation process, a total of 70,084 data points were obtained for each of the 19 variables. The variables used in the study, along with their characteristics and descriptive statistics, are summarized in Table 1.

These preprocessing methods rely on the assumption that HVAC variables exhibit relatively smooth temporal dynamics. While suitable for the 15-minute resolution of this dataset, they may oversmooth abrupt transitions or rare operating states. Nonetheless, this trade-off was considered acceptable given the physical characteristics of the system and the study's focus on long-term predictive performance.

Table 1. Description and statistical summary of variables used in the study

	unit	**# of missing values**	**# of outlier values**	**min**	**max**	**Avr.**	**Std. dev.**
HVAC load for North Wing	kW	5,232	897	0,00	63.30	27.63	13.95
Outdoor air temperature	°C	0	1,298	1.79	26.08	13.85	4.67
Outdoor dew point temperature	°C	0	2,374	-6.06	18.22	7.06	4.72
Outdoor relative humidity	%	0	989	14.47	95.50	67.44	21.04
Outdoor solar radiation	W/m^2	0	1,783	0.00	870.00	185.42	269.15
Mean indoor air temperature	°C	10,616	1,123	19.88	24.22	21.99	0.84
Heat pump heating water supply temperature	°C	11,933	663	21.71	54.06	40.94	8.20
RTU supply air temperature set point	°C	0	2,529	18.33	21.11	19.84	0.56
RTU supply air temperature	°C	12,151	2,019	15.61	23.56	19.65	1.19
RTU return air temperature	°C	0	1,832	19.11	25.50	22.28	0.93
RTU mixed air temperature	°C	0	2,393	13.44	22.78	18.16	1.56
RTU outdoor air temperature	°C	0	1,560	-0.78	25.44	12.25	5.02
RTU filtered supply air speed	m^3/s	12,159	1,287	3.90	9.11	6.69	0.78

RTU outdoor air flow rate	m^3/s	0	1,990	0.84	5.98	3.38	0.97
RTU outdoor air damper position	%	0	4,462	0.00	95.60	34.36	18.09
RTU economizer set point	°C	0	2,539	18.41	21.11	19.85	0.57
RTU mean plenum air pressure across two floors	N/m^2	0	1,116	-162.03	651.55	241.99	89.12
RTU supply fan speed	m^3/s	0	1,308	44.20	100.00	74.91	8.13
RTU return fan speed	m^3/s	0	4,372	33.90	96.10	62.62	9.06

To identify collinear variables in RTU-based HVAC operational data, Pearson correlation coefficients were calculated. The Pearson correlation coefficient ($\rho_{(X,Y)}$) between two random variables X and Y is defined as the covariance of X and Y (cov(X, Y)) divided by the product of their standard deviations (σ_X, σ_Y). Extreme values of ρ, namely +1 and -1, represent strong linear or inverse-linear relationships between variables. Thresholds of +0.80 and −0.80 were adopted, whereby values exceeding these bounds were considered indicative of strong positive or negative linear relationships, suggesting the presence of multicollinearity [44,45]. Based on this analysis, the following pairs of variables exhibited high correlations:

- RTU supply air temperature setpoint and RTU economizer setpoint (ρ = 0.98)
- Outdoor air temperature from the sensor and RTU outdoor air temperature (ρ = 0.85)
- RTU supply fan speed and RTU filtered supply air speed (ρ = 0.94)

To mitigate redundancy and avoid multicollinearity, only one variable from each highly correlated pair was retained: RTU supply air temperature setpoint, outdoor air temperature from the sensor, and RTU supply fan speed. Consequently, the number of input variables was reduced from 19 to 16. Finally, as the dataset comprises variables measured on different scales, min-max normalization was applied to rescale all features to a range between 0 and 1. This normalization method has been shown to be effective in preparing data for neural network training [46].

4. FINDINGS

4.1 Experimental Setup

In this study, HVAC system operational data are modelled to support the development of advanced control strategies, including reinforcement learning and deep reinforcement learning [47–52]. In such strategies, an "agent" learns to control the HVAC system by observing environmental and HVAC-related variables. These variables represent the "state," which is defined as a mathematical abstraction of the environment that contains relevant and informative features for decision-making [40]. For example, in HVAC control applications, the current indoor air temperature may serve as a key state variable considered by the agent. An "action" is then selected by the control policy to regulate the environment accordingly. In the context of HVAC control, actions may include setting the indoor air temperature setpoint, adjusting the supply air temperature, or modifying the fan speed [40].

The state variables are categorized into environmental and RTU state variables. Time-related variables—such as month, day, and hour—are treated as environmental state variables. Variables recorded by sensors embedded within the RTUs are defined as RTU state variables. In addition, indoor air temperature measurements taken from wall-mounted sensors located in the external thermal zones served by underfloor terminals (UFTs), as part of the building automation system, are also considered RTU state variables. On the other hand, the RTU outdoor air damper position, RTU supply air temperature setpoint, and RTU supply fan speed are defined as action variables (Table 2).

Table 2. State and action variables

Environmental state variables	month, day, time, outdoor air temperature, outdoor dew point temperature, outdoor relative humidity, outdoor solar radiation
RTU state variables	mean indoor air temperature, heat pump heating water supply temperature, RTU supply air temperature, RTU return air temperature, RTU mixed air temperature, RTU outdoor air flow rate, RTU floor plenum air pressure, RTU return fan speed, HVAC load for north wing
RTU action variables	RTU outdoor air damper position, RTU supply air temperature setpoint, RTU supply fan speed

Multi-input, multi-output architectures were developed using standard RNN, LSTM, and GRU models. The input to the models developed in this study consists of all 19 state and action variables, while the output includes only the 9 RTU state variables. This input-output structure was designed based on the agent. The agent observes the current state—comprising both

environmental and RTU variables—and selects an action accordingly. The training environment then takes this state–action pair and predicts the resulting RTU state. Notably, the environment is not responsible for predicting the agent's next action or the environmental variables, which are typically derived from historical data (e.g., weather). Instead, the environment's role is limited to forecasting the subsequent RTU state.

To train and evaluate the models, standard RNN, LSTM, and GRU architectures were implemented using TensorFlow version 2.19 and its built-in Keras API. The dataset was split chronologically, reserving the last 20% as a holdout set for final evaluation [53]. The remaining 80% was used for model selection and hyperparameter optimization using time series cross-validation, which divides the data into sequential folds: the training sets grow with each fold, incorporating all previous data, while the validation sets slide forward in time, preserving temporal order. Hyperparameter tuning was performed in two stages: first, network capacity (number of layers and hidden units) was optimized, and then, keeping the best architecture fixed, learning rate and batch size were refined. This approach ensures that evaluation remains unbiased and accounts for the temporal dependencies and seasonal variations present in the data.

