https://dx.doi.org/10.1016/j.buildenv.2023.110600

Graph Neural Network-based spatio-temporal indoor environment prediction and optimal control for central air-conditioning systems

Jing Zhang[1], Fu Xiao[1,2*], Ao Li[1], Tianyou Ma[1], Kan Xu[1], Hanbei Zhang[1], Rui Yan[3], Xing Fang[3] Yuanyang Li[3], and Dan Wang[4]

[1] Department of Building Environment and Energy Engineering, The Hong Kong Polytechnic University, Hong Kong, China

[2]Research Institute for Smart Energy, The Hong Kong Polytechnic University, Hong Kong, China

[3]Midea Building Technologies Division, Midea Group, Guangdong 528311, China

[4] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

Abstract

Model-based optimal control has proven its effectiveness in optimizing performance of central air-conditioning systems in terms of thermal comfort and energy efficiency. It was often assumed that temperature distribution in the entire air-conditioned space is uniform and can be represented by a single or averaged measurement in optimization. However, actual distribution in the air-conditioned space is usually uneven, which can affect thermal comfort and indoor air quality. The dynamics of air conditioning systems and their interactions with the built environment exist in both time and space domains. Conventional data-driven approaches typically concentrate on the temporal correlations only among building operation data, while neglecting the spatial correlations. This study proposed a spatio-temporal data-driven methodology for optimal control of central air conditioning systems, which aims to address the challenging issue of uneven spatial distributions of environmental parameters in air-conditioned space. The methodology consists of graph-based multi-source data integration, graph neural network-based indoor environment modeling and spatio-temporal model-based online optimal control. The proposed methodology is tested on real air handling unit-variable air volume (AHU-VAV) system in a high-rise office building through a cloud-based platform. Spatio-temporal data from building automation system (BAS), internet of things (IoT) devices and building information modeling (BIM) are integrated and organized as graph structure. Graph neural network models are developed for predicting the evolution of indoor air temperature distribution under different control settings. The developed graph neural network-recurrent neural network (GNN-RNN) model architectures show enhanced accuracy than conventional deep learning models (i.e., convolutional neural network-recurrent neural network (CNN-RNN), Dense-RNN). And the cloud-based online test demonstrates that the proposed model-based control strategy improves the percentage of time achieving Grade I thermal comfort from 36.5% (i.e., existing control strategy) to 81.3%.

Keywords: machine learning, graph neural network, indoor environment, optimal control, air conditioning system

Nomenclature

Abbreviation	Full name
AGCN	Adaptive Graph Convolutional Network
AHU	Air Handling Unit
ANN	Artificial Neural Network
B	Boolean
BAS	Building Automation System
BIM	Building Information Modeling
CFD	Computational Fluid Dynamics
ChebNet	Chebyshev Neural Network
CNN	Convolutional Neural Network
CO_2	Carbon Dioxide
DC	Diffusion Convolution
FFD	Fast Fluid Dynamics
GAT	Graph Attention Network
GA	Genetic Algorithm
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
HVAC	Heating, Ventilating, and Air Conditioning
IoT	Internet of Things
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
ST	Spatial-temporal
VAV	Variable Air Volume
W	Weighted
XGBoost	eXtreme Gradient Boosting

1. Introduction

The indoor environment has a great impact on human health and productivity. Maintaining thermal comfort for people is an important factor related to the general concept of comfort experienced in human life and activities [1,2]. Central air conditioning systems are provided to maintain a comfortable indoor environment, and the development of optimal control strategies for central air conditioning systems has become an increasingly important issue as people spend more time in the built environment [3,4].

Accurate indoor temperature predicting is essential for optimal control of central air conditioning systems [5]. It can be found that the well-mixing assumption was widely employed in managing central air conditioning systems. A single location measurement [6] or the averaged value of the measurements at multiple locations [7] were widely used in formulating the optimization objective functions and constraints. However, the real distribution of indoor air is usually uneven and the simplified well-mixing indoor environment cannot accurately reflect the spatial variation of the control variables [8]. If control variables vary significantly over distances, such an assumption can be unrealistic, and the control approach based on an unrealistic assumption may have difficulty in ensuring the indoor thermal comfort throughout the occupied zone [9]. Computational fluid dynamic (CFD) modelling has been widely adopted for air conditioning system design optimization and space environment analysis, as it can predict detailed spatial distributions of temperature. Some researchers coupled CFD with energy simulation to test the indoor environment control performance considering the uneven temperature distribution in space [9,10]. Although CFD (or FFD) can provide detailed simulation of temperature distributions, the unbearable complexity and computation time make it unpractical for online applications, especially for large target zones [11]. For enhancing building control performance, rapid and accurate indoor temperature distribution predicting remains a significant challenge.

Data-driven approaches have received increasing attention in indoor environment control, due to their fast inference, high accuracy and real-time adaptability for optimal control [12]. For instance, Wei et al. [6] developed multi-layer perceptron ensemble models for predicting total energy consumption, indoor average temperature, humidity and CO_2 in multiple zones to optimize the supply air temperature and static pressure of Air Handling Unit (AHU). Li et al. [13] embedded an Elman neural network-based indoor temperature prediction model into the conventional pressure-dependent variable air volume (VAV) terminal control loop to improve the control stability. However, almost all the data used in the existing data-driven methods for optimal control of central air conditioning systems are time-series data, which seldom carried any spatial information, such as sensor locations and configuration or topology of air conditioning system in the space. As a result, the data-driven models developed can only capture temporal correlations among variables, which is also called dynamic modeling.

Abundant spatial and temporal data are readily available in existing buildings, and more are emerging with the wide adoption of internet of things (IoT) devices in buildings. Researchers have made preliminary attempts to make use of the spatial information for enhancing air conditioning system operation. Wu et al. [14] partitioned the space of a floor into 4 groups by manual checking the locations of VAV boxes in design drawings, which contributed to providing tighter thresholds for fault detection and diagnostics of AHUs and VAVs. Geng et al. [15] conducted manual checking on the spatial distributions (i.e., sensor locations) of air temperatures in the same cluster for a better understanding of indoor environment. Abdelrahman et al. [16,17] integrated building data into a graph representation using graph embedding algorithm and adopted the spatio-temporal proximity to enhance personal thermal comfort preference prediction. Appropriate integration and utilization of multi-source spatio-temporal data (which can usually be heterogeneous) has a significant impact on the modeling task [18,19]. Graph is an intuitive and informative representation of complex physical systems, which is capable of describing relations between various physical entities (i.e., spatial attributes) and sequential information (i.e., temporal attributes) as paths in one data structure [20]. Using the graph representation of multi-source data as model input, the advanced graph neural networks (GNNs) serves as an powerful and promising solution for spatio-temporal modeling of physical systems[21,22]. Sanchez-Gonzalez et al. [23] proposed a recurrent GNN-based model for simulating complex physics, where physical states of each particle were represented by nodes and dynamic interactions between particles by edges in graphs. This model was trained by eight distinct physical systems and can simulate dynamics of fluids, rigid solids, and deformable materials. Hu et al. [24] developed a spatial-temporal graph convolutional network (ST-GCN) for energy consumption prediction of multiple neighboring buildings, where buildings and their dynamic solar impacts were represented by nodes and edges, respectively. The results show that the ST-GCN model outperforms eXtreme gradient boosting (XGBoost), multilayer perceptron (MLP) and gated recurrent unit (GRU) models, which achieves an average absolute percentage error of around 5%.

This study makes new contributions in the following two aspects. Firstly, we have developed a spatio-temporal data-driven modeling methodology for predicting the spatial distribution of key environmental parameters, such as temperature and CO_2 concentration. Secondly, we propose an optimal control of central air conditioning systems based on the developed model for indoor environment control, such as local thermal comfort and indoor air quality, which is not possible if using conventional well-mixing models or the temporal data-driven models. To achieve this, the proposed methodology utilized multi-source data from the building, including building information system (BIM), building automation system (BAS), and internet of things (IoT) sensors. The heterogeneous multi-source data were integrated into a graph structure that corresponds to the configuration of target system. Graph neural network-based spatio-temporal models were developed to predict indoor air environment and enabled online optimal control of air conditioning system. This kind of local environment control is specifically important for large spaces like airport terminals, theatres and open office floors. To evaluate the effectiveness of the proposed methods, a cloud-based online case test was conducted on an AHU-VAV system in a high-rise office building.

The remaining part of this paper is organized as follows. Section 2 describes the methodology developed for spatio-temporal model-based air conditioning system online control strategy. Section 3 presents the case study of a typical AHU-VAV system serving private offices and open area of a high-rise building in Foshan, Guangdong, China. Section 4 discusses the model performance, test results and limitations of the study. Section 5 concludes the paper.

2. Methodology

2.1 Overview of the research methodology

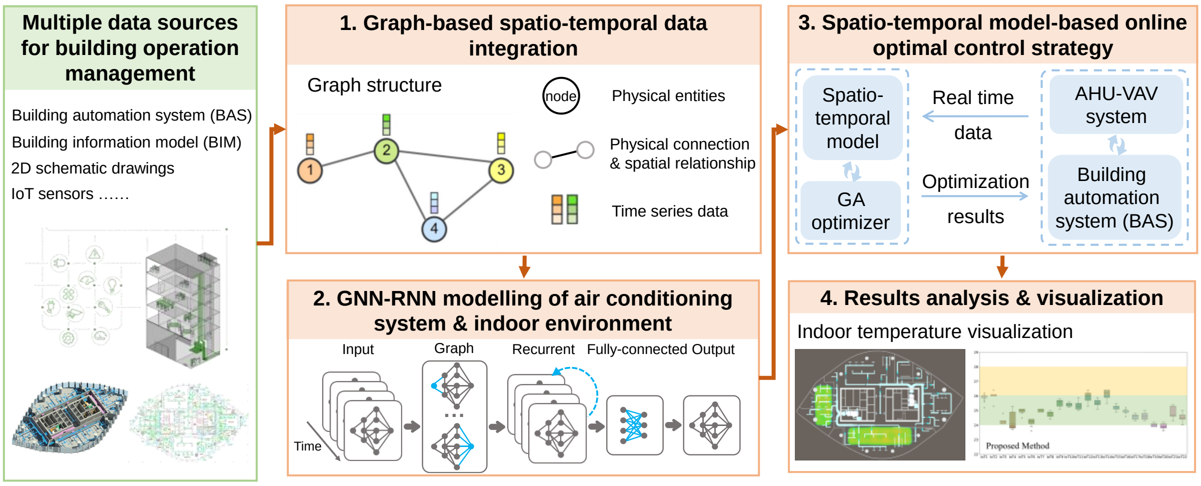

Multiple data sources for building operation management
Building automation system (BAS)
Building information model (BIM)
2D schematic drawings
IoT sensors ……
1. Graph-based spatio-temporal data integration
Graph structure
node
Physical entities
Physical connection & spatial relationship
Time series data
1
2
3
4
2. GNN-RNN modelling of air conditioning system & indoor environment
Input
Graph
Recurrent
Fully-connected
Output
Time
3. Spatio-temporal model-based online optimal control strategy
Spatio-temporal model
GA optimizer
Real time data
Optimization results
AHU-VAV system
Building automation system (BAS)
4. Results analysis & visualization
Indoor temperature visualization
Proposed Method

Figure 1. The block diagram of research methodology

The research methodology mainly consists of four steps as shown in **Figure 1**. The first step is graph-based spatio-temporal data integration, which aims to integrate the spatial and temporal data of a central air conditioning system using graph-based data representation. The second step is graph neural network-based spatio-temporal modeling of indoor environment, using graph neural network-based model. The graphs constructed in Step 1 will be used as the model input to capture the spatial and temporal relationships and predict the near-future indoor environment (i.e., the indoor temperature of air-conditioned space at multiple locations in the case study). The third step is to develop online optimal control strategy based on the model trained in Step 2 for balancing uneven indoor air distribution and enhancing indoor thermal comfort. The fourth step visualizes the indoor environment and analyzes the results.

