

Article

Estimating Spatio-Temporal Building Power Consumption Based on Graph Convolution Network Method

Georgios Vontzos ¹, Vasileios Laitos ¹, Avraam Charakopoulos ^{2,*}, Dimitrios Bargiotas ¹
and Theodoros E. Karakasidis ²

¹ Department of Electrical and Computer Engineering, University of Thessaly, 38334 Volos, Greece; gvontzos@uth.gr (G.V.); vlaitos@uth.gr (V.L.); bargiotas@uth.gr (D.B.)

² Department of Physics, University of Thessaly, 35100 Lamia, Greece; thkarak@uth.gr

* Correspondence: avcharak@uth.gr

Abstract: Buildings are responsible for around 30% and 42% of the consumed energy at the global and European levels, respectively. Accurate building power consumption estimation is crucial for resource saving. This research investigates the combination of graph convolutional networks (GCNs) and long short-term memory networks (LSTMs) to analyze power building consumption, thereby focusing on predictive modeling. Specifically, by structuring graphs based on Pearson's correlation and Euclidean distance methods, GCNs are employed to discern intricate spatial dependencies, and LSTM is used for temporal dependencies. The proposed models are applied to data from a multistory, multizone educational building, and they are then compared with baseline machine learning, deep learning, and statistical models. The performance of all models is evaluated using metrics such as the mean absolute error (MAE), mean squared error (MSE), R-squared (R^2), and the coefficient of variation of the root mean squared error (CV(RMSE)). Among the proposed computation models, one of the Euclidean-based models consistently achieved the lowest MAE and MSE values, thus indicating superior prediction accuracy. The suggested methods seem promising and highlight the effectiveness of GCNs in improving accuracy and reliability in predicting power consumption. The results could be useful in the planning of building energy policies by engineers, as well as in the evaluation of the energy management of structures.

Keywords: GCN; LSTM; building power prediction; adjacency matrix computation; graph



Citation: Vontzos, G.; Laitos, V.; Charakopoulos, A.; Bargiotas, D.; Karakasidis, T.E. Estimating Spatio-Temporal Building Power Consumption Based on Graph Convolution Network Method.

Dynamics **2024**, *4*, 337–356. <https://doi.org/10.3390/dynamics4020020>

Academic Editor: Cesare Biserni

Received: 26 February 2024

Revised: 23 March 2024

Accepted: 24 April 2024

Published: 2 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

At the global level, buildings consume around 30% of produced energy, and they are also responsible for 26% of energy-related emissions [1]. Furthermore, in the European Union, the energy utilized in buildings represents 42% of energy use and more than 30% of greenhouse gas emissions [2]. Consequently, buildings in general could be considered as one of the largest energy consumers. Therefore, accurate building energy prediction is vital for engineers to design policies that lead to high levels of energy efficiency, for stakeholders to make investment decisions, and for consumers and businesses to save energy and money.

Building energy prediction is based on three main methodologies: physical, hybrid, and data-driven models. Physical or “white box” models dynamically describe the thermal behavior of a building using heat and mass transfer equations. EnergyPlus, TRNSYS, and DOE-2 are some of the available software packages for building energy modeling [3]. Hybrid or “gray box” methods incorporate physical and data-driven approaches to predict building energy consumption, such as the proposal by Dong et al. [4], which combines building geometry (physical) and historical power consumption data to predict air conditioning and total power consumption for a group of residences. Furthermore, data-driven or “black box” models learn from the historical data of the target values, as well as environmental and exogenous factors, to forecast power utilization. Data-driven algorithms

are distinguished into statistical and machine learning (ML) models. The former include methods like linear regression (LR), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA), while the latter comprise methods such as regression tree, support vector regression (SVR), artificial neural networks (ANNs), deep learning (DL), and ensemble [5,6].

In the last decade, machine learning methods have gained the interest of researchers and have been extensively used in building power prediction. For instance, ML and DL algorithms were used for forecasting electrical demand in airports [7], offices [8–10], educational institutions [11–13], and residential [14,15] buildings. In more detail, the recurrent neural network (RNN) algorithms, namely long short-term memory (LSTM) [16], convolutional neural network (CNN), hybrid CNN-LSTM [17], and gated recurrent units (GRU) [18], perform well in power prediction due to their ability to extract the temporal dependencies between energy usage and other features such as occupancy, lighting conditions, and equipment usage. A brief summary of building energy prediction applications utilizing LSTM models is presented in [19].

