https://dx.doi.org/10.1016/j.aei.2024.102379

Design information-assisted graph neural network for modeling central air conditioning systems

Ao Li[1], Jing Zhang[1], Fu Xiao[1, 2*], Cheng Fan[3,4], Yantao Yu[5] and Zhe Chen[1]

[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] Key Laboratory for Resilient Infrastructures of Coastal Cities (Shenzhen University), Ministry of Education, China

[4] Sino-Australia Joint Research Center in BIM and Smart Construction, Shenzhen University, Shenzhen, China

[5] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China

Abstract

Buildings consume huge amounts of energy to create a comfortable and healthy built environment for people. The building engineering industry has benefitted from the advances in building informatics, including the rich data available in modern buildings and the rapid development in computing technology and data science, for building energy management. Dynamic modeling is often essential to online control and optimization of building energy systems. Data-driven modeling empowered by advanced machine learning has achieved ground-breaking performance in capturing temporal relationships among multivariate building operation data in recent years. However, the structural relationships among the physical entities, e.g., the topology of air conditioning ductworks and terminals, are generally overlooked in existing data-driven modeling

methods, although they are very helpful in capturing the relationships among building operation data. This study proposes to represent building air conditioning systems as graphs for machine learning, whose nodes and edges represent physical entities (e.g., VAV terminals) and their connections (e.g., ductwork), respectively. A novel graph neural network-based methodology is developed for dynamic modeling of central air conditioning systems, which consists of three steps, i.e., automated graph structure design, development of graph neural network, and model evaluation and explanation. Viable and generalizable graph structure design methods based on design information, e.g., design drawings and BIM models, and machine learning algorithms for model development and explanation are proposed.

A case study of dynamic modeling of a real central air conditioning system serving the tallest building in Hong Kong is carried out by adopting the methodology developed. Image identification techniques are employed to extract the topology of the air conditioning system from 2D schematic drawings, which is used as the prototype of the graphs. Graph neural network-based models, consisting of a graph layer and a recurrent layer, are developed to capture the structural and temporal relationships, separately. The graph layer learns structural relationships from the input graphs by using graph convolutional network (GCN) and graph attention network (GAT). The recurrent layer learns temporal relationships from massive historical operation data by using LSTM. As the models become much "darker", the Shapley additive explanations (SHAP) method is adopted to provide both global and local model explanations, i.e., the impact of each model input on outputs. The developed models exhibit improved capabilities of automated model architecture design, prediction accuracy, generalizability and interpretability. The methodology developed for dynamic modeling of air conditioning systems in this study leverages both traditional building

system design information and powerful machine learning algorithms, and exemplifies an ideal synergy between engineering domain expertise and machine learning.

Keywords: machine learning, graph neural network, dynamic modeling, image identification, air conditioning system

1. Introduction

Building energy management is critical for worldwide sustainable development due to buildings' high energy consumption. The building sector accounts for about 36% of global energy consumption and 37% of energy-related greenhouse gas emissions, providing enormous potential to achieve climate goals [1]. Meanwhile, buildings are becoming major users in electric power systems, affecting the supply-demand balance and grid reliability significantly. For instance, buildings contribute to approximately 94% of total electricity use in Hong Kong [2], calling for more attention and measures to building energy management. As a critical aspect of modern engineering informatics, there is a promising transition in building energy management, from mainly relying on domain and expert knowledge to harnessing the power of machine learning and artificial intelligence [3]. Building informatics, which is concerned with building information management throughout the whole building lifecycle relying on rich building data and powerful machine learning, plays an increasingly important role in improving building management and reducing energy consumption [4].

Model-based optimization has proven to be highly effective in reducing energy consumption of heating, ventilation and air conditioning (HVAC) systems [5,6,7,8], which are major source of energy consumption in buildings. However, developing accurate dynamic models for HVAC systems and the air-conditioned spaces is challenging due to the complex nonlinear nature of

HVAC systems, which are heavily influenced by meteorological conditions, building operating modes, occupant behaviour, etc [6]. The dynamic modeling approaches can be broadly classified into two categories, i.e., forward approach and data-driven approach. The data-driven approach constructs models from massive amount of historical building operation data by using advanced machine learning algorithms, which has gained increasing attention in recent years due to the computational efficiency and capability to tackle nonlinear tasks and capture temporal correlations among variables/features [9,10]. Yang et al. [10] developed a model predictive control (MPC) system using machine learning-based building dynamic models to control air-conditioning and mechanical ventilation systems. Two artificial neural network models, including a building dynamic model for predicting room air temperature and a thermal comfort model for predicting indoor PMV (Predicted Mean Vote), were developed for an office room. The MPC achieved a reduction of 58.5% cooling thermal energy consumption. Li et al. [11] 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. In real practice, for large and complex air-conditioned spaces, the temperature may vary widely between different areas. However, well-mixing assumption, which was widely employed in managing central air conditioning systems in previous studies [12,13], can insufficiently reflect the spatial variation of the control variables and potentially contribute to localized discomfort and undesirable indoor conditions [14]. The spatial dynamics are just as crucial to the control optimization of HVAC systems as the temporal dynamics [15].

However, existing research adopting the data-driven approach to modeling HVAC systems pays more attention to the temporal relationships among operation data, while generally overlooking the valuable spatial relationships among physical entities (such as sensor locations and

configuration or topology of air conditioning system in the space). One of the key issues is to incorporate these spatial relationships in the data-driven models to improve the model accuracy and interpretability, as well as the control performance. Typical physical entities in building air conditioning systems include variable air volume (VAV) boxes, air handling units (AHUs), heat exchangers, pumps and fans connecting by pipes and ducts. The structural information is rich and readily available in most building design drawings. Figure 1 shows the structures of three typical building energy systems, i.e., the configuration or connections of the physical entities. This kind of structural information is difficult to be leveraged by traditional data-driven modeling methods which typically receive 2-dimensiona tables or matrixes as model inputs. Dynamic modeling of building energy systems entails simulating the behaviour and interactions of many physical entities [7]. The interactions among those entities are closely related to how they are connected, as depicted in design drawings, a specific type of graphs. To this end, the graph-based machine learning algorithms, which accept graphs as inputs, have an intrinsic advantage in capturing structural relationships in modeling HVAC systems.

Graph-based machine learning algorithms have been adopted in previous research, e.g., graph-based data mining [19], probabilistic graphical model [24]. A graph used in graph-based machine learning algorithms usually consists of nodes and edges [17-20]. When applied to building energy systems, the nodes are physical entities like VAV boxes and heat exchangers, and the edges represent the physical connections among a collection of entities. The physical information (e.g., sensor measurements and control signals) is embedded in nodes or edges in the form of scalars. Conventional machine learning algorithms primarily take rectangular or grid-like samples/data as input, which are incapable of handling data of irregular graph structures [20,21]. And directly transforming the non-Euclidean structure data like the graphs of building energy systems into

Euclidean rectangular data cannot realize one-to-one mapping since the order of physical entities as features cannot be specified. This kind of data transformation, although has been adopted in existing research [22,23], may cause structure information loss. Moreover, the graph structures in those studies do not correspond to the structure of real physical systems and lack physical meanings.

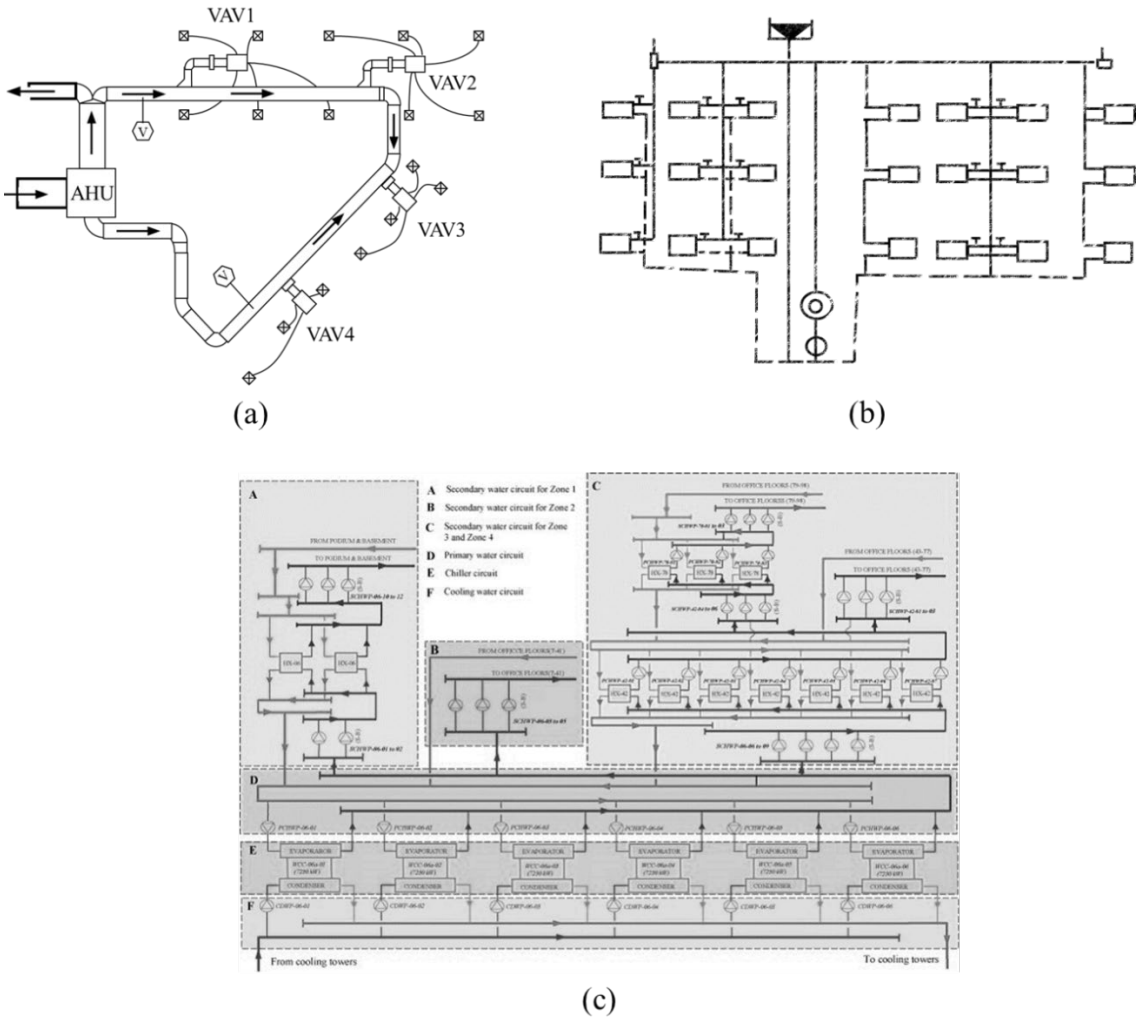


Figure 1. Examples of building energy systems with non-Euclidean structure (a) an AHU-VAV system, (b) a building hot water heating system, (c) a central chiller plant and chilled water distribution system [16]