2.2 Graph-based spatio-temporal data integration

2.2.1 Spatial and temporal data available in buildings

Today's buildings are information-intensive, storing a huge amount of spatial and temporal data. Heterogenous data stored in different databases are continuously generated at different stages of the building lifecycle and need to be extracted and integrated.

1) Spatial data. 2D schematic or layout drawings and 3D building information models (BIMs) contain the static and spatial data at the design and construction stage. 2D schematic drawings provide the connection (e.g., duct system) between different physical entities (e.g., air handling unit, variable air volume box, etc.), while 2D layout drawings provide the specific locations of them. 3D BIMs represent the geometry and spatial relationships of the building, making it easy to identify the location of entities and measure their distances [25].

2) Temporal data. Building automation systems, also known as building management system, contain the temporal data at the building operation stage. Building operational data in BAS are typically multivariate time series data, including energy consumption data, operating variables (e.g., real-time indoor temperature), environmental parameters (e.g., outdoor air temperature), and miscellaneous [26]. With the radical evolution of internet of things (IoT) networks, more environmental data from IoT sensors [27] and occupant feedback [17] are also available for air conditioning operation management.

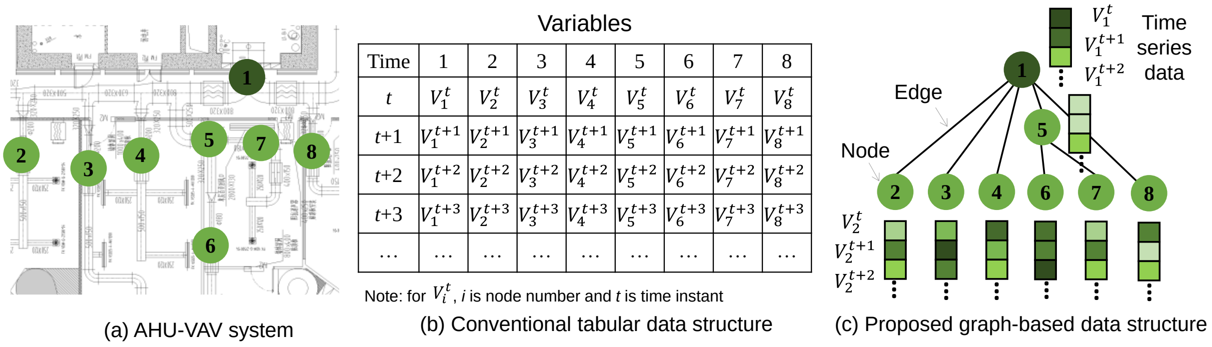
2.2.2 Graph-based data structure

Graph is recognized as one of the most generic, natural, informative and interpretable formats for data representation [28]. A graph $\boldsymbol{G}$ can be represented as Eq.(1).

$$\boldsymbol{G} = \left(\boldsymbol{V}, \boldsymbol{E}\right) \tag{1}$$

where $\boldsymbol{V}$ is the set of nodes and $\boldsymbol{E}$ is the set of edges. Here, $v_i \in \boldsymbol{V}$ denote a node and $e_{ij} = \left(v_i, v_j\right) \in \boldsymbol{E}$ denote an edge which connect node v_i and node v_j.

The graph structure is usually described by the adjacency matrix and the node attributes are described by the feature matrix. For an undirected and connected graph with N nodes, the adjacency matrix $\boldsymbol{A} = \left[a_{ij}\right] \in \mathbb{R}^{N \times N}$ can be a matrix of Boolean with $a_{ij} = 1$ if $e_{ij} \in \boldsymbol{E}$ (i.e., there is an edge from v_i to v_j) and $a_{ij} = 0$ if $e_{ij} \notin \boldsymbol{E}$ (i.e., there is no edge), or a weighted matrix with $a_{ij} \in \mathbb{R}$ (which is the weight of the edge e_{ij}) and $a_{ij} = 0$ if $e_{ij} \notin \boldsymbol{E}$. For a graph with M-dimensional nodes with time sequence T, the feature matrix $\boldsymbol{X} = \left[x_{i,k}^t\right] \in \mathbb{R}^{N \times M \times T}$ stores the node attributes.



Variables

Time	1	2	3	4	5	6	7	8
t	V_1^t	V_2^t	V_3^t	V_4^t	V_5^t	V_6^t	V_7^t	V_8^t
t+1	V_1^{t+1}	V_2^{t+1}	V_3^{t+1}	V_4^{t+1}	V_5^{t+1}	V_6^{t+1}	V_7^{t+1}	V_8^{t+1}
t+2	V_1^{t+2}	V_2^{t+2}	V_3^{t+2}	V_4^{t+2}	V_5^{t+2}	V_6^{t+2}	V_7^{t+2}	V_8^{t+2}
t+3	V_1^{t+3}	V_2^{t+3}	V_3^{t+3}	V_4^{t+3}	V_5^{t+3}	V_6^{t+3}	V_7^{t+3}	V_8^{t+3}
…	…	…	…	…	…	…	…	…

Note: for V_i^t, *i* is node number and *t* is time instant

(a) AHU-VAV system

(b) Conventional tabular data structure

(c) Proposed graph-based data structure

Figure 2. Comparison of conventional tabular structure and graph-based data structure proposed for spatio-temporal data

Figure 2 shows a comparison between a conventional tabular structure for temporal data only and a graph-based data structure for spatio-temporal data. Typically, building operational data (e.g., time, energy consumption data, operating variables, environmental parameters, etc.)

are stored in a two-dimensional tabular form, where each column represents a variable and each row stores the measurements or signals at the same time step [29], as is shown in **Figure 2** (b). However, the spatial information, like the locations of air conditioning devices and sensors as well as the connection of air conditioning devices, cannot be considered in conventional tabular data structure. On the other hand, graphs are more informative and suitable for spatio-temporal data, which consists of nodes representing physical entities (like VAV boxes and sensors, etc) and edges representing the relationship of the entities, as shown in **Figure 2** (c).

2.2.3 Graph representation of central air conditioning system

This paper proposes a generic graph representation for central air conditioning system, which integrates the related spatial (e.g., system configuration, sensor location, etc.) and temporal (e.g., system operational data, etc.) information. The nodes represent different physical entities, and the graph structure (i.e., connection between nodes) is designed based on physical connection and spatial relationship. The spatial and temporal information can be embedded in the graph structure, adjacency matrix, and feature matrix. Detailed introduction of the proposed generic graph representation is provided in this section, including the nodes, edges, adjacency matrix, and feature matrix. And a graph representation example of AHU-VAV system is presented.

1) Nodes

- **Physical entity node.** The nodes can represent the physical entities of target system, e.g., air handling unit, variable air volume box, indoor air. These nodes can be homogeneous or heterogeneous.
- **Global node (Optional).** Global node (also known as master node) is a special node which is connected to all other nodes in the graph. Global node can contain the global-level context of the target system which does not correspond to any specific physical

entity (e.g., outdoor environmental parameters). Existing researches have reported that well-designed global node can enhance the performance of graph neural network, e.g., document understanding [30] and recommendation system [31].

2) Edges

- **Physical connection.** The graph structure corresponds to physical configuration of the target system. That is, the nodes which are connected in the air ductwork (e.g., AHU and VAV boxes), are set to connect in the graph representation.
- **Spatial relationship.** The node connections can also be determined based on spatial relationships (even if there are no physical connection between them). For example, proximate IoT sensors and VAV boxes, proximate IoT sensors in space, are considered connected (i.e., there is an edge between these nodes). The distance threshold can be determined by the size of air-conditioned space and domain expertise.

3) Adjacency matrix

Adjacency matrix is the most commonly-used way for representing a graph. The elements of the matrix indicate whether pairs of nodes are adjacent or not in the graph (i.e., Boolean matrix) or the weights of edges (i.e., weighted matrix). For an undirected and connected graph with N nodes, the Boolean matrix $\boldsymbol{A}_{\text{Boolean}} = \left[a_{ij,\text{Boolean}} \right] \in \mathbb{R}^{N \times N}$ can be described as Eq.(2).

$$a_{ij,\text{Boolean}} = \begin{cases} 1 & e_{ij} \in \boldsymbol{E} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The similarities between the node vectors (i.e., node attributes) can be used as the quantitative measure of the correlation between nodes [32]. In this study, the absolute value of Pearson Correlation Coefficient (PCC) [33] is chosen as the similarity measure. For an undirected and connected graph with N nodes, the weighted matrix $\boldsymbol{A}_{\text{Weighted}} = \left[a_{ij,\text{Weighted}} \right] \in \mathbb{R}^{N \times N}$ can be described as Eq.(3).

$$a_{ij,\text{Weighted}} = \begin{cases} \left|\text{PCC}(v_i, v_j)\right| & e_{ij} \in \boldsymbol{E} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

4) Feature matrix

For node $v_i \in \boldsymbol{V}$ which has a M-dimensional vector of attribute values with time sequence T, the feature matrix can be described as Eq.(4)

$$\boldsymbol{X} = \left[x_{i,k}^{t} \right] \in \mathbb{R}^{N \times M \times T} \qquad 1 \le i \le N, 1 \le k \le M, t - T \le t \le t - 1 \tag{4}$$

where $x_{i,k}^{t}$ represent the attribute k of node i at timestep t.

For the air central conditioning system, the attributes of different nodes may be different, which brings challenges to capturing the spatio-temporal correlations among heterogeneous attributes. Several methods have been developed to handle heterogeneous graph input, such as zero-padding and context-aware heterogeneous graph attention block [34]. In this study, the zero-padding strategy is used to handle the different node attributes.

5) An example of an AHU-VAV system

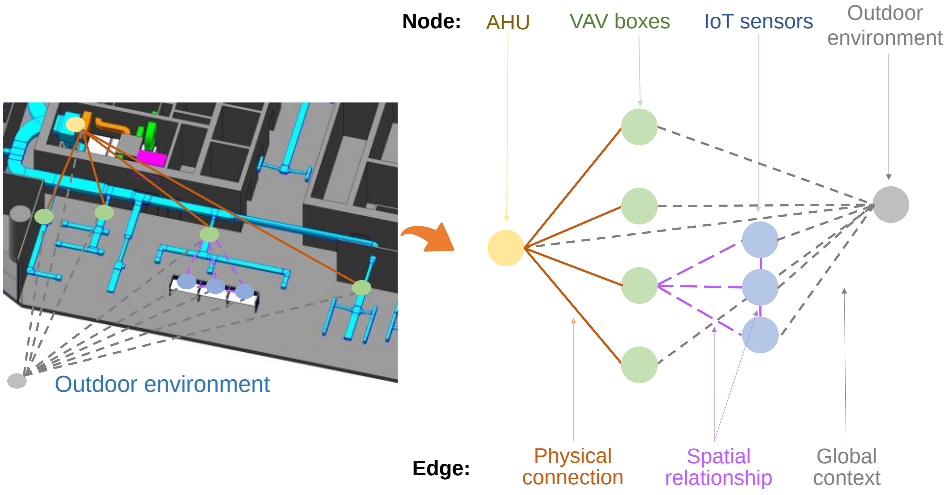


Figure 3. Diagram of graph representation of a simple AHU-VAV system

Figure 3 provides a graph representation example of a simple AHU-VAV system. The nodes represent physical entities of the system, and the node connections (graph structure) are determined based on physical connections and spatial interactions. The temporal information (i.e., system operation data) is encoded in the feature matrix and the spatial information (i.e., the edges) can be encoded in the adjacency matrix. In this way, the spatio-temporal data can be integrated into a graph representation.