Graph neural networks (GNNs) are deep learning algorithms for analyzing and modeling complex relationships, and they have been successfully applied to different scientific fields including social networks, citation networks, molecular structures, physics, chemistry, road networks, finance, etc. [20,21]. The fundamental idea behind GNNs is to extract the spatial correlation of the nodes based on their topological connections. GNNs demonstrate versatility and effectiveness in real-world challenges, and they have shown notable success in various applications, including chemical reaction prediction, question answering, image classification, disease classification, and time-series prediction [22].

Building-related prediction applications, such as energy usage, indoor environmental conditions, occupancy, etc., have known problems such as time series prediction; however, significant progress has been made with the incorporation of graph neural networks (GNNs), which represent a novel approach to modeling complex relationships within building structures. Inspired by the success of GNNs in capturing dependencies within graph-structured data, Hu et al. [23] introduced a novel graph-based hybrid model (a spatio-temporal graph convolutional network (ST-GCN)) to embed solar-based building interdependencies in urban building energy modeling in order to predict the energy usage of several buildings in a university campus. They performed a solar analysis to construct a directed-weighted graph of the project, where nodes represent buildings and edges represent the solar impacts. Guo et al. [24] combined GCN and GRU to predict the future energy consumption of 140 locations in a large-scale aluminum profile plant located in Guangdong, China. Lu et al. [25] proposed a GCN-based model for the estimation of the design loads of complex-shaped buildings.

Furthermore, Jia et al. [26] proposed a graph-based model to predict the thermal load of a building that incorporates graph attention neural networks (GATs) and gated recurrent units (GRUs) to extract spatial and temporal dependencies, respectively. This model was applied to a dataset from a simulated, single-story, and four-zone building with a general prediction accuracy (MSE) of nearly 0.01. The researchers depicted the thermal zones of the building as an unweighted and undirected graph to define the feature matrix. Moreover, Zhang et al. [27] presented a spatio-temporal, graph-based data-driven model (GNN-RNN), which showed enhanced accuracy compared to conventional deep learning models, for the indoor environment prediction and optimal control of air conditioning systems. The authors used a graph-based data representation of the central air conditioning system of the building to integrate spatial and temporal data to the forecasting model. Regarding spatial data, consider the air handling units' and variable air volume boxes' physical location and the connections between them. In terms of occupancy prediction, Xie and Stravoravdis [28] suggested a hybrid GCN-LSTM model, which is implemented in a generated occupancy profile. The authors mapped each office layout as an undirected, unweighted graph, where the nodes indicate each room in the floor plan and the edges represent their connection.

In summarizing the recent literature on GNNs for building energy usage prediction, to the best of the authors' knowledge, not much emphasis has been given to the impact of graph construction methods on the field of building power prediction. In this study, several graph computation methodologies are examined and implemented over a GNN-RNN model to forecast the zone-level and overall energy usage of a real-world multizone, multistory building dataset.

The main contributions of this work are as follows:

1. An adaptation of graph computation techniques, which have previously been utilized in other domains, for the purpose of building load prediction.
2. An investigation, adaptation, and implementation of generic methods to extract the spatial information of building zones to build a graph that represents the relationship between zones in pairs for a prediction application.
3. The generated graphs are implemented over a GNN-RNN forecasting model, and their results are compared with a variety of statistical, machine, and deep learning algorithms in terms of accuracy and the utilization of several error metrics.

This paper is divided into the following parts: Section 2 introduces and analyzes the necessary actions related to the graph computation techniques and presents the forecasting model. Dataset description, preprocessing, conducted experiments, and their results are presented in Section 3. In Section 4, the results and limitations are discussed while some directions for future work are suggested. This paper is then concluded in Section 5.

2. Materials and Methods

In this section, topics related to the prediction of building power consumption and several major concepts will be presented. The preliminary Section 2.1 serves as a foundational introduction in graph symbolization, prediction problem definition, and power problem forecasting formulation, thereby providing essential context for the subsequent analysis. Furthermore, in Section 2.2, the methodologies for adjacency matrix computation are presented, which involves the exploration of the various computational techniques employed to derive the crucial representation of graph structures that will be utilized in the forecasting model described in Section 2.3. Lastly, in Section 2.4, a comparison of the models that evaluate and contrast the efficacy of the proposed forecasting model is presented, thereby providing invaluable insight into the relative strengths and limitations of the proposed model.

2.1. Preliminaries

A multizone building power consumption prediction can be considered a spatio-temporal or, even better, a micro spatio-temporal prediction application. In order to describe the space layout or the quantitative relations between zones, a graph network approach is introduced.

