energies

MDPI

Article

Generating Occupancy Profiles for Building Simulations Using a Hybrid GNN and LSTM Framework

Yuan Xie [1,†] **and Spyridon Stravoravdis** [2,*,†]

[1] Institute for Environmental Design and Engineering, University College London, London WC1E 6BT, UK; yuan.xie.19@ucl.ac.uk
[2] School of Architecture, University of Liverpool, Liverpool L69 7ZN, UK
* Correspondence: s.stravoravdis@liverpool.ac.uk
† These authors contributed equally to this work.

Abstract: Building occupancy profiles are critical in thermal and energy simulations. However, determining an accurate occupancy profile is difficult due to its stochastic nature. In most simulations, the occupant activities are usually represented by fixed yearly schedules, which are often derived from guides and other similar sources and may not represent the simulated building accurately. Therefore, an inaccuracy in defining occupancy profiles can be a source of error in building simulations. Over the past few years machine learning has become very popular due to its ability to reveal hidden patterns and relationships between data and this makes it suitable for investigating patterns in occupancy data. This study proposes a novel hybrid model combining the Graph Neural Network and the Long Short-term Memory neural network (LSTM) to predict the occupancy of individual rooms on a typical office floor. The proposed Graph LSTM model can produce high-resolution occupancy profiles of an office that are in good agreement with the reference occupancy profiles of the same office. The reference occupancy profiles for this office were derived from an agent-based model using AnyLogic and were not used in the training of the neural network. The proposed Graph LSTM model outperformed other neural networks tested such as the Recurrent Neural Network (RNN), the Gated Recurrent Unit (GRU) and LSTM. When Graph LSTM is compared to the other neural networks tested, there is a range of improvement between 13.5 and 14.6% in the index of agreement, 38.3 and 46.8% in mean absolute error and 34.4 and 40.0% in root mean square error, when averaging the differences over the whole office.

Keywords: occupancy; energy simulation; neural networks; GNN; LSTM; RNN; GRU; RNN

check for updates

Citation: Xie, Y.; Stravoravdis, S. Generating Occupancy Profiles for Building Simulations Using a Hybrid GNN and LSTM Framework. *Energies* **2023**, *16*, 4638. https://doi.org/10.3390/en16124638

Academic Editor: Effrosyni Giama

Received: 5 April 2023
Revised: 5 June 2023
Accepted: 8 June 2023
Published: 11 June 2023

1. Introduction

The built environment is responsible for approximately 42% of all global carbon dioxide (CO_2) emissions [1], with 28% occurring during the building operation stage [2]. This includes energy used for heating, cooling, lighting, and other appliances. Recognized by the International Energy Agency [3], the energy consumption of a typical building is mainly determined by the following six parameters: the climate, the building envelope, the building services, the indoor environment quality, the building operation and maintenance, and the occupant behavior. The first four have been studied extensively, hence they can be predicted and represented accurately in a model. However, the latter two rely heavily on human behavior and activities, which are often unpredictable.

The occupancy of the space is the direct reflection of the occupant's activities. The number of people present in a space will contribute to the internal gain accordingly, therefore directly affecting the HVAC usage, and other electricity consumption, such as lighting and other appliances [4]. A reliable occupancy profile is critical in building thermal and energy simulations. Moreover, it can also be used when the building is in operation to aid the building services to operate more efficiently [5].

It is challenging to determine an accurate occupancy model due to its stochastic nature. In many simulations, the occupant activities are usually represented by fixed schedules, neglecting the high levels of uncertainty in human behavior. This leads to simulations where all occupants carry out the same actions, thus producing incorrect hourly demand peaks [6]. Therefore, inaccuracy in defining occupancy profile is a significant source of error in building thermal and energy simulations.

The most common occupancy profiles applied in energy simulation are the occupancy profile from ASHRAE standard 90.1 [7] and the EnergyPlus typical schedule [8]. These simple profiles fail to capture the complexity of actual circumstances, leading to more inaccuracy in simulation results [9]. Clevenger and Haymaker [10] found that the variation in energy use when altering occupancy profile and their environmental preference can be as high as 150%. With the weather data, building envelope, and building services defined and held constant, the variations are still significant [11]. This is also recognized by the International Energy Agency as occupant behavior and is one of the six factors that cause variation in energy usage in buildings [12]. Therefore, to produce a more reliable result through building energy modeling, a more complex occupancy profile with the consideration of occupant behavior will be more favorable [13].

The popularity of machine learning (ML) has raised dramatically over recent decades. This is due to the improvement in computational power and the availability of data. The application of ML in the building industry has also surged, from areas such as architecture and construction to building operations. Due to the ability of ML in interpreting hidden patterns and relationships between data, it was applied by a few studies on predicting occupancy, such as using K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Artificial Neural Network (ANN), and decision tree [14–17]. However, most studies focused on only the total occupancy of the whole office space, e.g., the total occupant count of the whole building or floor. However, different room types in an office may exhibit unique occupancy. Deep learning, as a part of a subset of ML, has proven to be highly effective for pattern recognition. The main tool for deep learning is neural networks. By including more features in the neural networks, they can be used for more targeted purposes. For example, Graph Neural Network (GNN) is where the data structure of the processed data is represented in graphs. The set of objects and their relationships are modeled as nodes and edges. The edges can be directed or undirected based on the relationships between the objects. Common applications of the GNN focus on node classification, link prediction, and clustering, in areas such as text generation, image recognition, science, knowledge graph, and graph generation. Another example is the Recurrent Neural Network (RNN), where a recurrent neuron is added to a standard ANN. As the recurrent neuron is unfolded through time, the hidden state of the last time step is fed to the current time step as additional input. It is particularly useful in cases where future data are dependent on the past, such as predicting stock prices. To further improve the RNN, a gated unit could be applied to the RNN, such as Long Short-Term Memory (LSTM) cell or Gated Recurrent Unit (GRU). An LSTM cell would decide which information is allowed on the cell state by learning which information is essential, and decide whether to keep or forget the information during training. The GRU cell is a simplified version of the LSTM cell. GRU does not have a cell state and instead uses only the hidden state to convey the information. There has been limited research conducted using neural networks for occupancy prediction.

This study proposes a hybrid model framework combining GNN and LSTM neural network based on the one adopted by Qi et al. [18] for $PM_{2.5}$ forecasting to predict the occupancy of individual rooms in an office environment. The proposed model was not applied in any previous study to predict room occupancy. The two deep-learning methods were chosen due to the outstanding performance of the GNN in representing spatial relationships by allowing the representation of the information as a graph structure to include information such as connectivity between rooms. And the LSTM in predicting temporal relationships by retaining essential information in its hidden state.

The rest of the paper is presented as follows: Section 2 provides the background of the previous work conducted on building occupancy studies using machine learning. Section 3 introduces the methodology of this work, which includes the proposed framework for the hybrid GNN and LSTM in the context of occupancy prediction in an office environment, and a simulated case study that was used to verify the framework. Section 4 presents and analyses the result produced. In Section 5, the discussion of the result is shown. Lastly, the conclusion sums up the work conducted and any potential future work to be performed.

2. Background

The energy used in a building is highly dependent on its occupancy; however, occupancy is hard to predict due to its stochastic nature. Several recent studies focus on studying occupancy using ML methods. A popular approach to predicting occupancy is using data from environmental sensors, such as the indoor air temperature, relative humidity, CO_2 concentration, VOC content, and air pressure. Applying an ML model to determine the relationship between the occupancy and the environmental data collected, then utilizing the established relationship to predict the occupancy in the future. The most common approach includes KNN, SVM, ANN [14–16], and several data-mining methods such as decision tree [19].

Anand et al. [20] produced a model where the real-time occupancy data were derived from Wi-Fi sensing, and processed through k-means clustering to identify occupancy patterns. The predicted result shows an error as low as 6.9% compared to the actual result. Wang et al. [14] used Wi-Fi data in combination with other environmental data and predicted occupancy using KNN, SVM, and ANN. The robustness of the occupancy predicted was improved by combining environmental sensor data with Wi-Fi data. Jiang et al. [21] used the CO_2 concentration measurement and applied it to Feature Scaled Extreme Learning Machine, the accuracy of the occupancy reported was up to 94%. Another work done by Szczurek et al. [16] used CO_2 concentration, indoor air temperature, and relative humidity as inputs, KNN was recognized as more efficient than linear discriminant functions. Yang and Becerik-Gerber [22] used data from light, sound, and motion detectors, along with CO_2 concentration, air temperature, relative humidity, and passive infrared sensor in three typical offices to the Autoregressive Moving Average model (ARMA), Neural Network, Markov Chain, and Logistic Regression algorithms for occupancy prediction. ARMA and Neural Network yielded more accurate results than the other model tested. Ryu and Moon [23] implemented Hidden Markov Model to the collected CO_2 data and produced occupancy prediction with an accuracy of between 85% and 93.2%. Peng et al. [15] investigated the use of motion signals as the input of a back propagation ANN. With the combination of ANN and look-up table, the result yielded the highest accuracy.

