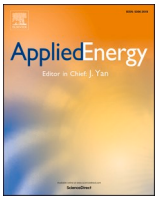
Applied Energy 338 (2023) 120936

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy



Data-driven predictive control for smart HVAC system in IoT-integrated buildings with time-series forecasting and reinforcement learning

Dian Zhuang [a], Vincent J.L. Gan [b,*], Zeynep Duygu Tekler [b], Adrian Chong [b], Shuai Tian [c], Xing Shi [c]

[a] School of Architecture, Southeast University, 2 Si Pai Lou, Nanjing 210096, China
[b] Department of the Built Environment, National University of Singapore, Singapore
[c] College of Architecture and Urban Planning, Tongji University, 1239 Si Ping Road, Shanghai 200092, China

HIGHLIGHTS

- Present a data-driven predictive control method for smart HVAC operations.
- Develop and validate 16 LSTM models with bi-directional processing, convolution, and attention mechanisms.
- Integrate optimal prediction models with a reinforcement learning agent to analyse sensor metadata and optimise the HVAC system.
- The influences of neural network configuration on recursive prediction are analysed.
- Improve 17.4% energy efficiency and 16.9% thermal comfort in IoT-enabled smart building.

ARTICLE INFO

Keywords:
Smart Facilities Management
Reinforcement Learning
Data-driven Control
Building Automation
Recursive Prediction
Time-Series Forecasting

ABSTRACT

Optimising HVAC operations towards human wellness and energy efficiency is a major challenge for smart facilities management, especially amid COVID situations. Although IoT sensors and deep learning were applied to support HVAC operations, the loss of forecasting accuracy in recursive prediction largely hinders their applications. This study presents a data-driven predictive control method with time-series forecasting (TSF) and reinforcement learning (RL), to examine various sensor metadata for HVAC system optimisation. This involves the development and validation of 16 Long Short-Term Memory (LSTM) based architectures with bi-directional processing, convolution, and attention mechanisms. The TSF models are comprehensively evaluated under independent, short-term recursive, and long-term recursive prediction scenarios. The optimal TSF models are integrated with a Soft Actor-Critic RL agent to analyse sensor metadata and optimise HVAC operations, achieving 17.4% energy savings and 16.9% thermal comfort improvement in the surrogate environment. The results show that recursive prediction leads to a significant reduction in model accuracy, and the effect is more pronounced in the temperature-humidity prediction model. The attention mechanism significantly improves prediction performance in both recursive and independent prediction scenarios. This study contributes new data-driven methods for smart HVAC operations in IoT-enabled intelligent buildings towards a human-centric built environment.

1. Introduction

The building and construction sector accounts for almost 40% of energy and process-related emissions [1]. Among various building facilities, the Heating Ventilation and Air Conditioning (HVAC) system contributes 50% of the energy consumption [2], and the energy demand is anticipated to triple by 2050 [3]. Since many preventive measures are implemented at workplaces nowadays to minimise the spread of COVID, it is vital to operate HVAC to maintain users' desired level of comfort while balancing energy efficiency [4]. Current building management systems use classical controllers such as rule-based control and proportional-integral-derivative control for HVAC. The classical controllers use fixed schedules together with manual control to establish the temperature set points, which can hardly respond to multi-dimensional

* Corresponding author.
E-mail address: vincent.gan@nus.edu.sg (V.J.L. Gan).

https://doi.org/10.1016/j.apenergy.2023.120936
Received 28 November 2022; Received in revised form 20 February 2023; Accepted 3 March 2023
Available online 8 March 2023

changes such as occupancy, climate conditions and electricity prices [5]. In addition, the implementation of classical controllers requires system adjustment manually according to the operating state, which may lead to wrong decisions and delayed responses [6,7]. With the advance of digital twins and artificial intelligence, researchers start to explore dynamic and intelligent control techniques [8]. Model predictive control (MPC) has been seen as one potential solution for intelligent system control in recent years. MPC is a well-established control method for complex interacting dynamic systems [9]. The control process relies on physics-based, grey-box or black-box models to achieve satisfactory robustness [7]. It has been implemented in the actual control with data from real buildings or generated from the simulation. It overcomes barriers of classical controllers which use fixed schedules together with manual control that can hardly respond to multi-dimensional changes such as occupancy, climate conditions and electricity prices [5].

Nowadays, there is a need towards a more adaptive approach to build a control-oriented system that can fit better with real environmental conditions [1011], and accelerate the computation of physics-based MPC [12]. Machine learning opens the way for transforming MPC towards data-driven predictive control. Reinforcement learning (RL) is one of the representative data-driven predictive control methods, wherein agents are established and trained to explore optimal control strategies from environment state information and "action-reward" loops [13]. The interaction environment may be a real-world environment corresponding to the online training process, or a virtual environment corresponding to the offline training process [14]. Ideally, the online training process avoids prior knowledge of real-world complex systems, combining with deep neural networks to quickly approach optimal control strategies. However, the flexibility of RL comes at the cost of increased complexity [15]. RL learns the optimal control by testing new strategies and evaluating their outcomes, thus some tested strategies might lead to an undesirable outcome which is unacceptable in the actual environment [16]. A common approach to overcome this problem is offline pretraining of RL agents within a surrogate environment [5].

Building information modelling (BIM) has been used to construct surrogate environments. However, the full range of real-world environmental features might not be correctly modelled, resulting in poor generalisability of the controller applied to actual buildings. As such, time-series forecasting (TSF) is seen as a viable solution to remedy the shortcoming. By integrating the building automation system and Internet of Things (IoT) sensing devices, time-series data about HVAC, indoor environment, outdoor weather, and occupant behaviour are continuously recorded to describe the dynamic environment [17,18]. Deep neural networks are subsequently used to achieve real-time prediction of future environmental changes. The data-driven environment realistically maps the environment interaction process and does not require rich professional knowledge to guide the model establishment [19]. In this regard, several neural network-based predictive models have been developed for HVAC energy efficiency, thermal comfort, and air quality [20–22].

While RL and TSF are promising in the literature, few studies have integrated them into HVAC optimal control. RL training obtains short-term and long-term experiences by continuously interacting the controller with the environment, which requires that TSF models continuously predict the environmental indicators. At the same time, it is difficult to get rid of the dependence on historical data in accurate time-series forecasting, which is characterized by data with multiple previous timestamps to predict the next timestamp. These results in predictive models constantly use the results of previous predictions as features for subsequent predictions, leading to exponential accumulations of prediction error [23–25]. Although several multi-step ahead prediction methods avoid error accumulation to a certain extent, it contradicts the continuous operation logic of the controller or overwhelms the control system by consuming large amounts of computational resources. This calls for a more intelligent approach for robust forecasting and optimising the building system operations.

This study aims to develop an automated data-driven predictive control method using TSF and RL to underpin smart HVAC operations in IoT-integrated buildings for energy efficiency and comfort optimisation. This involves the deep integration of a robust TSF model for training an RL agent to derive HVAC optimal control strategies. A model framework composed of 16 LSTM-based architectures is proposed, and a total of 48 prediction models corresponding to temperature, humidity, and system energy consumption prediction in the HVAC control problem are trained. Three TSF prediction scenarios including the recursive scenario are defined, and model robustness under the three prediction scenarios is evaluated. The optimal models are selected and the influences of different neural network architectures on model performance are discussed. Lastly, the optimal TSF models are used to train an RL agent for determining the predictive control strategies. The proposed new method is illustrated via a case study, the results of which indicate that the established data-driven control can realise 17.4% energy reduction and 16.9% PMV improvement. The rest of the paper is organised as follows. Section 2 presents the previous relevant studies, and Section 3 introduces the proposed new approach. Sections 4 and 5 illustrate the proposed method with detailed discussions on the predictive capacity of the deep neural network. Section 6 concludes the paper and describes future work.

2. Literature review

2.1. TSF models for building performance prediction

Wong and Li [26] presented the construction and validation of a selection evaluation model for intelligent HVAC control. This involves the evaluation of the candidate HVAC control system against certain selection criteria but also suggests a benchmark for the selection of the control system. Conventional models incorporated building performance with features independently, but the performance of a building is affected by short-term and long-term changes such as occupancy and season, thus having intrinsic temporal dependencies [23]. A recurrent neural network (RNN) in the field of speech recognition and natural language processing is designed for time-series prediction. An RNN stores long sequential information in hidden memory for proper processing, representation, and storage. Fig. 1 shows an example of recursion in RNN cells. X represents the learning data input by sequence, Y represents the prediction result of RNN cells, and h represents the system state of RNN cells to label the information observed up to the current moment. For time t, the system state of RNN cells is calculated as $h_t = f(\omega h_{t-1} + \mu X_t + b)$. Since solving the current system state requires information regarding the previous time step, the computation process contains recursion and therefore carries time series information. Its practical values in capturing long-term dependency are usually limited due to the problem of vanishing or exploding gradients [27]. Gated RNN is a feasible approach to cope with long-range dependencies. The idea is to empower RNN with the ability to control its internal information accumulation through a gated unit, which then masters both long-range dependencies and selectively forgets information to prevent

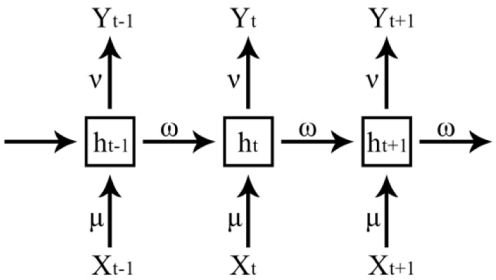


Fig. 1. Recurrent neural network (RNN) cells.

overloading. Long short-term memory (LSTM) is the most representative gated algorithm [28]. LSTM has been applied to the prediction of HVAC performance, and a variety of composite deep neural networks with LSTM as the core has emerged to improve the performance of prediction models.

Time-series forecasting of HVAC system performance mainly includes two targets: system energy consumption and thermal comfort. Table 1 summarizes the representative LSTM-based prediction models. The LSTM-based extension algorithms mainly include bidirectional long short-term memory (BiLSTM), convolutional neural networks (CNN), and attention mechanism (AM). The bidirectional operations enhance the prediction performance by adding a loop unit to achieve forward and backward movement for the identification of the impact of future information. Its contribution to the prediction accuracy of recursive prediction of HVAC energy consumption has been demonstrated in studies [23]. Attention mechanism coupled with LSTM/BiLSTM appears more frequently in the literature. The attention mechanism helps identify valid information more efficiently by assigning weights to different importance information. Besides, the attention mechanism can be added to the temporal [29,30] or feature [31] dimension, as well as to the front [31] or back [32] of LSTM layers. Anjun Zhao used a dual attention mechanism overlaid with the LSTM, where the feature attention layer is arranged in front of the LSTM layers and the temporal attention layer after the LSTM layers [33]. The enhancement of long prediction period stability in indoor temperature prediction has been demonstrated in [34], the accuracy improvement by LSTM [29] and BiLSTM [31,35] in energy consumption prediction and its resistance to overfitting [30] have been proved.