Additionally, the concept of "look-back" (lb), which refers to the number of previous time steps considered by the model when generating a prediction, was set to 672. This value was determined based on the sampling frequency of the building automation system, where data were recorded every 15 minutes. Given that there are 96 data points per day (24 variables recorded at 15-minute intervals), a look-back of 672 corresponds to one full week of historical data. The decision to use a one-week window is also rooted in the operational characteristics of HVAC systems in office buildings. Typically, these systems are programmed to cool or heat spaces during weekday working hours to maintain thermal comfort, while reducing energy consumption during off-hours and weekends. As such, HVAC systems often follow a weekly operational schedule. In order for HVAC control system to effectively capture these cyclical fluctuations and learn appropriate control strategies over time, it is essential for the episode length to encompass at least one full week [40]. Conversely, selecting a significantly longer episode window may lead to overfitting. Therefore, in this study, a one-week look-back value

(lb = 672) was adopted to balance the need for temporal learning with model generalizability [40].

4.2 Hyperparameter optimization

Hyperparameter optimization is a critical step in improving the predictive accuracy and generalization of machine learning models. In this study, a two-stage approach was adopted. The first stage focused on tuning the network capacity, namely the number of layers and the number of nodes per layer, while the second stage refined the learning rate and batch size. For both stages, the dataset was split chronologically, reserving the last 20% as a holdout set for final evaluation. The remaining 80% was used for the model selection through time series cross-validation. In each fold of the outer cross-validation, training sets grow with each fold, incorporating all preceding data, while validation sets slide forward in time, preserving temporal order. To implement early stopping, the training portion of each fold was further divided, allocating 80% for model training and 20% for monitoring the validation loss. Training was conducted for a maximum of 50 epochs, with early stopping applied if the inner-validation loss did not improve for 10 consecutive epochs. Model weights were restored to those corresponding to the epoch with the lowest inner-validation loss.

4.2.1 Stage-1: Network Capacity Tuning

The first stage focused on the number of layers and the number of nodes per layer. Based on previous studies indicating that RNNs typically do not require deep architectures to achieve reliable performance [40,54,55] two layer configurations (1-layer and 2-layer) and three node sizes (8, 16, and 32) per layer were considered, yielding six configurations per model. A total of 18 configurations were evaluated across standard RNN, GRU, and LSTM models using a grid search. MSE was employed as the primary performance indicator. During this stage, the learning rate and batch size were kept fixed at their default values (learning rate = 0.001, batch size = 32).

The numerical results for the stage-1 hyperparameter optimization are summarized in Table 3. The table reports several key metrics for each configuration of RNN, GRU, and LSTM models: model type identifies the recurrent network architecture whereas the number of layers and

the number of nodes per layer indicate the network depth and width, respectively. The average rank is calculated across the 5-fold time series cross-validation, where lower values indicate better relative performance; validation loss represents the mean across folds, reflecting expected predictive error; and the standard deviation of validation loss quantifies the stability and consistency of performance.

It is important to distinguish between average rank and average validation. While average validation measures absolute performance in terms of prediction error, average rank reflects the relative performance of a configuration within each fold, averaged across folds. Consequently, a configuration with the lowest average validation is not necessarily ranked first in every fold. A slightly higher average validation may correspond to a better average rank if the configuration consistently performs near the top across all folds. In this study, average rank is particularly favorable as a selection criterion because it accounts for variability across temporal folds and reduces the impact of occasional high or low validation losses, providing a more robust measure of consistent performance over time.

Table 3 shows that the 1-layer 16-node configuration consistently achieved the best performance for both GRU and LSTM models. The GRU with 1 layer and 16 nodes attained the lowest average rank (2.8) and a strong average validation (0.0054), closely followed by the LSTM with the same architecture (average rank of 4.0, average validation of 0.0055). In contrast, the RNN model exhibited slightly higher validation losses for the same configuration. The differences between GRU and LSTM are small, indicating that both architectures are highly competitive, though GRU shows a marginal advantage in consistency across folds.

These results suggest that shallow architectures with a moderate number of hidden units are sufficient to capture the temporal dependencies in the HVAC dataset, likely due to the repetitive weekly operational patterns and relatively smooth system dynamics. Deeper or larger architectures, such as 2-layer 32-node configurations, offered no significant improvements and in some cases led to higher validation losses, reflecting potential overfitting given the limited dataset size and the nature of the time series. Increasing the number of nodes beyond 16 resulted in minimal improvement, while a single layer was adequate for achieving competitive performance.

Table 3. Stage-1 hyperparameter optimization results for RNN, GRU, and LSTM architectures.

Model Type	# of layers	# of nodes per level	Avg. rank	Mean val.	Std. val.
GRU	1	16	2.8	0.0054	0.0014
LSTM	1	16	4.0	0.0055	0.0013
LSTM	2	32	5.6	0.0061	0.0024
LSTM	1	8	6.6	0.0054	0.0012
GRU	1	32	6.6	0.0065	0.0035
GRU	2	32	8.0	0.0064	0.0026
GRU	2	8	8.2	0.0066	0.0030
LSTM	1	32	8.4	0.0059	0.0014
GRU	2	16	8.6	0.0070	0.0028
LSTM	2	16	10.0	0.0067	0.0026
RNN	1	16	10.2	0.0071	0.0025
RNN	1	32	10.4	0.0074	0.0028
RNN	1	8	10.6	0.0069	0.0022
LSTM	2	8	10.8	0.0071	0.0035
GRU	1	8	12.8	0.0078	0.0034
RNN	2	8	15.2	0.0091	0.0031
RNN	2	16	15.6	0.0095	0.0041
RNN	2	32	16.6	0.0123	0.0085

Overall, these findings justify the selection of 1-layer 16-node architectures for GRU, LSTM, and RNN for stage-2 hyperparameter optimization. They also emphasize the importance of model simplicity, demonstrating that excessively deep or wide recurrent networks are unnecessary for accurate and reliable HVAC state forecasting.

4.2.2 Stage-2: Learning Rate and Batch Size Tuning

The second stage of hyperparameter optimization focused on refining the learning rate and batch size while keeping the network architecture fixed at the best-performing configuration identified in Stage-1. Specifically, the number of layers and nodes per layer were set to 1 and 16, respectively, for RNN, GRU, and LSTM models.

A grid search was conducted over three learning rates (0.0001, 0.001, 0.01) and four batch sizes (16, 32, 64, 128), chosen to balance convergence speed, training stability, and generalization. Smaller learning rates were included to allow careful adjustment of weights in the relatively shallow networks, while larger values were tested to potentially accelerate

convergence. Similarly, batch sizes were selected to explore the trade-off between noisy gradient estimates with smaller batches and smoother updates with larger batches, while keeping memory requirements manageable. A total of 36 configurations were evaluated across standard RNN, GRU, and LSTM models.

The results of Stage-2 hyperparameter optimization are summarized in Table 4. The table reports model type, learning rate (lr), batch size, average rank, mean validation loss, and the standard deviation of validation loss. As in Stage-1, average rank reflects relative consistency across folds, while mean validation loss quantifies predictive error.

According to the results, GRU proved the most accurate and stable, particularly with a learning rate of 0.001 combined with small to moderate batch sizes. The GRU achieved the lowest average rank (4.2) and a strong mean validation loss (0.0049), closely followed by the LSTM with the same settings (average rank of 6.8, mean validation of 0.0054), showing comparable predictive ability but slightly less consistency. RNN, while generally weaker, still provided reasonable baselines at lr = 0.001 with small to moderate batch sizes (best average rank of 12.0, mean validation of 0.0067).