This graph is symbolized as $G = (V, E)$, where V is a set of N nodes that denote each building's thermal zone, while E is an edges set that represents the association between a pair of nodes. A Boolean (binary) adjacency matrix A , whose elements are 0 or 1, is expressed as $A \in R^{N \times N}$, and it describes the links for each couple of nodes. The adjacency matrix element a_{ij} is equal to 1 if nodes i and j are directly connected, otherwise $a_{ij} = 0$. In this study, several methods are used for the computation of the adjacency matrix, and they will be analyzed in Section 2.2.

The estimation problem is defined as a given sequence of power consumption values in the time steps $t + 1, t + 2, \dots, t + T$, and the prediction of future power consumption values is represented in the time steps $t + T + 1, t + T + 2, \dots, t + T + h$, where T is the past historical input data and h the length of the prediction horizon.

The present and historical values of the power consumption of all zones are defined as $X = (X_1, X_2, \dots, X_T)$, while the values x_i^t for every zone i at every sampling time t are summarized in a feature matrix $X_t = (x_t^1, x_t^2, \dots, x_t^N) \in R^{N \times D}$, where D is the number of

input features. Furthermore, the sequence of the building power consumption estimation for a prediction time t is defined as $\hat{X}_t = (\hat{x}_t^1, \hat{x}_t^2, \dots, \hat{x}_t^N) \in R^{N \times h}$.

Consequently, in incorporating the abovementioned definitions, the power prediction is formulated as $\hat{X}_{t+T+h} = F(X, G)$, where F is the GNN-RNN model.

2.2. Adjacency Matrix Computation

Constructing a graph from the geographical location data of elements in a two-dimensional space is an unambiguous task. For instance, in traffic speed prediction research, the values a_{ij} of the adjacency matrix, which constructs the graph, could be the inverse proportion of the geographical distance of points i and j . In contrast, in a multistory building energy prediction study, the thermal zones are in a three-dimensional space, in which case, the impression to a graph is complicated. In this section, the methods that are used for adjacency matrix computation are presented and analyzed.

The methods that are used in this study are divided into two categories: correlation-based and distance-based.

2.2.1. Correlation-Based Methods

Pearson Correlation Coefficient (PCC)

As stated by Li et al. [29] and utilized by Zhang et al. [27], the similarities among node vectors (also known as node attributes) can serve as a quantitative measure of the correlation between nodes. Therefore, in this first method, the elements a_{ij} of the adjacency matrix A are calculated using Pearson's correlation coefficient [30] (Equation (1)) between the two zones v_i and v_j and the power consumption, which are described in Equation (2):

$$r = \frac{\sum_{k=1}^n (Pv_{ik} - \overline{Pv_i})(Pv_{jk} - \overline{Pv_j})}{\sqrt{\sum_{k=1}^n (Pv_{ik} - \overline{Pv_i})^2 \sum_{k=1}^n (Pv_{jk} - \overline{Pv_j})^2}}, \tag{1}$$

where Pv_i and Pv_j are the power consumption of zones v_i and v_j , Pv_{ik} and Pv_{jk} are the individual power consumption data points, $\overline{Pv_i}$, $\overline{Pv_j}$ are the mean values, and n is the number of data points.

$$a_{ij} = \begin{cases} 1, & PCC(v_i, v_j) > \sigma \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where PCC is the Pearson correlation coefficient and $\sigma \in [-1, 1]$ is a threshold to control the distribution and sparsity of the matrix A . Only the nodes that have a correlation value greater than the threshold are connected.

Absolute Pearson Correlation Coefficient (PCCA)

This method, as presented in Equation (3), is nearly similar to the previous one (PCC), with the only difference being that the absolute value of PCC is chosen, which will produce a graph different from the previous method in the case where the correlation matrix has a sufficient number of negatively correlated values.

$$a_{ij} = \begin{cases} 1, & |PCC(v_i, v_j)| > \sigma \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases}. \tag{3}$$

Pearson Correlation Coefficient Scaled (PCCS)

In this last correlation-based method, the entries of the PCC matrix are scaled (Equation (4)) in space $[0,1]$ using Equation (5).

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \cdot (max - min) + min, \tag{4}$$

where X_{min} and X_{max} are the minimum and maximum values of the input data X , and min and max are the desired range to transform the input data.

$$a_{ij} = \begin{cases} 1, & PCC_{scaled[0,1]}(v_i, v_j) > \sigma \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

2.2.2. Distance-Based Methods

Inspired from other works on traffic prediction and molecular science, we represent the Euclidean data in a non-Euclidean space of a graph using the following method. The floors of the building are placed in a three-dimensional Cartesian coordinate system, where the lower left corner of the ground floor of the building is considered as the point zero of the three axes while the coordinates of each zone's center is assumed.