In recent years, the advancement of graph neural networks (GNNs) has prompted researchers to further exploit the potential for physical representation of graphs [20,21]. In principle, GNNs employ a "graph-in, graph-out" paradigm based on an iterative scheme of information diffusion among neighbouring nodes, without changing the configuration of the input graph [17]. This enables graph neural networks to efficiently capture the structural dependency among the input features. Sanchez-Gonzalez et al. [25] proposed a graph neural network-based framework for simulating complex physical systems, in which physical states were represented by graphs of interacting particles, and complex dynamics were approximated by learned message-passing among nodes. Results showed that the GNN-based framework could accurately simulate a wide range of physical systems in which fluids, rigid solids, and deformable materials interact with one another. And the model developed was simpler and more accurate, and demonstrated better generalization ability than previous methods for modeling fluid dynamics. Hu et al. developed a spatial-temporal graph convolutional network (GCN) [18], one type of GNN, for energy consumption prediction of multiple neighbouring buildings. Each node in the graph represented a building, and edges represented their solar impacts with the width of the edges representing the impact level. The results showed that the GCN model outperformed time series machine learning models (including GRU, XGBoost and MLP). Not only have GNNs been successful in simulating physical systems, but they have also shown promising potential in the areas of traffic prediction [26], recommendation systems [27], and fake news detection [28]. Emerging studies have indicated that leveraging physical knowledge, like structural information, in machine learning-based modeling of physical systems may provide encouraging outcomes [18,25]. Given the rich design information of building energy systems, e.g., design drawings which can effectively assist to generate graphs as model input, as well as huge operation data for model training, GNN is

specifically suitable for dynamic modeling of building energy systems without losing structural information, which, however, has not been sufficiently investigated.

This study proposes a design information-assisted GNN-based methodology for dynamic modeling of HVAC systems. The GNN model uses the graphs representing the topology of building energy systems as input, which can be identified from the design information, such as the 2D design schematic drawings and Building Information Modelling (BIM) models. In this way, the structural information of the building energy systems can be captured by the model. In addition, a recurrent layer is designed in the GNN model to capture the dynamics of the system. The remaining part of this paper is organized as follows. Chapter 2 describes the methodology developed for dynamic modeling of HVAC systems and proposes suitable methods and algorithms. Chapter 3 presents the case study on adopting the methodology to develop dynamic models of the central air conditioning system in a high-rise commercial building in Hong Kong. The case study results and discussions are presented in Chapter 4. Chapter 5 concludes the paper.

2. Description of the methodology developed

2.1 Overview of the methodology

The design information-assisted GNN-based methodology is illustrated in Figure 2, which consists of three steps, automated graph structure design, development of graph neural network, and model evaluation and explanation. Graph structure design aims to represent the target system as a structural graph using image identification or semantic data transformation techniques. The measured data are attached to the corresponding nodes which form a matrix of input features. Although experienced building engineers can design the graph structure based on their understanding of the building energy system, this design is tailor-made for an individual building and does not apply to different buildings. An automated graph structure design method which can

be applied to various buildings is needed. Step 2 develops a GNN model, consisting of a graph layer, a recurrent layer and a fully-connected layer, for dynamic modeling of building energy systems. The model uses the graph structure data generated in Step 1 as input to capture the structural relationships, and a recurrent layer to capture the temporal relationships. Step 3 aims to evaluate the performance of the models and explain how the data-driven black-box models come to their outputs from the input.

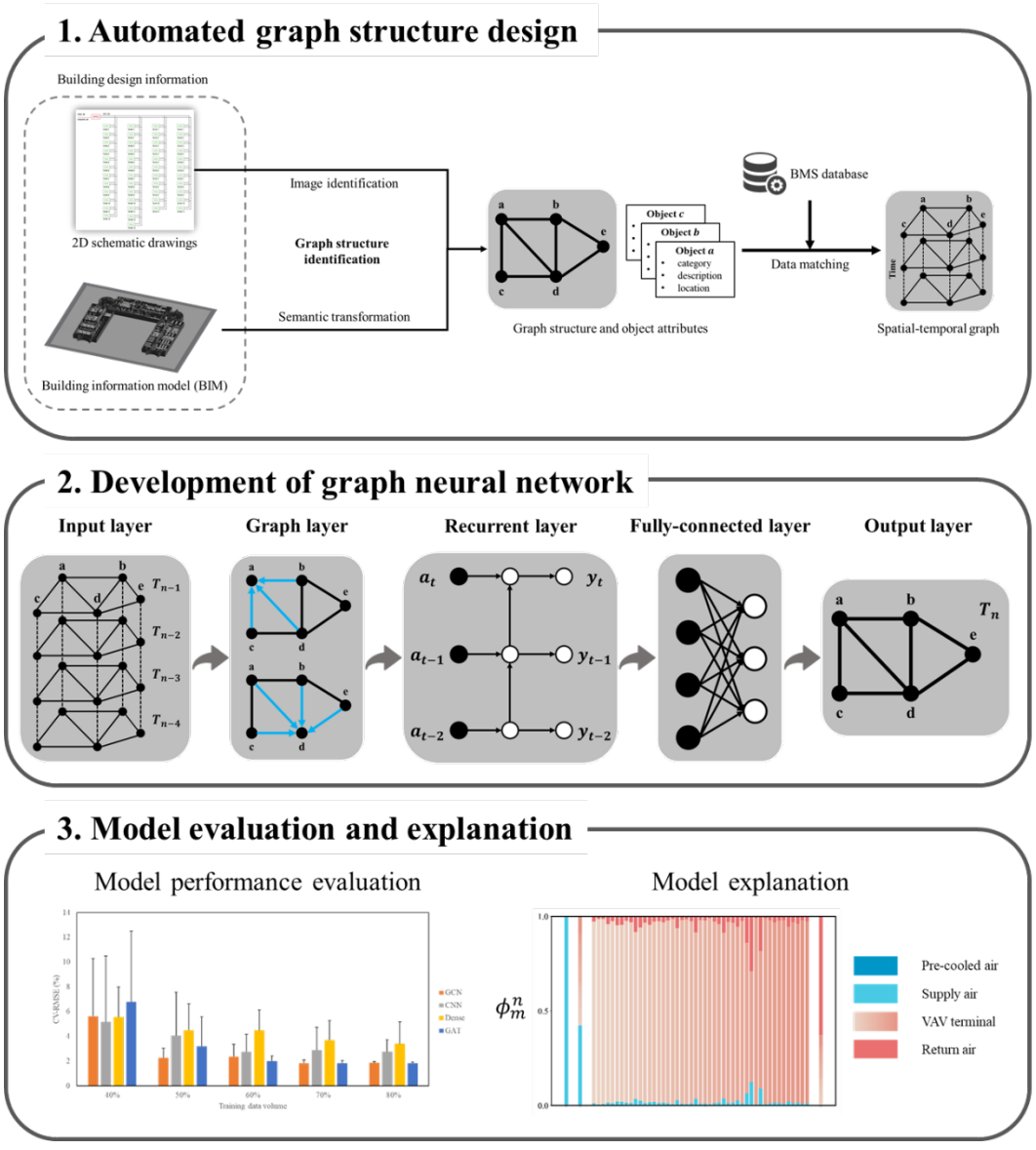


Figure 2. Block diagram of the methodology developed

Section 2.2 introduces the graph theory and proposes the methods for identifying graph structures from design drawings and BIM models. Section 2.3 provides a brief introduction to GNNs and their typical variations, and proposes the architecture for the GNN-model to capture both structural and temporal relationships. Section 2.4 elaborates on the significance of model explanation and the frequently adopted techniques for explaining machine learning models.

2.2 Automated graph structure design

A graph is recognized as one of the most generic, natural, informative and interpretable formats for data representation [19]. In graph theory, a graph can be represented as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is the set of nodes, and $\boldsymbol{E}$ is the set of edges [20]. Let $v_i \in \boldsymbol{V}$ denote a node and $e_{ij} = (v_i, v_j) \in \boldsymbol{E}$ denote an edge which connects the i^{th} and j^{th} nodes. When depicting the building energy system as a graph, the nodes in the graph can represent the physical entities [29] (e.g., VAV boxes, indoor spaces), while the edges represent their physical relationship [19] (e.g., heat transfer between neighbouring adjacent spaces, connections by piping and ductwork systems). The adjacency matrix, a matrix of Booleans, is the most common way of representing the node connections in a graph. The adjacency matrix $\boldsymbol{A}$ is a $N \times N$ matrix with $\boldsymbol{A}_{ij} = 1$ if $e_{ij} \in \boldsymbol{E}$ and $\boldsymbol{A}_{ij} = 0$ if $e_{ij} \notin \boldsymbol{E}$, and N is the total number of nodes (i.e., physical entities) concerned in the system. A graph is usually associated with a node attribute $\boldsymbol{X}$, where $\boldsymbol{X} \in \mathbb{R}^{N \times M}$ is a feature matrix with $\boldsymbol{X}_i \in \mathbb{R}^M$ representing the M-dimentional feature vector of node v_i. The graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{A})$ of the target system, to be used as the input of the model, is usually unavailable or incomplete at the beginning of model development. Since incorrect or incomplete graphs will pose difficulties throughout the modeling process, it is crucial to accurately extract the graph structure of the target system. Fewer researchers investigated extracting graph-based knowledge from Building Information Model (BIM) [30,31] and system schematic diagram manually prepared [32].