Table 4. Stage-2 hyperparameter optimization results for RNN, GRU, and LSTM architectures (top 18 configurations).

Model Type	lr	Batch	Avg. rank	Mean val.	Std. val.
GRU	0.001	64	4.2	0.0049	0.0008
GRU	0.0001	16	4.4	0.0050	0.0010
GRU	0.001	16	4.8	0.0050	0.0010
LSTM	0.001	64	6.8	0.0054	0.0014
LSTM	0.0001	16	7.4	0.0052	0.0011
GRU	0.001	128	8.4	0.0052	0.0009
LSTM	0.001	16	9.8	0.0062	0.0023
LSTM	0.001	128	9.8	0.0056	0.0013
LSTM	0.001	32	11.0	0.0059	0.0018
GRU	0.0001	32	11.2	0.0059	0.0019
GRU	0.001	32	11.2	0.0072	0.0048
RNN	0.001	16	12.0	0.0067	0.0027
LSTM	0.0001	32	12.4	0.0058	0.0011
GRU	0.010	128	16.6	0.0080	0.0041

RNN	0.001	32	17.8	0.0074	0.0025
RNN	0.001	64	18.2	0.0073	0.0023
LSTM	0.0001	64	18.6	0.0075	0.0020
LSTM	0.010	128	18.8	0.0079	0.0028

Two clear patterns emerged from Stage-2. First, a learning rate of 0.001 consistently provided the best balance of convergence speed and stability, whereas smaller rates were occasionally competitive but less reliable, and larger rates led to unstable training. Second, batch sizes of 16 and 64 were most effective, while 32 and especially 128 tended to increase errors or reduce stability. Based on these results, the best-performing configuration for each model was selected for further evaluation: GRU (lr = 0.001, batch size = 64), LSTM (lr = 0.001, batch size = 64), and RNN (lr = 0.001, batch size = 16).

4.3 Numerical Findings

After determining the optimal model configuration (one layer and 16 nodes), the final evaluation was performed on the holdout test set. Each model was retrained using 80% of the dataset (training + validation combined), while the remaining 20% was reserved exclusively for testing. Training was allowed for a maximum of 100 epochs, with early stopping applied using a patience threshold of 20 epochs.

As shown in Figure 3, the training and test loss trajectories reveal distinct behaviors across the three models. Both the LSTM and GRU achieved stable convergence, with training losses decreasing smoothly and test losses reaching a consistent plateau. The GRU displayed the most rapid decline in validation loss, stabilizing below 0.01 within fewer epochs compared to the LSTM. In contrast, the LSTM's test curve converged more slowly, although it ultimately approached a similar final accuracy.

The standard RNN exhibited a different trend. While its training loss decreased comparably to the other models, its test loss plateaued at a noticeably higher level. This gap indicates limited generalization capability and early stagnation, despite extended training.

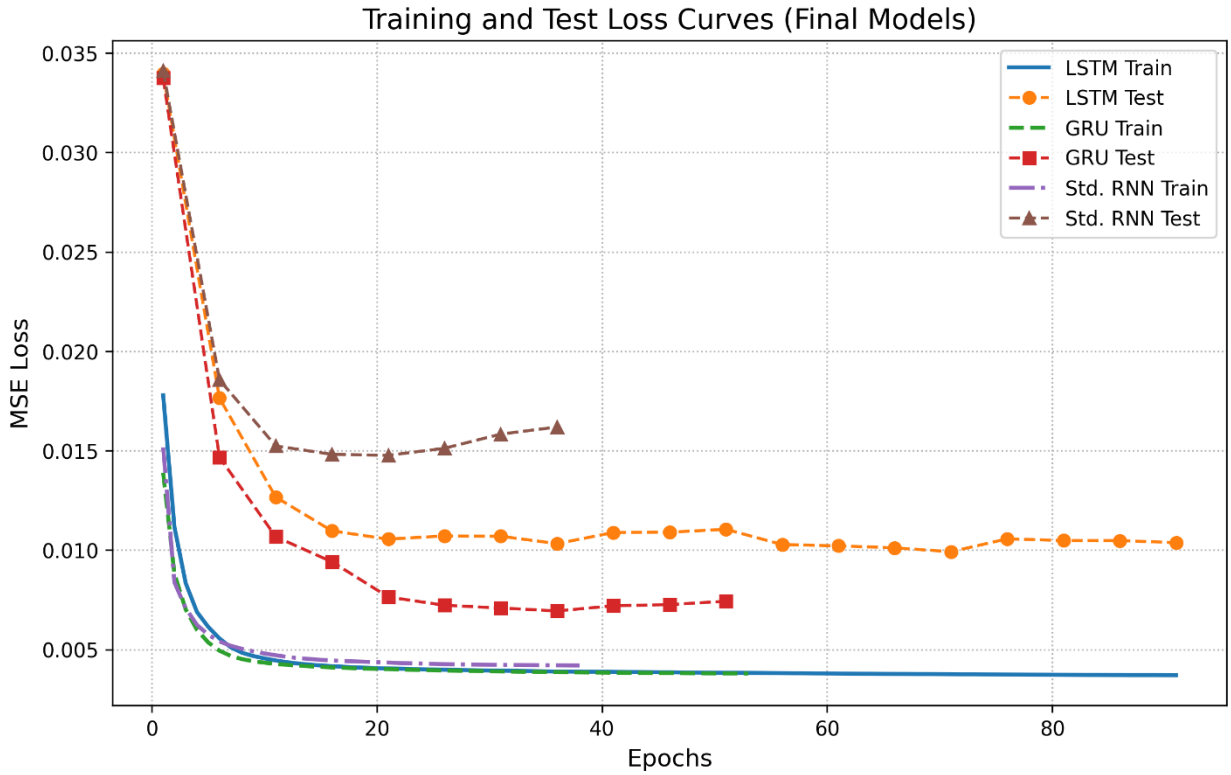

Figure 3. Training and holdout (test) loss curves for the RNN, GRU, and LSTM models using the best configurations selected from the two-stage hyperparameter optimization.

The numerical results for the GRU model obtained after Stage 2 hyperparameter tuning, evaluated on the holdout set, are presented in Table 5. The evaluation metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Squared Logarithmic Error (MSLE), and the coefficient of determination (R^2). Strong predictive performance was observed for most attributes—particularly for Heat pump heating water supply temperature, RTU supply air temperature, HVAC load for North Wing, and Mean indoor air temperature—with R^2 values exceeding 0.85, indicating a high level of accuracy.

Conversely, RTU floor plenum air pressure and RTU return fan speed exhibited relatively weaker predictive performance, with R^2 values of 0.26 and 0.69, respectively. This discrepancy may be attributed to potential distributional differences between the training and holdout datasets, which could hinder accurate generalization for variables with higher variability.

The results reveal that 7 out of 9 attributes can be predicted successfully, with R^2 scores above 0.81 (and a maximum of 0.99). This indicates the model's strong predictive capabilities for

most features. However, some attributes remain challenging to predict, suggesting room for improvement in the model. Possible areas for improvement could include refining the data collection process to better capture underlying patterns or addressing the distribution differences between training and holdout sets.

Table 5. Performance metrics of the GRU model on the test dataset for each RTU state variable.