Euclidean Distance Scaled (EDS)

A matrix with dimensions $N \times N$ was created, where each element is the spatial distance between zones i and j , which is calculated by Equation (6) and deals with the Euclidean distance in three-dimensional space, as is depicted in Figure 1. The matrix is then scaled (Equation (4)) in space $[0,1]$. A filter, i.e., the threshold value σ , is applied to the matrix elements, where only the values lower than the threshold are kept. The adjacency matrix is computed according to Equation (7).

$$ED = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, \quad (6)$$

where x_i, y_i, z_i and x_j, y_j, z_j are coordinates in the three-dimensional space of nodes i and j , respectively.

$$a_{ij} = \begin{cases} 1, & ED_{scaled[0,1]}(v_i, v_j) < \sigma \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases}, \quad (7)$$

where ED_{scaled} is the Euclidean distance between a pair of thermal zones and σ is a threshold value.

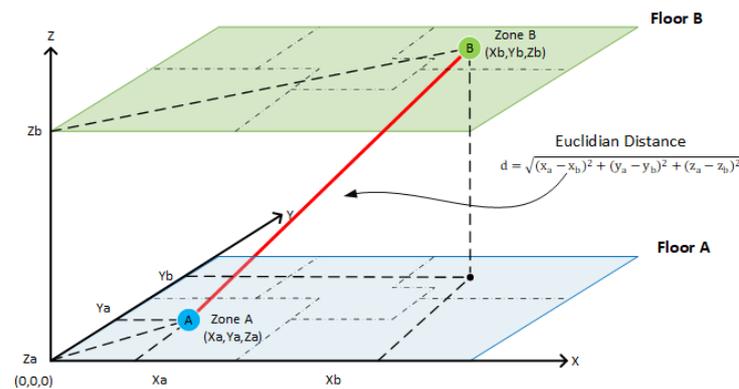


Figure 1. Euclidean distance computation in a Cartesian space between zones. This figure represents two floor plans, A and B, with the relevant zones of A and B.

Euclidean Distance with Threshold (EDT)

In this method, the matrix elements a_{ij} are equal to 1 when the Euclidean distance is smaller than the threshold value, as illustrated in Equation (8). In other words, only the nodes whose distance is less than the value of the threshold are connected.

$$a_{ij} = \begin{cases} 1, & ED(v_i, v_j) < \sigma \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Euclidean Distance with Gaussian Kernel (EDGK)

In this last distance-based method, the elements a_{ij} of the adjacency matrix are determined by the Gaussian kernel weighting function [31], which has been adapted in this study and is presented in Equation (9).

$$a_{ij} = \begin{cases} 1, & \exp\left(-\frac{[ED]^2}{2\theta^2}\right) \text{ if } ED < \sigma \text{ and } i \neq j, \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

where ED is the Euclidean distance, and σ and θ are the distance and distribution thresholds among the zones, respectively.

2.3. Forecasting Model

In the field of predicting building energy consumption, combining graph convolutional networks (GCNs) with long short-term memory (LSTM) models provides a powerful approach for capturing the complex spatial and temporal relationships found in building energy systems. GCNs are highly effective in analyzing complex relationships within graph-structured data, such as building networks, by aggregating the information from neighboring nodes. This capability is especially valuable for modeling the spatial relationships among various components of a building, such as rooms, floors, and zones. On the other hand, LSTM models stand out by capturing temporal patterns and dependencies over time, which are crucial for forecasting energy consumption dynamics. Thus, the utilization of a GCN-LSTM model in building energy prediction tasks extracts spatial and temporal variations, thereby representing an improved predictive performance and a deeper understanding of energy consumption dynamics in complex building environments.

2.3.1. Graph Convolutional Networks

Graph convolutional networks (GCN) are one of the most straightforward and extensively utilized variants of graph neural networks (GNNs). Their operation is based on the aggregation of the attributes of neighboring nodes via a weighted average, where the weights are determined by the edge connections [32].

A simplified version of a GCN [33] takes a graph as the input with a set of node features X , thereby utilizing this information to produce node embeddings that use the graph convolution operation, which is expressed as a nonlinear function, as is illustrated in the following layer-wise propagation formula:

$$H^{(l+1)} = f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)}), \quad (10)$$

where A is the adjacency matrix of graph G , $H^l \in R^{N \times C}$ and $H^0 = X \in R^{N \times D}$ represent the output and input vectors of l th GCN layer, $\sigma(\cdot)$ is denoted the activation function, and $W^{(l)}$ is a layer-specific trainable weight matrix.

In this study, in order to extract spatial information, a single-layer GCN was used in the prediction model, as is shown in Equation (11).

$$f(X, A) = ReLU\left(AXW^{(0)}\right), \quad (11)$$

where $ReLU(\cdot)$ is applied as an activation function and $W^{(0)}$ is randomly initialized using a Glorot initializer [34].