Convolutional options are often used in the field of computer vision, but they help process sequential data. A 1D convolution layer is often placed before RNN for data pre-processing [25]. The addition of CNNs has been proved to facilitate the screening of multiple features, eliminate noise, and improve prediction [36]. Such an approach can provide competitive results with much less computation time [23]. Although various HVAC performance prediction models have been proposed, the robustness of LSTM-based models under recursive prediction scenarios, which is crucial for the integration of predictive models with RL agents, has not been systematically evaluated. Some studies have mentioned the potential extension of LSTM-based algorithms towards recursive prediction [23,25], which requires further investigation.

2.2. TSF integrated with RL for HVAC predictive control

Reinforcement learning is a class of machine learning algorithms that specializes in solving control or sequential decision problems. Delayed feedback is a fundamental feature that distinguishes reinforcement learning [16]. It involves a set of interacting objects including the agent and the environment with their state (S), action (A), policy (P) and reward (R). As shown in Fig. 2, for any time t, the agent receives the current environmental state S_t and makes an action A_t based on the current policy P_t. The environment receives the action A_t and outputs a reward R_t back to the agent. The policy is then updated based on the reward, which completes a standard loop. In the above process, it is worth emphasising the short and long-term effects on the environment after any action made by the agent. Such effects could change as a whole following the new action, resulting in the inevitable problem of recursive prediction when the TSF models are used as an offline training environment.

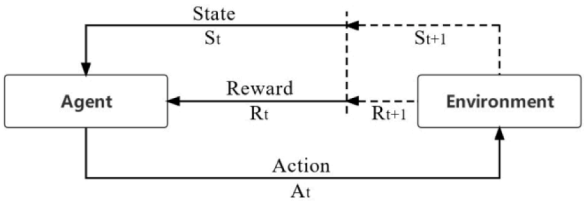


Fig. 2. Simplified interaction process between the agent and environment.

The coupled loops of TSF and RL in HVAC predictive control are shown in Fig. 3. The coloured blocks represent the updated features of the TSF model in each RL step: green blocks represent the features replaced by TSF prediction results such as energy consumption and air temperature; orange blocks represent the features replaced by RL actions such as temperature set point and supply air flow. Each standard loop involves three interaction processes as shown in the figure. (i) Current moment performance (such as HVAC energy) based on previous states is output by the predictive model to the agent. The immediate reward is then calculated for a policy update, and the performance is input as the current state for the next decision. (ii) Agent outputs the current performance as a new prediction feature to the TSF model for the next prediction. (iii) Agent outputs the current action (e.g. temperature set point) based on the updated policy as a new prediction feature to the TSF model for the next prediction.

The above process can well explain why several multi-step ahead prediction (MSAP) methods are not suitable for RL offline training environment. There are three main inference methods for MSAP, i.e. the recursive method, the direct method, and the multi-input and multi-output (MIMO) method [40]. For the direct method in the RL training environment, the number of models should be equal to the time step of an episode, which consumes substantial computational resources, making the control system extremely bloated. The MIMO model is considered to be the best solution in the existing performance prediction studies [29,37]. However, this approach is not available in RL because of its inability to provide feedback for continuous control operations. Specifically, the MIMO approach would output sequence from time T to T + N directly based on the data from T-M to T-1, so the prediction model cannot give feedback when the agent gives actions from time T to T + N. In summary, the coupled loops of TSF and RL make the recursive

Table 1
Existing LSTM-based models for HVAC performance prediction.

Performance indicator	References	No. of features	Time interval	Ahead prediction	Type of model
Energy consumption	Fan [23]	27	30 min	1 day	(CNN)-(Bi)LSTM
	Kim [36]	12	1 min	60 min	CNN-LSTM
	Sendra [37]	6	15 min	4 day	LSTM
	Li [29]	7	5 min	3, 6, 24 h	Attention-LSTM
	Fazlipour [30]	1	15 min	15, 30 min	Attention-LSTM
	Dai [31]	4	60 min	60 min	Attention-BiLSTM
	Li [35]	1	1 day	1–60 day	Attention-BiLSTM
	Zhao [33]	6	15 min	15 min	dual Attention-LSTM
	Chung [32]	71	1 h	1 h	CNN-LSTM-attention
	Xiao [24]	1	1 h	24 h	LSTM
	Jang [38]	11, 15, 19	1 h	24 h	LSTM
Thermal comfort	Jiang [34]	4	5 min	5, 30, 60, 90 min	Attention-LSTM
	Elmaz [25]	6	1 min	1, 30, 60, 120 min	CNN-LSTM
	Fang [39]	6	1 h	2, 7 day	LSTM

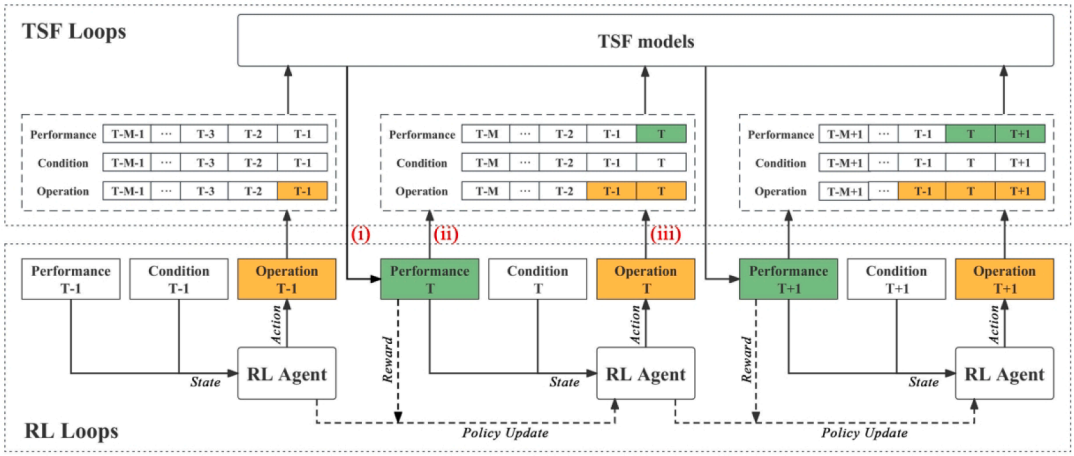


Fig. 3. Coupled loops between TSF and RL for HVAC predictive control.

method a feasible solution.

Several studies on the coupling of TSF and RL are listed below. Giuseppe Pinto used LSTM coupled with a Soft Actor-Critic (SAC) agent to manage the operation of heat pumps, chilled water and domestic hot water storage in four buildings to reduce the cost of electricity consumption while ensuring indoor temperature [41]. Zhengbo Zou [7] used LSTM coupled with Deep Deterministic Policy Gradient (DDPG) agent to control air handling units to reduce system energy consumption while ensuring thermal comfort. Both studies used recursive prediction with interacting LSTM and RL controllers. Christian Blad [42] coupled LSTM with a Q-learning-based multi-agent algorithm for online fine-tuning of HVAC systems to reduce heating costs. In this study, the direct approach was used to build 30 prediction models to predict 30 time-steps in an episode. The number of time steps was significantly less than that of mainstream studies. The prediction discontinuity due to multiple independent models is obvious in its results. They also mentioned that such a framework is more complex than any state-of-the-art solution, and measures must be taken to reduce the complexity. Some studies build prediction models for other conditions (e.g., outdoor temperature) [43,44] where the prediction process is independent of the RL training process and therefore out of the scope of this paper. It can be seen from established studies that the current mainstream coupled loops require the LSTM-based recursive prediction method. However, in the existing studies, the evaluation of the prediction models was limited to the traditional independent prediction scenario, which has limited significance in RL coupling. The focus of the discussion aims to spell out the importance of the robustness of the TSF model in RL offline training. While RL is not the only feasible solution, it is one of the promising approaches to optimise temperature set points or to be integrated with TSF predictions in other forms of MPC.

2.3. Summary of related work

In summary, a recursive approach is a viable solution for the coupled TSF and RL loops. However, considering the significant impact of recursive prediction on model robustness, the coupled TSF and RL loops for HVAC predictive control are still immature. Existing studies have yet to establish a model evaluation method in recursive prediction. In addition, although the LSTM-based algorithms show potential for robustness improvement, the optimal model architecture for HVAC has yet to be found. This study develops 16 TSF models using LSTM-based network architectures for HVAC indicators (such as indoor temperature, indoor humidity, and energy consumption) based on real-world operational data. This study then evaluates the established models to investigate the influence of neural network configurations on the model robustness, which supports the selection of the optimal model architectures. Lastly, this study couples the optimal TSF models with RL agents to optimize the operational efficiency of building facilities. The proposed new method is expected to enhance the efficiency, convergence, and optimality of operating building facilities. In addition, the study discovers new findings to the existing body of knowledge about how to leverage smart facility operations for building energy efficiency and comfort optimisation in the coupled TSF and RL loops.

3. Methodology

This study develops a data-driven predictive control method for energy efficiency and comfort optimisation, thus the control objectives include HVAC energy consumption and thermal comfort. Energy consumption is obtained constantly from monitoring the power consumption of HVAC equipment. Regarding thermal comfort, it is measured by the predicted mean vote (PMV) [45], which is the commonly used thermal comfort model [46]. Six variables are obtained as follows to enable the PMV computation. **Air temperature (ta)** and **relative humidity (rh)** are obtained directly from sensing devices and environmental monitoring equipment. **The mean radiant temperature (tr)** is 2.8 °C higher than air temperature according to section 5.3.1.2.1.b in ASHRAE Standard 55–2020 [47]. The boost in mean radiant temperature is adopted considering the actual situation of the test room. The building envelope and the exterior surfaces of the test room are mainly curtain walls with glass façades, so the influence of solar radiation on indoor thermal comfort could not be ignored. This study follows the ASHRAE Standard 55–2020 to increase the mean radiant temperature and incoproate the influence of solar radiation on comfort conditions of the test room. **Relative air speed (vel)**, **metabolic rate (met)**, and **clothing (clo)** are defined as 0.2 m/s, 1.1, and 0.61 respectively according to ASHRAE Standard 55–2020 [47] for indoor spaces which rely purely on the HVAC system and involve common office activities. This study aims to achieve optimal control of the HVAC system, and the thermal comfort parameters affected by HVAC operations are mainly indoor air temperature and humidity. Therefore, this study tries to ensure that only the above two variables are used, while other parameters remain unchanged when calculating the PMV. The proposed data-driven control aims to optimise the HVAC operational parameters to minimise HVAC energy consumption while maximising PMV.