Data mining is also a popular ML research methodology applied in pattern recognition for occupancy. Liang et al. [17] applied clustering to determine the patterns of the collected occupancy data from an office. A decision tree was used to learn the schedule rules, which were then used for predicting the occupancy profile. The model output results with a deviation lower than 5%. D'Oca and Hong [19] created a three-step data-mining framework using supervised and unsupervised learning. First, a decision-tree model was applied to the occupancy collected from 16 offices. Then, a rule induction algorithm was adopted for learning a pruned set of rules on the results from the first step. Lastly, patterns of occupancy schedules were extracted using k-means cluster analysis. Four typical occupancy profiles of the office environment were recognized.

The above studies provide valuable insight into possible ML techniques in occupancy prediction. Although they all provide promising results in terms of accuracy, the studies tend to focus on a larger scale. For example, they only investigated the occupancy of a whole office, instead of the individual rooms, despite the apparent differences in occupancy for each room type. Determining the relationships between each room in an office is a possible approach for predicting more accurate occupancy. However, this may require a more elaborate ML model to accomplish.

In terms of the work conducted on hybrid GNN and LSTM model, Qi et al. [18] introduced a hybrid model that combines Gated Graph Neural Network and LSTM (GC-LSTM) for modeling and forecasting the spatiotemporal variation of $PM_{2.5}$ concentrations between different observation station. Moreover, the combination of GNN and LSTM is also popular in traffic studies. Lu et al. [24] were the first to produce a Graph LSTM for capturing the spatial and temporal relationships simultaneously. The model is used for traffic speed prediction, where the results proved that the proposed method could apprehend the spatial–temporal dependencies and produced better prediction than the baseline methods, including standard Gated Recurrent Units (GRU) and LSTM.

3. Methodology

The modeling part of the project is divided into two main stages. The first stage is creating the occupancy data through agent-based and discrete event modeling. The second stage is the construction and training of the neural networks. The first stage of this study is a crucial enabling step to test the proposed neural network. The occupancy profiles used by D'Oca and Hong [19] and collected by Luo et al. [25] were analyzed, but the profiles did not have high enough resolution that covers individual office rooms, and was not suitable for the study of this paper. Duarte et. al. [26] reported on the difficulty of obtaining data on occupant behavior. Several other studies as reported by Anand et al. [27] have shown the growing interest in this field and the various monitoring methods that are being developed to capture occupant behavior in more detail. A lot of these studies, though, focus on the interaction of occupants with building systems, rather than just movement in space. Often in the literature and online databases, it is possible to obtain limited data for occupancy where the data available is either not of high resolution, or does not contain a suitable and more complex office type. As a result, the more robust way to obtain high-accuracy, high-resolution data for occupancy to be used as a reference case, would be through monitoring. However, that was not possible during the period of this research as due to the COVID-19 pandemic it was not possible to access offices and obtain such data. To overcome this problem and create a reference case to test how the proposed model would perform, occupancy data and profiles were derived through occupancy simulations.

3.1. Modeling and Simulation of Occupancy Data

To generate these occupancy profiles AnyLogic software [28] was used. Figure 1 illustrates the main steps in the first stage of the research. The agent-based and discrete event modeling was completed using AnyLogic. First, the AnyLogic model was created, then the geometric data of the five simulated office layouts was input. To simulate the office environment, agent-based simulation was conducted by defining logistics such as schedules and probability events. Then the simulation was conducted on all five layouts individually in a span of 200 simulated workdays. Finally, the model output the raw occupant count to be further implemented in the next stage.

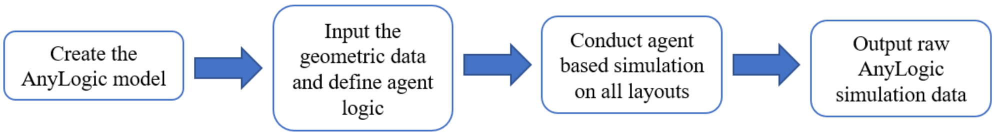

Create the AnyLogic model
Input the geometric data and define agent logic
Conduct agent based simulation on all layouts
Output raw AnyLogic simulation data

Figure 1. Flow chart of the AnyLogic simulation stage of the research.

Agent-based modeling was used to model the movement of people in a small office space for a day. To better reflect the real office environment, the most common office activities were modeled in AnyLogic, and the occupants were mapped as individual agents. The schedule of the agents is shown in Figure 2. The simulation is conducted between 6:00 and 23:00, with a total of 27 agents, including 25 employees and 2 managers. Every minute, the number of occupants in each room is recorded, resulting in 1021 data points per simulation per room. The model is built with the pedestrian library in AnyLogic, with the addition of JavaScript to specify the probability events.

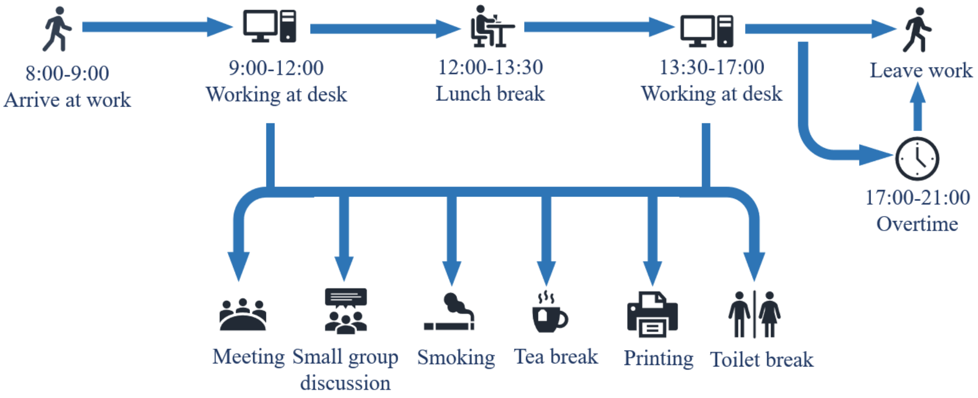


Figure 2. Flow chart of the schedule applied for the simulation of the office.

All agents arrive evenly between 8:00 and 9:00. Upon arrival, the role of the agents will be specified, and two agents will be set up as the manager and located in a separate office. Based on the population of smokers in the UK in 2018 [29], 15% of the agents will be set as smokers. This enables the possibility of triggering the smoking event. From 9:00 to 12:00, the agents will stay at their assigned tables. During this time, the specification of the possible events is listed in Table 1. Lunch break for all agents will happen between 12:00 and 13:30, and all events will be suspended during this period, agents will either stay at their Table (50% possibility), go to the staff lounge (30% possibility), or decide to leave the office (20% possibility). After a lunch break, all agents will go back to their desks, and the meeting event will resume. All possible events in the afternoon will be the same as in the morning. The regular working day is over at 17:00; however, the agents will have a 10% chance of staying for overtime. The length of overtime has a uniform distribution of between 30 min and 240 min. All agents will leave before 21:00.

Table 1. Specification of the possible events modeled.

Event	Event Probability	Numbers of Agent	Duration	Location
Printing	25%	1	5–10 min	Printing room
Tea break	25%	1	2–6 min	Staff lounge
Toilet break	25%	1	2–3 min	Toilet
Smoking	25%	1	5–15 min	Outside
Small discussion	every 30 min	2–3	5–10 min	Desk
Meeting	40% every 30 min	4–12	40–120 min	Meeting room
Overtime	10%	/	30–240 min	Own desk
Day off	10%	/	whole day	/

The events modeled can be categorized into two types, independent tasks, and group events. Printing, tea breaks, toilet breaks, and smoking are independent for each agent and only involve one agent at a time. Every half to one hour (triangularly distributed with 45 min as mean), the independent events listed above have the possibility of occurrence of 25%. Once this happens, a cool-down period of 2 h will be applied. Group events include small group discussions and meetings. Small group discussions will happen every 30 min involving 2 or 3 agents and will be located randomly at one of the agent's desks. The event time will be uniformly distributed between 5 and 10 min. For meetings, the number of people involved is uniformly distributed between 4 and 12, and the meeting length of 40 to 120 min. Every 30 min, there is a possibility of 40% to trigger a meeting. For cases with multiple meeting rooms, the meeting will be prioritized to be held in one meeting room first. Only when the first meeting room is occupied, the second one will be used.