2.3.2. Long Short-Term Memory

Long short-term memory (LSTM) [35] is a variation of recurrent neural networks. It presents tremendous prediction effectiveness in numerous tasks, such as time series prediction, speech recognition, and natural language processing, due to its ability to capture temporal dependencies in sequential data.

LSTM bears a similarity to the standard RNN formation, but it uses a specialized memory unit capable of retaining or discarding information over sequential data. The

memory unit or cell has four layers that interact in a special way. The computation method is explained in Equation (12) [36], and the architecture is depicted in Figure 2.

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
 \tilde{c}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
 c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\
 o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
 h_t &= o_t \cdot \tanh(c_t).
 \end{aligned}
 \tag{12}$$

More specifically, t is the time step and x_t is the input to the current time step. W , b , σ , and \tanh are the weight matrices, bias vectors, sigmoid activation function, and hyperbolic tangent activation function, respectively. Forget Gate f_t decides what information of the previous unit state to forget, while Input Gate i_t controls the new information to store in the memory unit. Furthermore, c_t is the memory unit that stores the long-term information from previous time steps and combines the information from the forget and input gates to update the memory unit. Finally, Output Gate o_t determines the output of the LSTM memory unit.

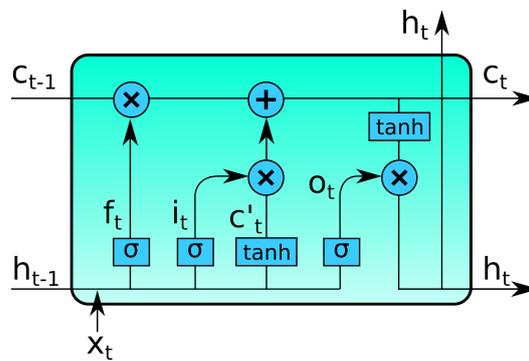


Figure 2. The LSTM memory unit architecture.

2.3.3. The Proposed Model: GCN-LSTM

The structure of the GCN-LSTM model utilized in this study, as shown in Figure 3, is a subset of the generic GNN-RNN prediction models. It consists of an input layer, a GCN layer, a LSTM layer, and a fully connected layer as the output. Each layer is extensively described in the following list.

Input layer: A graph G with N nodes. These represent the number of building zones that are generated using the similarities in the power consumption or distance information between the zones. The input of the GCN-LSTM model is the historical values of the N zones at T time points before the prediction window $T + h$.

GCN layer: A GCN layer with a ReLU activation function is used to extract the spatial correlations between the neighbor nodes. The GCN layer uses weights to aggregate the information from the neighbor nodes based on the acquired correlation between zones.

LSTM layer: This layer is adopted as a temporal feature extraction module to capture the long-term sequential dependencies of the power consumption between zones.

Output layer: A fully connected (Dense) layer returns the power consumption prediction sequence.

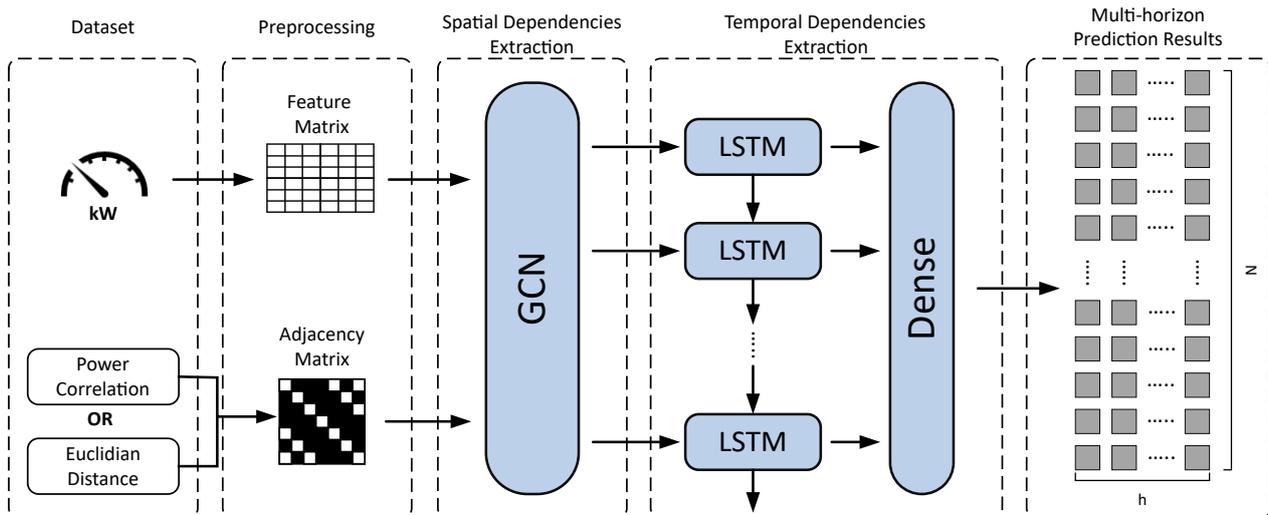


Figure 3. The GCN-LSTM model structure.