2D schematic drawing. 2D schematic drawings show physical connections between entities in the building energy system. The topology of the system, as the prototype of the graph, can be extracted from those drawings either manually based on domain expertise [32] or automatically using image identification techniques [33-36]. In addition, image identification aims to estimate the concepts and locations of objects in each image [35]. Following the identification of nodes and links, the overall graph structure can be generated based on their connections. However, there is no readily available reference and tools for identifying the graph from the building design drawings. Experience must be borrowed from other fields. Deep learning-based image identification is powerful and efficient when processing large amounts of images [34]. In the transportation industry, image identification is widely-used to extract the topology of roads for traffic prediction from satellite photographs [26]. However, it is worth noting that the training of a well-performing identifier or classifier requires large-scale labelled images. Currently, there appears to be no open and labelled image dataset for 2D building schematic drawings [34], which means manual labelling is required.

Meanwhile, the traditional template matching method provides an alternative to the deep-learning based methods that do not require any labelled images for training image identifier [37]. Within computer vision, template matching is a standard image identification method for finding small parts of an image (e.g., 2D design drawing) which match a template image (e.g., legend of objects) [38]. It slides the template image over the input image (as in 2D convolution) and calculates the similarity/correlation of the template image and the corresponding input image windows. Template matching is reliable and precise however computationally inefficient when a small scanning interval is employed (e.g., one pixel). And the scale of the template image and the object to be detected in the input image should be identical for this method to work.

Building Information Modelling (BIM). Extracting the topology of building energy systems from Building Information Modelling (BIM) based on semantic data transformation method is another option [39]. BIM models contain static information about the physical entity/object (e.g., name and object type), characteristics and attributes of object (e.g., material, color and thermal properties), and relationships between objects (e.g., locations, connections and ownership) [40,41]. Nodes can be defined by identifying the physical entities while edges can be defined by identifying the connections between the entities, while dynamic data (time sequences or streams) from Building Automation System (BAS) or IoT sensors need to be manually matched with graphs due to the heterogeneous data from different systems. More recently, Project Haystack [42], Brick [43], and Semantic web and ontologies [44] aim to address data heterogeneity in buildings that make it possible to automatically match dynamic data from BAS or IoT sensors with the graph.

2.3 Graph neural network

The earliest motivation for GNNs can root in the 1990s owing to graph analysis and graph representation learning [45,46], while the recent re-advancement of GNNs is mainly attributed to the success of deep neural networks, particularly convolutional neural networks (CNN) [46]. CNNs are capable of extracting multi-scale localized spatial characteristics and integrating them to construct highly expressive representations, which has led to breakthroughs practically in every aspect of machine learning and ushered in the era of deep learning [47]. The key elements of CNNs are local connections, shared weights and utilization of multiple layers [48], which are equally critical when handling tasks with graph data. However, CNNs can only work on Euclidean structure data such as text (1D sequences) and 2D grid images, while it is hard to generalize CNNs to graphs of non-Euclidean structures. In addition, if directly converting irregular graph data into

rectangular grid samples, the local connections in the graph domain will lose since the order of features (i.e., physical entities) varies.

Graph neural networks primarily adopt the matrix to represent the graph structure and to assist the message propagation within neural layers. In graph theory, an adjacency matrix is a square matrix used to represent a finite graph. The elements of the adjacency matrix indicate whether pairs of nodes are adjacent or not in the graph. Thus, when the rectangular data transformed from graph data is fed into GNNs alongside the adjacency matrix, the adjacency matrix preserves the connection relationships between nodes in the graph data. Based on the connection information provided by the adjacency matrix, graph neural networks can efficiently capture graphs' structural dependencies via message propagation between the nodes by aggregating information only from adjacent nodes.

Depending on the information aggregation method [20], GNNs can be further classified into several types, e.g., graph convolution networks (GCNs), graph attention networks (GATs), graph spatial-temporal networks, etc. Among these networks, GCNs play an essential role in capturing structural dependencies, whereas networks in other categories partially rely on GCNs in building blocks [20]. GCNs implement convolutional feature extraction through neighbourhood aggregation [51]. For instance, spectral-based graph convolutional layer follows the propagation rule:

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

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 $\widetilde{\boldsymbol{A}}$, $\widetilde{D_{ii}} = \sum_j \widetilde{\boldsymbol{A}}_{ij}$. $\boldsymbol{W}^{(l)}$ is a layer-specific trainable weight matrix.

$\sigma(\cdot)$ denotes an activation function (e.g., RELU and sigmoid). This propagation rule can be motivated via a first-order approximation of localized spectral filters on graphs [50,51]. The attention mechanism can be incorporated into the propagation step, which produces graph attention networks (GATs) [52]. The attention mechanism allows neural networks to pay attention to how different inputs influence outputs at each step of inference in the model development process [53,54]. One node usually has multiple neighbouring nodes, and these neighbours may have distinct impacts on it, as the center node. GATs compute the hidden states of each node by attending to its neighbours via a self-attention strategy. Overall, GCNs and GATs are the most prevalent forms of GNNs in use today.

Dynamic modeling of building energy systems is typically regarded as a time series forecasting task in data science. The model input can be represented as a graph, which has a global graph structure with inputs to each node evolving over time [55]. It is defined as $\boldsymbol{G'} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{A}, \boldsymbol{X'})$ with $\boldsymbol{X'} \in \mathbb{R}^{T \times N \times M}$ where T is the length of time squences. The most straightforward way to model a temporal graph, which include a sequence of smapshots, is to design a separate GNN to handle each snapshot of the graph and then feed the output of each GNN to a time series component/layer, such as an RNN [56,57]. Another way is to capture the temporal and structural relationships simultaneously by integrating GNN and RNN in one layer (DCRNN [55], GCRN-M2 [56]). For instance, inspired by the well-known ConvLSTM [58], GCRN-M2 amounts to ConvLSTM where graph convolutions substitute the conventional convolutions. There is no limitation of the architecture as any GNN (e.g., GCN, GAT, ChebNet) and RNN (e.g., LSTM, GRU) can be used [57]. Selection of the suitable architecture is usually based on the performance evaluation of the models for a target system.

2.4 Model explanation

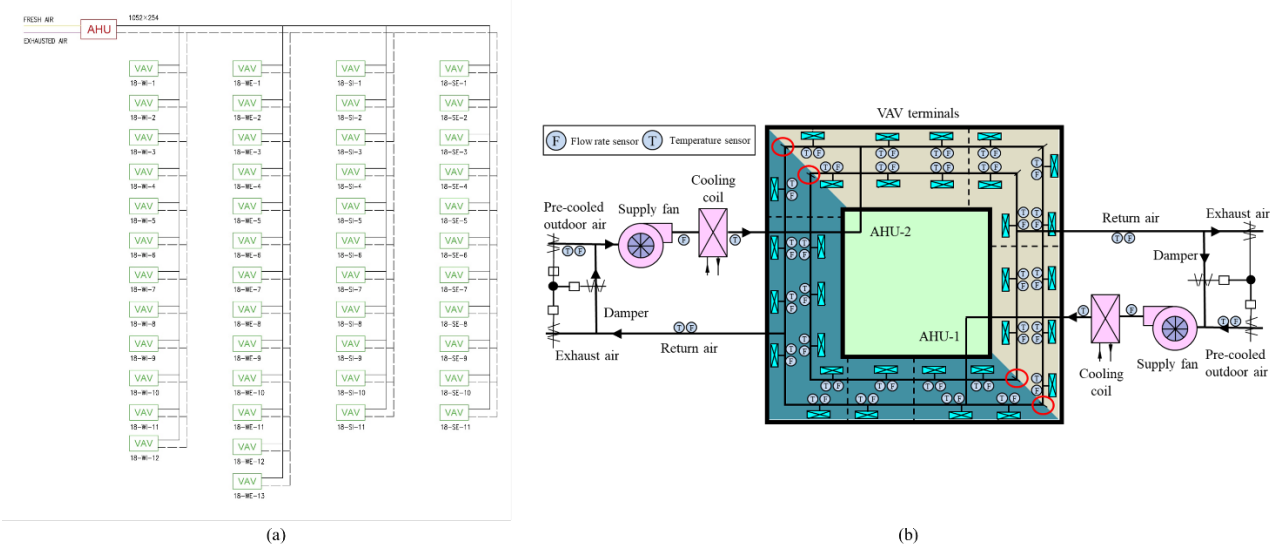
Along with the growing research interest, the intrinsic low interpretability of complex machine learning models has also resulted in significant and non-negligible skepticism in practical applications [59,86]. While machine learning models are becoming increasingly significant in smart building management, building professionals must first gain a basic understanding of how the models work and perform before confidently and widely embracing the technology [54]. Additional energy savings are possible if decision-makers are capable of understanding and trusting the underlying mechanisms of models [59,60]. The graph-based methodology developed aims to address the aforementioned problems from two aspects. First, As the message propagation of features is constrained by the adjacency matrix which is constructed according to the physical structure of a real building energy system, the graph neural network developed in this study is inherently interpretable on a modular level [61,62]. On the other hand, post-hoc model explanation methods will also be adopted to explain how a model works and which features are more important in the modeling process, after the model is established.