The office model consists of six types of rooms: main office space, manager office, meeting room, printing room, staff lounge, and toilet. A total of five layouts were created in AnyLogic to generate occupancy data, the 5 layouts have a different combination of main office space and meeting room count. The layouts shown in Figure 3 were applied as the training set for the training of the neural networks in the next stage. Layout 5, shown in Figure 4, was the testing set used for verifying the neural networks later.

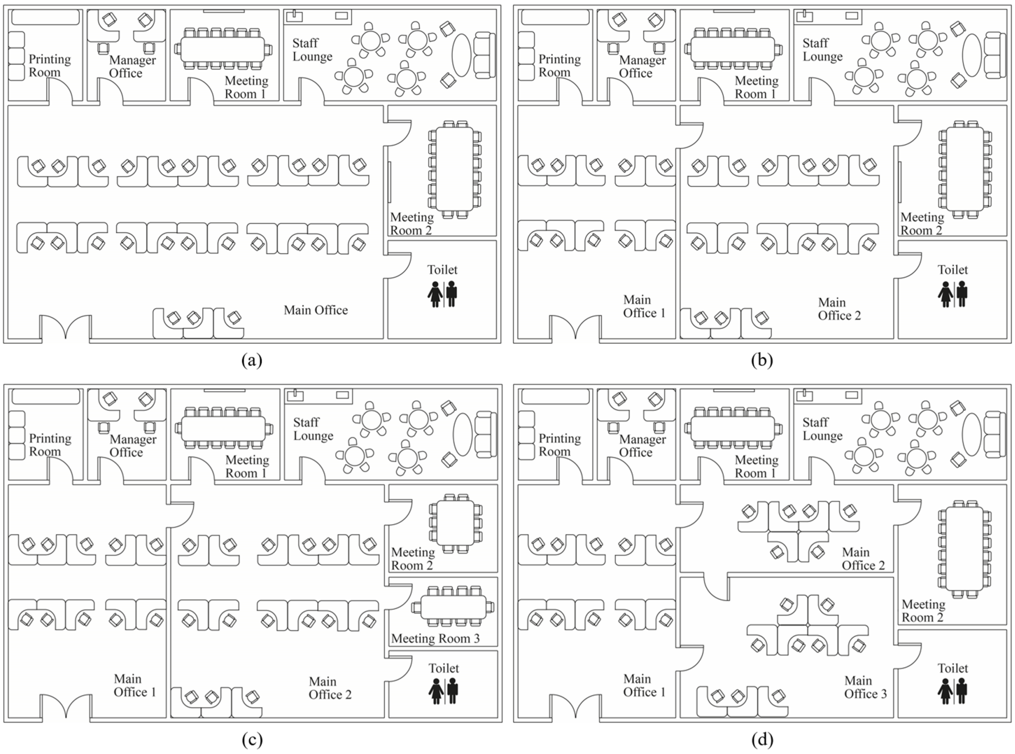


Figure 3. Office layouts used for the AnyLogic simulation and the training of the neural network: (**a**) Layout 1. (**b**) Layout 2. (**c**) Layout 3. (**d**) Layout 4.

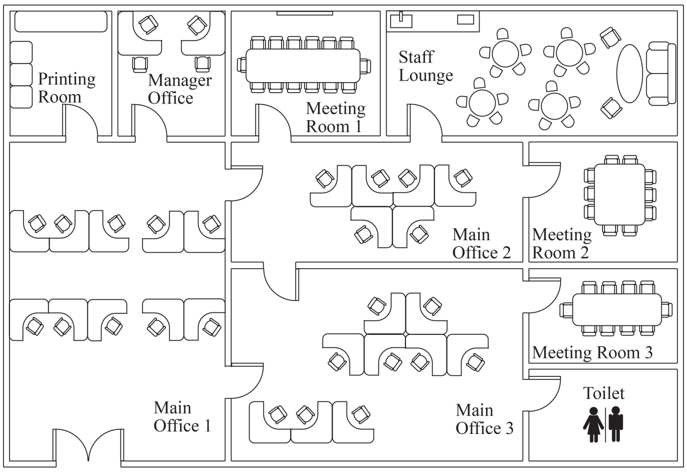


Figure 4. Office Layout 5, applied for the AnyLogic simulation and the testing for the neural network.

Due to the limitation of simulating and outputting in batch in AnyLogic, the total simulation conducted was 200 for each office layout, yielding a total of 1000 sets of data. Each set consists of the number of people in each room from 6:00 to 23:00 at a one-minute time step. As the events in the model are all based on the probability given, the output each time will be slightly different, but still maintain a similar trend throughout all cases.

3.2. *Neural Networks Construction and Testing*

After the occupancy profiles from AnyLogic simulations were obtained, they were used to research the proposed neural network framework. Figure 5 shows the flow chart of the steps performed in the neural network construction and testing stage of this research.

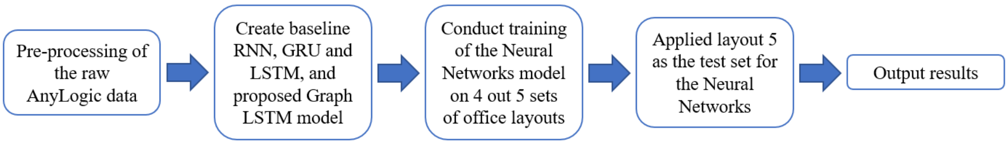

Pre-processing of the raw AnyLogic data
Create baseline RNN, GRU and LSTM, and proposed Graph LSTM model
Conduct training of the Neural Networks model on 4 out 5 sets of office layouts
Applied layout 5 as the test set for the Neural Networks
Output results

Figure 5. Flow chart of the neural network construction and testing stage of the research.

This study proposed a model combining GNN and LSTM to study and predict the occupancy of each room in office settings. Three baseline neural networks, RNN, GRU, and LSTM are also constructed for comparison. The neural networks are constructed using Python with PyTorch Geometric and TensorFlow. The methodology adopted by Qi et al. [18] on creating a hybrid GCN and LSTM model for forecasting $PM_{2.5}$ is a useful reference for this study. They applied a graph convolution operation to represent the spatial feature of each $PM_{2.5}$ monitoring station. Then, the temporal features of the data are processed by LSTM. The input used for the LSTM is the combination of the graph convolutional features and the original signals. This study employed a similar framework for the construction of the Graph LSTM to determine the spatial dependency of each office room and the occupancy time series.

To represent different layouts in the same matrix, all data are pre-processed using MATLAB before using as the input of the neural network. For each set, the occupancy data are combined with a room indicator, as shown in Figure 6. As the maximum number of rooms for this study is ten rooms, all occupancy is updated into matrix $X_{1021\times10}$, where each column represents the occupancy of each room. If the room is not present in the layout, the whole column will be zero. A matrix of dummy variables $I_{1021\times10}$ is also added to indicate if each room is presented in the layout. The whole column will be 0 if the room does not exist in the layout, and 1 if the room exists. The combination of the new occupancy matrix and the matrix of room indicator is then the input used to create the neural network.

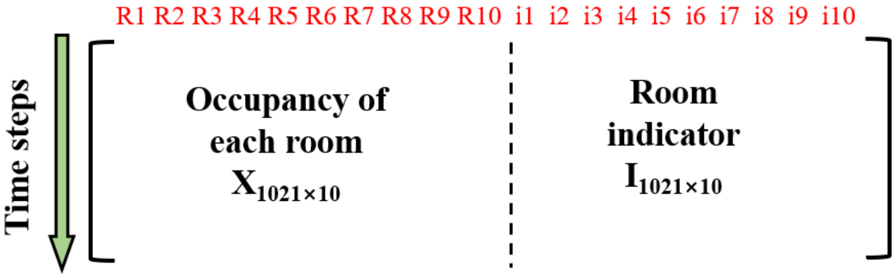


Figure 6. Pre-processed matrix for the baseline neural networks.

By stacking the matrix created above for all training sets together, a 3D array containing all training data is produced (Figure 7). This array is then input into the respective baseline neural networks for training. After training is completed, the 2D matrix from Figure 6 of the test set is fed to the trained neural network model, and the predicted occupant count for each room based on the test set is rounded to the nearest integer to better represent occupant count, and collected for later analysis.