State Variable	MAE	MSE	RMSE	MSLE	R^2
Heat pump heating water supply temperature	0.0410	0.0034	0.0580	0.0017	0.8531
RTU mixed air temperature	0.0832	0.0110	0.1051	0.0067	0.8505
RTU outdoor air flow rate	0.0696	0.0098	0.0992	0.0045	0.8180
RTU floor plenum air pressure	0.0771	0.0115	0.1072	0.0054	0.2581
RTU return air temperature	0.0300	0.0024	0.0487	0.0009	0.9439
RTU return fan speed	0.0854	0.0142	0.1191	0.0052	0.6859
RTU supply air temperature	0.0671	0.0073	0.0855	0.0039	0.8706
HVAC load for North Wing	0.0410	0.0034	0.0579	0.0022	0.9360
Mean indoor air temperature	0.0143	0.0004	0.0199	0.0002	0.9901
Avg.	0.0565	0.0070	0.0778	0.0034	0.8007

Figure 4 illustrates the comparison between the actual and predicted values for the nine RTU state variables using the final GRU model obtained after Stage 2 hyperparameter tuning. Each subplot displays the true values (solid blue line) alongside the model's predictions (dashed orange line) for the holdout dataset, providing insight into the model's generalization capability across different operational variables.

Overall, the GRU model demonstrates strong predictive accuracy for most of the variables, particularly those related to thermal dynamics. The predictions for mean indoor air temperature, RTU return air temperature, and HVAC load for the North Wing closely follow the actual values throughout the time series, with minimal deviations. This suggests that the GRU model is highly effective at capturing the temporal dependencies and control-response dynamics of temperature-related variables, which generally follow smoother and more predictable patterns over time.

Similarly, variables such as RTU supply air temperature, Heat pump heating water supply temperature, RTU mixed air temperature, and RTU outdoor air flow rate are also predicted

with reasonable accuracy. Although some discrepancies can be observed during transient periods or peak fluctuations, the model succeeds in replicating the temporal trends of these variables. This performance is consistent with their moderate-to-high R^2 values, reflecting the model's ability to generalize well for variables with stable dynamics.

In contrast, the model performs less effectively in predicting RTU return fan speed and, in particular, RTU floor plenum air pressure. The predicted values for these variables tend to deviate substantially from the actual values, failing to capture the sharp oscillations and variability present in the true data. Such underperformance may be attributed to several factors: first, these variables may be influenced by complex or non-deterministic control dynamics and external disturbances not fully captured by the available features; and second, their high variability could reflect higher-frequency fluctuations or sensor noise, making accurate generalization more difficult.

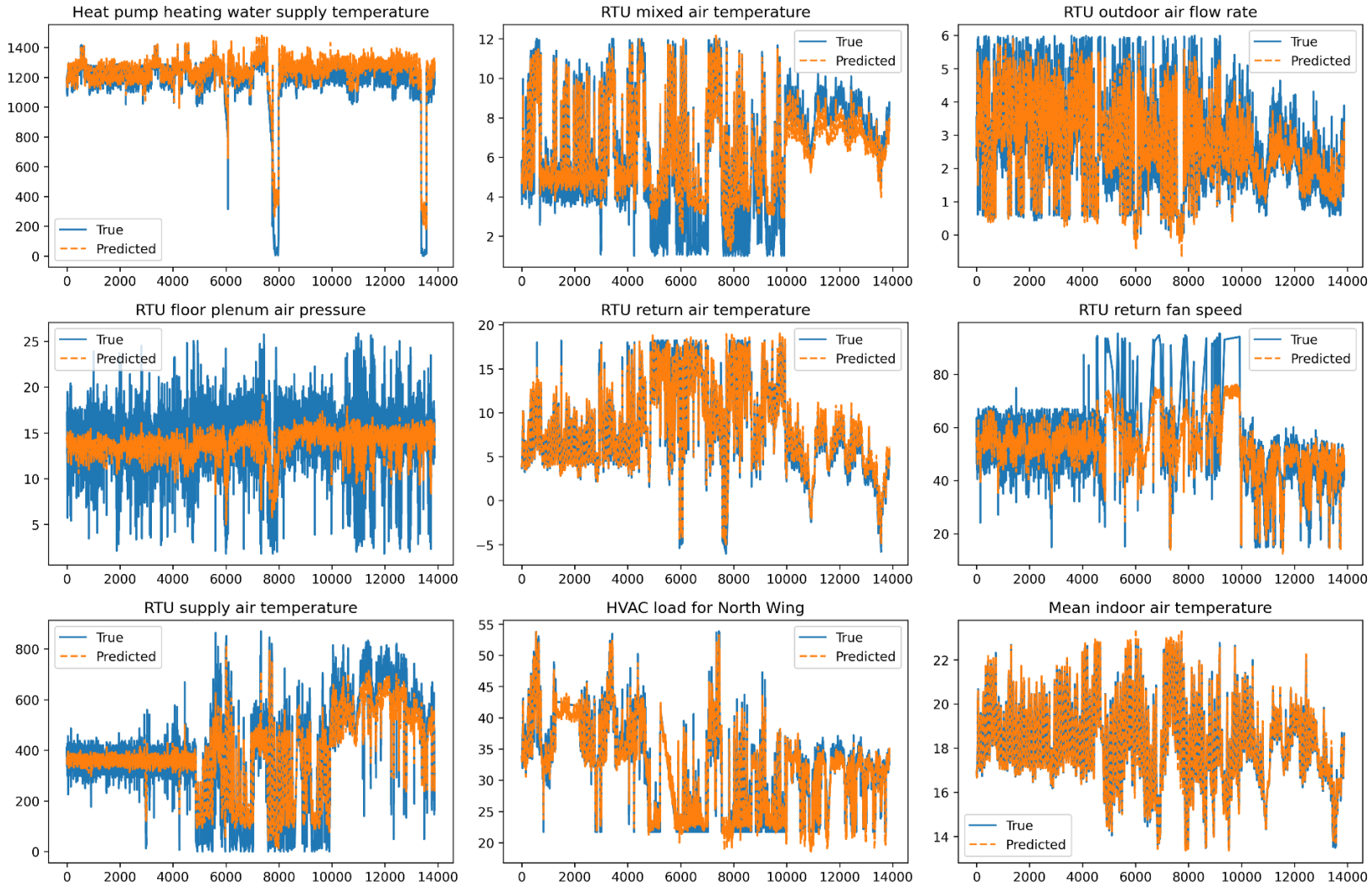

Figure 4. Comparison of true (solid blue) and predicted (dashed orange) values for the nine RTU state variables using the tuned GRU model on the holdout set.

5. DISCUSSION

5.1 Model Performance

This study compared the performance of three RNN architectures—standard RNN, LSTM, and GRU—for the prediction of RTU-based HVAC system's operational parameters. Among the tested models, the GRU architecture with one layer and 16 nodes consistently achieved the highest accuracy and efficiency. It outperformed both standard RNN and LSTM in terms of MSE, convergence speed, and R^2 scores across most predicted variables. These results support the effectiveness of GRU models in modeling multivariate time-series data in building automation systems, especially when fast convergence and computational efficiency are required.