2.4. Comparison Models

The presented adjacency matrix computation methods in combination with the GCN-LSTM model are compared with some statistical, machine, and deep learning models for multi-output multistep building power prediction. These baseline models are introduced below:

Historical average (HA): This is a statistical model that does not utilize online data for making predictions. The strategy underlying its prediction is as follows: For each prediction value, the average of historical values is used. Therefore, considering the mean value, the prediction horizon can be determined from one step up to multiple steps ahead. This model is quite simple but lacks the ability to capture abrupt value changes.

Multilayer perceptron (MLP) [37]: This neural network model works by taking the historical data as the input and using multiple layers of interconnected neurons to learn patterns and relationships within the data. These patterns help the model make predictions about future values. During the training process, the model adjusts its internal parameters through backpropagation, and it compares its predictions to the actual values and updates its synaptic weights to minimize the forecast errors.

Convolutional neural network (CNN) [38]: This is a kind of deep learning model that uses sequential data as the input and convolutional layers to extract the features and patterns from the data over time. These features capture local dependencies within the time series, thereby aiding in learning the relevant information for forecasting. The main architecture of this model consists of convolutional layers; pooling layers, which the model applies to reduce dimensionality and further extract essential features; and one or more fully connected layers, which are used to make the final predictions.

Long short-term memory (LSTM): LSTM is a variant of RNN, as described in Section 2.3.2.

3. Experiments

3.1. Dataset Description

In order to verify the effectiveness of our methods, several experiments were conducted on the following real-world dataset.

The CU-BEMS [39] dataset was collected from a seven-story building with an overall area of 11,700 m² located in Bangkok, Thailand. The building is divided into 33 thermal zones. The plan of each floor is shown in Figure 4. The dataset consists of measurements of the electricity consumption, in kW, of the individual air conditioning units, lighting, and plug loads of each zone. Furthermore, the indoor environmental conditions that were recorded include temperature (°C), relative humidity (%), and ambient light (lux) values for each zone. The data were gathered from 1 July 2018 to 31 December 2019, with a resolution of one minute.

This dataset has been used by other scholars to predict indoor temperature [40], real-time thermal comfort [41], and energy use [42].

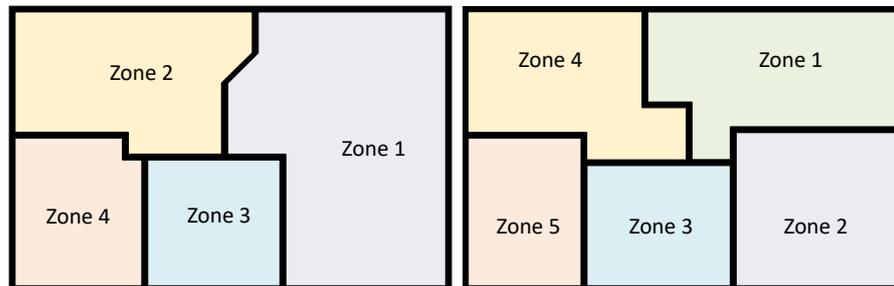


Figure 4. The floor plans on Floors 1–2 (left) and Floors 3–7 (right) [39].

3.2. Data Preprocessing

For this study, the load measurements of the same zone were aggregated to one value per time step, and the environmental measurements were discarded. The dataset does not present missing values. The null-empty values, due to null power consumption, were converted to zeros. The power consumption of Floor 1, Thermal Zone 3 appeared with a peak value (i.e., it is 160 times larger than the previous and next value) for a certain time step; thus, it was replaced with a linear interpolated value. The final dataset consists of 33 columns that represent the overall power consumption of each zone. Furthermore, the variables of the dataset were standardized using the Standard Scaler function from the Scikit Learn Python package before being used by the prediction algorithms. This function rescales the distribution of values such that the mean of the values is 0 and the standard deviation is 1.

3.3. Dataset Analysis

In this section, a brief analysis of the dataset is conducted. The dataset consists of 33 time series, with 790.560 observations, which correspond to the power consumption of the relevant zones. Therefore, this dataset is described as “large data”, and a letter value plot [43] is used to investigate the distribution and variability of the data. The letter–value plot is an extension of the box plots, which shows only actual values, and it labels fewer observations as outliers than the box plots.