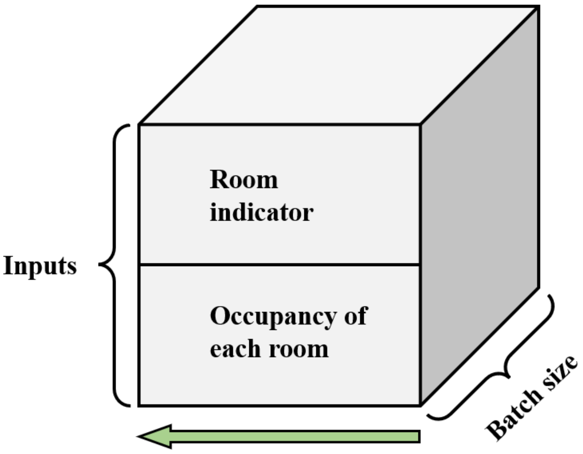
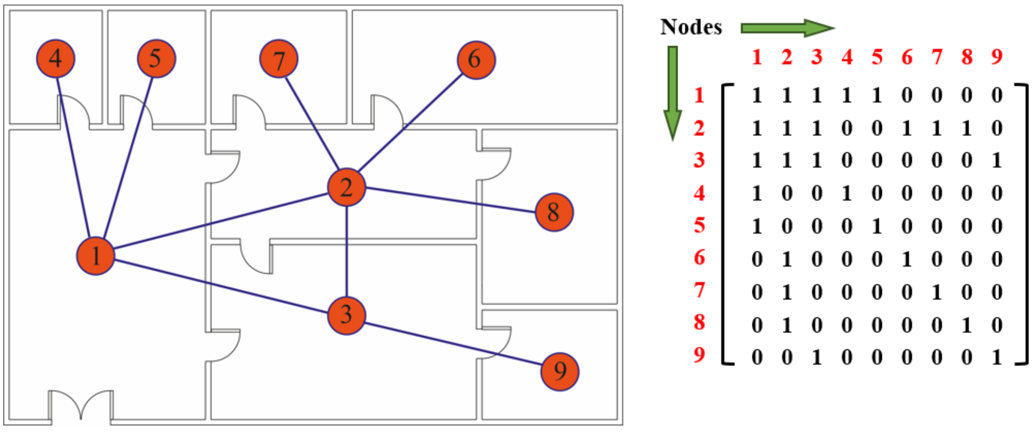


Figure 7. 3D array input for the training of the baseline neural network.

The Graph LSTM is constructed using PyTorch Geometric, an extension library for PyTorch built for deep learning on graphs and various other irregular structures [30]. To apply the GNN, the data are transformed into a graph structure, G = (V, E). Where the nodes V indicate each room in the office and the edges E are their connectivity. The graph structure is represented by their corresponding adjacency matrix A. Figure 8 shows the graph structure and the corresponding adjacency matrix for Layout 4. Please note that the graph structures illustrated below are only to demonstrate the connectivity between the nodes, and the number of nodes and edges, the length of the edge, and the location of the nodes are irrelevant in this case. Each row and column of the adjacency matrix represents the nodes in the structure, the corresponding value in the adjacency matrix will be 1 if a connection is present, and 0 otherwise. In this case, the rows and columns of the adjacency matrix represented each room of the office, '1' indicated the two rooms were connected and agents were able to move between the rooms, and '0' is where the rooms were not connected and agents were not able to move between the two rooms. As the edges are non-directed, all adjacency matrices applied are symmetrical. Moreover, the distance between the rooms was not considered in this problem to simplify the model and reduce training time.



$$
\begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
2 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
4 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
5 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
6 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
7 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
8 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
9 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

(a) (b)

Figure 8. The graph structure and adjacency matrix for Layout 4: (**a**) Graph structure. (**b**) Adjacency matrix A.

Figure 9 illustrated the proposed Graph LSTM model. The input layer consists of graph signal X_t, which includes the occupancy data of each room at each time step for all training sets, and adjacency matrix A. This input is then passed through a graph convolution layer to compute the spatial features H_t. The combination of the spatial features H_t calculated and the graph signal X_t, is then fed to the LSTM layer. Next, the product from the LSTM layer is directed through an output layer, where the predicted occupancy value will be computed and rounded to the nearest integer before outputting for further analysis.

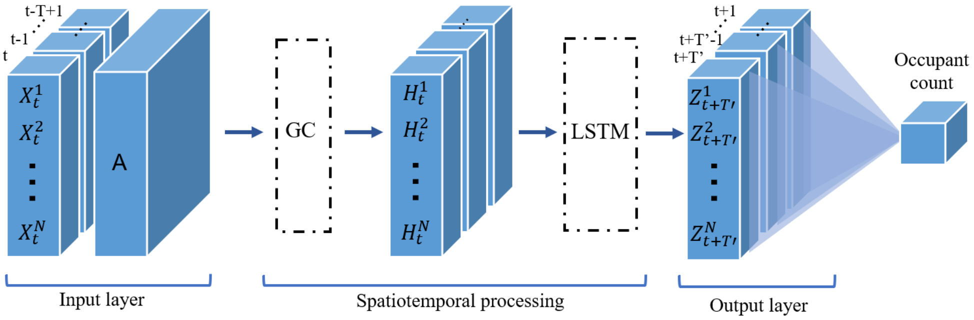


Figure 9. Proposed Graph LSTM architecture, modified from the framework created by Lu et al. [24].

The detailed settings for all experiments are kept the same and are listed in Table 2. For training and validation, the inputs are the data from layouts 1 to 4, and the test set is the data from Layout 5. This is to ensure the neural networks have learned the relationships between the different room and their occupancy pattern, by exposing them to multiple different layouts. In addition, the data chosen for the test set is the layout that was not used in training, hence, testing the ability of the neural network model in predicting occupancy of layouts with room combinations that the network has not seen before. Five hidden layers are applied for all neural networks, and each layer has 64 neurons. The learning rate is set as 1×10^{-3} , and the maximum epoch is 200. To prevent overfitting, a validation set is adopted to employ early stopping if there is no decrease in loss for ten consecutive epochs. Mean squared error is applied as the loss function for this study.

Table 2. Experimental settings for baseline neural networks.

Parameter	Value
Number of data points	10,005,800
Training set	64%
Validation set	16%
testing set	20%, Layout 5
Hidden layer	5
Neuron in each layer	64
Learning rate	1×10^{-3}
Early stopping patience	10
Maximum epoch	200
Loss function	Mean square error (MSE)

Three evaluation metrics are applied to the test set to assess the performance of the proposed model, Index of agreement (IA), mean absolute error (MAE), and root mean square error (RMSE). IA with a value closer to one implies a higher agreement between the predicted and the actual result. For MAE and RMSE, the lower the error, the better the model. IA, MAE, and RMSE are formulated as follows:

$$IA = 1 - \frac{\sum_{i=1}^{T}(o_i - p_i)^2}{\sum_{i=1}^{T}(|p_i - \bar{o}| + |o_i - \bar{o}|)^2} \tag{1}$$

$$MAE = \frac{1}{T}\sum_{i=1}^{T}|o_i - p_i| \tag{2}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(o_i - p_i)^2} \tag{3}$$

o_i: Ground truth occupancy at time step i.
p_i: Predicted occupancy at time step i.
$\bar{o}$: Average occupancy.

4. Results and Analysis

This section will start by presenting the office occupancy profiles produced from the AnyLogic simulation. Then, the predicted occupancy from the neural network is compared with the simulated data to examine the performance of the proposed Graph LSTM model.

4.1. AnyLogic Generated Occupancy Profiles

For all five office layouts, the simulation in AnyLogic produced a similar occupancy pattern for each room. Figure 10 shows a set of total occupant counts produced using Layout 2. The space is occupied between 8:00 and 21:00, with higher occupancy during regular working hours, a small drop during lunch break, and a sharp decrease when overtime started. This overall trend is as expected and is similar to the office occupancy provided by ASHRAE [7]. As seen in the figure below, compared to the ASHRAE profile, the occupancy data produced from the simulation fluctuate more during working hours, as the model accounted for the possibility of agents leaving the office for a short period.