To meet this aim, the control process first requires the sensing of necessary indoor and outdoor environmental parameters, followed by time-series forecasting (TSF) to predict the HVAC performance based on its historical and current states. Reward calculation can then be performed on the reinforcement learning (RL) agent to realise the HVAC predictive control for improved energy efficiency and thermal comfort. Fig. 4 illustrates the indoor and outdoor environmental parameters as

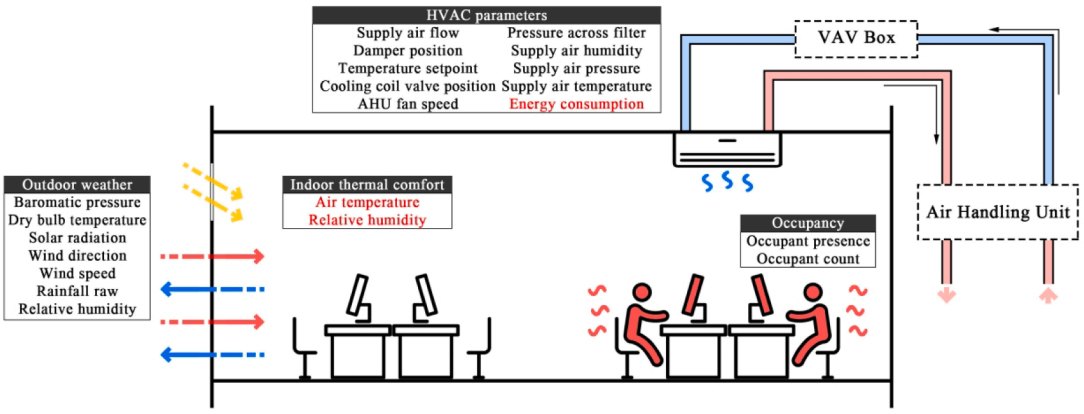


Fig. 4. Environmental and HVAC system parameters for data-driven predictive control with the target performance coloured in red.

well as the HVAC system parameters to achieve data-driven control.

- **Indoor environmental quality:** Indoor environmental quality is the fundamental parameter to support the time-series prediction and intelligent control of HVAC. In this study, the historical states of the indoor environment (such as temperature and relative humidity) are monitored and deployed as a part of inputs for the prediction.
- **Outdoor weather:** The proposed method also concerns the interaction of the outdoor environment including heat transfer through envelopes, solar radiation, and infiltration via doors and windows, which impact indoor thermal comfort. Outdoor air may also enter the air handling unit and influence the air-conditioning demand.
- **Occupancy:** Occupant heat production and equipment heat dissipation determine the heat gain and HVAC response, and therefore their impacts are considered in the proposed control method.
- **HVAC parameters:** HVAC operational parameters directly influence energy consumption, and they are the control targets in this study. As the main air exchange medium in a closed room, the supply air speed, temperature, humidity, etc. have decisive influences on indoor thermal comfort.

In this study, constant monitoring of the above-mentioned environmental parameters drives the development of TSF models and enables predictive control by coupling the optimal TSF model with an RL agent. Details of the methodology are presented separately in the following subsections.

3.1. TSF model construction

This section presents the development of TSF models based on 16 neural network architectures including BiLSTM, CNN, and Attention Mechanism (AM).

3.1.1. BiLSTM

LSTM is created to solve the gradient exploding and vanishing problems faced by conventional RNNs during long-term training [27]. LSTM adds cell state to the sequence data processing, together with the input data and the hidden state to calculate the output in each time-step. Specifically, three gating units are designed to guide information flow, namely the forget gate, input gate, and output gate. A forget gate is used to determine the proportion of information to be preserved. The flow of information is controlled using a sigmoid activation function, where all information can pass when the function output is 1, and no information can pass when the function output is 0. For any timestamp t, the output F_t of the forget gate is first calculated based on the hidden state H_{t-1} of the previous moment and the current input X_t, implying the proportion of the previous cell state retained. On the other hand, the proportion of new information is formulated through the input gate I_t. New information is transformed into the candidate cell state $C_{upd,t}$ by a tanh activation function. Up to this point, the new cell state C_t can be calculated from the original cell state C_{t-1} together with its preserve proportion F_t, and the candidate cell state $C_{upd,t}$ together with its preserve proportion I_t. An output gate O_t is used to define the preserve proportion of C_t to output the final hidden state H_t. The computational process of the LSTM network is shown in Equations (1) to (6).

$$F_t = \text{sigmoid}(W_{F1}X_t + W_{F2}H_{t-1} + B_F) \quad (1)$$

$$I_t = \text{sigmoid}(W_{I1}X_t + W_{I2}H_{t-1} + B_I) \quad (2)$$

$$C_{upd,t} = \tanh(W_{C1}X_t + W_{C2}H_{t-1} + B_C) \quad (3)$$

$$C_t = F_t{*}C_{t-1} + I_t{*}C_{upd,t} \quad (4)$$

$$O_t = \text{sigmoid}(W_{O1}X_t + W_{O2}H_{t-1} + B_O) \quad (5)$$

$$H_t = O_t{*}\tanh(C_t) \quad (6)$$

where W_F, W_I, W_C, and W_O represent the weight matrix for the calculation in forget gate F_t, input gate I_t, candidate cell state $C_{upd,t}$, and output gate O_t respectively. B_F, B_I, B_C, and B_O represent the biases for the calculation in F_t, I_t, $C_{upd,t}$, and O_t, respectively.

LSTM can only process information in the forward direction, meaning that it considers the effect of previous information on the future trend. However, the current state may be influenced by future information, such as the periodicity of building space usage and seasonal climate in HVAC control problems. In this study, BiLSTM [48] composed of a forward and a backward LSTM is leveraged. The information processed in the forward and backward directions is aggregated to output the current hidden state H_t. The two superimposed LSTMs can consider the information obtained from both the past and the future, improving the long-term dependence on learning to improve prediction accuracy.

3.1.2. Cnn

In the HVAC control problem, the prediction model contains multiple kinds of outdoor weather, operating parameters, and occupancy features. As such, feature extraction is an important step in developing high-performance predictive models. In this study, the convolutional layers are placed before the RNN layer to identify local features in sequence data for time-series data processing [25]. The original sequence data are transformed into shorter sequences of high-level features. To maintain the temporal characteristics of the information, a 1D convolution layer is used to process the feature dimension of the data, and the output new sequence allows the model to optimise the weight matrix for feature extraction. A 1D convolution layer is used to achieve the feature extraction purpose, and a dropout layer is added after CNN to avoid overfitting. The spatial features extracted from the

convolutional layer are fed into LSTM for subsequent processing.

3.1.3. Attention mechanism (AM)

The attention mechanism mimics the attentional characteristics of the human brain, focusing on more valuable details and reducing attention to less important information. The attention mechanism has been widely used in machine translation and other fields since it was proposed [49]. In this study, the attention mechanism selectively acts on the feature dimension and time dimension in time-series prediction, as the input or output side of RNN. For timestamp t, the attention mechanism is used to compute what is important for the input feature $X = [X_1, X_2, X_3, \ldots, X_n]$ at the current moment. For any feature X_i, H is the weight matrix created by the network layer, and then a score is derived by calculating the correlation between H and each input X_i using a score function. This study uses the dot product function as the score function, and its calculation process is shown in Equation (7). The next step is to use the activation function softmax to normalize these scores, the results of which constitute the attention distribution A_i of H on the input X_i. The normalization process is shown in Equation (8). Finally, according to the attention distributions, information can be selectively extracted from the input information X. The most common information extraction is to weigh the input information according to the attention distribution, in order to sum up the context reflecting what the model should focus on currently, as shown in Equation (9).

$$s(X_i, H) = X_i^T \mathrm{H} \tag{7}$$

$$\mathrm{a_i} = \mathrm{softmax}\left[s(X_i, H)\right] \tag{8}$$

$$\mathrm{context} = \sum_{i=1}^{n} a_i * X_i \tag{9}$$

3.1.4. Model framework

This study proposes 16 LSTM-based model architectures, which contain different combinations of bi-directional processing, convolutional processing, and attention mechanisms above-mentioned. Fig. 5 shows the proposed model framework, in which each structure can take LSTM or BiLSTM as the core layer, and eight structures in the figure form 16 neural network architectures. Two attention layers for the temporal dimension (Time AM) and feature dimension (Dim AM) are included. Referring to the way temporal features are stored in the network, a dimensional transpose is added to the temporal dimension attention layer before calculating the scores, and the reverse transpose is done after processing so that the data structure remains unchanged. In addition, attention layers as input and output of recurrent layer are included, when the attention layer is used after LSTM output, an extra Flatten layer is added to reduce the data dimension. A 1D convolution (Conv1D) layer is used as the pre-processing stage of the temporal data to achieve feature extraction. The 16 LSTM-based architectures are tested to identify the optimal TSF model.

3.2. Evaluation of TSF model robustness

3.2.1. Evaluation scenarios

In this study, three evaluation scenarios are set, namely independent prediction scenario, short-term recursive prediction scenario, and long-term recursive prediction scenario (see Table 2). The **independent prediction scenario** is the same as the evaluation scenario in most of the current studies. Each prediction is made exclusively using the original data from the validation set. The **short-term recursive prediction scenario** is designed for TSF model robustness under RL offline training process. The validation set is first sliced into several independent parts by the number of timestamps of one episode in RL training. Next, in each part, successive predictions are executed with recursive prediction mode, i.e., using each prediction result to overwrite the corresponding

Table 2
Comparison of three evaluation scenarios.

Scenario	Refreshing dataset	Resetting dataset	Role in RL agent
Independent prediction	No		Baseline
Short-term recursive prediction	Yes	One episode	Agent training
Long-term recursive prediction	Yes	Whole dataset	Agent implementation

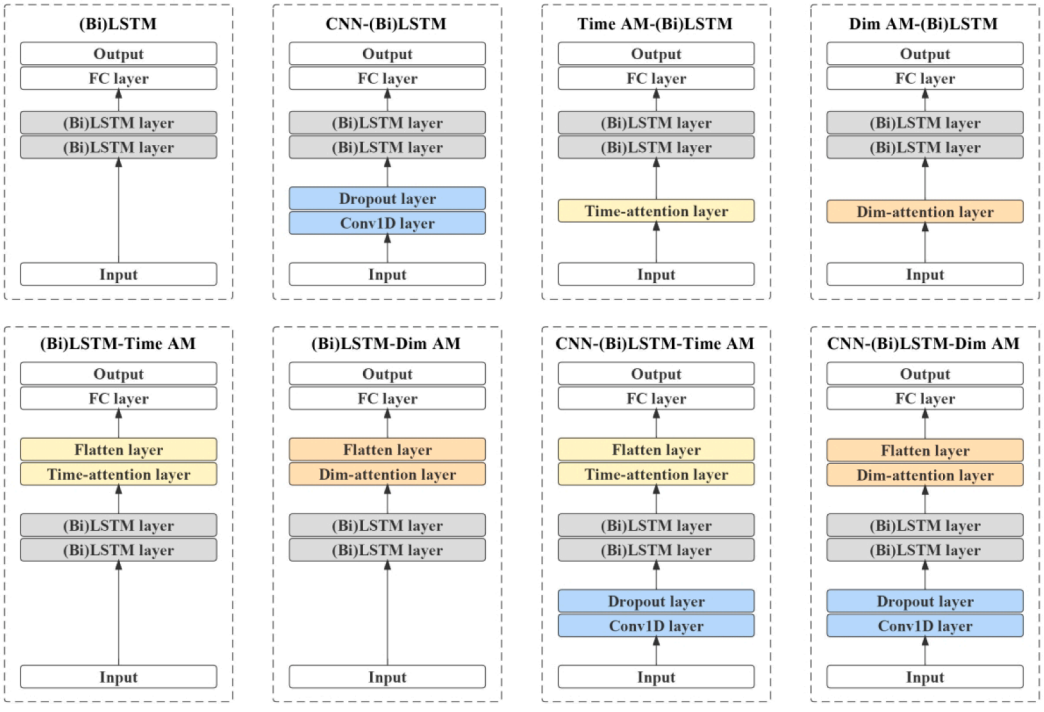


Fig. 5. Proposed model framework for HVAC performance prediction. Eight structures form 16 architectures with LSTM or BiLSTM as the core layer.

position of the validation set until the last prediction in each episode is finished. Next, the whole validation set is reset to its initial state. The above process repeats until the entire validation set is predicted. The **long-term recursive prediction scenario** is designed for continuous recursive prediction to ensure that its performance remains stable during the implementation of the reinforcement learning agent. The validation set is not divided in this scenario, and the prediction results overwrite the corresponding position of the dataset, iterating until the validation completes. Model prediction performance in each scenario is evaluated using multiple metrics, and model robustness in recursive prediction is quantified by calculating the variation of the metrics among different scenarios. For the illustrative example in this study, the validation set contains 4320 timestamps, and one RL episode corresponds to 288 timestamps. Therefore, short-term recursive evaluation splits the validation set with 288 timestamps, executing recursive prediction in each slice respectively for fifteen rounds of validation. The long-term recursive evaluation uses all 4320 timestamps to perform recursive prediction for one round of validation.