While LSTM has traditionally been the dominant architecture in HVAC modeling tasks—such as for predicting RTU or AHU performance [40,56,57]—an increasing number of recent studies have highlighted the potential of GRU. For instance, [58] reported that GRU reduced RMSE by 3.4% and MAPE by 14% compared to LSTM in peak electricity load forecasting. Similarly, [59] and [60] found that GRU achieved better predictive accuracy and lower error metrics than LSTM across various energy forecasting scenarios. In a separate study by [61], GRU not only yielded better R^2 and RMSE scores but also required 14% less computation time than LSTM. [19] further showed that GRU surpassed LSTM with 13% lower RMSE and 17% higher R^2 when applied to smart grid energy predictions.

In [62], energy consumption prediction in smart homes was conducted using both daily and hourly test data. The results revealed that GRU outperformed LSTM in daily predictions, achieving 4% lower MSE, 1.8% lower RMSE, and 4.4% lower MAE. However, for hourly predictions, LSTM demonstrated superior performance, with GRU's error metrics (MSE, RMSE, MAE) being respectively 19%, 10%, and 20% higher. These findings suggest that GRU's simpler architecture may be more effective in capturing short-term dependencies and handling less complex, fluctuating daily patterns, while LSTM's gated structure offers advantages in modeling longer-term dependencies and more intricate temporal relationships. Moreover, [63] compared the performance of LSTM, GRU, and Drop-GRU models for short- to medium-term electric load forecasting using real-world consumption data from French cities. Although

both LSTM and GRU achieved accurate results, GRU was found to be particularly advantageous due to its reduced parameter count, faster training time, and architectural simplicity.

In addition to the comparative performance metrics, in this study, notable differences in convergence behavior and generalization ability were observed among the models. The GRU model consistently demonstrated faster convergence and more stable validation loss curves compared to LSTM and standard RNN. While LSTM showed comparable training performance, it exhibited signs of overfitting, particularly during the later epochs, as indicated by increasing validation loss and less stable test accuracy. This suggests that although LSTM's more complex gating mechanisms may provide greater representational capacity, they can also lead to diminished generalization in scenarios with limited or noisy data. In contrast, the GRU model achieved better test performance with lower MSE and higher R^2 scores, indicating its ability to generalize effectively without overfitting. These findings support the notion that GRU's simpler structure—with fewer parameters and more direct gate mechanisms—not only reduces training time but also mitigates the risk of overfitting, particularly in multivariate, real-world building automation datasets.

Notably, the structure and richness of the input–output framework used in this study may also have played a role in GRU's superior performance. Unlike most studies that focus on single-variable forecasting (typically energy consumption), this study adopted a multi-input, multi-output (MIMO) architecture, using 19 input variables to predict 9 different RTU-related parameters. In such settings, the simplicity and efficiency of GRU—compared to LSTM's more complex gate mechanisms—may offer a practical advantage, especially for real-time or resource-constrained applications in smart buildings.

In conclusion, while LSTM remains a strong candidate for capturing long-term dependencies in time-series data, the findings of this study and the comparative evidence from recent literature suggest that GRU presents a compelling alternative for multi-variable HVAC prediction tasks. Its faster training, lower computational burden, and competitive predictive performance make it an attractive model, particularly when operational efficiency is a key consideration.

5.2 Practical Implications

The findings of this study demonstrate considerable potential for advancing HVAC operations in real-world applications, particularly in the context of facility management and building energy management. Accurate forecasting of HVAC operational parameters not only enhances algorithmic performance but also translates into tangible benefits such as reduced energy costs, improved thermal comfort, and increased system reliability. These benefits are especially critical in large-scale commercial and institutional buildings, where energy-intensive HVAC systems account for a substantial portion of operational expenses.

From a control perspective, precise multi-variable predictions enable advanced strategies such as Model Predictive Control (MPC) to operate more effectively. By anticipating future system states, MPC can proactively adjust HVAC operations to prevent comfort violations, minimize peak loads, and reduce unnecessary energy use. Similarly, reinforcement learning (RL) frameworks depend heavily on accurate short-term forecasts to develop stable and efficient control policies. Improvements in predictive accuracy reduce the likelihood of policy oscillations and accelerate convergence, thereby increasing the robustness and adaptability of RL-based building control systems.

At the operational level, the integration of GRU-based forecasting models into Building Automation Systems (BAS) provides facility managers with a practical decision-support tool. Such integration would allow building operators to anticipate demand peaks, optimize fan and supply air operations, streamline maintenance scheduling, and reduce occupant complaints. By embedding lightweight yet accurate recurrent models directly into existing BAS infrastructures, facilities can achieve scalable, real-time adaptability without incurring prohibitive computational costs. Moreover, the integration of such forecasting models with digital twin platforms can enable continuous monitoring and scenario testing, allowing operators to evaluate different energy management strategies before deployment. In practice, this could support the transition from static, rule-based HVAC schedules to dynamic, data-driven strategies that balance occupant comfort with energy efficiency goals.

Overall, the empirical results of this study not only highlight methodological advancements in recurrent neural network modeling but also underscore their real-world relevance. The

demonstrated improvements in predictive accuracy and computational efficiency contribute directly to the broader goals of sustainable building operation, energy efficiency, and occupant well-being. These findings reinforce the importance of data-driven approaches in shaping the next generation of intelligent, adaptive, and energy-conscious building environments.

5.3 Replicability Potential

The successful application of the GRU architecture in predicting multiple RTU-based HVAC operational parameters demonstrates its high replicability potential in advanced control frameworks. In data-driven HVAC control systems—such as Model Predictive Control (MPC), adaptive control, or hybrid reinforcement learning strategies—the reliability of the prediction model directly influences the quality of control actions. GRU's simpler architecture, faster convergence, and superior generalization ability (as evidenced by its performance in this study) make it particularly suitable for integration into these systems.

Once trained on historical building automation data, GRU models can be embedded within real-time control loops to anticipate system responses under different control scenarios. This predictive capacity enables proactive regulation of HVAC operations to maintain thermal comfort while minimizing energy consumption. Moreover, due to its computational efficiency, GRU is ideal for deployment in edge computing environments or smart controllers with limited processing capabilities.

Integrating the GRU model into operational HVAC systems requires access to a well-instrumented Building Automation System (BAS) or Building Management System (BMS). These systems must provide continuous streams of sensor data, such as temperature, humidity, air pressure, and energy consumption. A data preprocessing pipeline—including outlier detection, interpolation, and normalization—must be implemented prior to real-time inference. Software frameworks such as TensorFlow Lite, ONNX Runtime, or Edge Impulse can be used to deploy trained GRU models on embedded hardware. For cloud-based deployments, platforms like AWS IoT Greengrass or Azure IoT Hub can facilitate scalable integration.

Several recent studies support the replicability of such approaches. For instance, [18] successfully embedded GRU models into a smart grid control framework, achieving real-time energy balancing with reduced computational cost. Similarly, [51] demonstrated the feasibility of integrating GRU into an MPC-based HVAC controller, where it provided robust forecasts of temperature and energy demand under varying external conditions. These examples reinforce the practical viability of the GRU model beyond experimental settings, especially when computational scalability and fast retraining are desired.