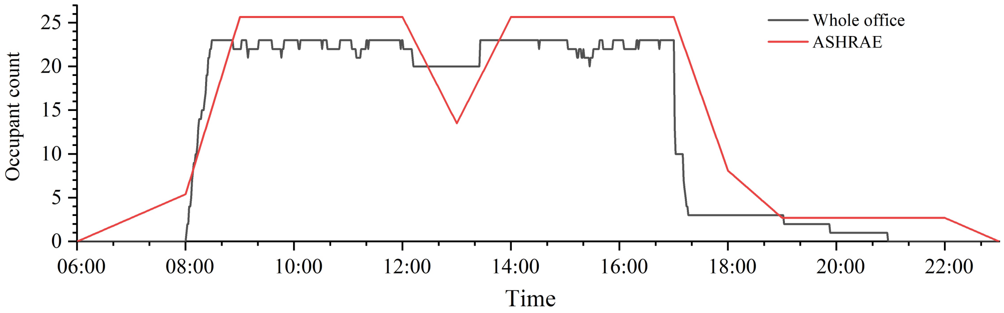


Figure 10. Whole office occupant count produced from AnyLogic compared with the occupancy profile provided by ASHRAE for Layout 2.

Figure 11 shows the occupancy profile of one simulated day of all individual rooms from layouts 1 and 5. From analyzing the overall pattern of the data presented, all rooms have a highly fluctuated profile due to events created in the model, where the agents had to travel to different rooms. The areas such as the main offices, where most rooms are connected, often show large peaks due to the occupants passing through to reach another location. Each simulation of the same layout generates slightly different occupancy due to the nature of the probability settings, but the overall trend remains similar.

The data produced using AnyLogic were in good agreement with the ASHRAE occupancy profile [7] at a whole office level. At an individual room level, the profiles created by AnyLogic appear realistic as the model captured the fluctuations in occupancy, which come from the activities and movement taking place within the office.

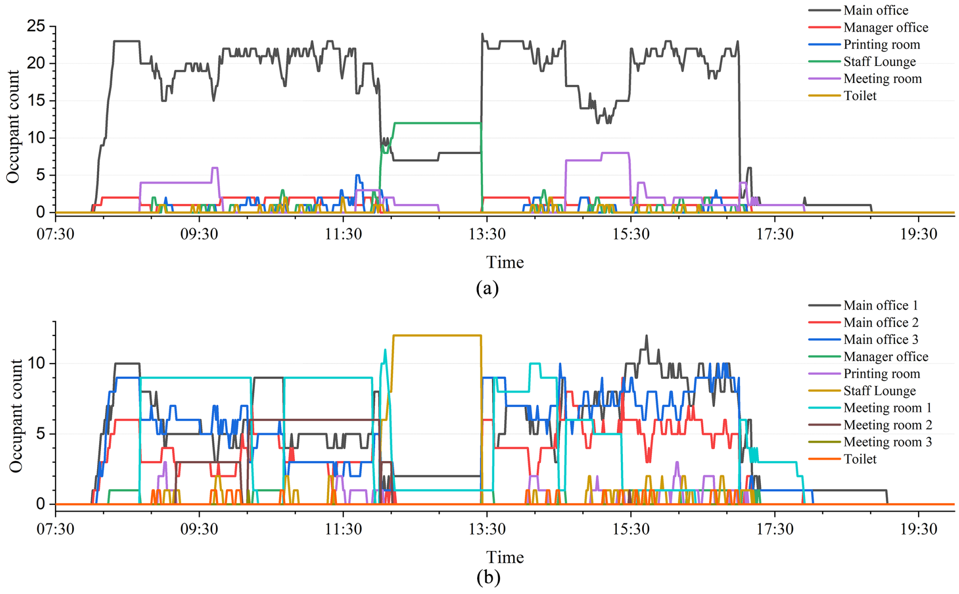


Figure 11. Occupancy result of an individual office room from AnyLogic simulation for (**a**) Layout 1. (**b**) Layout 5.

4.2. Predicted Occupancy Profiles from Neural Networks

The four predicted Neutral Network (proposed Graph LSTM, RNN, GRU, and LSTM) occupancy profiles for each room are plotted against the corresponding AnyLogic simulated occupancy profiles for comparison purposes.

Figure 12 shows the AnyLogic data and neural network predicted occupancies of the main office 1 in one set of data produced with Layout 5. All models predicted the overall occupancy trend of the space well. As the complexity of the neural network increases, the detail of the outputs improves. The three baseline neural networks, RNN, GRU, and LSTM, tend to produce a profile with relatively constant occupancy, where only the general trend and the larger spikes were predicted. Moreover, the forecasted occupant count is usually lower than the actual value. Although the Graph LSTM produced a highly noisy result, the overall trend fits the AnyLogic simulation result well.

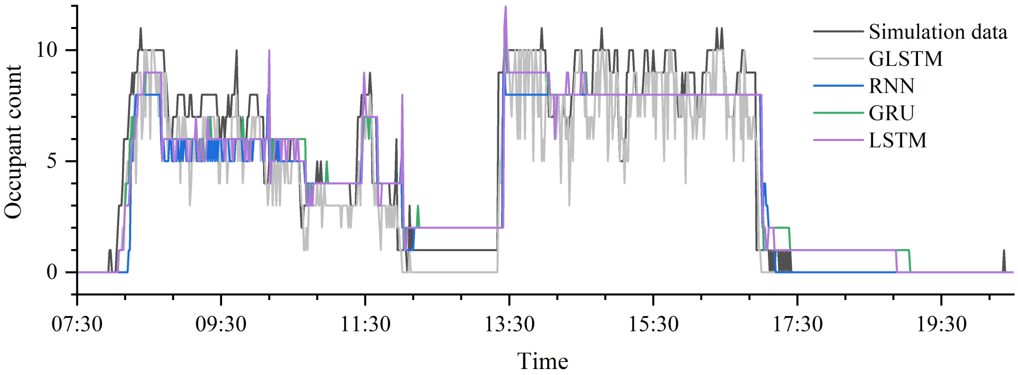


Figure 12. The simulated and predicted occupant count of Layout 5, main office 1.

The results of the meeting rooms are analyzed next. Figure 13 shows the predicted occupant count of meeting room 1. The pattern for meeting room 2 is similar. However, as meeting room 3 was underutilized in the AnyLogic simulation, all neural networks predicted 0 occupancy throughout the day. Therefore, the data from this room is deemed unsuitable for further analysis. Like the results for the main offices, all neural network models capture the overall trend of the space well. However, the baseline model tends to underestimate the occupant count, and the Graph LSTM produced a high amount of noise.

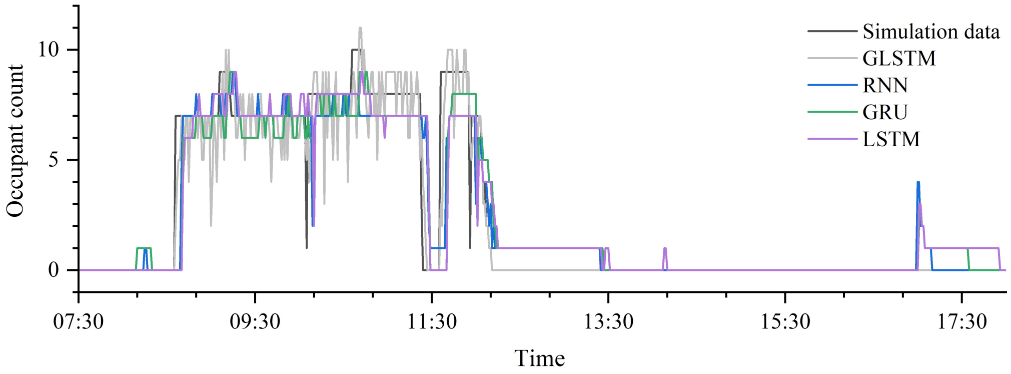


Figure 13. The simulated and predicted occupant count of meeting room 1.

As shown in Figure 14 all neural network models were able to forecast the trend of the occupancy of the staff lounge well, with one plateau during lunch break and several small spikes throughout working hours. As the break event in the AnyLogic simulation was based on probability and only lasted for a short period, the occurrence pattern is less distinct. Hence, all models failed to predict most of the small spikes during working hours. Similar to the results shown in the previously analyzed room types, the baseline neural network models produced inaccurate occupant count by overestimating the value, and although the Graph LSTM predicted the overall value correctly, the output was extremely noisy with fluctuation of up to 7 counts.