2.3 Spatio-temporal modeling of air conditioning system and indoor environment

A stacked GNN-RNN model architecture is adopted for spatio-temporal modeling of air conditioning system and indoor environment. The graph generated in Section 2.1 will be used as the model input. The graph layer and recurrent layer are capable of capturing the spatial dependency and temporal dependency among input features, respectively. The model developed can predict the entire indoor environment and will be used for the optimal control of air conditioning system.

2.3.1 Graph neural network (GNN)

The earliest motivation for GNNs can root in the 1990's owing to graph analysis and graph representation learning [35,36], while the recent re-advancement of GNNs is mainly attributed to the success of deep neural networks, particularly convolutional neural networks (CNN) [37]. CNNs are able to extract multi-scale localized spatial characteristics and integrate them to construct highly expressive representations, but can only work on structure data, such as text (1D sequences) and 2D grid images. Therefore, it is difficult to generalize CNNs to graphs of non-structural data.

A GNN [38] can be regarded as an extension of a CNN architecture, and provides an effective method to capture spatial dependencies among graph structured inputs. The adjacency matrix representing the graph structure can assists the message propagation within neural layers in the GNNs. Based on the connection information provided by the adjacency matrix, GNNs can efficiently capture graphs' structural dependencies via message propagation between the nodes by aggregating information only from connected nodes [39]. Depending on the information aggregation method [22], GNNs can be further classified into several types, e.g., chebyshev neural network (ChebNet) [40], graph convolution networks (GCNs) [41], adaptive graph convolution network (AGCN) [42], graph attention networks (GATs) [43], etc. Among them, GCNs play a crucial role in capturing structural dependencies, while networks in other categories partially rely on GCNs in building blocks [26]. GCNs accomplish convolutional feature extraction through neighborhood aggregation [45]. For instance, spectral-based graph convolutional layer follows the propagation rule

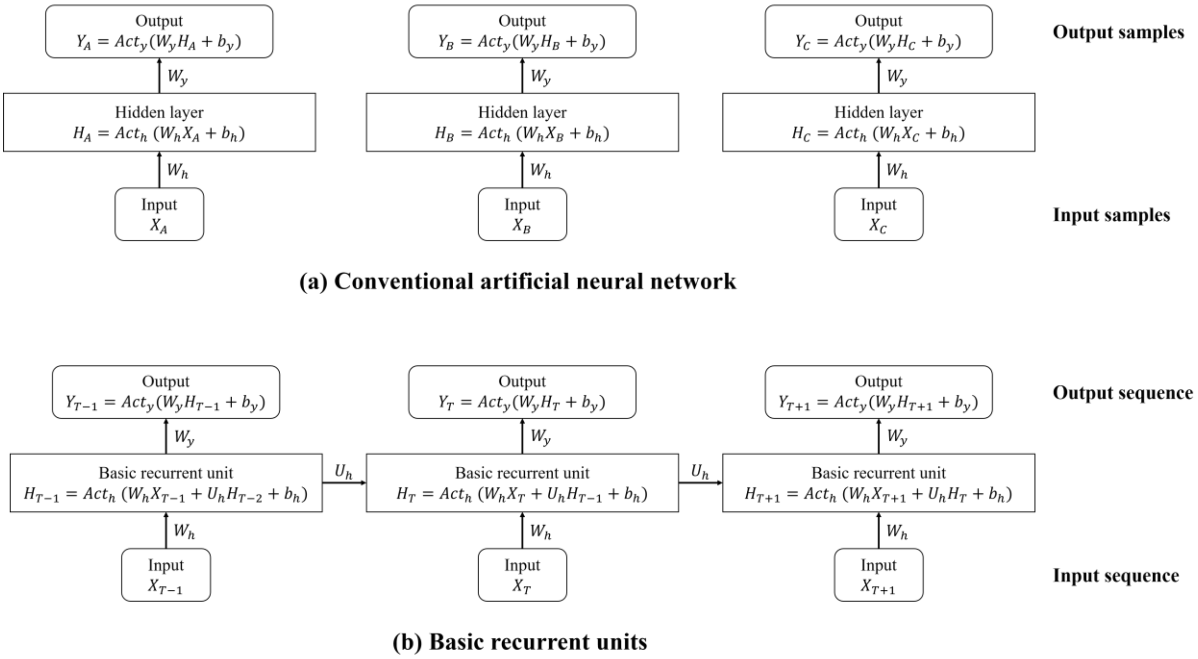
$$\boldsymbol{H}' = \sigma(\boldsymbol{D}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{W}^{(l)}) \tag{5}$$

where $\boldsymbol{H}$ and $\boldsymbol{H'}$ denote the input and output of the graph convolutional layer, respectively. $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$ is the adjacency matrix of the graph $\boldsymbol{G}$ with added self-connections $\boldsymbol{I}_N$ (i.e., the identity matrix). $\widetilde{\boldsymbol{D}}$ is the degree matrix of $\widetilde{\boldsymbol{A}}$, $\widetilde{D_{\iota\iota}} = \sum_j \widetilde{\boldsymbol{A}}_{ij}$. $\boldsymbol{W}^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function.

This propagation rule can be motivated via a first-order approximation of localized spectral filters on graphs [46]. The graph attention networks (GATs) are created by incorporating the attention mechanism into the propagation stage [47]. The attention mechanism enables the neural networks to pay attention how different inputs influence outputs at each step of inference [53,54]. In general, the most widely-used types of GNNs nowadays are GCNs and GATs.

2.3.2 Recurrent neural network (RNN)

Recurrent neural network (RNN) has been an important focus of research during the 1990's, which is developed for analyzing time-series data [44]. **Figure 4** shows the difference between conventional artificial neural network (ANN) and RNN. **Figure 4(a)** presents an example of a conventional ANN with one hidden layer. The W_1, W_2, b_1, and b_2 are weights and biases. $Act_1(z)$ and $Act_2(z)$ represent the activation functions. The processes of conventional ANN generating each output sample using the corresponding input sample are the same, and do not affect each other. On the other hand, RNN deals with input sequence/time-series data by individual vectors at each step and preserves the information it has captured at previous time steps in a hidden state. **Figure 4(b)** presents an example of RNN using the basic recurrent unit. The W_h, W_y, and U_h are matrices containing input weights, output weights and recurrent weights, respectively. $Act_h(z)$ and $Act_y(z)$ are activation functions, and b_h and b_y are biases. A hidden state vector H_t, which has the same length as the input variables, is defined to preserve the information which has been observed. Its values are set to zero at the initial time step. The hidden state vector at time step $T-1$ (H_{T-1}) will be used together with the input data at time step T (X_T) to calculate the hidden state at time step T (H_T). This allows RNNs to learn temporal relationships or dynamics from a sequence. However, the capability of RNNs in capturing long-term temporal dependencies can be limited due to vanishing or exploding gradient problem, i.e., the model will become untrainable within increase in recurrent operations [45]. To overcome this problem, the two special types of RNN called long short-term memory (LSTM) [45] and gated recurrent units (GRU) [46] were created by introducing the concept of gates to control the flow of information and learn the important information in the memory unit, which can pass the information in long sequences. LSTM enables the reinjection of past information later by calculating what percentage will the past information be allowed to affect the present information, thus contributing to deal with the problem of vanishing or exploding gradients. On the other hand, GRU is a simplified and efficient alternative to the LSTM, which uses only an update gate and a reset gate. In this study, the LSTM is used as the RNN layer.


Output
$Y_A = Act_y(W_y H_A + b_y)$
W_y
Hidden layer
$H_A = Act_h\,(W_h X_A + b_h)$
W_h
Input
X_A
Output
$Y_B = Act_y(W_y H_B + b_y)$
W_y
Hidden layer
$H_B = Act_h\,(W_h X_B + b_h)$
W_h
Input
X_B
Output
$Y_C = Act_y(W_y H_C + b_y)$
W_y
Hidden layer
$H_C = Act_h\,(W_h X_C + b_h)$
W_h
Input
X_C
Output samples
Input samples
Output
$Y_{T-1} = Act_y(W_y H_{T-1} + b_y)$
W_y
Basic recurrent unit
$H_{T-1} = Act_h\,(W_h X_{T-1} + U_h H_{T-2} + b_h)$
W_h
Input
X_{T-1}
U_h
Output
$Y_T = Act_y(W_y H_T + b_y)$
W_y
Basic recurrent unit
$H_T = Act_h\,(W_h X_T + U_h H_{T-1} + b_h)$
W_h
Input
X_T
U_h
Output
$Y_{T+1} = Act_y(W_y H_{T+1} + b_y)$
W_y
Basic recurrent unit
$H_{T+1} = Act_h\,(W_h X_{T+1} + U_h H_T + b_h)$
W_h
Input
X_{T+1}
Output sequence
Input sequence

(a) Conventional artificial neural network

(b) Basic recurrent units

Figure 4. Schemata of (a) conventional ANN, (b) RNN

2.3.3 GNN-RNN model for spatio-temporal modeling

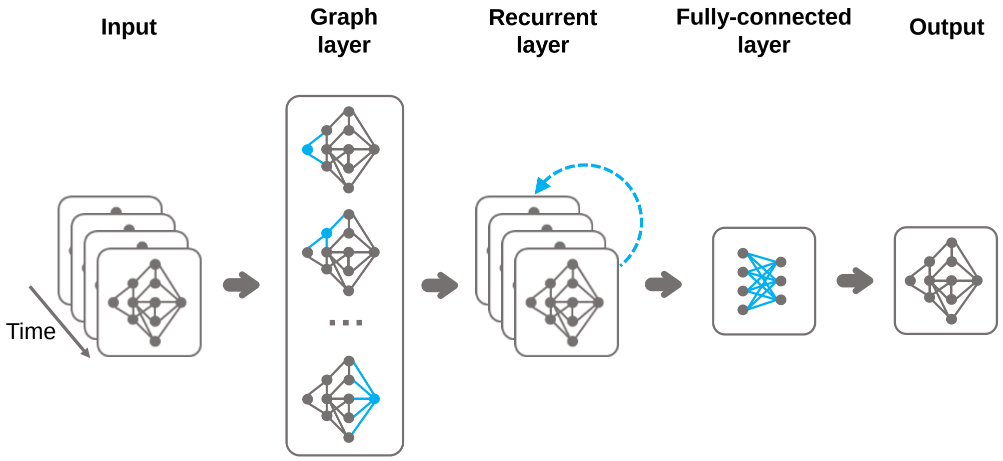


Figure 5. Spatio-temporal model for AC system and indoor environment

As show in **Figure 5**, a stacked GNN-RNN architecture is adopted for the spatio-temporal modeling of central air conditioning system which predicts the indoor environment parameters at multiple locations/regions. The developed GNN-RNN model consists of a graph layer, a recurrent layer, and a fully connected layer. The graph layer is responsible for capturing the spatial relationship within the air conditioning system and the air-conditioned space, while the recurrent layer is responsible for handling the temporal relationship.