Post-hoc explanation methods for machine learning models can be divided into global explanation and local explanation. Global explanations are made based on a holistic view of the model architecture and parameters (e.g., permutation feature importance [63] and partial dependency test [64,65]). Global explanation facilitates the understanding of the whole logic of a model and follows the entire reasoning leading to all the different possible outcomes [66]. For instance, the data permutation method calculates the feature importance by computing the decrease of prediction accuracy resulting from randomly permuting the values of a feature. The higher the drop in prediction accuracy, the more important a variable is, and vice versa. However, in practice, it is very challenging to achieve accurate global model interpretations/explanations, especially

when there exist many correlated input variables. On the other hand, local explanations focus on individual data instances and investigate why a certain prediction is made for each instance [66,67]. Local explanation methods are more frequently employed in deep neural networks than global explanation methods [66]. Widely-used local explanation methods include individual conditional expectation curves [62,65], local surrogate models (LIME) [59,67], counterfactual explanations [69], and Shapley additive explanations (SHAP) [60,70]. For instance, LIME aims to learn a local interpretable surrogate model (e.g., linear regression and decision trees), of the globally complex/black-box model, in the neighborhood of a certain data instance [59,67].

3. Case study

3.1 Design information and data retrieved from a real central air conditioning system



(a) (b)

Figure 3. The target AHU-VAV system (a) simplified 2D schematic drawing; (b) simplified layout drawing

A case study is carried out, which adopts the developed GNN-based methodology for dynamic modeling of the central air conditioning system serving a typical floor of the tallest building in

Hong Kong, the International Commerce Centre (ICC). This building is about 490m high with 108 floors, accommodating retail shops, offices, and luxury hotels. The typical office floor with a total gross floor area of 3600 m^2 is served by the variable air volume (VAV) air conditioning system. Two identical air-handling units (AHUs) are installed on each typical floor which separately serves half of the office area. Only half of the AHU-VAV system, consisting of one AHU and 47 VAV terminals, is considered in the case study. The simplified schematic drawing and layout of the AHU-VAV system concerned are illustrated in Figure 3.

The thermodynamic-based approach for the detailed simulation of indoor temperature distribution requires the temperature and flow rate of supply air as inputs [7]. The dynamic modeling of this AHU-VAV system aims to predict the one-step ahead temperature at each VAV box (a node in the graph) based on the previous temperatures and flowrates of all VAV boxes, PAU and AHU, which is regarded as a node-level time series prediction task.

$$\boldsymbol{Y} = f\left(\boldsymbol{X}'\right) \tag{2}$$

$$\boldsymbol{X}' = \left[x_{i,m}^{t}\right], t \in [t-T, \dots, t-1], i \in \boldsymbol{V}, m \in [Temp, Flow] \tag{3}$$

$$\boldsymbol{Y} = \left[x_{i,Temp}^{t}\right], i \in \boldsymbol{V} \tag{4}$$

Where $\boldsymbol{X}'$ and $\boldsymbol{Y}$ represent the model input (i.e., graph associated feature matrix) and model output. $x_{i,Temp}^{t}$ and $x_{i,Flow}^{t}$ represent the temperature and flowrate at node i at time t, respectively. For each VAV box, measurements of the return air temperature and the supply air flowrate are collected and stored in the Building Automation System (BAS). T represents the length of time sequence, which is set as four timesteps in this study. And $\boldsymbol{V}$ is the collection of all nodes (i.e., the VAV box, PAU and AHU) concerned.

One-week data from 14th to 20th October 2016 with a collection interval of 5 minutes were retrieved from BAS. The whole dataset (containing more than six hundred observations) is transformed into subsequences via a sliding-window manner, with the sliding window interval set as 10 minutes. Keeping sufficient data for model training while maintaining a certain level of difference between the data samples is the goal here. 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.

3.2 Automated graph structure design based on 2D schematic drawings using image identification techniques

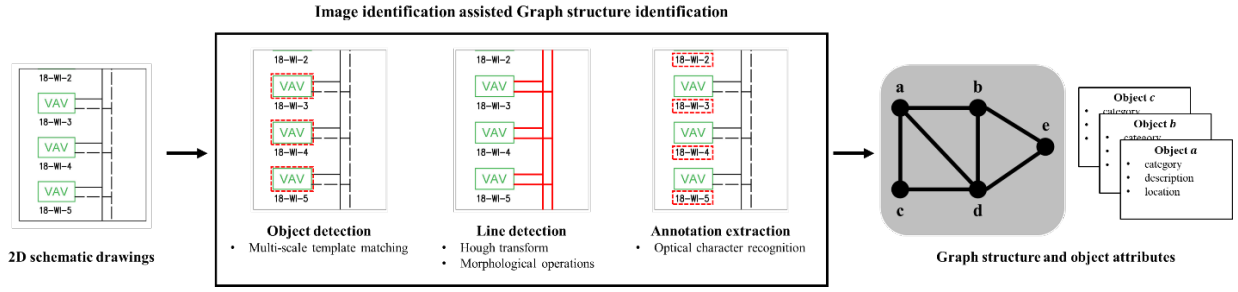


Figure 4. Graph structure identification scheme (from 2D schematic drawings) based on image identification techniques

Most existing buildings lack BIM models but do keep design drawings in either electronic or printed versions. An automated graph structure design method from 2D schematic drawings based on image identification techniques is proposed in this study, as shown in Figure 4. Although the graph structure design in this case study can be done by an experienced building engineer, the automated method is a must for mass and convenient deployment of the modeling methodology in various buildings. The automated graph structure design method needs to accomplish three tasks:

object detection based on multi-scale template matching, line detection based on Hough transform and morphological operations, and annotation extraction based on optical character recognition.

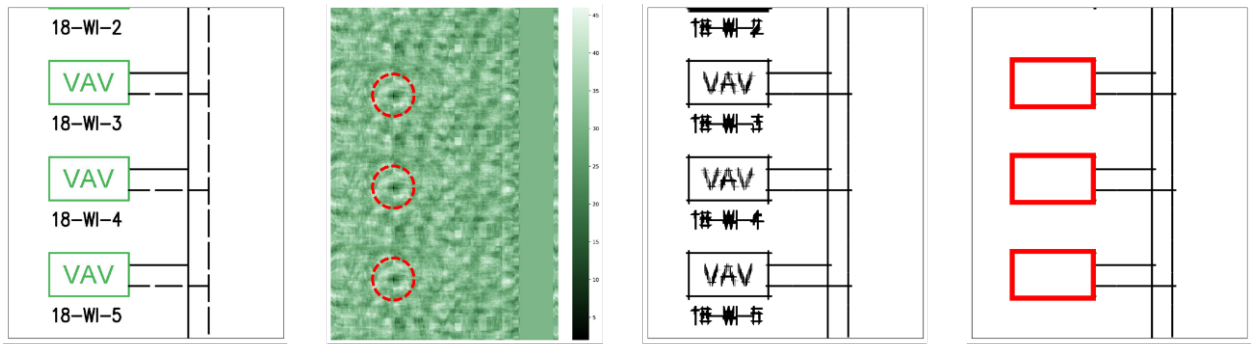


Figure 5. Image identification tasks using partial schematic drawing as an example (a) raw image; (b) heatmap of Hamming distance between the template image and sliding window; (c) preliminary line detection results after Hough transformation and morphological operations; (d) combination of detected objects and filtered straight lines.

Object detection. Object detection aims to detect the objects (nodes) and their location information in the 2D schematic drawings. Due to the lack of sufficient labeled image datasets for training a deep-learning based image identifier, the template matching method is adopted here to fulfill this task, which has been introduced in Section 2.1. Template matching slides the template over the input image and identifies the target objects based on the similarity between the template and the covered window on the image. The object detectors (i.e., the template image) can be obtained from the legend of the schematic drawings. After that, multiple detectors are generated by rescaling and rotating the original template to address the scale mismatch problem. Figure 5 illustrates the image identification procedure using part of the AHU schematic drawing as an example. Figure 5(a) depicts the raw image with three VAV terminals as well as the air ducts connecting them. The template of VAV terminal, obtained from the legend, scans through the

whole AHU schematic drawing. The image similarity identification is conducted by using the perceptual hash algorithm [70], which establishes the "perceptual equality" (i.e., a digital representation) of the image. The Hamming distance between the perceptual hash values of each sliding window and the template is computed. The heatmap of the Hamming distance is presented in Figure 5(b), with each point representing the window centered on it. Darker colors represent a higher degree of similarity between the sliding window and the template. By configuring a suitable similarity threshold, target object can be identified in the three red circles in Figure 5(b). However, when the scanning interval in template matching is small (e.g., one pixel), several neighborhood windows around an object could get similar scores and are all considered as candidate regions (i.e., the similarity score is higher than a pre-defined threshold). This study adopts non-maximum suppression [72] to address this problem by selecting the window with the highest similarity score in the neighborhood and suppressing those windows with low scores.

Line detection. Line detection aims to detect the pipelines based on morphological methods and Hough transform [73]. Morphological transformations are some simple operations based on the image shape, e.g., Erosion and Dilation. Morphological transformation needs two inputs, i.e., the original image and a structuring kernel which decides the nature of operation. Erosion operation is adopted in this research to extract straight lines (i.e., the air pipelines). In Erosion operation, the kernel slides through the images (as in 2D convolution). A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel are 1, other it is eroded (made to zero). By setting kernels as a horizontal line or vertical line, the straight lines in original images can be detected. Another challenge is that dashed lines are also used in building 2D schematic drawings to represent the pipelines. Without detecting and filling in the dashed lines, the algorithm cannot automatically and correctly identify the object connection in the physical system. Hough transform,

as the classical approach for lane line detection in automatic driving, is adopted to solve this problem via OpenCV module [67]. Figure 5(c) depicts the preliminary line detection results after Hough transform and morphological operations. Figure 5(d) shows the outcome of deleting short lines and combining the remaining long lines (i.e., air ducts) with the detected objects (i.e., VAV terminals).