3.2.2. Evaluation metrics

The model robustness evaluation is conducted based on the variation of the same index under different prediction scenarios. Mean absolute error (MAE), root mean square error (RMSE), and success rate (SR) are used to measure the model accuracy. MAE and RMSE are commonly-used metrics for evaluating time-series-forecasting models, which are both related to the magnitude, and their calculation processes are shown in Equations (10) and (11). SR is an additional percentage metric to measure how likely the established model is to keep the error within an acceptable range. Tolerance here refers to the acceptable error range for the prediction of HVAC performance. The calculation for SR is shown in Equation (12). The reason why mean absolute percentage error (MAPE) is not selected as a dimensionless metric is that the real-time energy prediction would appear with many values that converge to zero, in which cases MAPE may lose interpretability. Lower MAE and RMSE represent higher model accuracy, whereas higher SR represents higher model accuracy.

$$MAE(y,\widehat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \widehat{y_i}| \tag{10}$$

$$RMSE(y,\widehat{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \widehat{y_i})^2} \tag{11}$$

$$SR(y,\widehat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}1(|y_i - \widehat{y_i}|\langle tolerance\rangle)*100\% \tag{12}$$

where 1(x) represents the indicator function, that is, the subset of all prediction results where the prediction error is smaller than the tolerance. Tolerance refers to the acceptable range, and in this study, indoor temperature tolerance is 0.5℃, relative humidity tolerance is 5%, and energy consumption tolerance is 1 kWh. The settings of tolerance take into account value ranges and the comparability of SR calculation results.

3.3. RL agent

3.3.1. Soft Actor-Critic (SAC) algorithm

An RL agent is performed to realise the HVAC control for improved energy efficiency and thermal comfort. In previous relevant studies, DDPG has been implemented in HVAC optimal control [44,50,51]. However, DDPG is sensitive to hyperparameters, and its performance is difficult to generalise to other problems. As such, Tuomas Haarnoja proposed the SAC algorithm [52], which supports continuous action space, high data utilisation efficiency, and low hyperparameter dependence. For HVAC control, SAC has been proven to be more stable in balancing temperature and energy with a smaller amount of data compared to other RL algorithms [53] and therefore is leveraged for training the RL agent in this study.

SAC leverages the idea of maximum entropy, which is interpreted as the degree of chaos and randomness. The higher the entropy, the more chaotic it is and the more information it contains. If a variable $\times$ obeys the distribution P, the entropy H(P) of $\times$ is calculated using Equation (13). The benefits of entropy are that the policy can be made as random as possible, as the agent explores the state space S and avoids the policy falling into a local optimum. The policy calculation with entropy maximisation is shown in Equation (14), where ρ_π denotes the distribution obeyed by the state-action pair that the agent encounters under the control of policy π, α is a hyperparameter named temperature coefficient that adjusts the importance given to the entropy. In the implementation of SAC, the value function and the policy are each fitted by a neural network. The function receives the input state action on a state-action pair and outputs value; the strategy receives the input state and outputs a distribution about the action as the strategy. When an action is needed, a Gaussian distribution with mean value and the standard deviation is sampled, and the sampling result is used as the decision action of the strategy. Further details can be obtained from [54].

$$H(P) = E_{x\ P}[-logP(x)] \tag{13}$$

$$\pi^*_{MaxEnt} = argmax_\pi \sum_t E_{(s_t,a_t)\ \rho_\pi}[r(s_t,a_t) + \alpha H(\pi(\bullet|s_t))] \tag{14}$$

3.3.2. Agent offline training

The optimal TSF models selected based on robustness evaluation are used to build an interactive training environment for RL agent offline training. OpenAI GYM platform in python is selected to build the training environment. The RL training environment needs to dynamically output states and rewards while receiving actions. To meet this purpose, the dataset features are classified into performance (e.g. energy consumption), operation (e.g. supply air flow), and other conditions (e.g. outdoor temperature) Performance and other conditions constitute the states, whereas HVAC operations are the actions received from the agent. For the current moment t, S_t is first extracted from the dataset and then input into the agent to receive action A_t. A_t subsequently replaces the action data of time t in the dataset to enable the prediction of HVAC performance. The prediction result P_{t+1} replaces the performance data of time $t + 1$ in the dataset, and then S_{t+1} is extracted from the new dataset for the next iteration. Appendix A shows the complete TSF-integrated RL offline training environment.

The reward function is designed to achieve the best trade-off between HVAC energy consumption and PMV values, and its overall calculation is shown in Equation (15). The reasons for using PMV values rather than uncomfortable hours in this study are as follows: As a basis, thermal comfort under existing control is in a very bad state. In this case, the uncomfortable hour cannot be used as an RL reward indicator, because RL training is to gradually optimise itself by exploring feasible solutions, and the reward function must be able to identify minor environmental improvements. If RL successfully reduces PMV from 1 to 0.8, this improvement cannot be recognised by uncomfortable hours but can be recognised by PMV values. Only by giving the agent clear feedback on all kinds of operations can the agent be guided to continuously optimise itself. The control is divided into occupancy and non-occupancy cases. The occupancy period should satisfy either the daily operating hours or the occupancy state captured by the monitoring equipment. For the occupancy period, both energy consumption and thermal comfort are controlled, and their rewards are weighted and summed with weights α, β summing to 1. Multiple weight combinations are evaluated in the implementation stage. For the non-occupancy period, the energy consumption weight is set to 1 to ensure the same max reward. In addition, since the reward range for both performances is limited to [-1, 0] (described in detail in the next paragraph), the reward is shifted and scaled up so that the range of values is [-8, 2]. A reasonable number of reward levels and a certain degree of positive and

negative reward allocation are conducive to the convergence of the RL agent [55].

$$R = \begin{cases} (\alpha^* R_E + \beta^* R_T + 0.2)^* 10 | occupancy = 1 \\ (R_E + 0.2)^* 10 | occupancy = 0 \end{cases} \quad (15)$$

Since weights are assigned to sub-rewards, the sub-reward value should range the same with [-1, 0]. The normalised energy is used for sub-reward calculation according to Equation (16). Regarding thermal comfort, the PMV and auxiliary parameter settings are applied. With the indoor temperature and relative humidity, the pythermalcomfort library in python is used to compute the PMV in a real-time fashion. According to ASHRAE Standard 55–2020 [47], the indoor thermal comfort acceptable range is $-0.5 < \text{PMV} < 0.5$, so the rewards for PMV within this range are set to the highest reward 0. In addition, considering the fluctuating range of PMV in our dataset, the PMV rewards over 1.5 are set to the lowest reward −1. A smooth linear transition is established in [0.5, 1.5] as the main control range. The complete thermal comfort calculation process is shown in Equation (17).

$$R_E = -normalize(energyconsumption) \quad (16)$$

$$R_T = \begin{cases} 0 | abs(PMV) \le 0.5 \\ -abs(PMV) + 0.5 | 0.5 \le abs(PMV) \le 1.5 \\ -1 | abs(PMV) \ge 1.5 \end{cases} \quad (17)$$

4. Illustrative example

4.1. Study area and IoT sensor description

The proposed new methods are illustrated via a case study on an IoT-enabled smart building at the National University of Singapore. An office spanned over 141.9 m^2 with a capacity of 25 occupants is selected. The office is equipped with a dedicated VAV system supplying an airflow rate of 3,192 CMH to maintain the room temperature. It is air-conditioned by an Air Handling Unit (AHU) with a total supply airflow rate of 14,560 CMH, providing chilled air to eleven other rooms in the same building. The HVAC operating hours for the room are from 08:30 to 18:40. A building management system (BMS) is currently deployed to monitor and manage the building's mechanical and electrical systems. As part of BMS, miscellaneous types of IoT sensors are installed to automatically collect information about the building's energy consumption, HVAC operations and outdoor weather conditions. The BACnet Protocol is used to retrieve these sensor measurement data to be stored in the PI Data Archive. It should be noted that despite the deployment of BMS to collect information about the building's room-level HVAC power consumption data, this is not always feasible due to the building's cooling system configuration and the need to deploy sensors for each air handling unit servicing each room within the building. The energy consumption data in this study combines two measurement items namely "Chilled water energy" and "AHU fan energy". Standalone IEQ sensors are installed to measure the indoor environment, and Wi-Fi Routers to count the occupancy using the number of newly connected devices via Wi-Fi. An independent article about the dataset named ROBOD has been published in this regard [56]. The data of all working days from September 7, 2021 to April 9, 2022 (120 days) are measured and collected at a sampling frequency of 5 min. The initially screened parameters and their value ranges are shown in Table 3.

4.2. Data processing

The dataset contains a total of 34,560 timestamps, each containing 19 features. Data with anomalies are removed from the original dataset. Missing value refill and outlier replacement, caused by an intermittent sensor failure, are then performed. The missing value is refilled using the previous timestamp data, and a total of 13 items of indoor temperature and relative humidity are filled. For outlier determination and replacement, a reasonable range of values for each feature is set, and data that are outside the acceptable range are replaced with data from previous timestamps. Four outliers in HVAC energy consumption data are recognized and replaced. For the IEQ data (indoor air temperature and relative humidity) to be predicted, data smoothing is carried out to improve the model convergence. The indoor environment is affected by a variety of uncontrollable features and by the HVAC system with time delay. Data smoothing can eliminate the above influence to a certain extent. The moving average method as shown in Equation (18) is used to process the dataset. To preserve the original information as intact as possible, the moving window radius is set to 1. The average value is calculated using data at the current moment, and one time-stamp before and after (10 min in total). For energy consumption data, the measured HVAC operating parameters can explain most of the data changes. Their influences are direct and without time delay. Therefore, energy consumption data is directly used for TSF models without data smoothing.

$$p_t = \frac{\sum_{i=1}^{n} (x_{t-i} + x_{t+1}) + x_t}{2n + 1} \quad (18)$$

where p_t represents the filtering result at moment t, x_{t-1} represents the observation at moment $t-1$, and n represents the moving window radius.