A set of graphs with adjacency matrix $\boldsymbol{A}=\left[a_{ij}\right]\in\mathbb{R}^{N\times N}$ and feature matrix $\boldsymbol{X}=\left[x_{i,k}^{t}\right]\in\mathbb{R}^{N\times M\times T}$ constructed in **Section 2.2** will be used as the input of the GNN-RNN model. The output of the model is the one-step ahead indoor environment prediction $\hat{Y}$, which can be described as Eq.(6) and Eq.(7).

$$\hat{Y}=[\hat{y}_{i,k}] \qquad 1\le i\le N, 1\le k\le M \tag{6}$$

$$\hat{Y}=f\left(\boldsymbol{X}'\right) \tag{7}$$

where $\hat{y}_{i,k}$ represent the predicted attribute *k* of node *i* at the next timestep.

Root mean square error (RMSE) is adopted to assess the model prediction accuracy and can be described as Eq.(8)

$$RMSE=\sqrt{\sum_{i=1}^{n}\frac{(y_{i,k}-\hat{y}_{i,k})^{2}}{n}} \tag{8}$$

where $y_{i,k}$ represent the actual value of attribute *k* of node *i* at the next timestep.

2.4 Model-based online optimal control

Based on the indoor environment dynamic model (developed in Section 2.2) and genetic algorithm (GA) optimization, a model-based online optimal control strategy is developed for enhancing the control performance of central air conditioning system. **Figure 6** shows the diagram of proposed model-based online optimal control strategy.

The optimizer fetches real-time system operation data from the BAS. The GNN-RNN model simulates the indoor temperature distribution evolution within a prediction horizon Δt_{pred} , which is used by the cost estimator to compute the 'overall cost'. Through many generations of computation, the GA optimizer finds the optimal control actions (i.e., indoor temperature setpoint of each VAV box in the case study) which minimizes the overall cost over the entire prediction period.

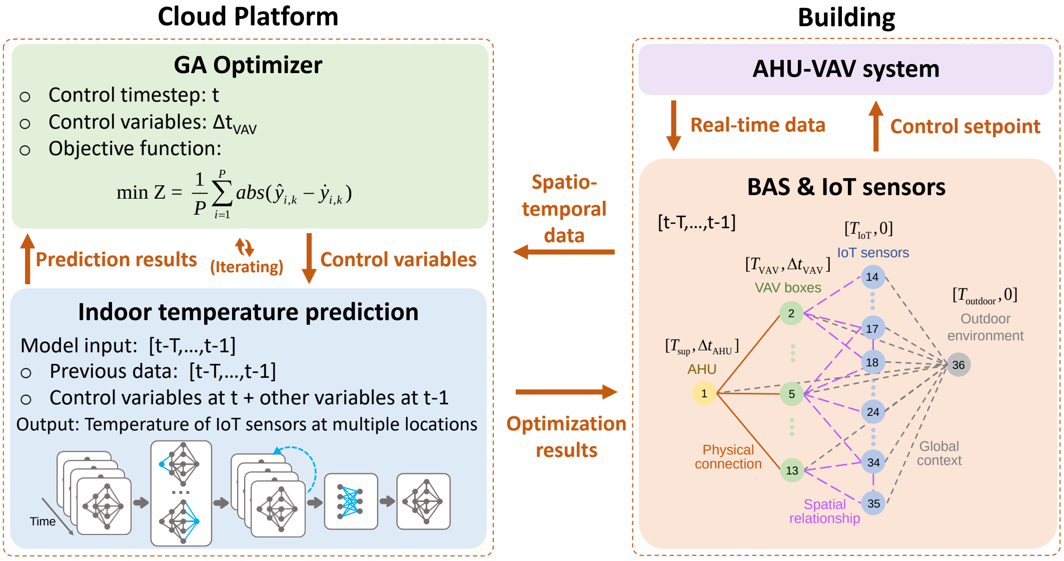


Figure 6. Proposed model-based optimal control strategy

The requirements for the indoor environment can be indicated by user and described as Eq.(9).

$$\dot{Y} = [\dot{y}_{i,k}] \qquad 1 \le i \le P, 1 \le k \le M \tag{9}$$

The objective function, accounting for thermal comfort, is constructed as Eq.(10).

$$\min \mathrm{Z} = \frac{1}{P}\sum_{i=1}^{P} \left| \hat{y}_{i,k} - \dot{y}_{i,k} \right| \tag{10}$$

where $\hat{y}_{i,k}$ and $\dot{y}_{i,k}$ are the prediction results and the requirements for indoor environment (i.e., occupied zone temperature measured by IoT sensors in the case study), respectively. *P* represents the number of requirements and can be decided by users.

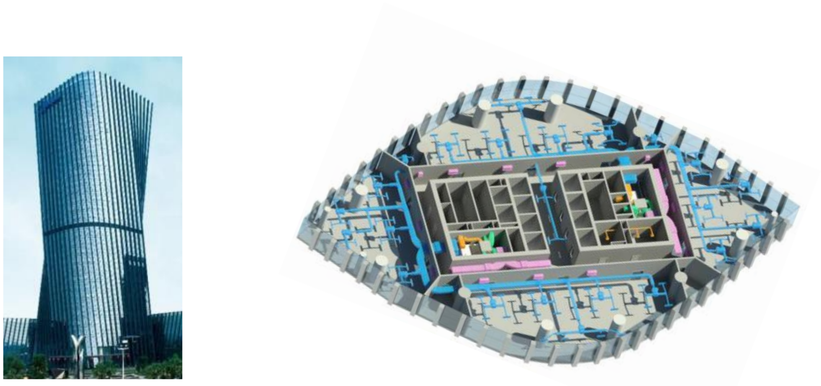
In the case study, the genetic algorithm (GA) optimizer is chosen to optimize the control variables (i.e., temperature difference Δt_{VAV} between the return air temperature and the indoor air temperature setpoint of all VAV boxes). According to *Design code for HVAC of civil buildings* GB50736-2012 [47], the indoor air temperature thermal comfort Grade I is from 24 to 26°C and Grade II is from 26 to 28°C. The target temperature range of the air-conditioned space (i.e., the occupied zone temperature) is from 24-26°C. As suggested by the building owner, the requirements for indoor air temperature in the objective function were set at 25°C (i.e., $\dot{y}_{i,k} = 25$).

3. Case study

A case study is conducted on a real AHU-VAV system in an office building to test the proposed modeling method and online optimal control strategy.

3.1 Introduction of target air-conditioning system and its existing control strategy

The proposed strategy is tested on a typical floor in a high-rise office building (as shown in **Figure 7(a)**) in Foshan, Guangdong, China. The office building has 31 floors and a height of 128m. The typical floor consists of 260 m^2 private offices and 605 m^2 open-plan area. The building information model (BIM) of the typical floor is shown in **Figure 7(b)**. Two identical air-handling units (AHUs) are installed on each typical floor to serve the symmetric half spaces, separately.



(a) The target building (b) Building information model (BIM) of the typical floor

Figure 7. Target building and its typical floor

Figure 8 shows the target AHU-VAV system and its air-conditioned space. The AHU is connected with 12 VAV boxes (pressure-independent type) and serves half of the air-conditioned space of the typical floor. The area highlighted in green is the private office area and the area highlighted in blue is the open-plan area (i.e., open office space with cubicles).

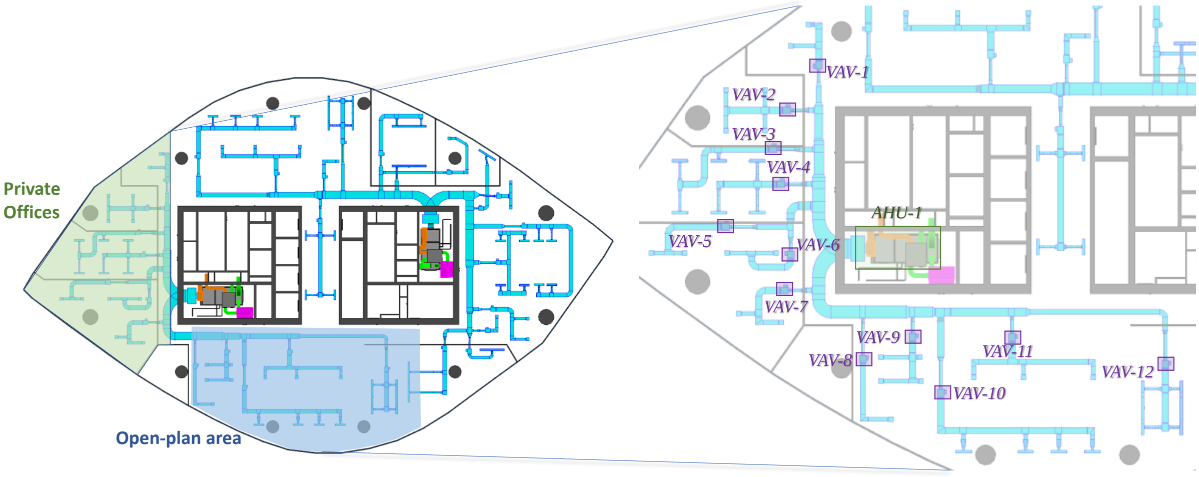


Figure 8. Target AHU-VAV system and its air-conditioned space

The target AHU-VAV system serves both private office area and open-plan area with a mixed and complex occupancy pattern. The private offices are usually used as personal workspaces, although small group meetings may also be conducted in them. The desks in the open area were always fully occupied. From our observation on the operational data in a long period, the thermal environment of the private office area is more sensitive to the number of people while that in the open area is more sensitive to the outdoor weather condition.

In the existing control strategy, the return air temperatures at the VAV boxes were measured and compared with the setpoints to control the openings of VAV boxes. The air temperature setpoints used to control the VAV boxes were determined based on previous experiences, and usually kept unchanged unless office workers complained about the thermal sensation. The temperature setpoints of different VAV boxes were different, mainly varied from 23°C to 26°C. Occasionally, the setpoints were as low as 18°C. A very low setpoint was usually because the operator received complaints about the environment being too hot.

3.2 Description of data for spatio-temporal model development

22 smart IoT temperature sensors were installed in the occupied zone (i.e., on the office desks) served by the target AHU-VAV system. The sensor parameters are shown in **Table 1**.