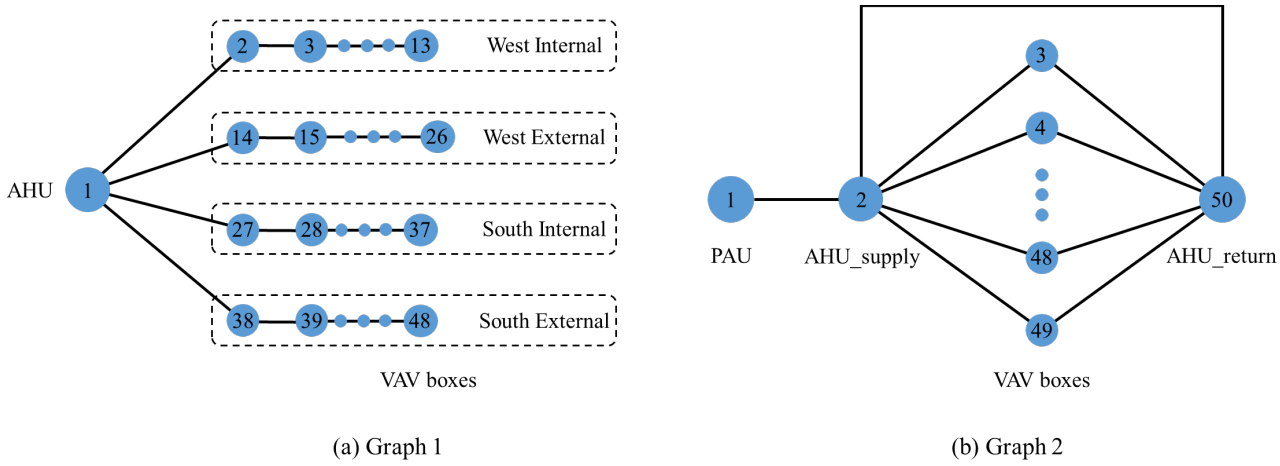


Figure 6. (a) Graph 1: automatically generated graph structure using image identification based on 2D schematic drawing of target AHU-VAV system; (b) Graph 2: modified graph structure

Figure 6(a) shows the automatically generated graph structure from Figure 3(a) via image identification techniques, which represents the AHU-VAV system and consists of 48 nodes. One AHU node and forty-seven VAV boxes are identified from the schematic drawing via image identification methods. The identified physical entities are only connected when they are adjacent. According to the physical knowledge about AHU-VAV system, another graph is designed based on the automatically generated version, as shown in Figure 6. Two nodes (PAU and AHU_return) are added and represent pre-cool outdoor air duct from Pre-cooled air unit (PAU) and return air duct from Air Handling Unit (AHU). The in-sequence connection between nodes are also modified. Both two graph structures are tested in the case study. After designing the graph structure, construction of graph corresponding feature matrix (i.e., model input and output) requires

matching the nodes with sensor measurements in BAS dataset. This process is automatically fulfilled by annotation extraction and fuzzy string matching. Annotation extraction based on optical character recognition is conducted to automatically extract the attribute annotations close to the objects (e.g., "18-W1-3" in Figure 5) and enrich the semantic information of objects [34]. Optical character recognition [75] is a technology that detects and converts characters in the image into computer text through operations such as denoising, feature extraction and classification. And fuzzy string matching is employed to match the extracted annotations of nodes with sensor descriptions in BAS dataset.

3.3 Development of graph neural network

In dynamic modeling of the AHU-VAV air conditioning systems, the model input is organized as a structural graph with fifty nodes (i.e., N= 50), each containing two features (i.e., $M = 2$) along four timesteps (i.e., $T = 4$). As shown in Figure 7, the model is designed via the straightforward "stacking" way to have a graph layer capture structural dependency and a recurrent layer capture temporal dependency separately. This order follows the majority of existing literature [17,20], i.e., a graph layer in front and a recurrent layer following it. A fully-connected layer is used as the regression head following the recurrent layer.

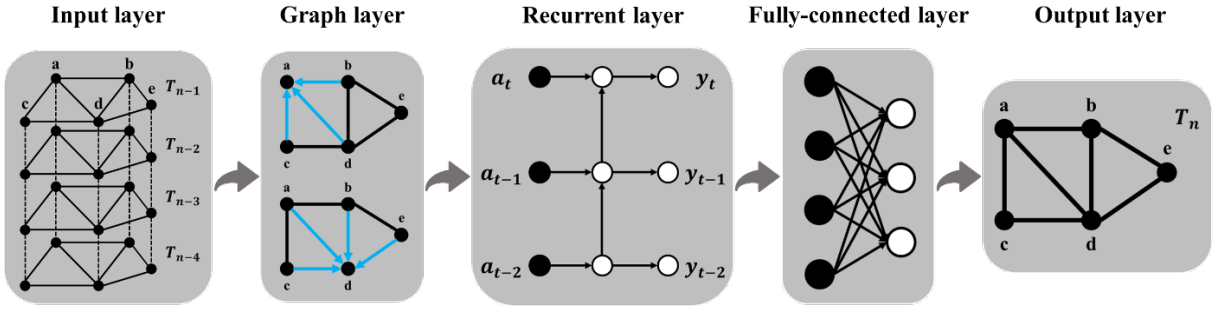


Figure 7. The architecture of the graph neural network-based model

In this study, GAT and GCN are tested in the graph layer for their wide application and ground-breaking performances on many deep learning tasks [21]. For recurrent layer, LSTM is selected for its ability to capture and learn long-term dependencies in the data. For fair performance comparison, two additional similar networks using the "stacking" architecture will be tested which replace the graph layer by a fully-connected layer and a conventional convolutional layer, respectively, while the recurrent layer is kept. The reasons for exclusion of other time series prediction models is mainly twofold. First off, it is challenging to determine whether the performance discrepancies are due to the models' ability to handle spatial dependency or temporal dependency. Second, some models cannot flexibly handle such multivariate time series input in a multi-input multi-output manner. The order of input nodes is randomly shuffled for CNN-RNN model in this case study to reveal the impact of this transformation. In total, there are four neural architectures for comparison in case study, including GCN-RNN, GAT-RNN, CNN-RNN and Dense-RNN. GCN-RNN and GAT-RNN use the graph attributed feature matrix $\boldsymbol{X}$ and its corresponding adjacency matrix as model input. The hyperparameters are identified based on a grid search in preliminary test, and the detailed grid-search settings are provided in the Appendix. Sigmoid function is selected as activation function in the graph layer. The output number of GCN layer is set as 10. For CNN-RNN model, 10 filters with the stride of 1 and the window length of 3 is selected. The output dimension of graph layer is set as 10. Five attention heads are used in GAT layer. And the unit number of RNN layer is set as 15. Moreover, the performance of different model architectures with varying training data availability (i.e., from 40% to 80%) is investigated. RMSE (Root mean square error) is adopted to assess the model prediction accuracy.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}} \tag{5}$$

Where y_i is the actual (temperature, in this study) value at the i^{th} VAV box, $\widehat{y_i}$ is the model predicted value.

3.4 Model explanation using SHAP method

This study adopts Shapley Additive exPlanation (SHAP) method [76] to explain the GNN models, due to its wide application in existing research [60,70] and its capability to explain all kinds of machine learning models. Proposed by Lundberg and Lee [76], SHAP is a powerful model explanation method based on ideas from game theory [77] and local explanations [78]. SHAP assigns each feature, i.e., temperature and flow rate measurement at each node, an important value for a particular prediction. As an additive feature attribution method, SHAP develops an explanation model g as a sum of values attributed to each feature. The simplified formula can be expressed as below:

$$g(x') = \boldsymbol{\phi}_0 + \sum_{i=1}^{T\times N\times M} \boldsymbol{\phi}_i \tag{6}$$

Where $T \times N \times M$ is the total number of input features. $\boldsymbol{\phi}_i$ is the feature importance value. Large ϕ_i infers higher impact on the model output. $\boldsymbol{\phi}_0$ represents the model output with all simplified inputs toggled off (i.e., missing). While SHAP is a local explanation method, a global model explanation can be obtained by averaging the absolute Shapley values for every instance [76,77]:

$$\boldsymbol{I}_m = \frac{1}{Q}\sum_{q=1}^{Q}\left\|\boldsymbol{\phi}_q(x_m)\right\| \tag{7}$$

where $\boldsymbol{I}_m$ is the average Shapley value of the m^{th} feature. Q is the total number of instances in the dataset. $\boldsymbol{\phi}_q(x_m)$ refers to the Shapley value of the m^{th} feature in the q^{th} instance.

Furthermore, in this case study, $\boldsymbol{\phi}_{t,i,m}^{q}$ denotes the feature importance of $x_{i,m}^{t}$ on q^{th} instance in the output. Considering the large number of model input features (i.e., $50 \times 4 \times 2$), it makes little sense to directly analyze and visualize the SHAP value of each input feature on the model output. As a compromise, the dimension-level global model explanations are calculated via aggregating the SHAP values on each input dimension (i.e., time t, node i, feature m) for visualization and further analysis:

$$\boldsymbol{\phi}_{t}^{q} = \sum_{i=1}^{N=50} \sum_{m=1}^{M=2} \left\| \boldsymbol{\phi}_{t,i,m}^{q} \right\| \tag{8}$$

$$\boldsymbol{\phi}_{i}^{q} = \sum_{t=1}^{T=4} \sum_{m=1}^{M=2} \left\| \boldsymbol{\phi}_{t,i,m}^{q} \right\| \tag{9}$$

$$\boldsymbol{\phi}_{m}^{q} = \sum_{t=1}^{T=4} \sum_{i=1}^{N=50} \left\| \boldsymbol{\phi}_{t,i,m}^{q} \right\| \tag{10}$$

4. Results and discussions

4.1 Model prediction performance

In total, four model architectures (i.e., CNN-RNN, Dense-RNN, GCN-RNN and GAT-RNN) and two graph structures are tested and compared in this study. Graph 1 refers to the automatically generated graph with 48 nodes (i.e., Figure 6(a)), while Graph 2 refers to the modified graph with 50 nodes (i.e., Figure 6(b)). All the methods and machine learning algorithms are programmed using Python, and Keras and Spektral packages [80]. The training data and testing data are randomly divided from the whole dataset. As described in Section 3.3, the performance of different model architectures with varying training data availability (i.e., from 40% to 80%) is investigated. During the model training process, 20% of the training data is randomly selected for validation.