Feature selection is performed on the collected outdoor weather parameters and HVAC operation parameters, and the selection process is completely based on the results of statistical analysis. For the outdoor weather, the linear regression method is used to test the fitted relationship between the outdoor parameters at moment t and the three performance indicators at moment $t + 1$. The significance index p-value ($p < 0.01$) is used, and the VIF value is also checked for multicollinearity among the parameters. Finally, one parameter, wind speed, is removed and five weather parameters are retained. For the HVAC operation parameters, we believe that each HVAC operation parameter with low correlation with other parameters may carry effective information which is helpful to the prediction of indoor environment and energy consumption. Therefore, the Pearson correlation coefficient is used to test, and the parameters with an absolute value of correlation coefficient greater than 0.9 are set to have multicollinearity problems. The visualization of the correlation analysis results is shown in Fig. 6. The results show that six AHU parameters (cooling coil valve position, AHU fan

Table 3
Initially screened parameters from the sensors.

Parameter type	Parameter	Range	Parameter	Range
IEQ	Air temperature (℃)	25.1–29.7	Relative humidity (%)	59.7–88.8
Occupancy	Wi-Fi connected devices (Number)	1–13		
Outdoor weather	Barometric pressure (hPa)	997.6–1008.6	Dry bulb temperature (℃)	22.5–35.5
	Horizontal solar radiation (W/m^2)	0–1288.5	Wind speed (m/s)	0–7.7
	Relative humidity (%)	39.8–100	Rainfall raw (mm)	0–24.7
HVAC parameters	HVAC energy consumption (kWh)	0–12.2	Damper position (%)	0–100
	Supply air flow (CMH)	0–1546.4	Cooling coil valve position (%)	7.6–100
	Temperature setpoint (℃)	25.2–28.0	Offcoil air temperature (℃)	15.9–28.3
	AHU fan speed (Hz)	0–41.2	Pressure across the filter (Pa)	0–108.0
	Supply air pressure (Pa)	0–157.6	Supply air temperature (℃)	17.8–28.9

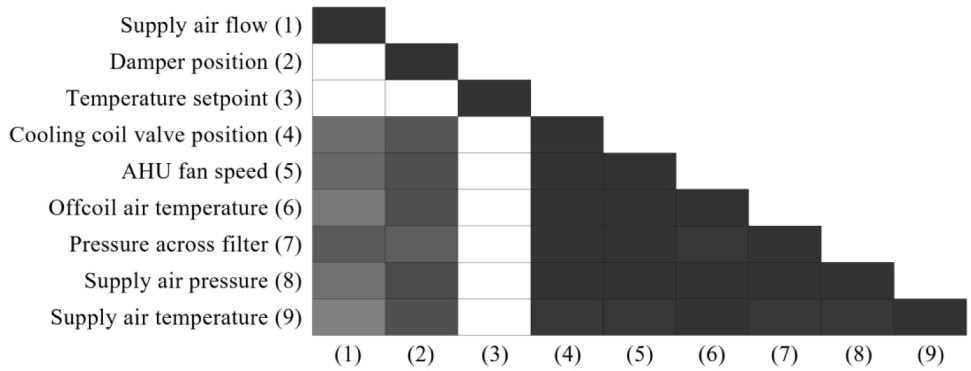


Fig. 6. Visualisation of Pearson correlation coefficients among HVAC operation parameters. The white block presents the coefficient in 0 and the black block presents the absolute value of the coefficient in 1.

speed, off-coil air temperature, pressure across filter, supply air pressure, supply air temperature) have co-collinearity with each other. Finally, only the AHU fan speed was preserved, together with temperature setpoint, damper position and supply air flow for further training.

Based on the dataset completed by the above processing, the coupled loops of RL and TSF for HVAC predictive control were carried out. TSF model training is first carried out using the established model framework towards three HVAC performances. The prediction features including HVAC operation parameters, time parameters, outdoor weather parameters, occupancy parameters, and the prediction target itself. It is important to emphasize that, given the need to implement recursive prediction, the other two performance parameters are not included in the prediction model for each performance to reduce the dependence on past prediction results. The trained prediction models will be evaluated for model robustness, and the selected optimal models will be used to build the offline training environment. For the RL agent, indoor environmental quality and other condition parameters will be used as states and HVAC operation parameters as actions. The roles of all parameters in the coupled loop are shown in Table 4.

5. Experimental results and discussion

5.1. TSF model evaluation

As shown in Table 4, each TSF model includes 12 features which are 4 HVAC operation features, 7 other condition features, and 1 prediction target feature. The prediction period in this study is the same as the operating frequency of the HVAC controller, so we choose a shorter prediction period to ensure the immediate response of the control system to environmental changes. For any moment t, the model outputs the performance for moment $t + 1$ (after 5 min). We set the lookback time to 24 timestamps, i.e., the model identifies the parameter changes within the past 2 h to make the performance prediction for 5 min later. We have tried using a longer lookback time, e.g., the past 288 timestamps (1 day), and found that although the model indicates a small improvement in independent prediction accuracy, it performs poorly when coupled with RL offline training. One possible reason is that the model builds a stronger dependence on periodic features (e.g., daily fixed operating patterns in the dataset) and weakens the sensitivity to short-term parameter changes (e.g., adjusting supply air flow). The training process divided the dataset for 6:1:1, i.e., the first 90 days (25,920 timestamps) were selected as the training dataset, 91–105 days (4320 timestamps) as test dataset, and 106–120 days (4320 timestamps) as validation dataset. It should be emphasized that this study uses fixed datasets for model training, testing and validation. From a practical perspective, this is because we always hope that the model can gain experience from the previous data to make predictions and decisions about future operations. Such a model is of practical significance. From a model perspective, the dataset cannot be rearranged because the existing dataset contains long-term and short-term dependencies that are helpful for model training. The three prediction targets in the first 10 days of the validation set are shown in Fig. 7.

Table 4
Parameter roles in coupled loops of RL and TSF.

Role in RL	Parameters	Role in TSF
State	Indoor air temperature, Indoor relative humidity, HVAC energy consumption	Target (Feature for itself)
	Current hour, Wi-Fi connected devices, Barometric pressure, Dry bulb temperature, Horizontal solar radiation, Relative humidity, Rainfall raw	Feature
Action	Damper position, Supply air flow, Temperature setpoint, AHU fan speed	

To ensure the consistency of the evaluation process, we designed standardized neural network layers for different types of neural network architectures. As mentioned in the previous section, three main types of neural network layers are included in the design, namely RNN layers, CNN layers, and Attention layers. For the BiLSTM, a two-layer neural network architecture is used, and the number of neurons in each layer is set to 64. It is tested that two LSTM layers with 64 neurons provide more stable convergence speed and prediction performance when working together with other types of neural network layers. For the CNN layer, a 1D convolution layer (Conv1D) is used. One convolutional layer is stacked with 64 kernels, kernel sizes of 3, and stride of 1. Padding is set to "same" to preserve temporal dimensions. A dropout layer with a scale of 0.3 is designed after convolutional processing to avoid overfitting. For the attention layer, a fully connected layer with a softmax activation function is used to calculate the weight of inputs and normalize them. Next, inputs are assigned importance weights using matrix multiplication. For the processing of temporal and feature dimensions, since the dot product operation is based on the first dimension of a two-dimensional matrix, the data dimensions are transposed before the attention calculation for the feature dimension and transposed again after the operation to reset to the original arrangement. For the attention layer before or after RNN layers, when the attention layer is placed after the RNN layer, an additional flatten layer is added to transform it into 1D before outputting results.

We use the TensorFlow2 framework in a Python environment to perform the training of the prediction models. MSE is used as the loss metric for the training process, and the Adam optimizer is used to update the network weights. The batch size is set to 64, the epoch is 100, and the learning rate is 0.001. The model checkpoint is also set to automatically save the best model, i.e., the prediction model with the best performance in the test dataset is always saved instead of the last trained prediction model. The model training is implemented in a cloud computational environment Kaggle [57]. The training environment provides a CPU

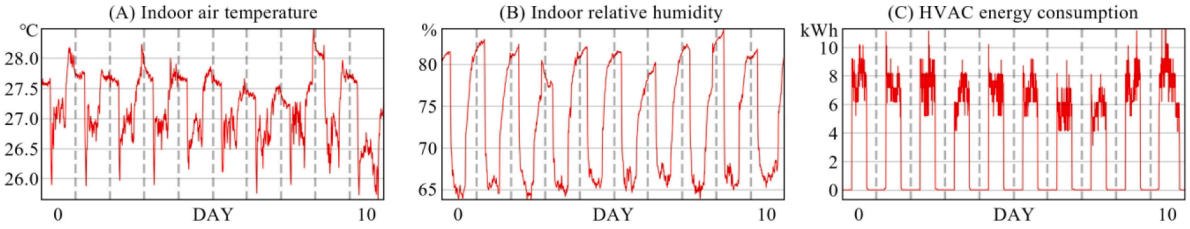


Fig. 7. Three HVAC performance indicators in the first 10 validation days.

with 4 cores and 16G RAM, together with a single GPU of NVIDIA Tesla P100. The time to complete the model training for two parallel operations in the above platform is 15–20 min.

5.1.1. Independent prediction evaluation

Fig. 8 shows the MSE distribution of the test dataset over 100 training iterations. There are 48 prediction models corresponding to the implementation of 16 neural network architectures on three HVAC performances. All the models converge well to a steady state during the training process, and the best prediction accuracy of every single performance is similar. Since the data used to calculate the MSE are normalized, the accuracy distributions of different types of performance can be compared. It can be seen that the overall accuracy of the energy consumption prediction model is lower than that of the IEQ. The first reason is that the IEQ data are smoothed before the training process, some of the sharply varying sample errors are eliminated, and a more continuous distribution is also beneficial for model training. In addition, similarly, the energy consumption data under the current control strategies present abrupt data changes at the beginning and end of the daily office operation time, which is not conducive to the identification of the RNN model. If the prediction model identifies the abrupt change time earlier or later, it will make a small number of results around the abrupt change time suffer very large errors, which have a greater impact on the overall prediction accuracy. Another significant trend is that although a dropout layer has been added, the CNN-LSTM and CNN-BiLSTM models in the air temperature prediction model showed a significant accuracy decrease after about the 80th generation of training, which indicates that using only the CNN layer coupled with the RNN layer would increase the risk of model overfitting. In this study, due to the automatic determination of model checkpoints, the models used in the following sections were the models corresponding to the lowest points of the respective loss curves in the figure.

The independent prediction performance is first analysed, and the RMSE value of the normalized validation data is used as the main metric here. Table 5 shows the top three model architectures for the three HVAC performances. Overall, the model architectures incorporating the 1D convolutional layer did not achieve high prediction accuracy in any of the independent prediction scenarios. This study did not use multiple stacked CNN layers and a larger number of filters, while the contribution

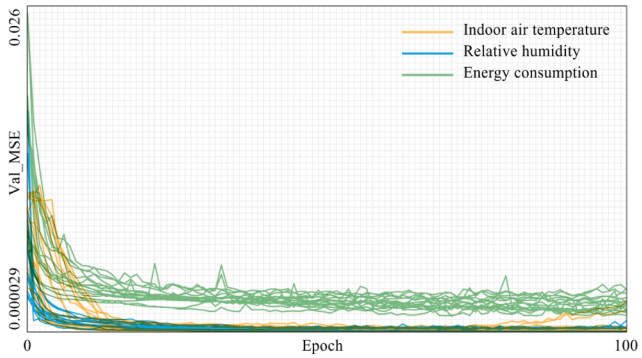


Fig. 8. MSE loss distribution for the testing data during the training process.