Despite its promising results, several factors must be considered before deploying GRU models in live HVAC control systems. First, model performance is highly dependent on the quality and representativeness of the training data; biased or incomplete datasets may lead to poor generalization. Second, system drift over time (e.g., equipment aging, occupancy changes) may require periodic retraining or online learning capabilities. Third, multi-input, multi-output models, as used in this study, require careful feature selection to avoid overfitting and ensure interpretability. Finally, integration into existing HVAC infrastructures may be constrained by proprietary hardware, communication protocols, or legacy software systems, requiring additional adaptation efforts.

Furthermore, the model's ability to predict multiple variables simultaneously in a MIMO configuration, as used in this study, enhances its compatibility with integrated building automation platforms, where multiple outputs such as zone temperatures, supply air flow rates, and power loads must be considered simultaneously. By coupling GRU-based forecasts with optimization algorithms, control strategies can be continuously refined based on real-time feedback, enabling adaptive and context-aware HVAC regulation.

In summary, the GRU model selected in this study not only demonstrates strong predictive performance but also offers significant advantages for real-world deployment in intelligent building environments. It should be noted that runtime performance benchmarks (e.g., latency, memory footprint, and hardware constraints) were not evaluated in this study. Future work will focus on testing GRU deployment on representative platforms such as embedded controllers (TensorFlow Lite) or edge devices (ONNX Runtime), to complement the predictive findings with concrete runtime evidence. Its replicability potential lies in its balance between

accuracy, efficiency, and adaptability, making it a practical choice for the next generation of smart HVAC control systems.

6. CONCLUSION

This study conducted a comprehensive comparative analysis of three RNN architectures, namely standard RNN, LSTM, and GRU to predict multiple key operational variables of HVAC systems in a MIMO framework. Using high-resolution sensor data from a real-world office building, the performance of these models was evaluated based on multiple metrics including MSE, RMSE, MAE, MSLE, and R^2. The key findings of this research are as follows:

- GRU demonstrated stronger overall performance than both LSTM and standard RNN models in test accuracy, convergence speed, and overall efficiency, making it a relatively more effective architecture for predicting HVAC operational parameters within the scope of this study.
- The optimal configuration for all models was a single-layer structure with 16 nodes, highlighting that deeper architectures do not necessarily yield better performance and may lead to overfitting.
- The GRU model achieved R^2 scores above 0.81 for 7 out of 9 predicted variables, with peak accuracy of 0.99 for indoor air temperature, demonstrating its strong generalization ability.
- In contrast, standard RNN models consistently underperformed, due to their inability to capture long-term dependencies.
- Although LSTM models showed competitive performance, they exhibited higher computational demands and greater risk of overfitting in later epochs.
- The proposed MIMO prediction approach effectively captured the interdependencies among various HVAC operational variables, which is critical for developing advanced control strategies such as reinforcement learning.

These findings underscore the practical applicability of GRU-based models in intelligent building systems. By enabling accurate and real-time predictions of multiple operational parameters, the GRU model supports the development of energy-efficient and occupant-

centered HVAC control strategies. Its lightweight architecture and fast convergence also make it suitable for deployment in resource-constrained environments, such as embedded devices or edge computing platforms.

Nevertheless, the conclusions of this study should be interpreted in light of its limitations. Only three recurrent neural network architectures were tested, and hyperparameter tuning was limited in scope**,** which constrains the breadth of comparisons that can be drawn. While the GRU's reduced complexity suggests potential for broader generalizability, performance may vary across buildings with different operational patterns, climates, or occupant behaviors. External factors such as maintenance schedules, equipment degradation, or policy-driven operational changes could also introduce additional variability in real-world deployment.

In addition, a deeper comparative analysis with other studies would have provided richer insights; however, this was not feasible due to the lack of prior work employing the same dataset or comparable experimental configurations. Future research should address these gaps by developing standardized benchmark datasets, applying more comprehensive hyperparameter optimization, and incorporating a broader set of model families. Such efforts would not only facilitate deeper comparative analyses but also enhance the generalizability and replicability of findings across diverse contexts. Moreover, future work could focus on applying this methodology to a broader range of building types and climate zones, coupled with cross-validation over diverse datasets. This would help validate its generalizability and replicability, confirm its effectiveness in various real-world scenarios, and strengthen its potential for large-scale deployment. Incorporating external drivers such as occupant behavior data and equipment maintenance records could further strengthen predictive reliability. Another promising direction is the integration of GRU models into online learning frameworks, enabling adaptation to long-term system drift caused by equipment aging, evolving occupancy patterns, or shifting user preferences. Hybrid approaches that couple GRU-based forecasting with optimization algorithms could also facilitate Model Predictive Control (MPC), supporting proactive and context-aware decision-making in HVAC operations. Furthermore, extending GRU architectures to include uncertainty quantification would allow for risk-aware control strategies, particularly in dynamic and uncertain environments.

In conclusion, this study demonstrates that GRU models present a robust, efficient, and scalable solution for HVAC system prediction tasks, while acknowledging the limited range of architectures and hyperparameter search as constraints**.** These findings lay a strong foundation for incorporating GRU-based forecasting into the next generation of smart building control systems, with future work needed to confirm scalability and generalizability**.**

Data availability
Data will be made available on request.

Declaration of competing interest
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] E.K. Simpeh, J.G. Pillay, R. Ndihokubwayo, D.J. Nalamu, Improving energy efficiency of HVAC systems in buildings : a review of best practices, Int. J. Build. Pathol. Adapt. 40 (2022) 165–182. https://doi.org/10.1108/IJBPA-02-2021-0019.

[2] International Energy Agency, International Energy Agency (IEA) World Energy Outlook 2022, Https://Www.Iea.Org/Reports/World-Energy-Outlook-2022/Executive-Summary. (2022) 524. https://www.iea.org/reports/world-energy-outlook-2022.

[3] A. Saadallah, S. Ali, H. Abdulkarim, Q.A. Ali, K.A. Ayad, A. Koca, Air thermal management platform assessment in centralized and decentralized air - conditioning systems, J. Therm. Anal. Calorim. 149 (2024) 12399–12415. https://doi.org/10.1007/s10973-024-13546-1.

[4] Y. Choi, X. Lu, Z.O. Neill, F. Feng, T. Yang, Optimization-informed rule extraction for HVAC system : A case study of dedicated outdoor air system control in a mixed-humid climate zone Cooling coil outlet Air, Energy Build. 295 (2023) 113295. https://doi.org/10.1016/j.enbuild.2023.113295.

[5] S.B. Sai, H. Srinaath, S. Sanjunath, A review and analysis of control techniques in HVAC systems, Sustain. Agri, Food Environ. Res. 10 (2022) 1–8. https://doi.org/10.7770/safer-v10n1-art2491.

[6] Y. Du, H. Zandi, O. Kotevska, K. Kurte, J. Munk, K. Amasyali, E. Mckeea, F. Lia, Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning, Appl. Energy. 281 (2021) 116117.

[7] H.M.N.B. Herath, P. Paranamana, D.C.N. Chandrasena, Development of A Demand Control Algorithm for HVAC Systems to Conserve Energy in Office Buildings, in: 2023 IEEE Int. Conf. Energy Technol. Futur. Grids, IEEE, 2023: pp. 1–6. https://doi.org/10.1109/ETFG55873.2023.10407435.