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. For each combination of model architecture and training data availability, 20 trials are tested. The mean RMSE and their standard deviations on testing dataset are illustrated in Table 1 and Figure 8. The results of a baseline prediction strategy are also provided, i.e., $Y_t = Y_{t-1}$. The results indicate that, GAT-RNN and GCN-RNN outperform CNN-RNN and Dense-RNN in general. When Graph 1 is used, GAT-RNN and GCN-RNN both perform consistently (i.e., around 0.20℃ RMSE), even under low training data availability. However, for the in-sequence connection in Graph 1, the information of the AHU node must pass through numerous graph layers (of message propagation) before reaching the subsequent VAV boxes, which could be less efficient. When using Graph 2, GAT-RNN and GCN-RNN demonstrate obvious prediction performance improvement over CNN-RNN and Dense-RNN (with 70% and 80% training data availability). When the training data accounts for 80% of the whole dataset, GAT-RNN and GCN-RNN outperform CNN-RNN and Dense-RNN in terms of both bias and variance. GAT-RNN achieves RMSE of 0.12±0.00℃, while GCN-RNN achieves RMSE of 0.13±0.01℃. GAT-RNN and GCN-RNN achieve approximately 40% of accuracy improvement over Dense-RNN. The results indicate that the change in training data volume has a limited impact on GCN-RNN and GAT-RNN when using Graph 1. Both models achieve good performance under small amount of data. As for the remaining four cases, the model prediction accuracy improves gradually with an increase in training data. Among them, both GCN-RNN and GAT-RNN (using Graph 2) show faster enhancement in model accuracy and lower requirement for the amount of training data, compared to CNN-RNN and Dense-RNN. Moreover, the performance differences between ten independent trainings are also very small, e.g., 0.01 for GCN-RNN (Graph 2).

It can be observed that Dense-RNN perform the worst out of the four models. The most likely reasons for this result are that connecting unrelated nodes might disturb the model's inference mechanism, and increased model parameters make the training process more complicated. CNN-RNN demonstrates its superiority in several existing research [22,23], however, it fails in this case study. As mentioned in Section 3.3, the order of input nodes is randomized in the CNN-LSTM model which is incompetent for modeling physical systems with non-Euclidean structures. This random order makes it difficult for the CNN filters to extract useful features/information (from the neighborhood of each node). This also demonstrates that utilizing conventional data-driven modeling techniques without arranging data according to the physical system structure would result in the loss of a substantial amount of crucial structural information, and consequently decreases the model accuracy.

Table 1. Model prediction accuracy (RMSE: ℃) under different training data ratio

Training data ratio	CNN-RNN	Dense-RNN	GCN-RNN (Graph 1)	GAT-RNN (Graph 1)	GCN-RNN (Graph 2)	GAT-RNN (Graph 2)	Baseline
40%	0.45 ± 0.42	0.40 ± 0.17	0.20 ± 0.06	0.28 ± 0.22	0.39 ± 0.34	0.37 ± 0.37	0.21
50%	0.27 ± 0.23	0.31 ± 0.15	0.19 ± 0.08	0.20 ± 0.05	0.24 ± 0.18	0.18 ± 0.09	
60%	0.21 ± 0.17	0.29 ± 0.11	0.21 ± 0.08	0.18 ± 0.05	0.17 ± 0.18	0.14 ± 0.03	
70%	0.17 ± 0.08	0.24 ± 0.11	0.19 ± 0.08	0.18 ± 0.04	0.13 ± 0.04	0.12 ± 0.01	
80%	0.20 ± 0.14	0.20 ± 0.08	0.20 ± 0.05	0.22 ± 0.06	0.13 ± 0.01	0.12 ± 0.00	

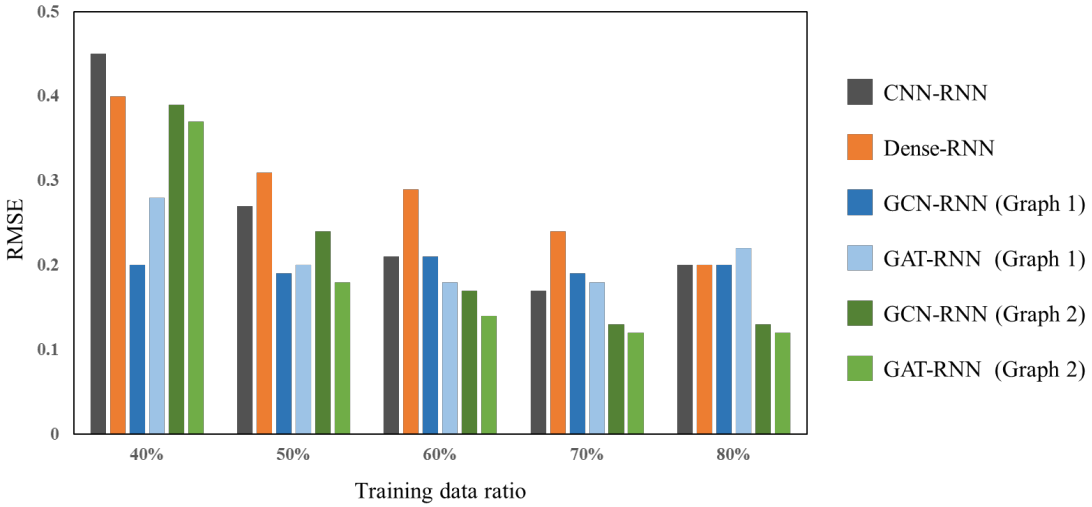


Figure 8. Model prediction performance (RMSE) under different training data ratio

4.2 Model explanation

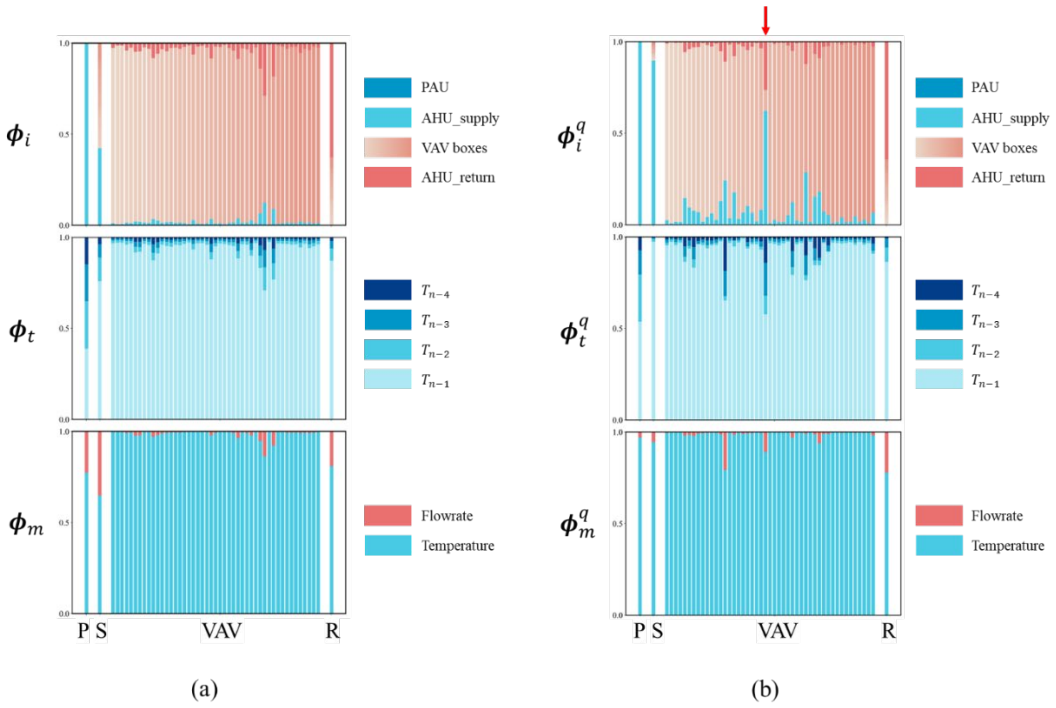


Figure 9. SHAP values of GCN-RNN model (a) global explanation on the whole dataset; (b) local explanation for model output at 08:25 am on Oct 14th, 2016

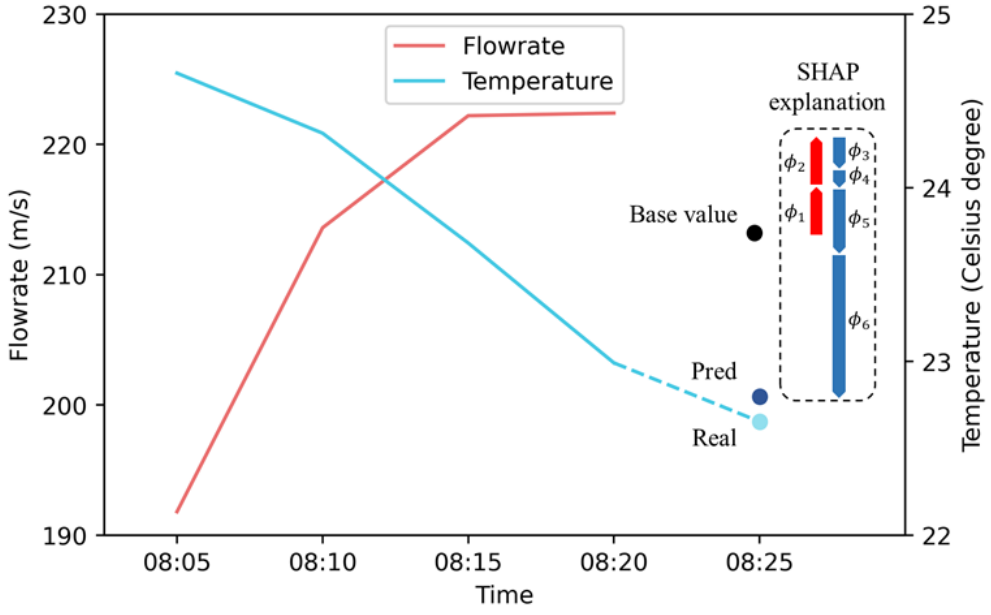


Figure 10. Local SHAP explanation for temperature prediction of VAV box "18-WE-11" at 08:25 am on Oct 14th, 2016