Figure 5a presents a letter–value plot for each thermal zone, and Figure 5b shows the horizontal bar plot of the median and mean values for each zone of the building. From these two plots, it is noticeable that most of the zones presented a mean power consumption that was lower than 10 kW and a median that was nearly 2 kW. Only a couple of the zones presented mean and median values greater than 20 kW.

Furthermore, Figure 6 shows the letter–value plot for the total power consumption of the building. It was discerned that the median was around 300 kW and that half of the total observations were distributed at 300 kW and below; meanwhile, the remainder of the total observed values were allocated between 300 kW and around 1750 kW.

Additionally, Figure 7 depicts a correlation plot that visualizes the relationships between the power consumption of the zones. It seemed that Zone 1 to Zone 4, which are located on the first floor, had a low correlation compared to the other zones.

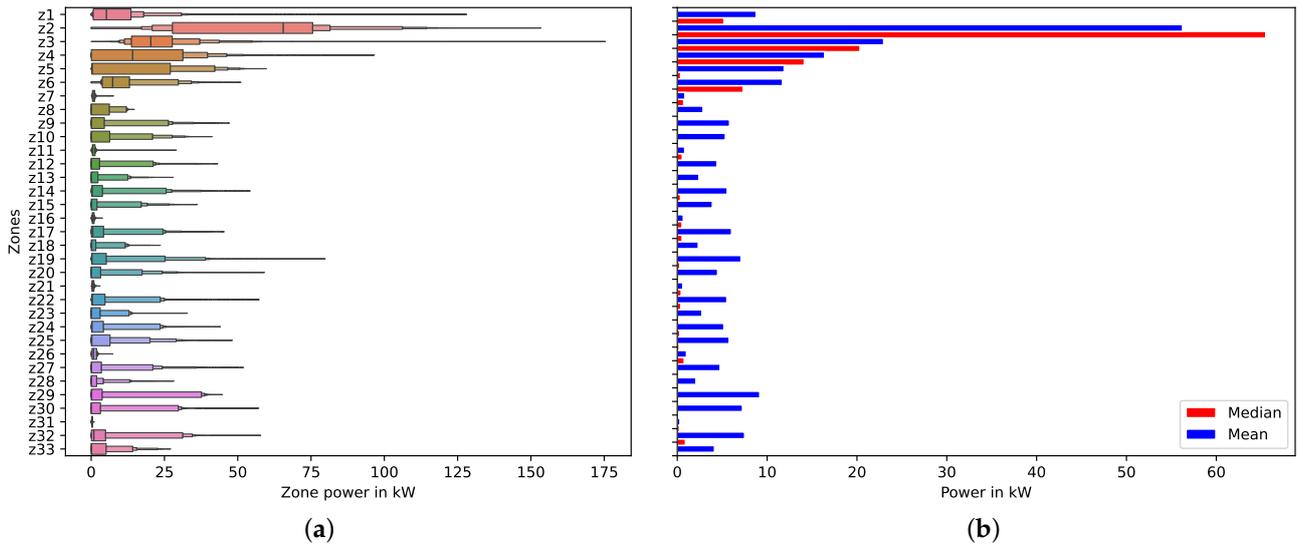


Figure 5. (a) Letter-value plot of the thermal zones. (b) Plot of the mean and median values of each zone.

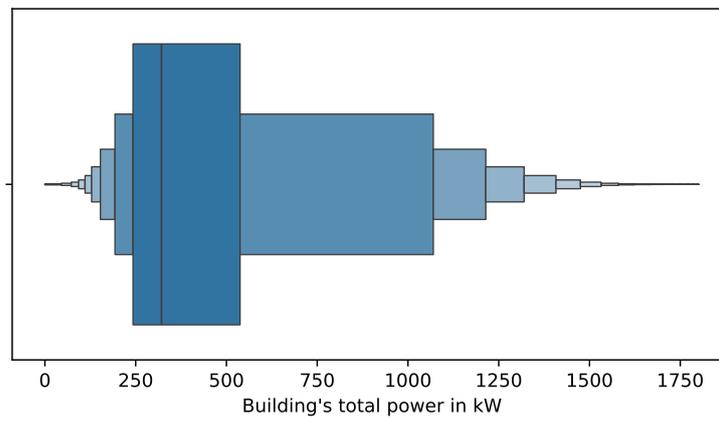


Figure 6. Letter-value plot of the building's total power.