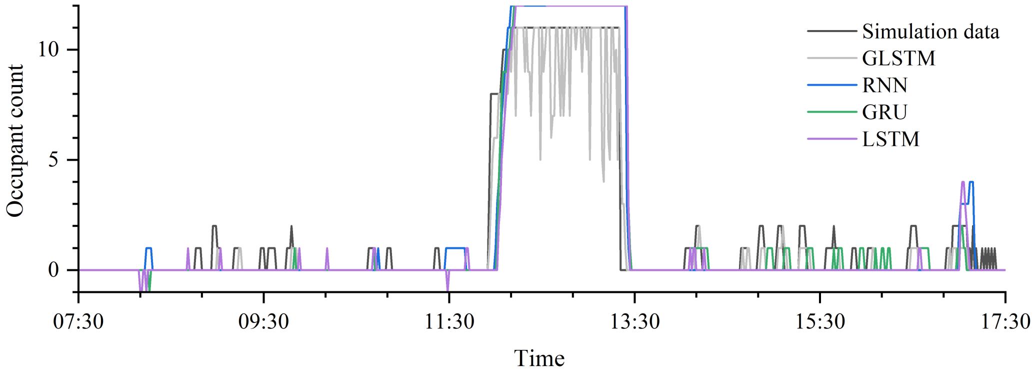


Figure 14. The simulated and predicted occupant count of staff lounge.

The results from the neural networks for the manager's office are shown in Figure 15. The Graph LSTM can represent the overall trend decently, despite the noise presented. However, the baseline results produced relatively constant occupancy and failed to include more details. A possible explanation is that the room is always occupied at a meager occupant count. This limited the amount of detail that could be included in the training data.

Figure 16 presented the predicted occupancy of the printing room. The Graph LSTM produced a highly accurate result that models all the larger fluctuations and most of the small variations compared to the simulated result. However, the baseline neural networks all output a relatively flat occupant count with a maximum of one.

The predicted occupant count of the toilet is plotted in Figure 17. All baseline neural networks, RNN, GRU, and LSTM, produced a constant zero occupant count throughout the simulation day. The Graph LSTM was the only one to forecast the day, with higher accuracy in the afternoon.

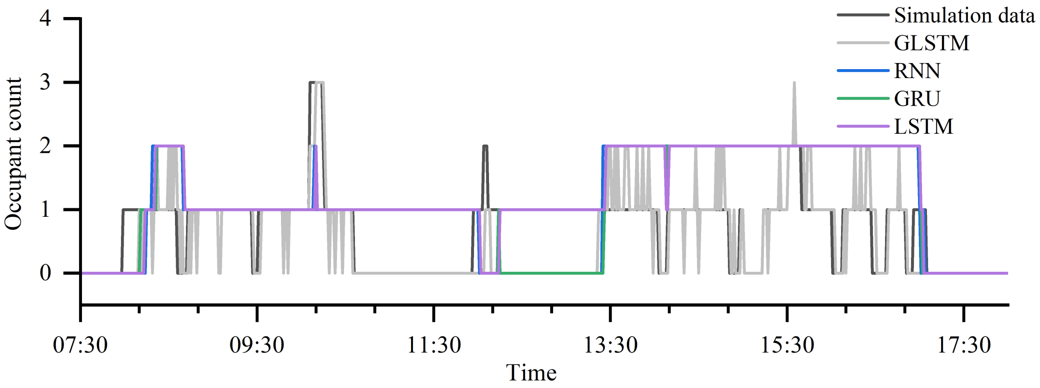


Figure 15. The simulated and predicted occupant count of manager office.

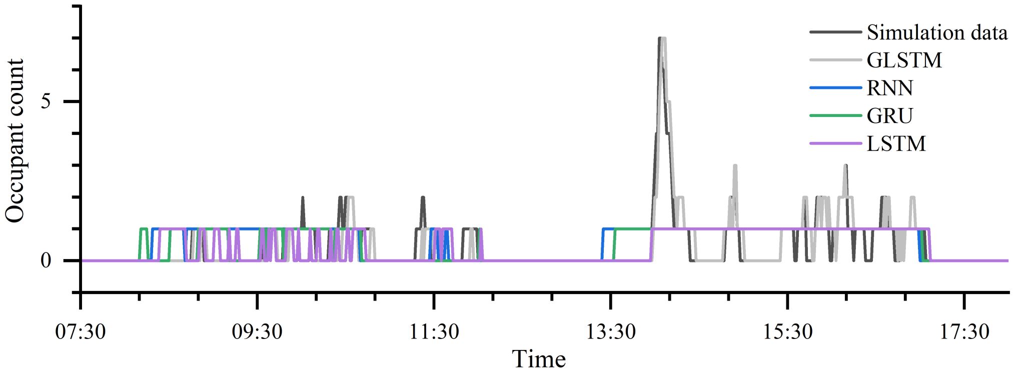


Figure 16. The simulated and predicted occupant count of printing room.

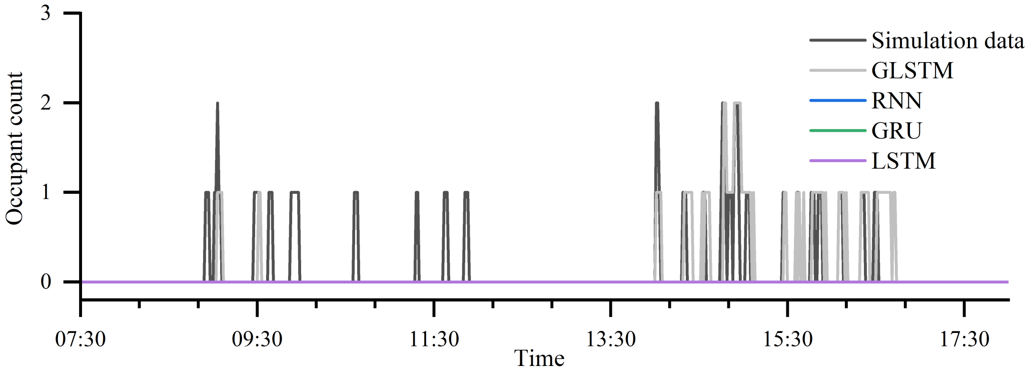


Figure 17. The simulated and predicted occupant count of toilet.

To further analyze the predicted occupancy profile for individual rooms under different neural networks, the evaluation matrices, IA, MAE, and RMSE, were computed and shown in Table 3. The Graph LSTM presented the best performance across all three evaluation matrices for all rooms.

For IA, the closer the value is to one, the lower the model prediction error. IA is a unit value; therefore, it is used to compare the performance of the neural network models between the application in different rooms. Figure 18 plotted the IA of all rooms in Layout 5. For all main offices, meeting rooms, and staff lounge, all neural networks show promising results, with the Graph LSTM slightly outperforming the baseline neural networks by between 0.03% to 5.75%. In the rest of the rooms, all neural network models did not perform as well as in the previously analyzed rooms. The manager room shows the lowest IA of 0.6741 for RNN and the highest IA of 0.8769 for Graph LSTM. For the printing room, all baseline neural networks produced an agreement of 63–65%, which is around 30% lower than the 95% agreement for the Graph LSTM. Among all room types, the forecast of the toilet produced the worst IA, of 0.2058 for all baselines and 0.7592 for Graph LSTM. The Graph LSTM was only able to forecast most of the afternoon value accurately (Figure 17), but it was already a huge jump in performance with a 72.9% increase in IA.

Table 3. Performance of all models on individual rooms.