This section analyzes the SHAP explanations of a GCN-RNN model trained with 80% data of the whole dataset. As explained in Section 3.4, two aggregation strategies are designed for easier visualization of SHAP values, including aggregation along different input dimension, and local explanation to global explanation aggregation (i.e., averaging the SHAP values for every instance in the testing dataset). Figure 9(a) presents the heatmap of global SHAP explanations of the GCN-RNN model outputs. The upper, middle, and lower sub-figures show the global SHAP values $\boldsymbol{\phi}_i^q$, $\boldsymbol{\phi}_t^q$ and $\boldsymbol{\phi}_m^q$ for each model output (along the x axis), calculated according to Equations (8-11). As described in Section 3.4, $\boldsymbol{\phi}_i^q$, $\boldsymbol{\phi}_t^q$ and $\boldsymbol{\phi}_m^q$represent the aggregated feature importance at the node, time and feature level, respectively. In Figure 9, P, S, VAV, R represent the PAU, AHU_supply, VAV boxes, and AHU_return, respectively. Taking one VAV column in the upper sub-figure of Figure 9(a) as example, the SHAP values $\boldsymbol{\phi}_i^q$ are aggregated at the node level, which represent the

relative feature importance of each node in the model input (i.e., pre-cooled air, supply air, VAV boxes and return air) when the model generates the prediction on this VAV box. As indicated in Figure 9(a), the SHAP values on model output of each VAV box are dominated by its temperature at the previous timestep. This is due to the fact that the status of each node does not fluctuate much most of the time, that is, the temperature is very close to the previous timestep.

Figure 9(b) shows the SHAP local explanations of the GCN-RNN model output at 08:25 am on October 14th, 2016. An obvious difference between the global explanation and the local explanation at a specific time instant can be observed for several nodes (i.e., VAV boxes). Take the VAV box "18-WE-11" pointed by the red arrow in Figure 9(b) as an example for in-depth analysis. When the GCN-RNN model generates its temperature prediction at this moment (i.e., 08:25), the input features of the supply air and return air at earlier times play a remarkable influence. In this case study, the model generates the output at 08:25 based on the system status from 08:05 to 08:20. Figure 10 shows the temperature and air flowrate of this VAV box during this period. The right side of Figure 10 illustrates how different input features contribute to model decision-making process (i.e., from base value to model output), while the calculated SHAP values are shown in Figure 9(b). It can be observed from Figure 10 that this VAV box is rapidly cooling down during this period. This corresponds to the SHAP local explanation results: continuous and gradually increasing cold air supply causes a significant temperature decrease in the space served by this VAV box. This observation inspires greater confidence in the model output among users. In addition, model explanations can also assist model users in identifying where the model go wrong when the model output deviates from the real measurement.

4.3 Discussions

In this case study, the template matching-based object detection method is adopted to aid the graph structure design based on building 2D schematic drawings, considering that manually labelling images for training deep learning object detectors is labor-consuming. To achieve high detection precision, the template images should be rotated and scaled, and a small sliding interval should be specified. In this case, a heavy computation load is inevitable. It is possible to save some time by decreasing the resolution of both the template image and the input image. Moreover, there exist several special challenges associated with image identification-based graph structure design when handling building 2D schematic drawings. For instance, pipelines of irregular shape are hard to be detected; different pipelines may be intertwined and overlapped; one physical system may be depicted in multiple CAD schematic drawings. Overall, relatively few studies have been conducted on this topic. The case study represents pioneering efforts on this topic.

Graph symbolization of building HVAC systems or other subsystems is still an open question with no fixed answer. Although some exploratory attempts are performed in this research, it is difficult to determine whether the graph designed is the most "correct" or "appropriate". The target systems can also be represented as different types of graphs according to the tasks, e.g., spatial-temporal graph, dynamic graph, heterogeneous graph, etc. Spatial-temporal graphs have static topology but ever-changing node or edge features, while the nodes and edges of dynamic graphs appear and/or disappear over time [82]. A heterogeneous graph is another special kind of information graph, which contains either multiple types of objects as nodes or multiple types of links as edges. In real applications, dynamic graphs and heterogeneous graphs are also commonly used. For instance, the system topology of a hydraulic system in buildings can be represented as a dynamic graph as the water valves' on-off status in some pipelines vary under different operation modes which causes

some links to disappear. How to design proper graph structure according to system configuration and handle dynamic or heterogeneous graph structures in building energy systems appear to be challenging yet promising future research directions.

This paper utilizes the SHAP method to explain outputs of developed graph neural networks. The SHAP values, which quantify the impact of each model input on model outputs, are similar in form to causal inference. Noteworthily, traditional machine learning algorithms can only learn correlation-based patterns and relationships from data. That is, if interpreted from the perspective of causal inference, the obtained SHAP values are likely to be spurious, misleading or contrary to the physics principles [83]. Therefore, it is more appropriate to interpret these SHAP values from the perspective of correlation rather than causal inference. And this is an inevitable issue of using post-hoc approaches to explain data-driven models rather than developing white-box models or inherently interpretable models [83]. Another limitation of this study is that the global explanation is the aggregated absolute SHAP values, which neglect the information in their signs (i.e., larger or smaller than zero). This simplification is made to facilitate the results visualization under such a large number of input features (i.e., $50 \times 4 \times 2$). Additionally, not all model explanation methods for traditional machine learning models apply to GNNs [84,85]. For reference, Yuan et al. [85] provided a systematic and comprehensive review of existing explanation techniques for deep graph models. Therefore, the applicability and working principles must be prioritized when selecting suitable explanation techniques for GNNs.

5. Conclusion

With the increasing volume of multi-source data available in the building lifecycle and powerful machine learning, building informatics plays a more important role in building energy management, such as dynamic modeling of building energy systems which is a classical task in

building control and optimization. The dynamics of air conditioning systems and their interactions with the built environment exist within both time and space domains. Traditional data-driven approaches typically focus only on the temporal correlations among building operation data, while ignoring the spatial correlations. This study proposes a promising modeling method that captures both temporal and structural relationships in building energy systems through graph neural network-based models, exemplifying the value of incorporating physical knowledge into machine learning models for dynamic modeling of central air conditioning systems. To capture the structural relationships in the physical configuration of building energy systems, this paper attempts to symbolize the physical entities and their relationships as nodes and edges in a graph based on graph theory, and then uses the graphs as the input of machine learning models. A novel design information-assisted graph neural network-based methodology for dynamic modeling of central air conditioning system is proposed, which consists of graph structure design, development of graph neural network, and model evaluation and explanation. The proposed methodology is tested on historical operation data of the central air conditioning system in a high-rise commercial building in Hong Kong. The dynamic modeling of this air conditioning system is regarded as a node-level time series prediction task. Image identification methods are employed to extract the topology of the air conditioning system from 2D schematic drawings, which are used as the structure of the graph. GCN-RNN and GAT-RNN show improved prediction performance compared with conventional deep learning models.

The graph-based perspective introduced in this research provides a novel way to symbolize, analyze and model HVAC systems based on their physical configuration, opening up exciting possibilities for future research in this area. It enables efficient integration and utilization of building operation data and design information, which can be extracted from a variety of sources

including schematics drawings as well as BIM models. The specifically designed procedures and model interpretability benefit to mass implementation of the methodology in various buildings in an automated and digitized manner. The model is valuable for optimal control and fault detection diagnosis of the central air conditioning system. In conclusion, this study provides a valuable methodology for incorporating engineering knowledge of building HVAC systems into machine learning modeling, and demonstrates an ideal synergy between machine learning and engineering domain expertise, which opens up exciting possibilities for future research in building informatics.

Acknowledgement

The authors gratefully acknowledge the support of this research by Innovation and Technology Fund (ITP/002/22LP) of the Hong Kong SAR.

Appendix

Table 2. The grid-search settings for model hyperparameters

Parameters	Grid-search values
The activation function in hidden layers	ReLU, Sigmoid, Tanh
The number of output channels in GCN layer	5, 10, 15, 20
The Attention head number in GAT layer	5, 10, 15, 20
The filter number in each 1D convolutional layer	10, 20, 30, 40
Kernel size of 1D convolutions	2, 3, 4
Stride size of 1D convolutions	1, 2
The number of recurrent units in each recurrent layer	5, 10, 15, 20

References

1. IEA. (2019). 2019 Global status report for buildings and construction. United Nations Environment Programme
2. EMSD, Energy end-use data, 2021. https://www.emsd.gov.hk/filemanager/en/content_762/HKEEUD2021.pdf
3. Hartmann, T., & Trappey, A. (2020). Advanced Engineering Informatics-Philosophical and methodological foundations with examples from civil and construction engineering. Developments in the built environment, 4, 100020.
4. 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.
5. Luo, X. J., & Oyedele, L. O. (2021). Forecasting building energy consumption: Adaptive long-short term memory neural networks driven by genetic algorithm. Advanced Engineering Informatics, 50, 101357.
6. Li, X., & Wen, J. (2014). Review of building energy modeling for control and operation. Renewable and Sustainable Energy Reviews, 37, 517-537.
7. Harish, V. S. K. V., & Kumar, A. (2016). A review on modeling and simulation of building energy systems. Renewable and sustainable energy reviews, 56, 1272-1292.
8. Luo, X. J., Oyedele, L. O., Ajayi, A. O., Monyei, C. G., Akinade, O. O., & Akanbi, L. A. (2019). Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands. Advanced Engineering Informatics, 41, 100926.
9. Schreiber, T., Netsch, C., Eschweiler, S., Wang, T., Storek, T., Baranski, M., & Müller, D. (2021). Application of data-driven methods for energy system modelling demonstrated on an adaptive cooling supply system. Energy, 230, 120894.
10. Yang, S., Wan, M. P., Chen, W., Ng, B. F., & Dubey, S. (2020). Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization. Applied Energy, 271, 115147.
11. 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.