Table 1 Parameters of IoT temperature sensor

Item	Parameters
Dimensions	56.2×56.2×12.8mm
Weight	32g
Wireless Connection	Bluetooth 5.0
Temperature measurement range	0°C ~60°C
Temperature display resolution	0.1°C
Temperature measurement accuracy	0.3°C

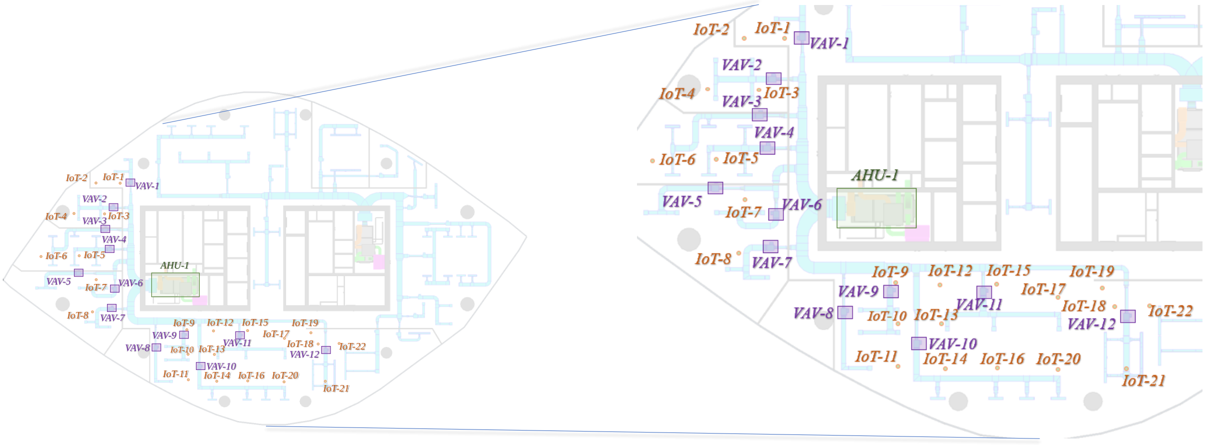


Figure 9. Schematic of the AHU-VAV system and sensor locations

There are 2 IoT sensors in each private office (a total of 8) and 14 IoT sensors in the open area. The locations of the 12 VAV boxes and 22 IoT sensors are illustrated in **Figure 9**. The IoT sensors were installed in the occupied zone at a height of 0.8m and the VAV boxes were placed above the ceiling at a height of 2.4m.

The historical data (during working hours) from June 12 to September 8, 2022 are collected from BAS and IoT sensors for development of indoor environment prediction model. The raw data is sampled at 1 minute interval. The 10 minutes average of the data is chosen to represent each timestep as the indoor environment does not change that quickly. The whole dataset (containing 4,826 observations) is transformed into subsequences via a sliding-window manner. Data preprocessing is performed to enhance data quality. The missing values are filled in using the moving average method, while the outliers are identified with domain expertise. Min-max normalization is adopted to transform the data into a suitable scale for further analysis. The historical data are randomly divided into training dataset and testing dataset. The training dataset and testing dataset account for 80% and 20% of the total data, respectively. **Table 2** provides a statistical summary of the historical data.

Table 2. Description of the historical data

Device	Quantity	Measurements	Max.	Min.	Avg.
AHU	1	Supply air temperature T_{sup} (°C)	19.6	12.9	16.3
		Return air temperature T_{rtn} (°C)	30.6	23.6	25.8
VAV box (at 2.4m)	12	Indoor air temperature setpoint $T_{\mathrm{VAV,sp}}$ (°C)	29.1	18.0	24.2
		Return air temperature T_{VAV} (°C)	33.9	22.0	26.1
IoT sensor (at 0.8m)	22	Indoor temperature T_{IoT} (°C)	33.4	20.8	25.9
N/A	1	Outdoor air temperature T_{outdoor} (°C)	38.4	24.7	31.7

3.3 Description of the cloud-based online test platform

A cloud-based online test is carried out on the proposed model-based optimal control strategy, as shown in **Figure 10**. A cloud platform in Hong Kong is used to collect and store the spatio-temporal data from the AHU-VAV system in Foshan, Guangdong Province, and optimize the control setpoints of VAV boxes. The optimization results are then sent to the building automation system (BAS), which in turn sends them to the VAV boxes. The spatial data are extracted manually through building information model (BIM) by domain knowledge and stored in a MySQL spatial database, which is a relational database that stores the adjacency matrix representing the data structure. The temporal data, including the AHU-VAV system operational data from building automation system (BAS) and the indoor temperature measurements from internet of things (IoT) sensors, are collected and stored in an influxdb temporal database, which is efficient at querying time-series data and require less storage space.

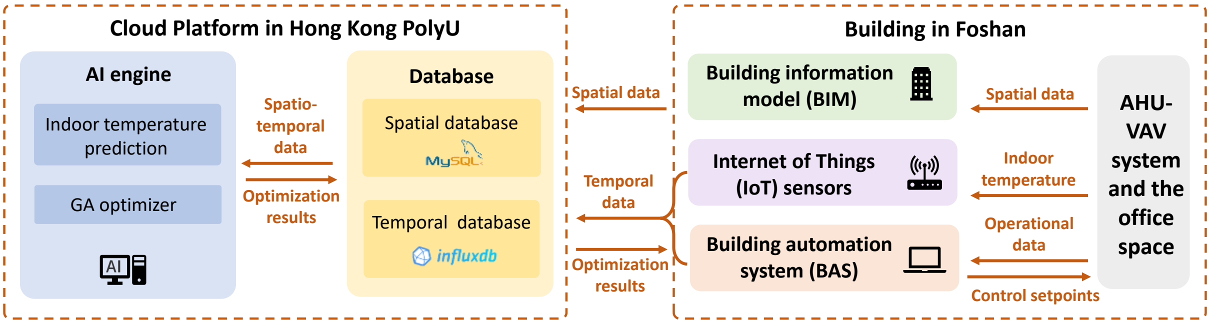


Figure 10. Cloud-based online test

The spatio-temporal modelling of the target AHU-VAV system aims to predict occupied zone temperature for the next timestep based on the operational data of AHU, VAV boxes and outdoor environment of previous 3 timestep (i.e., T=3).

As shown in **Figure 11**, each graph contains 36 nodes (i.e., N=36) including 1 AHU node, 12 VAV box nodes, 22 IoT sensor nodes, and 1 outdoor environment node with 2-dimensional vector of attribute values (i.e., M=2).

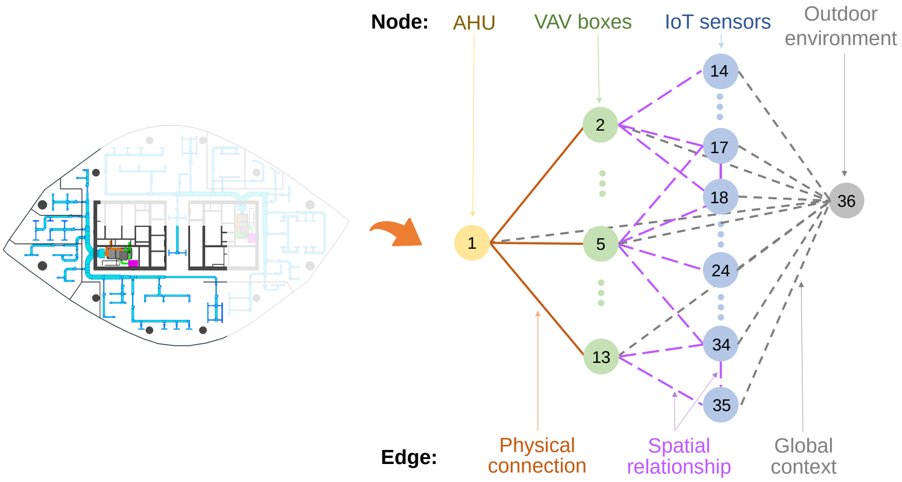


Figure 11 Graph representation in the case study

As shown in **Figure 12**, for the small private offices, the IoT sensors in each room are connected to the VAV box in the corresponding room, and they are also interconnected. For the large open area, the IoT sensors are connected to the VAV boxes within a threshold distance (i.e. 4m in this study) and interconnected within a threshold distance (i.e. 2m in this study).

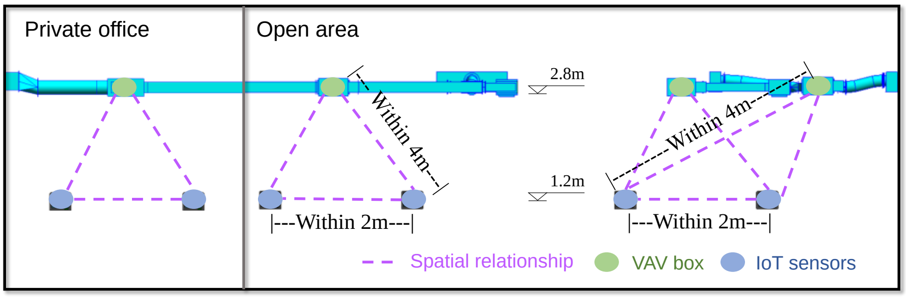


Figure 12 Graph structure in private office and open area

The feature matrix is constructed using Eq.(11).

$$\boldsymbol{X}' = \begin{cases} [x^t_{i,T_{\text{sup}}}, x^t_{i,\Delta t_{\text{AHU}}}] & i \text{ is AHU} \\ [x^t_{i,T_{\text{VAV}}}, x^t_{i,\Delta t_{\text{VAV}}}] & i \text{ is VAV box} \\ [x^t_{i,T_{\text{IoT}}}, 0] & i \text{ is IoT sensor} \\ [x^t_{i,T_{\text{outdoor}}}, 0] & i \text{ is outdoor environment} \end{cases} \tag{11}$$

where $\Delta t_{\text{AHU}} = T_{\text{rtn}} - T_{\text{sup}}$ represents the temperature difference between supply air temperature and return air temperature of AHU, and $\Delta t_{\text{VAV}} = T_{\text{VAV}} - T_{\text{VAV,sp}}$ represents the temperature difference between indoor air temperature setpoint and return air temperature of VAV box.

For the proposed GNN-RNN neural architecture, three widely-used graph layers are tested, including graph convolutional layer (i.e., GCN-RNN) [41], diffusion convolution layer (i.e., DC-RNN) [48], and graph attention layer (i.e., GAT-RNN) [43]. For comparison, two conventional deep learning model architectures, i.e. CNN-RNN and Dense-RNN, are tested, which replace the graph layer with convolutional layer and fully-connected layer.

In summary, five model architectures (i.e., CNN-RNN, Dense-RNN, GCN-RNN, DC-RNN, GAT-RNN) with two adjacency matrices (i.e., Boolean matrix and weighted matrix) are tested and compared in this study. For graph attention networks, the attention mechanism enables specifying different weights to different nodes in a neighborhood [43]. Therefore, the GAT-RNN model is only tested with Boolean adjacency matrix. For the recurrent layer, long short-term memory (LSTM) is chosen in all models.

3.4 Development and validation of spatio-temporal models

All the models and methods are tested using Python programming language, as well as Keras (CNN and RNN), Spektral (GNN), and Scikit-opt (genetic algorithm) packages. The hyperparameters are identified based on a grid search in preliminary test, as shown in **Table 3**. To prevent overfitting, an early-stopping training scheme is adopted, i.e., terminating the model training process when the resulting accuracy in validation data stops increasing after a specified number of iterations.