[8] A.T. Nguyen, D.H. Pham, B.L. Oo, M. Santamouris, Y. Ahn, B.T.H. Lim, Modelling

building HVAC control strategies using a deep reinforcement learning approach, Energy Build. 310 (2024) 114065. https://doi.org/10.1016/j.enbuild.2024.114065.
[9] Y. Zhao, W. Li, C. Jiang, Thermal sensation and occupancy-based cooperative control method for multi-zone VAV air-conditioning systems, J. Build. Eng. 66 (2023) 105859. https://doi.org/10.1016/j.jobe.2023.105859.
[10] G. Calis, S.D. Atalay, M. Kuru, N. Mutlu, Forecasting occupancy for demand driven HVAC operations using time series analysis, J. Asian Archit. Build. Eng. 16 (2017) 655–660. https://doi.org/10.3130/jaabe.16.655.
[11] C. Fan, Y. Ding, Cooling load prediction and optimal operation of HVAC systems using a multiple nonlinear regression model, Energy Build. 197 (2019) 7–17. https://doi.org/10.1016/j.enbuild.2019.05.043.
[12] T. Ahmad, H. Chen, Y. Guo, J. Wang, A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review, Energy Build. 165 (2018) 301–320. https://doi.org/10.1016/j.enbuild.2018.01.017.
[13] C. Deb, F. Zhang, J. Yang, S.E. Lee, K.W. Shah, A review on time series forecasting techniques for building energy consumption, Renew. Sustain. Energy Rev. 74 (2017) 902–924. https://doi.org/10.1016/j.rser.2017.02.085.
[14] K. Amasyali, N.M. El-Gohary, A review of data-driven building energy consumption prediction studies, Renew. Sustain. Energy Rev. 81 (2018) 1192–1205. https://doi.org/10.1016/j.rser.2017.04.095.
[15] M. Bourdeau, X. qiang Zhai, E. Nefzaoui, X. Guo, P. Chatellier, Modeling and forecasting building energy consumption: A review of data-driven techniques, Sustain. Cities Soc. 48 (2019) 101533. https://doi.org/10.1016/j.scs.2019.101533.
[16] A. Afram, F. Janabi-Sharifi, Review of modeling methods for HVAC systems, Appl. Therm. Eng. 67 (2014) 507–519. https://doi.org/10.1016/j.applthermaleng.2014.03.055.
[17] W.T. Ho, F.W. Yu, Predicting chiller system performance using ARIMA-regression models, J. Build. Eng. 33 (2021). https://doi.org/10.1016/j.jobe.2020.101871.
[18] C. Nichiforov, I. Stamatescu, I. Fagarasan, G. Stamatescu, Energy consumption forecasting using ARIMA and neural network models, Proc. - 2017 5th Int. Symp. Electr. Electron. Eng. ISEEE 2017. 2017-Decem (2017) 1–4. https://doi.org/10.1109/ISEEE.2017.8170657.
[19] I. Amalou, N. Mouhni, A. Abdali, Multivariate time series prediction by RNN architectures for energy consumption forecasting, Energy Reports. 8 (2022) 1084–1091. https://doi.org/10.1016/j.egyr.2022.07.139.
[20] J. Song, G. Xue, Y. Ma, H. Li, Y. Pan, Z. Hao, An Indoor Temperature Prediction Framework Based on Hierarchical Attention Gated Recurrent Unit Model for Energy Efficient Buildings, IEEE Access. 7 (2019) 157268–157283. https://doi.org/10.1109/ACCESS.2019.2950341.
[21] F. Kurniawan, S. Sulaiman, S. Konate, M.A.A. Abdalla, Deep learning approaches for MIMO time-series analysis, Int. J. Adv. Intell. Informatics. 9 (2023) 286–300. https://doi.org/10.26555/ijain.v9i2.1092.
[22] I.D. Mienye, T.G. Swart, G. Obaido, Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications, Inf. 15 (2024) 1–34. https://doi.org/10.3390/info15090517.
[23] F. Mtibaa, K.K. Nguyen, M. Azam, A. Papachristou, J.S. Venne, M. Cheriet, LSTM-based

indoor air temperature prediction framework for HVAC systems in smart buildings, Neural Comput. Appl. 32 (2020) 17569–17585. https://doi.org/10.1007/s00521-020-04926-3.
[24] M. Cordeiro-Costas, D. Villanueva, P. Eguía-Oller, M. Martínez-Comesaña, S. Ramos, Load Forecasting with Machine Learning and Deep Learning Methods, Appl. Sci. 13 (2023). https://doi.org/10.3390/app13137933.
[25] H. Liu, Y. Liu, X. Guo, H. Wu, H. Wang, Y. Liu, An energy consumption prediction method for HVAC systems using energy storage based on time series shifting and deep learning, Energy Build. 298 (2023) 113508. https://doi.org/10.1016/j.enbuild.2023.113508.
[26] L. Gao, T. Liu, T. Cao, Y. Hwang, R. Radermacher, Comparing deep learning models for multi energy vectors prediction on multiple types of building, Appl. Energy. 301 (2021) 117486. https://doi.org/10.1016/j.apenergy.2021.117486.
[27] Q. Yin, C. Han, A. Li, X. Liu, Y. Liu, A Review of Research on Building Energy Consumption Prediction Models Based on Artificial Neural Networks quently , this literature review aims to provide a thorough analysis o types , and temporal characteristics in energy consumption poses guidelines fo, Sustainability. 16 (2024) 1–30.
[28] L. Xu, M. Hu, C. Fan, Probabilistic electrical load forecasting for buildings using Bayesian deep neural networks, J. Build. Eng. 46 (2022) 103853. https://doi.org/10.1016/j.jobe.2021.103853.
[29] N. Fatehi, A. Politis, L. Lin, M. Stobby, M.H. Nazari, Machine Learning based Occupant Behavior Prediction in Smart Building to Improve Energy Efficiency, 2023 IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. ISGT 2023. (2023) 1–5. https://doi.org/10.1109/ISGT51731.2023.10066411.
[30] H. Liu, Y. Liu, H. Huang, H. Wu, Y. Huang, Energy consumption dynamic prediction for HVAC systems based on feature clustering deconstruction and model training adaptation, (2024) 1439–1460.
[31] U.N. Mandiri, R. Sciences, Hybrid GRU-KAN Model for Energy Consumption Prediction in Commercial Building Cooling, 1 (2025) 69–78.
[32] H. Wang, X. Chen, N. Vital, E. Duffy, A. Razi, Energy optimization for HVAC systems in multi-VAV open offices : A deep reinforcement learning approach, Appl. Energy. 356 (2024) 122354. https://doi.org/10.1016/j.apenergy.2023.122354.
[33] U. Kingdom, HVAC-DPT : A Decision Pretrained Transformer for HVAC Control, (2024).
[34] L. Wen, K. Zhou, S. Yang, Load demand forecasting of residential buildings using a deep learning model, Electr. Power Syst. Res. 179 (2020) 106073. https://doi.org/10.1016/j.epsr.2019.106073.
[35] Z. Wang, Y. Zhao, C. Zhang, P. Ma, X. Liu, A general multi agent-based distributed framework for optimal control of building HVAC systems, J. Build. Eng. 52 (2022) 104498. https://doi.org/10.1016/j.jobe.2022.104498.
[36] R. DiPietro, G.D. Hager, Deep learning: RNNs and LSTM, in: Handb. Med. Image Comput. Comput. Assist. Interv., Elsevier, 2020: pp. 503–519. https://doi.org/10.1016/B978-0-12-816176-0.00026-0.
[37] F.M. Shiri, T. Perumal, N. Mustapha, R. Mohamed, A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU, (2023). http://arxiv.org/abs/2305.17473.
[38] N. Luo, Z. Wang, D. Blum, C. Weyandt, N. Bourassa, M.A. Piette, T. Hong, A three-year