12. 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.
13. 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.
14. Sun, Z., & Wang, S. (2010). A CFD-based test method for control of indoor environment and space ventilation. Building and Environment, 45(6), 1441-1447.
15. Zhang, J., Xiao, F., Li, A., Ma, T., Xu, K., Zhang, H., ... & Wang, D. (2023). Graph neural network-based spatio-temporal indoor environment prediction and optimal control for central air-conditioning systems. Building and Environment, 242, 110600.
16. Sun, Y., Wang, S., & Xiao, F. (2013). Development and validation of a simplified online cooling load prediction strategy for a super high-rise building in Hong Kong. Energy Conversion and Management, 68, 20-27.
17. Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. B. (2021). A gentle introduction to graph neural networks. Distill, 6(9), e33.
18. Hu, Y., Cheng, X., Wang, S., Chen, J., Zhao, T., & Dai, E. (2022). Times series forecasting for urban building energy consumption based on graph convolutional network. Applied Energy, 307, 118231.
19. 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.
20. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1), 4-24.
21. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. AI Open, 1, 57-81.
22. Somu, N., MR, G. R., & Ramamritham, K. (2021). A deep learning framework for building energy consumption forecast. Renewable and Sustainable Energy Reviews, 137, 110591.
23. Niu, D., Yu, M., Sun, L., Gao, T., & Wang, K. (2022). Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism. Applied Energy, 313, 118801.

24. Li, T., Zhao, Y., Yan, K., Zhou, K., Zhang, C., & Zhang, X. (2021, October). Probabilistic graphical models in energy systems: A review. In Building Simulation (pp. 1-30). Tsinghua University Press.
25. Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., & Battaglia, P. (2020, November). Learning to simulate complex physics with graph networks. In International Conference on Machine Learning (pp. 8459-8468). PMLR.
26. Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. Expert Systems with Applications, 117921.
27. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., & Yin, D. (2019, May). Graph neural networks for social recommendation. In The world wide web conference (pp. 417-426).
28. Zhang, J., Dong, B., & Philip, S. Y. (2020, April). Fakedetector: Effective fake news detection with deep diffusive neural network. In 2020 IEEE 36th International Conference on Data Engineering (ICDE) (pp. 1826-1829). IEEE.
29. Abdelrahman, M., Chong, A., & Miller, C. (2020). Build2Vec: Building representation in vector space. arXiv preprint arXiv:2007.00740.
30. 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.
31. Ismail, A., Strug, B., & Ślusarczyk, G. (2018, June). Building knowledge extraction from BIM/IFC data for analysis in graph databases. In International Conference on Artificial Intelligence and Soft Computing (pp. 652-664). Springer, Cham.
32. Zhang, C., Zhao, Y., Li, T., & Zhang, X. (2020). A post mining method for extracting value from massive amounts of building operation data. Energy and Buildings, 223, 110096.
33. Qiao, L., Zhang, L., Chen, S., & Shen, D. (2018). Data-driven graph construction and graph learning: A review. Neurocomputing, 312, 336-351.
34. Zhao, Y., Deng, X., & Lai, H. (2021). Reconstructing BIM from 2D structural drawings for existing buildings. Automation in Construction, 128, 103750.
35. Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.

36. Cho, Y. K., Ham, Y., & Golpavar-Fard, M. (2015). 3D as-is building energy modeling and diagnostics: A review of the state-of-the-art. Advanced Engineering Informatics, 29(2), 184-195.
37. Brunelli, R. (2009). Template matching techniques in computer vision: theory and practice. John Wiley & Sons.
38. Slade, J., Jones, C. B., & Rosin, P. L. (2017). Automatic semantic and geometric enrichment of CityGML building models using HOG-based template matching. In Advances in 3D Geoinformation (pp. 357-372). Springer, Cham.
39. Gao, X., & Pishdad-Bozorgi, P. (2019). BIM-enabled facilities operation and maintenance: A review. Advanced engineering informatics, 39, 227-247.
40. Bazjanac, V., & Crawley, D. B. (1999, September). Industry foundation classes and interoperable commercial software in support of design of energy-efficient buildings. In Proceedings of Building Simulation'99 (Vol. 2, pp. 661-667). Boston: Addison-Wesley.
41. Werbrouck, J., Pauwels, P., Bonduel, M., Beetz, J., & Bekers, W. (2020). Scan-to-graph: Semantic enrichment of existing building geometry. Automation in Construction, 119, 103286.
42. Project Haystack. <http://project-haystack.org/>.
43. Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., ... & Whitehouse, K. (2018). Brick: Metadata schema for portable smart building applications. Applied energy, 226, 1273-1292.
44. 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.
45. Sperduti, A., & Starita, A. (1997). Supervised neural networks for the classification of structures. IEEE Transactions on Neural Networks, 8(3), 714-735.
46. 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.
47. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
48. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
49. Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

50. Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2), 129-150.
51. Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems, 29.
52. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.
53. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
54. Li, A., Xiao, F., Zhang, C., & Fan, C. (2021). Attention-based interpretable neural network for building cooling load prediction. Applied Energy, 299, 117238.
55. Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.
56. Seo, Y., Defferrard, M., Vandergheynst, P., & Bresson, X. (2018, December). Structured sequence modeling with graph convolutional recurrent networks. In International Conference on Neural Information Processing (pp. 362-373). Springer, Cham.
57. Skarding, J., Gabrys, B., & Musial, K. (2021). Foundations and Modeling of Dynamic Networks Using Dynamic Graph Neural Networks: A Survey. IEEE Access, 9, 79143-79168.
58. Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 28.
59. Jin, X., Xiao, F., Zhang, C., & Li, A. (2022). GEIN: An interpretable benchmarking framework towards all building types based on machine learning. Energy and Buildings, 260, 111909.
60. Arjunan, P., Poolla, K., & Miller, C. (2020). EnergyStar++: Towards more accurate and explanatory building energy benchmarking. Applied Energy, 276, 115413.
61. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., & Yang, G. Z. (2019). XAI—Explainable artificial intelligence. Science Robotics, 4(37), eaay7120.
62. Molnar, C. (2020). Interpretable machine learning. Lulu. com.
63. Wang, Z., Wang, Y., Zeng, R., Srinivasan, R. S., & Ahrentzen, S. (2018). Random Forest based hourly building energy prediction. Energy and Buildings, 171, 11-25.

64. Vartholomaios, A., Chatzidimitriou, A., & Ioannidis, K. (2020). METAMODELING THE INFLUENCE OF FORM AND SHADING ON THE HEATING AND COOLING LOADS OF A RESIDENTIAL ZONE IN THE MEDITERRANEAN.
65. Kotevska, O., Munk, J., Kurte, K., Du, Y., Amasyali, K., Smith, R. W., & Zandi, H. (2020, December). Methodology for Interpretable Reinforcement Learning Model for HVAC Energy Control. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 1555-1564). IEEE.
66. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE access, 6, 52138-52160.
67. Fan, C., Xiao, F., Yan, C., Liu, C., Li, Z., & Wang, J. (2019). A novel methodology to explain and evaluate data-driven building energy performance models based on interpretable machine learning. Applied Energy, 235, 1551-1560.
68. Aswani, A., Master, N., Taneja, J., Smith, V., Krioukov, A., Culler, D., & Tomlin, C. (2012, June). Identifying models of HVAC systems using semiparametric regression. In 2012 American Control Conference (ACC) (pp. 3675-3680). IEEE.
69. Lin, B., & Liu, H. (2015). A study on the energy rebound effect of China's residential building energy efficiency. Energy and Buildings, 86, 608-618.
70. Papadopoulos, S., & Kontokosta, C. E. (2019). Grading buildings on energy performance using city benchmarking data. Applied Energy, 233, 244-253.
71. Niu, X. M., & Jiao, Y. H. (2008). An overview of perceptual hashing. ACTA ELECTONICA SINICA, 36(7), 1405.
72. Neubeck, A., & Van Gool, L. (2006, August). Efficient non-maximum suppression. In 18th International Conference on Pattern Recognition (ICPR'06) (Vol. 3, pp. 850-855). IEEE.
73. Illingworth, J., & Kittler, J. (1988). A survey of the Hough transform. Computer vision, graphics, and image processing, 44(1), 87-116.
74. Bradski, G. (2000). The openCV library. Dr. Dobb's Journal: Software Tools for the Professional Programmer, 25(11), 120-123.
75. Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence, 39(11), 2298-2304.
76. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

77. Š trumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. Knowledge and information systems, 41(3), 647-665.
78. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).
79. Lubo-Robles, D., Devegowda, D., Jayaram, V., Bedle, H., Marfurt, K. J., & Pranter, M. J. (2020, October). Machine learning model interpretability using SHAP values: Application to a seismic facies classification task. In SEG International Exposition and Annual Meeting. OnePetro.
80. Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An explainable machine learning framework for intrusion detection systems. IEEE Access, 8, 73127-73141.
81. Grattarola, D., & Alippi, C. (2021). Graph neural networks in tensorflow and keras with spektral [application notes]. IEEE Computational Intelligence Magazine, 16(1), 99-106.
82. Jain, A., Zamir, A. R., Savarese, S., & Saxena, A. (2016). Structural-rnn: Deep learning on spatio-temporal graphs. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 5308-5317).
83. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206-215.
84. Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.
85. Yuan, H., Yu, H., Gui, S., & Ji, S. (2022). Explainability in graph neural networks: A taxonomic survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.
86. Chen, Z., Xiao, F., Guo, F., & Yan, J. (2023). Interpretable machine learning for building energy management: A state-of-the-art review. Advances in Applied Energy, 100123.