Table 5
Top 3 models and their validation RMSE in independent prediction scenario.

Performance	Architecture	RMSE
Air temperature	BiLSTM-Time AM	0.0080
	Dim AM-LSTM	0.0094
	LSTM-Time AM	0.0095
Relative humidity	LSTM-Time AM	0.0066
	LSTM-Dim AM	0.0067
	BiLSTM-Time AM	0.0069
Energy consumption	LSTM	0.0452
	BiLSTM-Dim AM	0.0461
	Dim AM-BiLSTM	0.0484

of CNN layers in improving training efficiency is emphasized in the existing literature [23]. In addition, the use of parallel CNN has been shown to more fully exploit the feature extraction capabilities of the convolutional layers [32]. The significant contribution of the attention layer to prediction accuracy in independent prediction scenarios is obvious, and almost all of the high-accuracy models have an added attention layer. The attention mechanism for feature dimension is more effective when it is used as the input of RNN, and the processing for both time and feature dimensions can work well when used as the output of RNN.

5.1.2. Recursive prediction evaluation

In recursive prediction scenarios, the short-term recursive prediction period is set to 1 day (288 timestamps) since the length of one episode for RL offline training is 1 day in this study. The long-term prediction refers to the dataset division result, and the entire validation dataset (15 days, 4320 timestamps) is used for the long-term recursive prediction scenario. Fig. 9 shows the variation of validation RMSE for each model under three prediction scenarios. It is clear from the figure that the error of the model rises significantly after the change from independent to recursive prediction, which is also widely recognized by existing studies. However, different model architectures have dramatic effects on recursive prediction accuracy and may show several-fold differences for the same prediction target, which is quite different from the trend found in the independent prediction that model accuracies are similar for a different architecture. Moreover, the accuracy evaluation results in independent prediction cannot be generalized to recursive prediction at all: the model with the highest independent prediction accuracy may show poor performance in the recursive prediction process. The above phenomenon also validates the significance of this study to some extent. In general, it is unreasonable to directly use a TSF model with high independent prediction accuracy to couple with an RL agent, because the TSF model predicts recursively during the RL training process, and the recursive prediction accuracy always differs from independent prediction accuracy. A high-performance model in independent prediction may have significant performance degradation in the RL coupled loops.

The independent prediction accuracy of temperature and humidity is higher, but their decay in recursive prediction scenarios is also greater. Conversely, the independent prediction accuracy of the energy consumption model is lower, but the accuracy decay in recursive prediction is less. Collectively, the normalized RMSEs of the optimal models all fell

Fig. 9. Variations of normalized validation RMSE for each model under three prediction scenarios.

into the range of 0.07–0.09. For the comparison of long-term and short-term recursive prediction, the long-term accuracy of IEQ shows further decay compared with short-term accuracy, but for HVAC energy consumption, the accuracies are almost the same. The reason is that the energy consumption data under the current control strategy follows a strong daily periodicity, with frequent fluctuations in the high level during the operation time, and a near-zero state for more than 10 h (120-time stamps) at night, which is several times longer than the model lookback time (2 h). Even if the recursive prediction accuracy of some models is low, such long silent periods can be recognized, and the data in the silent period gradually overwrite the historical energy consumption, achieving the effect of resetting the energy consumption input data for prediction models.

For the model architectures, a significant improvement of the CNN layer on the robustness of IEQ prediction can be seen. All the temperature and humidity prediction models with optimal prediction performance contain an antecedent CNN layer. Moreover, the analysis in the previous paragraph confirms that the bidirectional processing does not significantly affect the humidity-independent prediction performance, but it indicated an additional boost in its recursive prediction. Focusing on all CNN architectures, the attention mechanism again plays a role in recursive prediction and deploying both CNN and attention layers outperforms CNN-LSTM. And the attention layer for the feature dimension slightly outperforms the temporal dimension. For energy consumption prediction, the changing pattern is similar to that of the independent prediction scenario: including the CNN layer does not play a significant role and the BiLSTM network is more advantageous. It is worth mentioning that the LSTM model performs best in independent prediction scenarios, but decays more in the recursive context.

5.1.3. *Optimal TSF models*

To further compare the overall performance of the optimal models, three models with the best short-term recursive RMSE were selected as candidates. Their complete metric distributions are listed in Table 6. Since the purpose of model development is to train RL agents for HVAC control, it is also crucial for the prediction success rate (SR) measure, especially when making a selection of similar models. It can be seen from the table that for air temperature prediction, although the CNN-LSTM-Dim AM architecture performs better in conventional evaluation, its SR (error$<$0.5℃) is lower than that of the CNN-LSTM-Time AM architecture. The same phenomenon can also be seen in the comparison of Dim AM-BILSTM architecture and BiLSTM-Dim AM architecture for energy prediction.

The prediction results of the candidate model for the first three days

Table 6
Evaluation metrics of candidate models.

Performance	Architecture	Scenario	MAE	RMSE	SR
Air temperature	CNN-LSTM-Dim AM	Independent	0.013	0.019	100
		Short-term	0.065	0.089	77.535
		Long-term	0.067	0.090	76.493
	CNN-LSTM-Time AM	Independent	0.013	0.019	100
		Short-term	0.067	0.094	80.625
		Long-term	0.076	0.100	75.208
	Time AM-LSTM	Independent	0.010	0.014	100
		Short-term	0.077	0.096	74.201
		Long-term	0.083	0.102	71.563
Relative humidity	CNN-LSTM-Time AM	Independent	0.015	0.021	99.965
		Short-term	0.054	0.076	94.236
		Long-term	0.059	0.079	94.236
	CNN-BiLSTM-Dim AM	Independent	0.018	0.024	100
		Short-term	0.066	0.090	92.014
		Long-term	0.072	0.094	91.806
	CNN-BiLSTM-Time AM	Independent	0.014	0.020	100
		Short-term	0.068	0.109	93.056
		Long-term	0.072	0.113	92.257
Energy consumption	Dim AM-BiLSTM	Independent	0.024	0.048	91.875
		Short-term	0.034	0.073	87.431
		Long-term	0.034	0.073	87.431
	BiLSTM-Dim AM	Independent	0.025	0.046	92.708
		Short-term	0.035	0.076	88.368
		Long-term	0.035	0.076	88.368
	LSTM-Time AM	Independent	0.024	0.050	86.944
		Short-term	0.036	0.084	80.104
		Long-term	0.036	0.084	80.104

Note: Bolded represents the selected optimal models.

of the validation set are shown in Fig. 10. Model performance can be better recognized from some key error points. For the indoor temperature, the error of the Time AM-LSTM model is often reflected in the inability to make accurate judgments for higher night time temperatures, i.e., failure to quickly return the temperature to an unconditioned state. The error of ANN-LSTM-Dim AM is the persistently high prediction of indoor temperature during some cooling periods, which is adverse for RL agent training. For relative humidity, it can be seen that the CNN-BiLSTM-Time AM model is unable to correctly identify the rapid change from a high point to a low point during cooling time, and it tends to drop the humidity earlier, while the CNN-LSTM-Dim AM model tends to drop the humidity too much in the afternoon. For energy prediction. Dim AM-BiLSTM architecture overestimates the system energy consumption at the cooling starting point. Finally, combining the analysis of evaluation metrics and prediction results, the CNN-LSTM-Time AM model was used to predict indoor air temperature, the CNN-LSTM-Time AM model to predict indoor relative humidity, and the BiLSTM-Dim AM model to predict system energy consumption. The above models will be used for the subsequent training of the RL agent. The reason for selecting different models to predict different objectives is that one TSF model outputting multiple indicators will lead to a significant decline in prediction accuracy due to the lack of targeted model architecture design. Only the high-performance model-driven surrogate environment can truly replace the real environment for RL offline training.

5.2. RL agent for Data-Driven control

This study used the SAC to train an RL agent for HVAC optimal

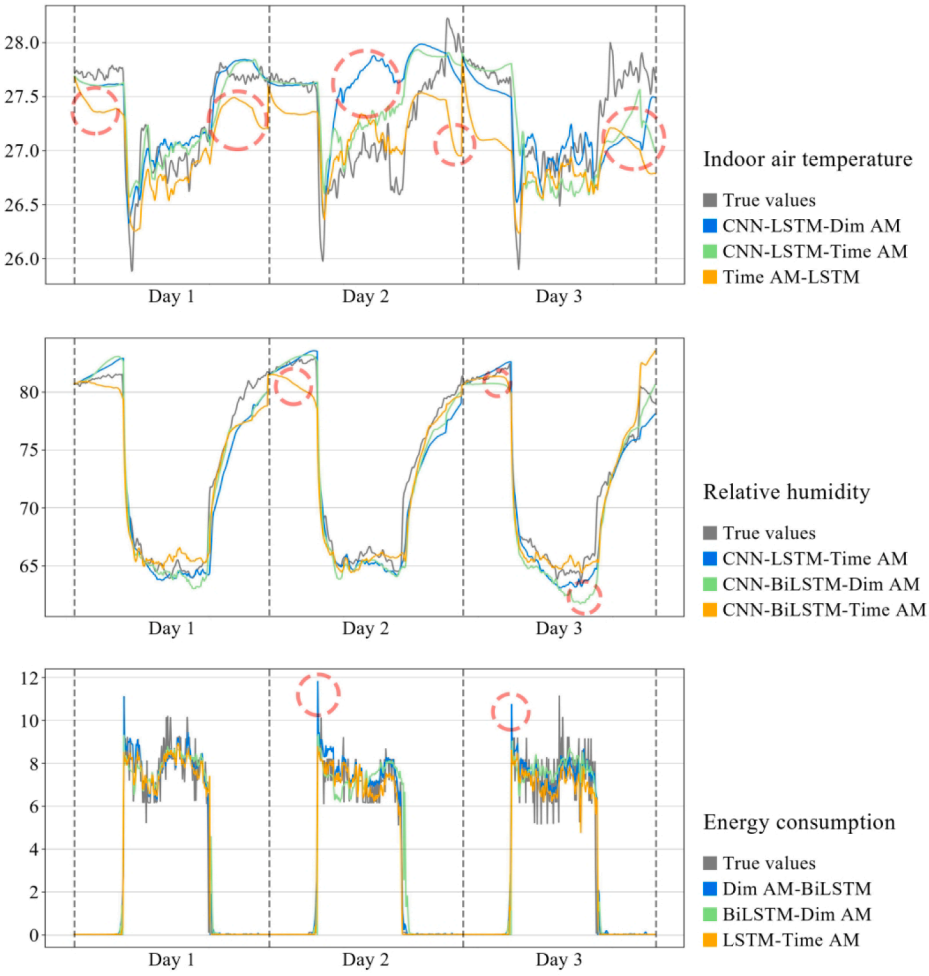


Fig. 10. Prediction results of the candidate models for the first 3 days of the validation set with key error points marked in red circles.

control. Two-layer neural networks are used, with 256 neurons in the hidden layer and a batch size of 256. The learning rate of each neural network is set to 0.0003. The discount rate is set to 0.99. The temperature coefficient starts at 1 and ends at 0.1. We use the entire 120-day dataset to train the controller, with a training period of one episode per day (288 timestamps), for a total of 120 episodes. The training is also implemented in the Kaggle cloud platform, using the same configuration as the prediction model training. Under the above conditions, it takes about 2 h to complete the training of the RL agent for all 120 episodes in a two-line execution.