Table 3. Grid-search settings for model hyperparameters

Hyperparameters	Grid-search values	Best setting
Activation function in hidden layers	ReLU, Sigmoid, Tanh	Sigmoid
Number of output channels in GCN layer	5, 10, 15, 20	10
Attention head number in GAT layer	5, 10, 15, 20	10
Number of output channels in DC layer	5, 10, 15, 20	10
Filter number in each 1D convolutional layer	5, 10, 20, 30	10
Stride size of 1D convolutions	1, 2	1
Number of recurrent units in each recurrent layer	5, 10, 15, 20	15

Each combination of model architectures is tested for 20 times, and the prediction accuracy of the models on testing dataset are shown in **Table 4** and **Figure 13**.

As shown in **Table 4** and **Figure 13**, the GNN-RNN models (i.e., GAT-RNN, GCN-RNN(B), GCN-RNN(W), DC-RNN(B), DC-RNN(W)) achieve better prediction performance than the conventional deep learning models (i.e., CNN-RNN, Dense-RNN), where B represents the boolean matrix and W represents the weighted matrix. GAT-RNN achieves the best prediction performance with an RMSE of 0.39±0.14, which shows a 19% improvement over the Dense-RNN architecture. The node connection in the neural networks determines the information propagation between different nodes. The results demonstrate that node connections can be designed based on prior information of physical connection (system topology) and spatial relationship between the physical entities, to alleviate the interference of unnecessary information between irrelevant nodes and ease the complexity of model training. Detailed discussion on this topic will be provided in Section 4.

In addition, a weighted adjacency matrix is tested, where each edge is weighted by the absolute value of the PCC of the node attributes. In real-world practice, the cooling demand of a large occupied zone is usually be satisfied by multiple VAV boxes. This means that one VAV box can influence the temperatures at multiple IoT sensors, and each IoT sensor measurement can be affected by multiple VAV boxes. The PCC calculation results are able to indicate, in a data-driven manner, which IoT sensors are more susceptible to the influence of a specific VAV box, or which VAV box has a greater impact on a particular IoT sensor. The weighted adjacency matrix assigns higher weights to node pairs with strong correlations. Among all GNN-RNN models, models with weighted edges (i.e., GAT-RNN, GCN-RNN(W), DC-RNN(W)) achieve better prediction performance than models with Boolean edges (i.e., GCN-RNN(B), DC-RNN(B)), indicating that the weighted edges improve the performance of equally weighted edges.

Table 4. Model prediction accuracy (RMSE: °C)

	CNN-RNN	**Dense-RNN**	**GAT-RNN**	**GCN-RNN (Boolean matrix)**	**GCN-RNN (Weighted matrix)**	**DC-RNN (Boolean matrix)**	**DC-RNN (Weighted matrix)**
RMSE (°C)	0.48±0.19	0.52±0.15	**0.39**±0.14	0.43±0.14	0.40±0.09	0.46±0.11	0.41±0.17

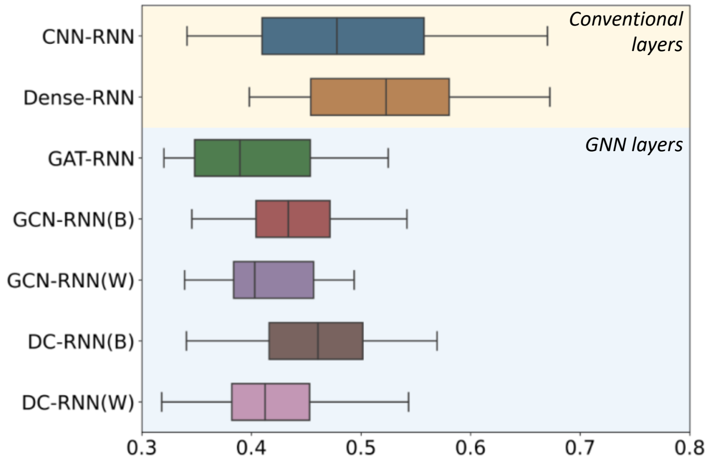


Figure 13. Model prediction accuracy (RMSE: °C)

3.5 Online test results and analysis

The GAT-RNN model which achieves the best performance is selected as the indoor environment prediction model in the online test. One-day cloud-based online test (from 10:00 to 17:00) is conducted and another day with similar outside weather condition is chosen as the reference day (under existing control strategy). **Figure 14** shows the outdoor temperature profiles of these two days. It can be seen that the outdoor air temperature profile of the test day during the testing hours is similar to that of the reference day, which ensures a comparably fair comparison on indoor environment control. Moreover, air temperatures during the first hour (10: 00-11:00) on the test day are significantly higher than that in the reference day, and the highest temperature of the test day is also higher than the reference day, indicating more unfavorable operating conditions.

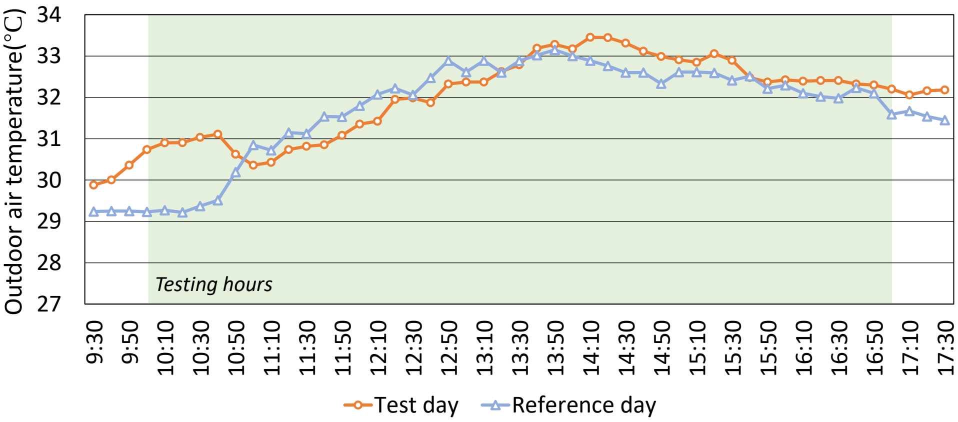
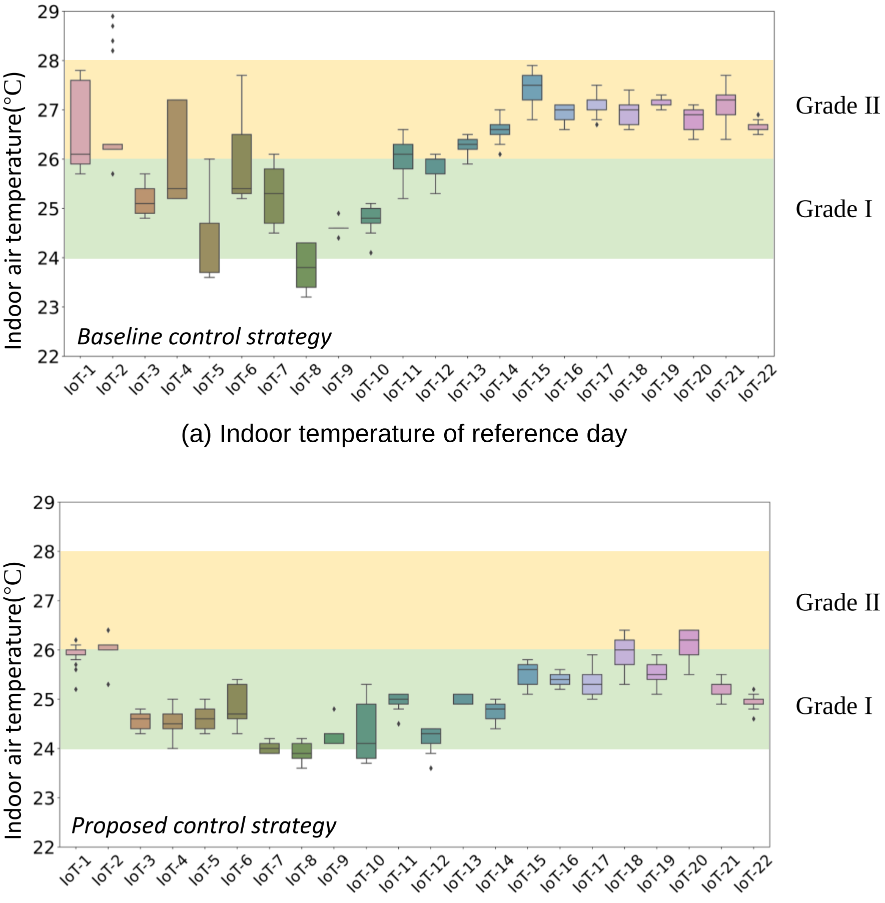


Figure 14. Weather condition on test day and reference day

The control performances from 10:00 to 17:00 of the proposed model-based online optimal control strategy and existing control strategy are shown in **Figure 15** and **Table 5**.



(a) Indoor temperature of reference day

(b) Indoor temperature of test day

Figure 15. Box plots of indoor temperature measured by IoT sensors on reference day and test day

As shown in **Figure 15**, the IoT sensors' measurements reveal significant spatial temperature variations in the occupied zone, indicating uneven distribution. In view of this, indoor environment modeling and control should account for this uneven distribution rather than relying on a single average temperature in the space. On the reference day without adopting the prediction-based temperature setpoint reset strategy, approximately half of the indoor area had a temperature within the Grade II range. On the test day, the lower figure shows that the proposed model-based control strategy can better control the indoor temperature in the occupied zone near the setpoint ($\dot{y}_{i,k} = 25$), and the indoor temperature in most area is well controlled within the Grade I range. It is worth noting that the improvement of thermal comfort was achieved without any hardware retrofit or on-site calibration.

Table 5. Control performance of the proposed model-based optimal control strategy and existing control strategy

Indoor temperature measured by IoT sensors	**Reference day**	**Test day**	**Improvement**
Max. (°C)	28.9	26.4	
Min. (°C)	23.2	23.6	
Average (°C)	26.1	25.0	
Percentage of time achieving Grade I thermal comfort (24-26°C)	36.5%	81.3%	122.3%
Percentage of time achieving Grade I & II thermal comfort (24-28°C)	94.2%	96.4%	2.4%

As shown in **Table 5**, compared with existing control strategy, the proposed model-based online optimal control strategy improves the percentage of achieving Grade I thermal comfort from 36.5% to 81.3%, and the percentage of achieving Grade I & II thermal comfort from 94.2% to 96.4%. This remarkable improvement in achieving Grade I thermal comfort can be attributed to the proposed model-based online optimal control strategy's ability to adapt to changing environmental conditions in real time. By continuously optimizing the control setpoints based on the predictions from the spatio-temporal model, the proposed strategy can maintain a consistently high level of thermal comfort during the test day.

Figure 16 visualizes the hourly indoor temperature distribution from 10:00 to 17:00 of the reference day and test day. In this case study, the outdoor temperature peaks around 14:00, while the indoor temperature peaks around 15:00 due to the delay caused by the building envelopes. It can be seen that there exists an obvious problem of uneven indoor air temperature distribution on the reference day. The indoor air temperature in small private offices (i.e., the left area) is lower than that in most open area (i.e., the lower area in the figure). As the outdoor air temperature rises, the fixed indoor temperature setpoints cannot respond to the changing cooling loads, resulting in poor indoor thermal comfort. However, the proposed strategy can alleviate the problem of uneven room temperature and maintain a comfortable indoor environment when outdoor air temperature changes.