		Main Office 1	**Main Office 2**	**Main Office 3**	**Meeting Room 1**	**Meeting Room 2**	**Staff Lounge**	**Manager Office**	**Printing Room**	**Toilet**
RNN	IA	0.9609	0.9692	0.9058	0.9380	0.9451	0.9675	0.6741	0.6305	0.2058
	MAE	0.8041	0.2880	0.2929	0.711	1.0382	0.3408	0.4089	0.3105	0.0833
	RMSE	1.4290	1.0073	0.7647	1.1651	1.5427	1.0958	0.7774	0.6792	0.3145
GRU	IA	0.9687	0.9650	0.9447	0.9416	0.9538	0.9678	0.6779	0.6505	0.2058
	MAE	0.8012	0.3379	0.2302	0.6983	0.9354	0.3585	0.4721	0.2733	0.0833
	RMSE	1.2843	1.0691	0.6806	1.1288	1.4013	1.1188	0.7704	0.6512	0.3145
LSTM	IA	0.9680	0.9659	0.9374	0.9388	0.9530	0.9666	0.7183	0.6357	0.2058
	MAE	0.7933	0.3232	0.2086	0.7023	0.9138	0.3359	0.4055	0.2644	0.0833
	RMSE	1.3009	1.0677	0.6892	1.1596	1.4034	1.1149	0.7246	0.6520	0.3145
Graph LSTM	IA	0.9690	0.9818	0.9611	0.9758	0.9660	0.9801	0.8769	0.9537	0.7592
	MAE	0.7444	0.2782	0.1675	0.3937	0.7914	0.2370	0.1459	0.0891	0.0676
	RMSE	1.2820	0.7780	0.5190	0.7767	1.1851	0.7717	0.3846	0.3018	0.2637

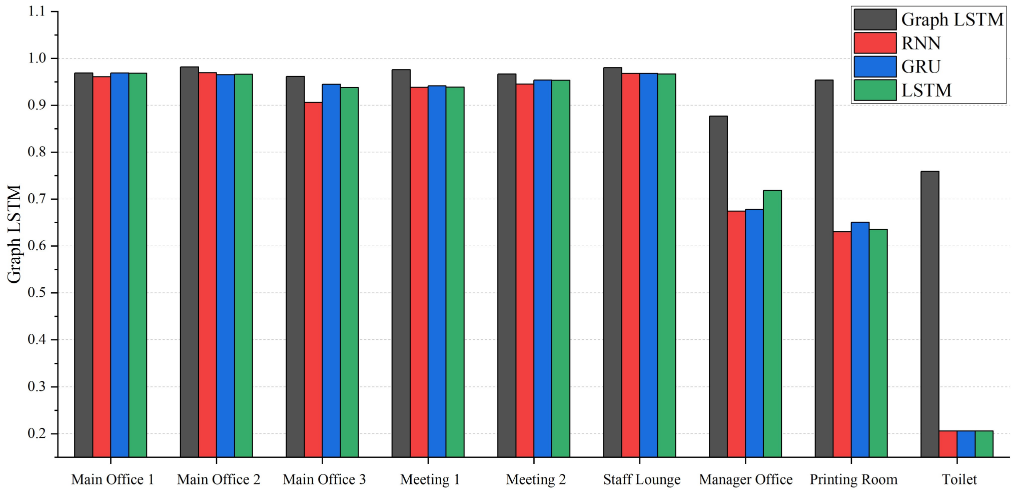


Figure 18. Index of agreement of each room in Layout 5.

The overall performance of the proposed Graph LSTM model and the baseline models are summarized in Table 4. This is calculated by taking the average value of the evaluation metrics for the rooms presented. The Graph LSTM shows the best performance among all neural networks tested from all three evaluation metrics. The differences in IA between the three baselines are very close or less than 0.1, with LSTM presenting a slightly higher IA among them all. The IA of the Graph LSTM is 13.5% higher than the IA of the LSTM. Due to the square difference, the IA is more sensitive to extreme values, implying the presence of more extreme error in the baseline results. The MAE of all models is lower than 0.5 occupant count. Graph LSTM produced the lowest value of 0.3239, which is 38.3% lower than the LSTM baseline and 46.8% lower than the RNN result. Like IA, due to the square difference, RMSE penalizes large errors more severely. The four models computed all resulted in higher RMSE than MAE, indicating the presence of some extreme errors. Compared to the baseline neural networks, the Graph LSTM produced an RMSE of 40.1%, 34.4%, and 34.6% lower than RNN, GRU, and LSTM, respectively. All neural network models created produced an IA value very close to one, and a relatively low MAE and

RMSE value, indicating the models' ability to predict occupancy, with the proposed Graph LSTM slightly outperforming the baseline neural networks.

Table 4. Overall performance of all models constructed.

Model	IA	MAE	RMSE
RNN	0.8000	0.4753	0.9750
GRU	0.8084	0.4656	0.9354
LSTM	0.8099	0.4478	0.9636
Graph LSTM	0.9360	0.3239	0.6958

5. Discussion

The baseline neural network models (RNN, GRU, and LSTM) all show promising results for room types such as main office space and meeting room, where the occupancy profiles have a more stable and recognizable pattern. However, they failed to model the rooms that depend on more random events, such as the toilet and printing room. Hence, the occupancy profiles that did not have a distinct pattern for the neural networks to learn from resulted in poor prediction. Compared to the baseline neural networks, the Graph LSTM model predicted highly detailed and accurate occupant counts. This prediction provided a valuable reference for cases where a detailed occupancy pattern is required. For cases where highly detailed occupancy patterns are not needed, the baseline models offer sufficient performance with a simpler approach.

To apply the occupancy in thermal and energy simulations, the implementation of a more detailed occupancy profile can directly improve the accuracy of heat gain from humans in a space and therefore outputs such as thermal and energy performance, carbon dioxide concentration, and HVAC sizing and operation. Occupancy profiles can also have an impact on daylight and artificial lighting simulations. As different types of rooms may require various levels of occupancy profile details, which may also change throughout the lifetime of a building, it is good practice to include a range of potential profiles as part of simulations. The Graph LSTM model could provide those alternative occupancy profiles, depending on how it is trained. Therefore, more training from telephone center offices would produce different profiles versus training from a standard office setup.

Although the Graph LSTM produced a more accurate prediction of occupancy profiles, the results produced were very noisy. The highly fluctuated result from the Graph LSTM could be caused by overfitting. Overfitting occurs when the deep-learning network tries to fit all the noise during training leading to highly fluctuated results. As this model is highly complex and involves many more parameters than the baseline neural networks, it may include more unwanted noise. Moreover, the training sets applied all have similar trends, but all present various minor fluctuations due to the small number of occupants traveling between spaces randomly. To reduce overfitting, a range of measures exist that can be tested in the future, but which were beyond the scope of this research.

- First, the model could be simplified by reducing the number of hidden layers or neurons, which reduces the total number of parameters, and hence, limiting the ability of the model in fitting the noise.
- Second, the training data can be pre-processed to smooth out some noise; however, this may not be ideal for rooms such as the toilet or the printing room, where the occupancy only consists of small spikes. Furthermore, the pre-processing may smooth out the occupant count spikes when they are passing through a space. Therefore, this can only be used where the study does not require the consideration of room connectivity.
- Thirdly, adjusting the settings for early stopping could reduce overfitting. Although early stopping was already applied for this experiment, the results were still overfitted. This implies the settings may not be ideal for this Graph LSTM model. Possible

adjustments that could be tested in the future include reducing the number of epochs before triggering early stopping when there is no decrease in loss.

The proposed model can also be improved by adding more features as input and increasing the amount of training data, e.g., the inclusion of the room size or the distance between different rooms as input. As the structure of the Graph data is particularly suitable for representing spatial relationships, the inclusion of room size can aid the neural network in determining the travel time of occupants between rooms. The type of office can also be specified as the feature. The training for this study only included a basic office; the data of different types of offices with various occupancy patterns can be included for training to extend the representation of the model. As a result, different versions of the Graph LSTM could exist, each one trained for a specific office type, thus producing alternative occupancy profiles, as needed for simulations.

Lastly, the produced model is highly dependent on the data used for training. For this study, the training data were simulated from AnyLogic and were based on a range of logical assumptions. If monitored data were used for training, it is possible that the performance of the model would be different.

Regarding the accuracy of occupancy predictions against actual occupancy, the latter might vary significantly over time based on climate changes, changes in occupant behavior (i.e., different occupants over time, another company taking office space, etc.), and, as a result, making it difficult to assess the accuracy of an occupancy prediction. If the training data given to a model captures all these variations over time and the future occupancy of an office is in line with past behavior, it can be expected that the predicted occupancy would be fairly accurate. Therefore, the authors see that the proposed model can be very useful when used for generating occupancy profiles that are specific to a given office design so that they can be used in thermal and energy simulations to better understand building performance. This would be invaluable when occupancy data do not exist, if ASHRAE occupancy profiles are not appropriate for usage, or when higher-resolution occupancy data are needed.