Referring to the reward function equations (16) and (17), we identified occupied and non-occupied conditions, where the occupied condition implements the control of both energy consumption and thermal comfort, and the non-occupied condition implements the maximum energy saving control. We assigned weights to energy consumption (α) and thermal comfort (β) in the occupied condition, nine weight combinations from 0.1 to 0.9 for α (corresponding to 0.9 to 0.1 for β) were tried. For model decisions, the first should be whether the corresponding parameter combinations can make the agent controller converge to a stable state. We observe the training results for 120 days and find that the RL model cannot converge smoothly when the energy consumption weight is greater than 0.3, so the above parameter combinations are discarded. For the three models with energy consumption weights of 0.1, 0.2, and 0.3, we calculated the average energy consumption and the occupied average PMV of the environment in 100–120 days, and the calculation results are shown in Table 7. Since the PMV weights are all larger than the energy consumption weights, the obtained result data are more inclined toward thermal comfort optimization compared with the existing studies. It is not possible to judge the optimal model directly because this is a multi-objective optimization problem with multiple Pareto front solutions. Fig. 11 shows the Pareto front of this problem. We finally chose the controller with an energy consumption weight of 0.2 and a PMV weight of 0.8 due to its best overall performance and convergence.

The convergence of the selected model throughout the training process is shown in Fig. 12, with the blue line recording the sum of the cumulative rewards for the past 5 episodes. Referring to the design of the reward function in the previous section, the range of rewards that can be obtained for each timestamp is [-8, 2] and the cumulative rewards for each episode are [-2304, 576]. The training process of the agent as a whole is in a stage-wise ascending state. The training process shows fluctuations in the early stage and gradually stabilizes in the 50th generation. The subsequent training process falls into local optimal solutions twice and finally reaches an equilibrium state after 90–100 generations.

The difference between the surrogate environment under RL agent control and the original dataset under existing control is compared after the RL agent reaches stability. Data for episodes 110–115 are extracted for visualisation. The energy consumption as well as the PMV variations and their comparison with the existing controller are presented as shown in Fig. 13. Compared to the existing control strategy, the RL controller turns on the cooling system in advance of the daily operation time. When operation time begins, the HVAC system under the existing controller quickly reached the peak energy consumption, but the RL controller will gradually supply cooling, thus reducing the system energy consumption and ensuring a more stable thermal comfort. On some days, the RL agent

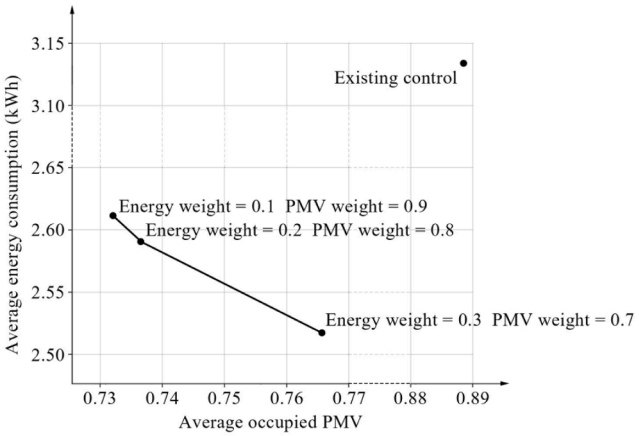


Fig. 11. Pareto front indicating the impact of different weights.

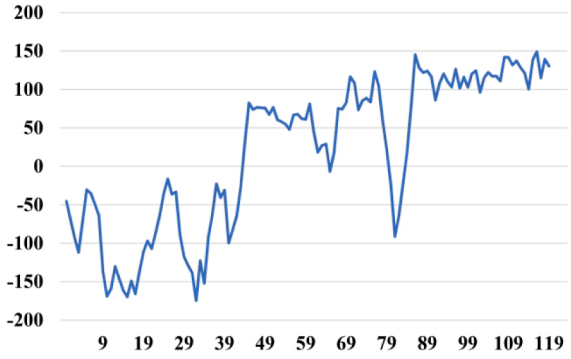


Fig. 12. RL agent convergence of 5 episodes rewards.

can also be found to end the cooling operation early, using indoor thermal storage to ensure thermal comfort for the rest of the operation time. Overall, the RL controller presents a more stable thermal comfort control, while avoiding established controllers from dropping the room temperature too much in the initial phase. In terms of energy saving, the RL agent shows the flexibility of intelligent control for the ability to break through the limits of operation time. Early preheating and early shutdown operations are not possible using traditional control methods. The built RL controller in this study finally saved 17.4% energy consumption and improved by 16.9% thermal comfort compared with the existing controller. It is worth emphasizing that the comparison is focusing on the RL intelligent agent and the original PID control. The use of other intelligent control methods such as MPC may achieve similar results, but they are not modelled in this study due to space constraints.

6. Conclusions

This paper presents a data-driven predictive control method driven by real-world data in building system operations. The main scope is improving the accuracy and stability of the coupled TSF and RL loops in HVAC predictive control scenarios. Even though the severe impacts of

Table 7
Average energy consumption and occupied PMV in 100–120 days and their comparisons with existing control.

Energy weight	PMV weight	Average energy consumption	Average energy improvement	Average occupied PMV	Occupied PMV range	Average PMV improvement
Existing control		3.139 kWh		0.887	[0.70, 1.26]	
0.3	0.7	2.519 kWh	19.8%	0.766	[0.57, 1.08]	13.6%
0.2	0.8	2.594 kWh	17.4%	0.737	[0.60, 1.06]	16.9%
0.1	0.9	2.610 kWh	16.8%	0.732	[0.57, 1.03]	17.5%

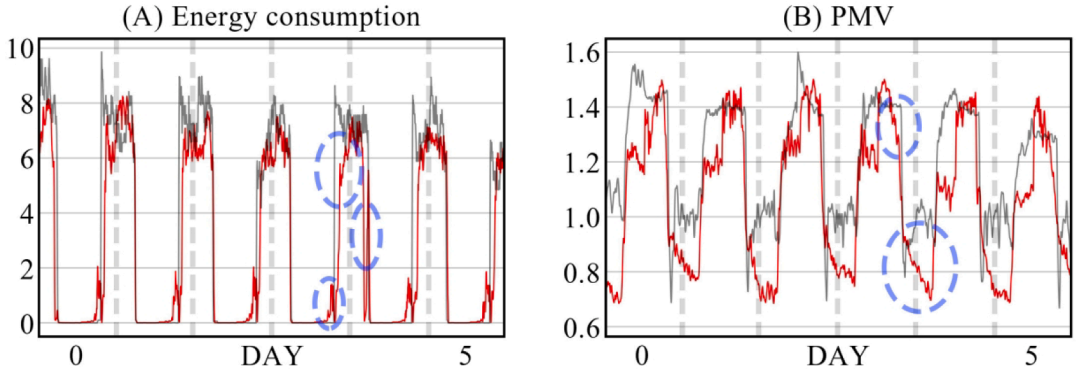


Fig. 13. Comparison of RL controller (red lines) and existing controller (grey lines) in episodes 110–115 with representative differences marked in blue circles.

recursive prediction on TSF have been well recognized, relevant studies often use independent prediction to carry out the model evaluation, which constrains the control effectiveness and rationality of RL agent implementation. This paper fills the research gap of recursive prediction, which is important for deep learning algorithms and the integration of RL agents into HVAC control. The influence of various algorithms in recursive prediction is systematically evaluated to support future studies in the related field.

Focusing on the recursive prediction of TSF models in RL-coupled loops, this paper presents 16 LSTM-based model architectures, including various combinations of CNN, bidirectional processing, and attention mechanisms. Three optimal TSF model architectures for HVAC indicators are then selected. The results show that recursive prediction significantly affects model accuracy. The degradation of accuracy varies widely for different model architectures, while the optimal model architectures can control the RMSE error within 0.1. The overall decay of indoor environmental prediction accuracy is more serious than that of energy consumption. CNN play an important role as a pre-processing layer in improving the robustness of environmental quality prediction, but its impacts on energy consumption prediction are limited. The improvement by bi-directional processing on the energy consumption prediction model is found, but it is not suitable for indoor environment prediction, especially indoor temperature. The attention mechanism plays a key role in all HVAC performance predictions, including the improvement of independent prediction accuracy and recursive prediction stability. The optimal models are CNN-LSTM-Time AM architecture for indoor environment prediction, and BiLSTM-Dim AM architecture for energy consumption prediction. The agents trained with the optimal models are then implemented for HVAC control in an office. The established RL agent can achieve energy savings through early preheating and early shutdown while achieving higher levels of thermal comfort and thermal stability.

The main limitations and future works of this paper are summarised in the following. First, deeper networks and more targeted hyperparameter optimization may further improve the performance of the model. As part of our follow-up work, more stacking layers and hyperparameter settings for the selected optimal model architecture will be explored to optimise the accuracy and stability of the TSF model. Secondly, this study used the number of Wi-Fi devices as the occupancy data. The follow-up study will further improve the model by including time-series prediction for occupancy information [58], on top of energy consumption and thermal comfort prediction. Thirdly, only indoor air temperature and relative humidity are considered variables in the PMV calculation process to access thermal comfort. The follow-up study will involve more factors such as dynamic attire and online thermal comfort surveys to develop more human-centric intelligent controls. Fourthly, while the performance of the RL agent is discussed in this study, its applications for HVAC control in a real building which contains many spaces still require further analysis. Our future work will include the deployment of the RL controller for more suitable rooms to test its applicability. Modelling and comparison of different intelligent control will be expanded as a part of future work.

CRediT authorship contribution statement

Dian Zhuang: Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **Vincent J.L. Gan:** Conceptualization, Writing – review & editing, Supervision, Project administration. **Zeynep Duygu Tekler:** Investigation, Resources, Writing – review & editing. **Adrian Chong:** Conceptualization, Resources, Writing – review & editing. **Shuai Tian:** Methodology, Software. **Xing Shi:** Supervision, Resources.

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.

Data availability

Data will be made available on request.

Acknowledgements

This research is supported by the NUS Start-up Grant (No. A-0008324-01-00). Any opinions and findings are those of the authors, and do not necessarily reflect the views of the grantor.