It should be pointed out that the performance of existing control strategy is rather poor, but it was the real control strategy adopted by building operators in the past few years. The control strategy for the VAV boxes is commonly used, but the VAV box controllers were not well tuned. As the building in the case study was occupied, it is not allowed to conduct in-situ commissioning and tuning on the VAV box controllers.

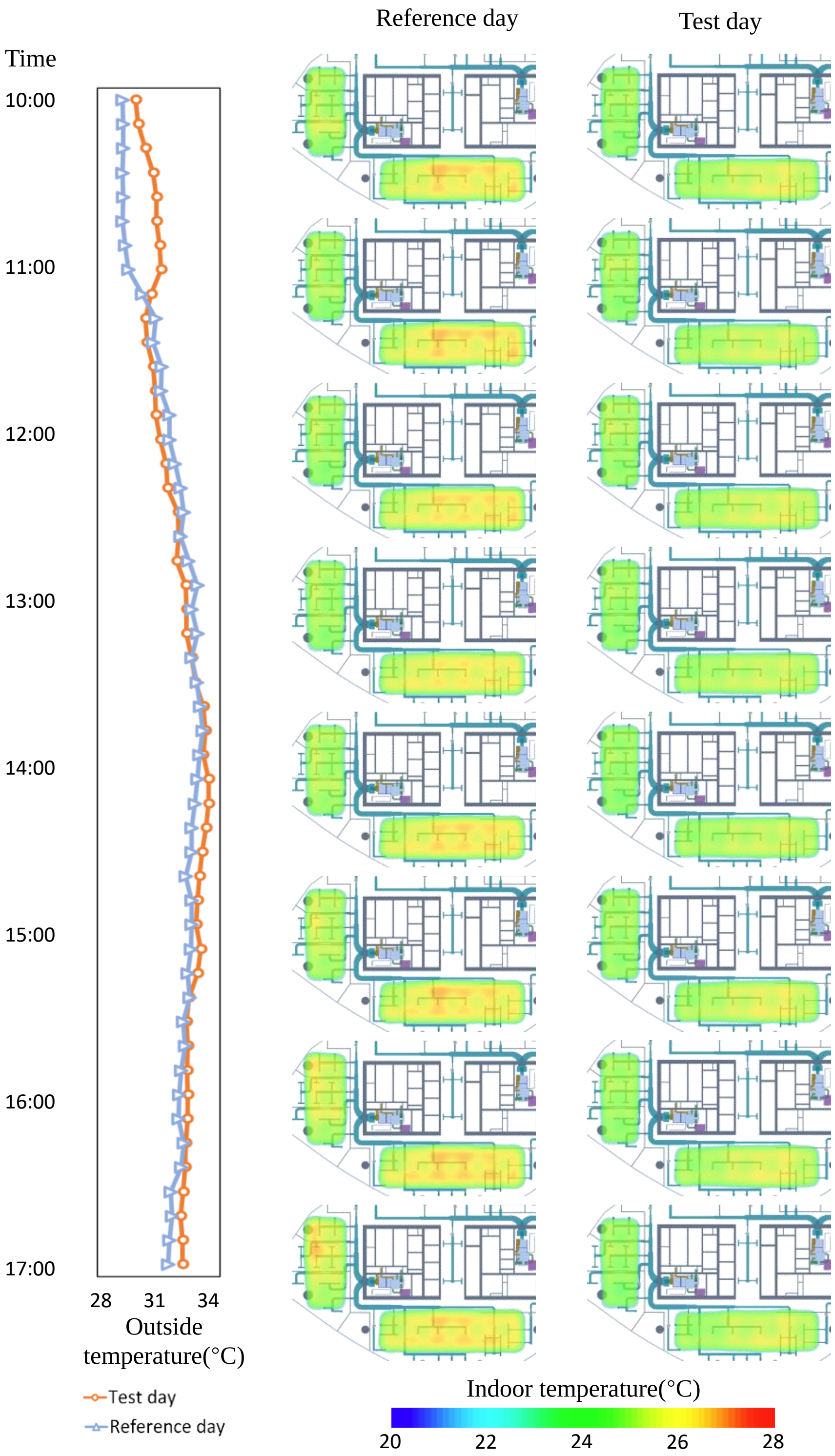


Figure 16 Indoor temperature visualization of the reference day and test day

4. Discussions and Limitations

In this study, the spatio-temporal data are manually organized by domain knowledge. However, if the entire air distribution network is complex and large, designing suitable graph structure corresponding to the system configuration could be time-consuming and labor-intensive. Image processing techniques (e.g., image identification, template matching, skeleton extraction) may help identify labels, locations, annotations, and connections of physical entities from 2D building design drawings, and design the graph structure automatically or semi-automatically [51,52]. More recently, common data representation schema for different building systems, for example, Haystack [53], Brick [54], semantic web and ontologies [55], make it possible to automatically match dynamic data from BAS or IoT sensors with the graph. The full potential of GNN in modeling spatio-temporal indoor environment can be realized in different air conditioning systems by utilizing the advanced techniques and schema mentioned above.

Two conventional layers (i.e., CNN and Dense) and three graph layers (i.e., GAT, GCN and DC) with two adjacency matrices (i.e., Boolean and Weighted) are tested in this study. Based on local connectivity, the information flow in convolutional layers is determined by the sequencing of features. Fully-connected layers connect all features to each other. It can be observed that conventional layers perform worse than the graph layers, and the Dense-RNN model performs the worst out of all models. The most likely reasons are that connecting unrelated nodes might disturb the model's inference mechanism, and the increased model parameters make the training process more complicated. For graph neural networks, on the other hand, the information propagation between nodes is determined by the graph structure. If two nodes are not directly connected, there will be no direct information propagation between them. The graph structure proposed in this study is designed based on physical connection and spatial relationship between the entities in AHU-VAV system. Compared with a graph with pairwise connections between nodes, the graph structure designed in this study removes the association between many apparently irrelevant nodes. This can significantly reduce the difficulty for the model to learn the complex correlations between multiple physical entities by aggregating information only from adjacent nodes.

The GNN-RNN model organizes the spatio-temporal data with physical information, which helps the model to propagate useful information among connected nodes and improved the model performance. Among all GNN-RNN models, the models with weighted edges (i.e., GAT-RNN, GCN-RNN(W), DC-RNN(W)) perform better than models with Boolean edges (i.e., GCN-RNN(B), DC-RNN(B)), suggesting that detailed weighted edges can further help each node to better extract information from its neighbors. Among these models with weighted edges, the GAT-RNN achieved best performance. The GAT-RNN attention mechanism is incorporated into the propagation step, allowing neural networks pay attention to how different inputs influence outputs at each step of inference in the model development process and assign different importance to different nodes [49,50], which makes it perform better. The results prove that the GNN-based spatio-temporal data-driven methods can model the uneven spatial distributions of environmental parameters in air-conditioned space and consider the spatial correlations to better understand indoor spatially distributed dynamics.

The application of the proposed method in real buildings is restricted by the availability of spatially-high-resolution sensors. For buildings which do not install IoT sensors in occupied zones, the IoT nodes and edges can be removed from the graph structure, and the remainder of the graph can still be used to a GNN model of the air conditioning system. However, the model cannot capture the interactions between the air conditioning system and the environment served by it, nor the uneven distribution of environment parameters in the occupied zone. With more and more IoT sensors deployed in smart buildings for built environment control, data availability will not be a problem for the GNN-based method developed in this study.

In actual building operation, especially for large buildings with complex energy systems, the control performance of HVAC systems is often below expectations, particularly after long operations without frequent calibrations. The reasons are multi-fold, e.g., the installation process did not adhere to the design specifications; the cooling/heating demand exceeds the design specifications; the space layout has been modified and the air conditioning system was not altered accordingly. This research presented the operation status of a typical office building. The HVAC maintenance staff adopt the fixed-setpoint control method, as a common practice, and do not change the setpoints until receiving complaints. The real operation data collected from Building Automation System and IoT sensors show that the actual control performance is barely satisfactory. For a large air-conditioned area, the control effect of different locations varies. Almost half of the area is in Grade II thermal comfort, though the AHU setpoint lies in Grade I thermal comfort. For the HVAC system control in large and open spaces, the uneven distribution of indoor air temperature is always a challenging issue. On the other hand, the proposed GNN-based predictive control method serves as an effective high-level control action that sends the optimal temperature setpoints to VAV box controllers via network communication. This eliminates the need for any in-situ work in the office area, making it a low-cost and non-intrusive solution to upgrade the performance of the existing AHU-VAV system.

5. Conclusions

This study developed a novel spatio-temporal data-driven methodology for optimal control of central air conditioning systems, which consists of graph-based multi-source data integration, graph neural network-based spatio-temporal modeling method and model-based online optimal control strategy. This study integrates multi-source spatio-temporal data as graph structure instead of traditional tabular format. Graph neural network-based spatio-temporal models are then developed for indoor environment prediction and model-based online optimal control. The GNN-based spatio-temporal models of the built environment and its air conditioning system empower the model-based optimal control strategies to optimize the thermal comfort in the entire space and in a real-time manner. It advances the conventional model-based optimal control of air conditioning systems concerning indoor environment, where indoor temperature is usually represented by a single measurement or averaged value of several measurements.

An AHU-VAV system serving a typical floor of a high-rise office building was chosen for test and validation of the proposed method. Spatio-temporal data from BAS, IoT devices and BIM are integrated into graph structure. Three GNN-RNN models (i.e., GAT-RNN, GCN-RNN and DC-RNN) with two adjacency matrices (i.e., Boolean and weighted matrix) were compared with two conventional models (i.e., CNN-RNN, Dense-RNN). GAT-RNN achieved the best performance with RMSE of 0.39±0.14°C in predicting indoor air temperature at multiple locations in the conditioned space. One day cloud-based online test was carried out in the real AHU-VAV system. The existing control strategy always kept fixed indoor air temperature setpoints based on the experience of building operators, resulting in an uneven indoor air distribution and failure to achieve the required Grade I thermal comfort in most cases. The proposed strategy used GA optimizer to optimize the indoor air temperature setpoint based on the prediction of indoor air temperature. Test results showed that the proposed spatio-temporal model-based control strategy well alleviated the problem of uneven room temperature and can achieve Grade I thermal comfort for most of the time.

This study provides a graph-based perspective to symbolize, analyze and model the AHU-VAV system and air-conditioned space together based on the physical configuration and represents a significant step towards leveraging spatio-temporal data in built environment modeling and control and highlights the need for further research and development to fully exploit the potential of such data. The spatio-temporal model can well capture the spatio-temporal relationship of central air conditioning system and link the system with the indoor environment. Using the modeling method developed, the building design information and operational data can be effectively integrated and utilized. In addition, this study provides a valuable methodology for incorporating the physical knowledge of the air conditioning system, especially the design information, into machine learning modelling, demonstrating an ideal synergy between machine learning and domain expertise.

In future work, further research will be conducted in three areas, including automating the design of graph structures of various air conditioning systems from BIM models and design drawings by taking advantage of advanced imagine processing techniques, exploring and comparing different graph structures to recommend the most suitable ones for typical AHU-VAV systems and taking account of energy consumption in the proposed optimal control strategy.

Acknowledgement

The authors gratefully acknowledge the support of this research by Innovation and Technology Fund (ITP/002/22LP) and the Research Grants Council (C5018-20GF; 15220323) of the Hong Kong SAR, China.