dataset supporting research on building energy management and occupancy analytics, Sci. Data. 9 (2022) 1–15. https://doi.org/10.1038/s41597-022-01257-x.
[39] J.E. Seem, Using intelligent data analysis to detect abnormal energy consumption in buildings, Energy Build. 39 (2007) 52–58. https://doi.org/10.1016/j.enbuild.2006.03.033.
[40] Z. Zou, X. Yu, S. Ergan, Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network, Build. Environ. 168 (2020) 106535. https://doi.org/10.1016/j.buildenv.2019.106535.
[41] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, M. Kolehmainen, Methods for imputation of missing values in air quality data sets, Atmos. Environ. 38 (2004) 2895–2907. https://doi.org/10.1016/j.atmosenv.2004.02.026.
[42] J. Park, S. Kim, applied sciences Improved Interpolation and Anomaly Detection for Personal PM 2 . 5 Measurement, (2020).
[43] S. Moshenberg, U. Lerner, B. Fishbain, Spectral methods for imputation of missing air quality data, Environ. Syst. Res. (2015). https://doi.org/10.1186/s40068-015-0052-z.
[44] N. Shrestha, Detecting Multicollinearity in Regression Analysis, Am. J. Appl. Math. Stat. 8 (2020) 39–42. https://doi.org/10.12691/ajams-8-2-1.
[45] D.S. Young, Handbook of regression methods, CRC Press, Boca Raton, FL, 2017.
[46] T. Jayalakshmi, A. Santhakumaran, Statistical Normalization and Back Propagationfor Classification, Int. J. Comput. Theory Eng. (2011) 89–93. https://doi.org/10.7763/ijcte.2011.v3.288.
[47] Z. Deng, Q. Chen, Reinforcement learning of occupant behavior model for cross-building transfer learning to various HVAC control systems, Energy Build. 238 (2021) 110860. https://doi.org/10.1016/j.enbuild.2021.110860.
[48] D. Azuatalam, W.-L. Lee, F. de Nijs, A. Liebman, Reinforcement learning for whole-building HVAC control and demand response, Energy AI. 2 (2020) 100020. https://doi.org/10.1016/j.egyai.2020.100020.
[49] X. Ding, A. Cerpa, W. Du, Multi-zone HVAC Control with Model-Based Deep Reinforcement Learning, ArXiv:2302.00725. (2023) 1–13. https://doi.org/10.48550/arXiv.2302.00725.
[50] M. Esrafilian-najafabadi, F. Haghighat, Transfer learning for occupancy-based HVAC control : A data-driven approach using unsupervised learning of occupancy profiles and deep reinforcement learning, Energy Build. 300 (2023) 113637. https://doi.org/10.1016/j.enbuild.2023.113637.
[51] W. Li, Y. Zhao, J. Zhang, C. Jiang, S. Chen, L. Lin, Y. Wang, Indoor temperature preference setting control method for thermal comfort and energy saving based on reinforcement learning, J. Build. Eng. 73 (2023) 106805. https://doi.org/10.1016/j.jobe.2023.106805.
[52] A. Silvestri, D. Coraci, S. Brandi, A. Capozzoli, E. Borkowski, J. Köhler, D. Wu, M.N. Zeilinger, A. Schlueter, Real building implementation of a deep reinforcement learning controller to enhance energy efficiency and indoor temperature control, Appl. Energy. 368 (2024) 123447. https://doi.org/10.1016/j.apenergy.2024.123447.
[53] A. Ghareeb, A. Hussein, A. Saadallah, A. Kakei, E. Canli, A. Chiasson, J. Choi, A. Selim, Prediction of the operational performance of a vehicle seat thermal management system using statistical and machine learning techniques, Case Stud. Therm. Eng. 60 (2024) 104626. https://doi.org/10.1016/j.csite.2024.104626.
[54] B. Cortez, B. Carrera, Y.J. Kim, J.Y. Jung, An architecture for emergency event

prediction using LSTM recurrent neural networks, Expert Syst. Appl. 97 (2018) 315–324. https://doi.org/10.1016/j.eswa.2017.12.037.
[55] T. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control learning control with deep reinforcement learning, in: 4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc., 2016: pp. 1–14.
[56] A. Azhan, W. Tan, M. Gan, S. Yip, Predictive AC Control Using Deep Learning : Improving Comfort and Energy Saving, Int. J. Informatics Vis. 7 (2023) 1066–1073. https://doi.org/10.30630/joiv.7.3-2.2345.
[57] D. Zhuang, V.J.L. Gan, Z. Duygu, A. Chong, S. Tian, Data-driven predictive control for smart HVAC system in IoT-integrated buildings with time-series forecasting and reinforcement learning, Appl. Energy. 338 (2023) 120936. https://doi.org/10.1016/j.apenergy.2023.120936.
[58] A.A. Pierre, S.A. Akim, A.K. Semenyo, B. Babiga, Peak Electrical Energy Consumption Prediction by ARIMA, LSTM, GRU, ARIMA-LSTM and ARIMA-GRU Approaches, Energies. 16 (2023). https://doi.org/10.3390/en16124739.
[59] G. Goui, A. Zrelli, N. Benletaief, A Comparative Study of LSTM/GRU Models for Energy Long-Term Forecasting in IoT Networks, Proc. - 23rd IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2023. (2023) 60–64. https://doi.org/10.1109/ICIS57766.2023.10210257.
[60] M. Abumohsen, A.Y. Owda, M. Owda, Electrical Load Forecasting Using LSTM, GRU, and RNN Algorithms, Energies. 16 (2023) 1–31. https://doi.org/10.3390/en16052283.
[61] L.Y. Chen, Y.T. Chen, Y.H. Chen, D.S. Lee, Applicability of energy consumption prediction models in a department store: A case study, Case Stud. Therm. Eng. 49 (2023) 103380. https://doi.org/10.1016/j.csite.2023.103380.
[62] N. Bhoj, R. Singh Bhadoria, Time-series based prediction for energy consumption of smart home data using hybrid convolution-recurrent neural network, Telemat. Informatics. 75 (2022) 101907. https://doi.org/10.1016/j.tele.2022.101907.
[63] S. Mahjoub, L. Chrifi-Alaoui, B. Marhic, L. Delahoche, Predicting Energy Consumption Using LSTM, Multi-Layer GRU and Drop-GRU Neural Networks, Sensors. 22 (2022) 1–20. https://doi.org/10.3390/s22114062.