Appendix A. . TSF integrated RL offline training environment

Algorithm A1. RL offline training environment.

```
Inputs:
prediction model m, prediction lookback n_in
dataset D[I_A, I_P, I_C] with HVAC operation index I_A, performance index I_P, other condition index I_C
Initialize environment:
action space A_S, observation space O_S, counts = 0
for each episode do
    Reset environment:
    reset dataset D
    extract initial state S_0 from D using the index [counts + n_in −1, [I_p, I_c]]
    reset episode length L
    return S_0
        for each step do
            get current state S
            get action A from agent for current state S
            replace D in index [counts + n_in −1, I_A] with A
            predict HVAC performance P using the data in D[counts: counts + n_in,:]
            calculate reward R for performance P using reward function
            replace D in index [counts + n_in, I_P] with P
            counts +=1
            state S for the next step: D[counts + n_in −1, [I_p, I_c]]
            L -=1
            if self.length <= 0:
              done = True
            else:
              done = False
            end if
            return S, R, done
        end for
end for
```

References:

[1] IEA, 2019 Global Status Report for Buildings and Construction. 2019.
[2] Pérez-Lombard L, Ortiz J, Pout C. A review on buildings energy consumption information. Energ Buildings 2008;40(3):394–8. https://doi.org/10.1016/j.enbuild.2007.03.007.
[3] IEA, The Future of Cooling. 2018: Paris.
[4] Wang T, et al. Digital twin-enabled built environment sensing and monitoring through semantic enrichment of BIM with SensorML. Autom Constr 2022;144:104625. https://doi.org/10.1016/j.autcon.2022.104625.
[5] Brandi S, Fiorentini M, Capozzoli A. Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management. Autom Constr 2022;135:104128. https://doi.org/10.1016/j.autcon.2022.104128.
[6] Yu X, Ergan S, Dedemen G. A data-driven approach to extract operational signatures of HVAC systems and analyze impact on electricity consumption. Appl Energy 2019;253:113497. https://doi.org/10.1016/j.apenergy.2019.113497.
[7] Zou Z, Yu X, Ergan S. Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network. Build Environ 2020;168:106535. https://doi.org/10.1016/j.buildenv.2019.106535.
[8] Wang W, et al. Energy conservation through flexible HVAC management in large spaces: An IPS-based demand-driven control (IDC) system. Autom Constr 2017;83:91–107. https://doi.org/10.1016/j.autcon.2017.08.021.
[9] Afram A, Janabi-Sharifi F. Theory and applications of HVAC control systems – A review of model predictive control (MPC). Build Environ 2014;72:343–55. https://doi.org/10.1016/j.buildenv.2013.11.016.
[10] Yu L, et al. A Review of Deep Reinforcement Learning for Smart Building Energy Management. IEEE Internet Things J 2021;8(15):12046–63. https://doi.org/10.1109/JIOT.2021.3078462.
[11] Blum D, et al. Field demonstration and implementation analysis of model predictive control in an office HVAC system. Appl Energy 2022;318:119104. https://doi.org/10.1016/j.apenergy.2022.119104.
[12] Schwingshackl D, Rehrl J, Horn M. LoLiMoT based MPC for air handling units in HVAC systems. Build Environ 2016;96:250–9. https://doi.org/10.1016/j.buildenv.2015.11.011.
[13] Fu Q, et al. Applications of reinforcement learning for building energy efficiency control: A review. J Build Eng 2022;50:104165. https://doi.org/10.1016/j.jobe.2022.104165.
[14] Jang Y, Kim Y, Catalao JPS. Optimal HVAC System Operation Using Online Learning of Interconnected Neural Networks. IEEE Trans Smart Grid 2021;12(4):3030–42. https://doi.org/10.1109/TSG.2021.3051564.
[15] Kakade SM. On the sample complexity of reinforcement learning. University of London; 2003.
[16] Wang Z, Hong T. Reinforcement learning for building controls: The opportunities and challenges. Appl Energy 2020;269:115036. https://doi.org/10.1016/j.apenergy.2020.115036.
[17] Li N, Calis G, Becerik-Gerber B. Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations. Autom Constr 2012;24:89–99. https://doi.org/10.1016/j.autcon.2012.02.013.
[18] Li W, Li H, Wang S. An event-driven multi-agent based distributed optimal control strategy for HVAC systems in IoT-enabled smart buildings. Autom Constr 2021;132:103919. https://doi.org/10.1016/j.autcon.2021.103919.
[19] Alanne K, Sierla S. An overview of machine learning applications for smart buildings. Sustain Cities Soc 2022;76:103445. https://doi.org/10.1016/j.scs.2021.103445.
[20] Goyal, M., M. Pandey and R. Thakur. Exploratory Analysis of Machine Learning Techniques to predict Energy Efficiency in Buildings. 2020: IEEE.
[21] Somu N, et al. A hybrid deep transfer learning strategy for thermal comfort prediction in buildings. Build Environ 2021;204:108133. https://doi.org/10.1016/j.buildenv.2021.108133.
[22] Yang G, Yuan E, Wu W. Predicting the long-term CO2 concentration in classrooms based on the BO–EMD–LSTM model. Build Environ 2022;224:109568. https://doi.org/10.1016/j.buildenv.2022.109568.
[23] Fan C, et al. Assessment of deep recurrent neural network-based strategies for short-term building energy predictions. Appl Energy 2019;236:700–10. https://doi.org/10.1016/j.apenergy.2018.12.004.
[24] Xiao Z, et al. Impacts of data preprocessing and selection on energy consumption prediction model of HVAC systems based on deep learning. Energ Buildings 2022;258:111832. https://doi.org/10.1016/j.enbuild.2022.111832.
[25] Elmaz F, et al. CNN-LSTM architecture for predictive indoor temperature modeling. Build Environ 2021;206:108327. https://doi.org/10.1016/j.buildenv.2021.108327.
[26] Wong JKW, Li H. Construction, application and validation of selection evaluation model (SEM) for intelligent HVAC control system. Autom Constr 2010;19(2):261–9. https://doi.org/10.1016/j.autcon.2009.10.002.
[27] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput 1997;9(8):1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
[28] Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge: The MIT Press; 2016.
[29] Li Y, et al. A data-driven interval forecasting model for building energy prediction using attention-based LSTM and fuzzy information granulation. Sustain Cities Soc 2022;76:103481. https://doi.org/10.1016/j.scs.2021.103481.
[30] Fazlipour Z, Mashhour E, Joorabian M. A deep model for short-term load forecasting applying a stacked autoencoder based on LSTM supported by a multi-stage attention mechanism. Appl Energy 2022;327:120063. https://doi.org/10.1016/j.apenergy.2022.120063.
[31] Dai Y, et al. Improving the Bi-LSTM model with XGBoost and attention mechanism: A combined approach for short-term power load prediction. Appl Soft Comput 2022:109632. https://doi.org/10.1016/j.asoc.2022.109632.
[32] Chung WH, Gu YH, Yoo SJ. District heater load forecasting based on machine learning and parallel CNN-LSTM attention. Energy 2022;246:123350. https://doi.org/10.1016/j.energy.2022.123350.
[33] Zhao A, et al. Prediction of functional zones cooling load for shopping mall using dual attention based LSTM: A case study. Int J Refrig 2022. https://doi.org/10.1016/j.ijrefrig.2022.07.020.
[34] Jiang B, et al. Attention-LSTM architecture combined with Bayesian hyperparameter optimization for indoor temperature prediction. Build Environ 2022:224. https://doi.org/10.1016/j.buildenv.2022.109536.

[35] Li D, et al. Short-mid term electricity consumption prediction using non-intrusive attention-augmented deep learning model. Energy Rep 2022;8:10570–81. https://doi.org/10.1016/j.egyr.2022.08.195.
[36] Kim T, Cho S. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 2019;182:72–81. https://doi.org/10.1016/j.energy.2019.05.230.
[37] Sendra-Arranz R, Gutiérrez A. A long short-term memory artificial neural network to predict daily HVAC consumption in buildings. Energ Buildings 2020;216:109952. https://doi.org/10.1016/j.enbuild.2020.109952.
[38] Jang J, Han J, Leigh S. Prediction of heating energy consumption with operation pattern variables for non-residential buildings using LSTM networks. Energ Buildings 2022;255:111647. https://doi.org/10.1016/j.enbuild.2021.111647.
[39] Fang Z, et al. Multi-zone indoor temperature prediction with LSTM-based sequence to sequence model. Energ Buildings 2021;245:111053. https://doi.org/10.1016/j.enbuild.2021.111053.
[40] Ben Taieb S, et al. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Syst Appl 2012;39(8):7067–83. https://doi.org/10.1016/j.eswa.2012.01.039.
[41] Pinto G, Deltetto D, Capozzoli A. Data-driven district energy management with surrogate models and deep reinforcement learning. Appl Energy 2021;304:117642. https://doi.org/10.1016/j.apenergy.2021.117642.
[42] Blad C, Bøgh S, Kallesøe CS. Data-driven Offline Reinforcement Learning for HVAC-systems. Energy 2022;261:125290. https://doi.org/10.1016/j.energy.2022.125290.
[43] Wu Z, et al. Towards comfortable and cost-effective indoor temperature management in smart homes: A deep reinforcement learning method combined with future information. Energ Buildings 2022;275:112491. https://doi.org/10.1016/j.enbuild.2022.112491.
[44] Liu X, et al. A multi-step predictive deep reinforcement learning algorithm for HVAC control systems in smart buildings. Energy 2022;259:124857. https://doi.org/10.1016/j.energy.2022.124857.
[45] Fanger PO. Thermal comfort: Analysis and applications in environmental engineering. Appl Ergon 1972;3(3):181. https://doi.org/10.1016/S0003-6870(72)80074-7.
[46] Du H, et al. Evaluation of the accuracy of PMV and its several revised models using the Chinese thermal comfort Database. Energ Build 2022;271:112334. https://doi.org/10.1016/j.enbuild.2022.112334.
[47] ASHRAE (2020). ANSI/ASHRAE Standard 55, Thermal Environmental Conditions for Human Occupancy, Atlanta, GA, United States.
[48] M., S. and K.P. K., Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 1997. 45(11): p. 2673-2681. https://doi.org/10.1109/78.650093.
[49] Mnih, V., et al., Recurrent Models of Visual Attention, in Advances in Neural Information Processing Systems. 2014.
[50] Du Y, et al. Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning. Appl Energy 2021;281:116117. https://doi.org/10.1016/j.apenergy.2020.116117.
[51] Liu T, et al. A novel deep reinforcement learning based methodology for short-term HVAC system energy consumption prediction. Int J Refrig 2019;107:39–51. https://doi.org/10.1016/j.ijrefrig.2019.07.018.
[52] Haarnoja, T., et al., Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, in INTERNATIONAL CONFERENCE ON MACHINE LEARNING, VOL 80, J. Dy and A. Krause, J. Dy and A. Krause^Editors. 2018: 35th International Conference on Machine Learning (ICML).
[53] Biemann M, et al. Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control. Appl Energy 2021;298:117164. https://doi.org/10.1016/j.apenergy.2021.117164.
[54] OpenAI, Soft Actor-Critic. 2018.
[55] Laud, A. and G. DeJong, The influence of reward on the speed of reinforcement learning: An analysis of shaping, in Proceedings of the 20th International Conference on Machine Learning (ICML'03). 2003.
[56] Tekler ZD, et al. ROBOD, room-level occupancy and building operation dataset. Build Simul 2022. https://doi.org/10.1007/s12273-022-0925-9.
[57] Inc, K., Kaggle Notebooks. 2022.
[58] Tekler ZD, Chong A. Occupancy prediction using deep learning approaches across multiple space types: A minimum sensing strategy. Build Environ 2022;226:109689. https://doi.org/10.1016/j.buildenv.2022.109689.