References

[1] Ormandy, D., & Ezratty, V. (2012). Health and thermal comfort: From WHO guidance to housing strategies. Energy Policy, 49, 116–121. https://doi.org/10.1016/j.enpol.2011.09.003.

[2] Enescu, D. (2017). A review of thermal comfort models and indicators for indoor environments. Renewable and Sustainable Energy Reviews, 79, 1353–1379. https://doi.org/10.1016/j.rser.2017.05.175.

[3] Li, W., & Wang, S. (2020). A multi-agent based distributed approach for optimal control of multi-zone ventilation systems considering indoor air quality and energy use. Applied Energy, 275, 115371. https://doi.org/10.1016/j.apenergy.2020.115371.

[4] Zhang, Y., Shan, K., Li, X., Li, H., & Wang, S. (2023). Research and Technologies for next-generation high-temperature data centers – State-of-the-arts and future perspectives. Renewable and Sustainable Energy Reviews, 171, 112991. https://doi.org/10.1016/j.rser.2022.112991.

[5] Feng, Z., Yu, C. W., & Cao, S.-J. (2019). Fast prediction for indoor environment: Models assessment. Indoor and Built Environment, 28(5), 727–730. https://doi.org/10.1177/1420326X19852450.

[6] Wei, X., Kusiak, A., Li, M., Tang, F., & Zeng, Y. (2015). Multi-objective optimization of the HVAC (heating, ventilation, and air conditioning) system performance. Energy, 83, 294–306. https://doi.org/10.1016/j.energy.2015.02.024.

[7] Zhou, P., Huang, G., Zhang, L., & Tsang, K.-F. (2015). Wireless sensor network based monitoring system for a large-scale indoor space: data process and supply air allocation optimization. Energy and Buildings, 103, 365–374. https://doi.org/10.1016/j.enbuild.2015.06.042.

[8] Sun, Z., & Wang, S. (2010). A CFD-based test method for control of indoor environment and space ventilation. Building and Environment, 45(7), 1441–1447. https://doi.org/10.1016/j.buildenv.2009.12.007.

[9] Zhu, X., Shi, T., Jin, X., & Du, Z. (2021). Multi-sensor information fusion based control for VAV systems using thermal comfort constraints. Building Simulation, 14(4), 1047–1062. https://doi.org/10.1007/s12273-020-0736-9.

[10] Du, Z., Xu, P., Jin, X., & Liu, Q. (2015). Temperature sensor placement optimization for VAV control using CFD–BES co-simulation strategy. Building and Environment, 85, 104–113. https://doi.org/10.1016/j.buildenv.2014.11.033.

[11] Zhu, H.-C., Ren, C., & Cao, S.-J. (2021). Fast prediction for multi-parameters (concentration, temperature and humidity) of indoor environment towards the online

control of HVAC system. Building Simulation, 14(3), 649–665. https://doi.org/10.1007/s12273-020-0709-z.

[12] Fan, C., Yan, D., Xiao, F., Li, A., An, J., & Kang, X. (2021). Advanced data analytics for enhancing building performances: From data-driven to big data-driven approaches. Building Simulation, 14(1), 3–24. https://doi.org/10.1007/s12273-020-0723-1.

[13] Li, X., Zhao, T., Zhang, J., & Chen, T. (2017). Predication control for indoor temperature time-delay using Elman neural network in variable air volume system. Energy and Buildings, 154, 545–552. https://doi.org/10.1016/j.enbuild.2017.09.005.

[14] Wu, S., & Sun, J.-Q. (2011). Cross-level fault detection and diagnosis of building HVAC systems. Building and Environment, 46(7), 1558–1566. https://doi.org/10.1016/j.buildenv.2011.01.017.

[15] Geng, Y., Ji, W., Xie, Y., Lin, B., & Zhuang, W. (2022). A sub-sequence clustering method for identifying daily indoor environmental patterns from massive time-series data. Automation in Construction, 139, 104303. https://doi.org/10.1016/j.autcon.2022.104303.

[16] Abdelrahman, M., Chong, A., & Miller, C. (2020). Build2Vec: Building Representation in Vector Space. arXiv preprint arXiv:2007.00740. Retrieved from http://simaud.org/2020/proceedings/102.pdf (accessed February 28, 2022).

[17] Abdelrahman, M. M., Chong, A., & Miller, C. (2022). Personal thermal comfort models using digital twins: Preference prediction with BIM-extracted spatial–temporal proximity data from Build2Vec. Building and Environment, 207, 108532. https://doi.org/10.1016/j.buildenv.2021.108532.

[18] Wang, S., Cao, J., & Yu, P. S. (2022). Deep Learning for Spatio-Temporal Data Mining: A Survey. IEEE Transactions on Knowledge and Data Engineering, 34(12), 3681–3700. https://doi.org/10.1109/TKDE.2020.3025580.

[19] Zhou, S., Huang, X., Liu, N., Chung, F.-L., & Huang, L.-K. (2022). Improving Generalizability of Graph Anomaly Detection Models via Data Augmentation. arXiv preprint arXiv:2209.10168. Retrieved from http://arxiv.org/abs/2209.10168 (accessed February 13, 2023).

[20] Esser, S., & Fahland, D. (2021). Multi-Dimensional Event Data in Graph Databases. Journal of Data Semantics, 10, 109–141. https://doi.org/10.1007/s13740-021-00122-1.

[21] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. AI Open, 1, 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001.

[22] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386.

[23] Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., & Battaglia, P. (2020). Learning to Simulate Complex Physics with Graph Networks. In Proceedings of the 37th International Conference on Machine Learning (ICML) (pp. 8459–8468). PMLR. https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html (accessed November 29, 2022).

[24] Hu, Y., Cheng, X., Wang, S., Chen, J., Zhao, T., & Dai, E. (2022). Time series forecasting for urban building energy consumption based on graph convolutional network. Applied Energy, 307, 118231. https://doi.org/10.1016/j.apenergy.2021.118231.

[25] Azhar, S. (2011). Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. Leadership and Management in Engineering, 11(3), 241–252. https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127.

[26] Xiao, F., & Fan, C. (2022). Building information modeling and building automation systems data integration and big data analytics for building energy management. Research Companion to Building Information Modeling (pp. 525–549).

[27] Javed, A., Larijani, H., Ahmadinia, A., Emmanuel, R., Mannion, M., & Gibson, D. (2017). Design and Implementation of a Cloud Enabled Random Neural Network-Based Decentralized Smart Controller With Intelligent Sensor Nodes for HVAC. IEEE Internet of Things Journal, 4(2), 393–403. https://doi.org/10.1109/JIOT.2016.2627403.

[28] Fan, C., Xiao, F., Song, M., & Wang, J. (2019). A graph mining-based methodology for discovering and visualizing high-level knowledge for building energy management. Applied Energy, 251, 113395. https://doi.org/10.1016/j.apenergy.2019.113395.

[29] Fan, C., Xiao, F., Li, Z., & Wang, J. (2018). Unsupervised data analytics in mining big building operational data for energy efficiency enhancement: A review. Energy and Buildings, 159, 296–308. https://doi.org/10.1016/j.enbuild.2017.11.008.

[30] Sun, M., Li, S., Zhang, Y., Liu, Y., He, S., & Rao, G. (Eds.). (2020). Chinese Computational Linguistics: 19th China National Conference, CCL 2020, Hainan, China, October 30 – November 1, 2020, Proceedings. Springer International Publishing. https://doi.org/10.1007/978-3-030-63031-7.

[31] Wang, Z., Wei, W., Cong, G., Li, X.-L., Mao, X.-L., & Qiu, M. (2020). Global Context Enhanced Graph Neural Networks for Session-based Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 169–178). ACM. https://doi.org/10.1145/3397271.3401142.

[32] Li, A., Fan, C., Xiao, F., & Chen, Z. (2022). Distance measures in building informatics: An in-depth assessment through typical tasks in building energy

management. Energy and Buildings, 258, 111817. https://doi.org/10.1016/j.enbuild.2021.111817.

[33] Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson Correlation Coefficient. In Noise Reduction in Speech Processing (pp. 1–4). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-00296-0_5.

[34] Han, J., Liu, H., Zhu, H., Xiong, H., & Dou, D. (n.d.). Joint Air Quality and Weather Predictions Based on Multi-Adversarial Spatiotemporal Networks, 9.

[35] Sperduti, A., & Starita, A. (1997). Supervised neural networks for the classification of structures. IEEE Transactions on Neural Networks, 8(3), 714–735. https://doi.org/10.1109/72.572108.

[36] Frasconi, P., Gori, M., & Sperduti, A. (1998). A general framework for adaptive processing of data structures. IEEE Transactions on Neural Networks, 9(5), 768–786. https://doi.org/10.1109/72.712151.

[37] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791.

[38] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. IEEE Transactions on Neural Networks, 20(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605.

[39] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., & Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks.

arXiv preprint arXiv:1806.01261. Retrieved from http://arxiv.org/abs/1806.01261 (accessed May 13, 2022).

[40] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Advances in Neural Information Processing Systems (pp. 3844–3852). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html (accessed December 26, 2022).

[41] Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. arXiv preprint arXiv:1609.02907. https://doi.org/10.48550/arXiv.1609.02907.

[42] Li, R., Wang, S., Zhu, F., & Huang, J. (2018). Adaptive Graph Convolutional Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11691.

[43] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks. arXiv preprint arXiv:1710.10903. https://doi.org/10.48550/arXiv.1710.10903.

[44] Medsker, L., & Jain, L. C. (1999). Recurrent Neural Networks: Design and Applications. CRC Press.

[45] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[46] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv preprint arXiv:1412.3555. https://doi.org/10.48550/arXiv.1412.3555.

[47] MHURD. (2012). GB 50736-2012: Design code for heating ventilation and air conditioning of civil buildings.

[48] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. arXiv preprint arXiv:1707.01926. http://arxiv.org/abs/1707.01926 (accessed December 28, 2022).

[49] Bahdanau, D., Cho, K., & Bengio, Y. (2016). Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473. https://doi.org/10.48550/arXiv.1409.0473.

[50] Li, A., Xiao, F., Zhang, C., & Fan, C. (2021). Attention-based interpretable neural network for building cooling load prediction. Applied Energy, 299, 117238. https://doi.org/10.1016/j.apenergy.2021.117238.

[51] Braun, A., & Borrmann, A. (2019). Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning. Automation in Construction, 106, 102879. https://doi.org/10.1016/j.autcon.2019.102879.

[52] Lu, Q., Chen, L., Li, S., & Pitt, M. (2020). Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings. Automation in Construction, 115, 103183. https://doi.org/10.1016/j.autcon.2020.103183.

[53] Project Haystack. (2014). Home - Project Haystack. https://project-haystack.org/.

[54] Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Bergés, M., Culler, D., Gupta, R. K., Kjærgaard, M. B., Srivastava, M., & Whitehouse, K. (2018). Brick: Metadata schema for portable smart building applications. Applied Energy, 226, 1273–1292. https://doi.org/10.1016/j.apenergy.2018.02.091.

[55] Zhang, C., Romagnoli, A., Zhou, L., & Kraft, M. (2017). Knowledge management of eco-industrial park for efficient energy utilization through ontology-based approach. Applied Energy, 204, 1412–1421. https://doi.org/10.1016/j.apenergy.2017.03.130.