6. Conclusions

This study proposed a novel neural network by combining the Graph Neural Network and the LSTM to predict the occupancy of individual rooms in an office layout. The Graph LSTM structure was chosen due to its excellent performance in high spatiotemporal dependency tasks. The following was concluded from the research:

- The proposed Graph LSTM model and all the baseline models tested are all capable of predicting the occupancy for each room of a given office layout unseen by the model before.
- The Graph LSTM model outperformed the baseline neural networks tested when evaluated using metrics such as IA, MAE, and RMSE. The IA of the Graph LSTM is 13.5% higher than the IA of the LSTM, 14.6% higher than RNN, and 13.6% higher than GRU. The MAE of all models is lower than the 0.5 occupant count. Graph LSTM produced the lowest value of 0.3239, which is 38.3% lower than the LSTM baseline and 46.8% lower than the RNN result. The Graph LSTM produced RMSE of 40.1%, 34.4%, and 34.6% lower than RNN, GRU, and LSTM, respectively. All neural network models created produced an IA value very close to 1, and a relatively low MAE and RMSE value, indicating the models' abilities to predict occupancy, with the proposed Graph LSTM slightly outperforming the baseline neural networks.
- All baseline neural network models were able to capture the occupancy of the rooms with more regular occupancy patterns, such as the occupancy of main office spaces and meeting rooms well with IA of between 0.9374 to 0.9687. But they failed to predict the occupancy of rooms that were based on more random events, such as the toilet and printing room. On the other hand, the Graph LSTM produced occupancy more accurately for all room types.

Although the proposed model shows excellent results compared to the baseline test, the result produced by the Graph LSTM was very noisy, due to the occurrence of overfitting in the trained model. Measures to reduce noise in the results were discussed, which include reducing the number of hidden layers, pre-processing training data, and adjusting the settings for early stopping. These measures could be tested in future developments of the model.

Overall, the Graph LSTM model shows a high potential in the application of predicting office occupancy. In future studies, the model can benefit from improving its representability by adding more features such as room size and office type. It will also be beneficial to be trained and tested using monitored occupancy data as it may exhibit different performance compared to using simulated data for training.

Author Contributions: Conceptualization, Y.X. and S.S.; methodology, Y.X.; software, Y.X.; analysis, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, S.S.; supervision, S.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
ARMA	Autoregressive Moving Average model
CO_2	Carbon dioxide
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
IA	Index of agreement
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory Neural Network
MAE	Mean absolute error
ML	Machine Learning
RNN	Recurrent Neural Network
RMSE	Root mean square error
SVM	Support Vector Machine

References

1. Climate Change Mitigation. Available online: https://ukgbc.org/our-work/climate-change-mitigation/ (accessed on 4 April 2023).
2. New Report: The Building and Construction Sector can Reach Net Zero Carbon Emissions by 2050. Available online: https://www.worldgbc.org/news-media/WorldGBC-embodied-carbon-report-published (accessed on 4 April 2023).
3. Yoshino, H.; Chen, A. Total Energy Use in Buildings: Analysis and Evaluation Methods (Annex 53). Available online: https://www.iea-ebc.org/Data/publications/EBC_PSR_Annex53.pdf (accessed on 4 April 2023).
4. Ding, Y.; Wang, Q.; Wang, Z.; Han, S.; Zhu, N. An occupancy-based model for building electricity consumption prediction: A case study of three campus buildings in Tianjin. *Energy Build.* **2019**, *202*, 109412. [CrossRef]
5. Salimi, S.; Hammad, A. Critical review and research roadmap of office building energy management based on occupancy monitoring. *Energy Build.* **2019**, *182*, 214–241. [CrossRef]
6. He, M.; Lee, T.; Taylor, S.; Firth, S.K.; Lomas, K.J. Coupling a stochastic occupancy model to EnergyPlus to predict hourly thermal demand of a neighbourhood. In Proceedings of the 14th International Conference of the International Building Performance Simulation Association, Hyderabad, India, 7–9 December 2015; pp. 2101–2108.
7. NSI/ASHRAE/IES Standard 90.1-2019—Energy Standard for Buildings Except Low-Rise Residential Buildings. Available online: https://www.ashrae.org/technical-resources/bookstore/standard-90-1 (accessed on 4 April 2023).
8. EnergyPlus Documentation: Output Details and Examples EnergyPlus Outputs, Example Inputs and Data Set Files. Available online: https://energyplus.net/sites/default/files/pdfs_v8.3.0/OutputDetailsAndExamples.pdf (accessed on 4 April 2023).
9. Dong, B.; Lam, K.P. A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. *Build. Simul.* **2014**, *7*, 89–106. [CrossRef]

10. Clevenger, C.M.; Haymaker, J. The Impact of The Building Occupant on Energy Modeling Simulations. In Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, QC, Canada, 14–16 June 2006; pp. 1–10.
11. Yan, D.; O'Brien, W.; Hong, T.; Feng, X.; Gunay, H.B.; Tahmasebi, F.; Mahdavi, A. Occupant behavior modeling for building performance simulation: Current state and future challenges. *Energy Build.* **2015**, *107*, 264–278. [CrossRef]
12. IEA World Energy Balances 2018. Available online: https://webstore.iea.org/world-energy-balances-2018 (accessed on 4 April 2023)
13. Khazail, J. Modeling occupation behaviour. *ASHRAE J.* **2016**, *58*, 72–74.
14. Wang, W.; Chen, J.; Hong, T. Occupancy prediction through machine learning and data fusion of environmental sensing and Wi-Fi sensing in buildings. *Autom. Constr.* **2018**, *94*, 233–243. [CrossRef]
15. Peng, Y.; Rysanek, A.; Nagy, Z.; Schlueter, A. Case Study Review: Prediction Techniques in Intelligent HVAC Control Systems. In Proceedings of the 9th International Conference on Indoor Air Quality Ventilation and Energy Conservation in Buildings, Seoul, Republic of Korea, 23–26 October 2023.
16. Szczurek, A.; Maciejewska, M.; Pietrucha, T. Occupancy determination based on time series of CO2 concentration, temperature and relative humidity. *Energy Build.* **2017**, *147*, 142–154. [CrossRef]
17. Liang, X.; Hong, T.; Shen, G.Q. Occupancy data analytics and prediction: A case study. *Build. Environ.* **2016**, *102*, 179–192. [CrossRef]
18. Qi, Y.; Li, Q.; Karimian, H.; Liu, D. A hybrid model for spatiotemporal forecasting of PM 2.5 based on graph convolutional neural network and long short-term memory. *Sci. Total Environ.* **2019**, *664*, 1–10. [CrossRef] [PubMed]
19. D'Oca, S.; Hong, T. Occupancy schedules learning process through a data mining framework. *Energy Build.* **2015**, *88*, 395–408. [CrossRef]
20. Anand, P.; Yang, J.; Sekhar, C. Improving The Accuracy Of Building Energy Simulation Using Real-Time Occupancy Schedule And Metered Electricity Consumption Data. In Proceedings of the ASHRAE Annual Conference, Long Beach, CA, USA, 24–28 June 2017.
21. Jiang, C.; Masood, M.K.; Soh, Y.C.; Li, H. Indoor occupancy estimation from carbon dioxide concentration. *Energy Build.* **2016**, *131*, 132–141. [CrossRef]
22. Yang, Z.; Becerik-Gerber, B. Modeling personalised occupancy profiles for representing long term patterns by using ambient context. *Build. Environ.* **2014**, *78*, 23–35. [CrossRef]
23. Ryu, S.H.; Moon, H.J. Development of an occupancy prediction model using indoor environmental data based on machine learning techniques. *Build. Environ.* **2016**, *107*, 1–9. [CrossRef]
24. Lu, Z.; Lv, W.; Xie, Z.; Du, B.; Huang, R. Leveraging Graph Neural Network with LSTM For Traffic Speed Prediction. In Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, Leicester, UK, 19–23 August 2019; pp. 74–81.
25. Luo, N.; Wang, Z.; Blum, D.; Weyandt, C.; Bourassa, N.; Piette, M.A.; Hong, T. A three-year dataset supporting research on building energy management and occupancy analytics. *Sci. Data* **2022**, *9*, 156. [CrossRef] [PubMed]
26. Duarte, C.; Wymelenberg, K.V.D.; Rieger, C. Revealing occupancy patterns in an office building through the use of occupancy sensor data. *Energy Build.* **2013**, *67*, 587–595. [CrossRef]
27. An P.; Cheong, D.; Sekhar, C. A review of occupancy-based building energy and IEQ controls and its future post-COVID. *Sci. Total Environ.* **2022**, *804*, 150249. [CrossRef] [PubMed]
28. AnyLogic Simulation Software. Available online: https://www.anylogic.com/ (accessed on 1 June 2023).
29. Adult Smoking Habits in the UK: 2018. Available online: https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandlifeexpectancies/bulletins/adultsmokinghabitsingreatbritain/2018 (accessed on 4 April 2023).
30. PyTorch Geometric Documentation. Available online: https://pytorch-geometric.readthedocs.io/en/latest/ (accessed on 4 April 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.