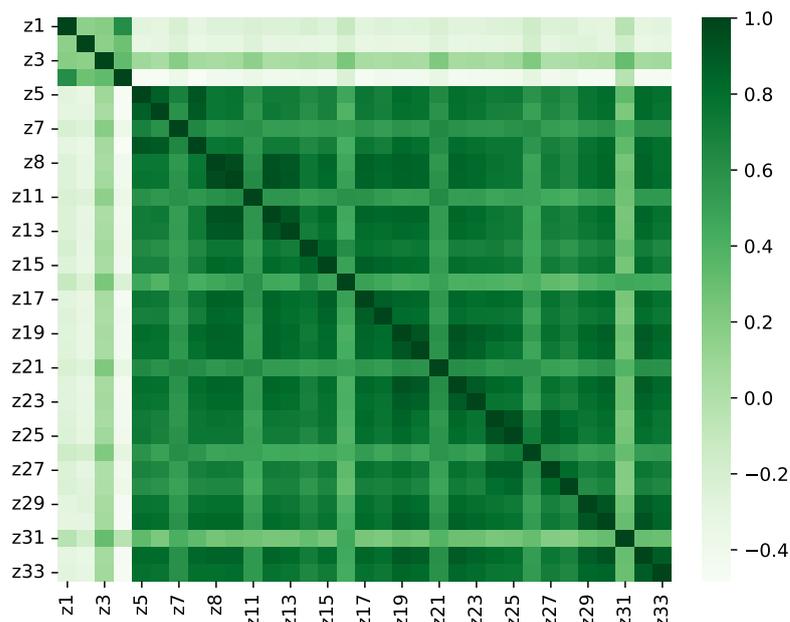


Figure 7. Correlation plot of the power between zones.

3.4. Predictions Evaluation

The performance of each model was evaluated by the following metrics: mean absolute error in kW (MAE), mean squared error in kW (MSE), R-squared in percentage (R^2), and the coefficient of variation of the root mean squared error in percentage (CV(RMSE)). These are also the most commonly used metrics in building power prediction tasks.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (13)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (14)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (15)$$

$$\text{CV(RMSE)} = \frac{\text{RMSE}}{\bar{y}}, \quad (16)$$

where y_i denotes the actual values, \bar{y} is the mean of the y_i values, \hat{y}_i represents the predicted values, and n presents the observed samples. The RMSE is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (17)$$

The lower values of MAE and MSE [44] indicate a better prediction performance. The R^2 metric, when it is closer to 100%, represents a better model fitting. Moreover, as stated by Chicco et al. [45], R-squared is more informative in regression analysis evaluation than other metrics. Furthermore, the CV(RMSE) metric, when it shows values lower than 25%, represents a model with satisfactory prediction according to ASHRAE Guideline 14 [46].

3.5. Software Environment and Experimental Setup

The research experiments were conducted in a Google Colab Platform with a Python 3 Google Compute Engine backend, and it utilized a GPU NVIDIA T4 with RAM 15.0 GB system with RAM 12.7 GB and an available disk space of 78.2 GB. The 3.10 Python programming language was used for code development, with the incorporation of the open-source software library Tensorflow 2.15.0 and the Keras 3.0.2 high-level API for training and testing the algorithms. Furthermore, the following Python libraries were used: Pandas 2.1.0 and Numpy 1.26.0 for data analysis, as well as Seaborn and Matplotlib for visualizing the exploratory analysis and predicted results, respectively. Additionally, the NetworkX Python library was used for graph analysis and visualization.

The architecture of our forecasting model consists of one graph convolutional layer, an activation function ReLU, one LSTM layer with dropout, and one dense layer as the output. The datasets were divided into 70%, 10%, and 20% for training, validation, and testing, respectively. The root mean squared propagation (RMSprop) optimizer was utilized to update the prediction model parameters with a learning rate of 0.001, while mean absolute error was chosen as a loss function. The batch size was set to 256, and the historical data were set to 10. Finally, an early-stopping regularization technique with a patience parameter was used during the training process to prevent model overfitting. The patience parameter value signifies the number of iterations at which no further enhancement in the prediction performance is observed, thus leading to the termination of the training process. Multiple trial intervals were experimented with, ranging from 5 to 10, in order to determine the best fit for the patience parameter, which was determined to be 8.

3.6. Experimental Results

In this section, the different adjacency matrices and the results of the multistep prediction experiments are presented. For the correlation-based adjacency matrix computation

(AMC) methods, the value of the threshold σ was selected as 0.7 so as to compare the three methods on the same basis. Figure 8 illustrates the adjacency matrices. As is presented in Figure 8a,b, the PCC and PCCA methods produced the same graph with 220 edges. This was due to the fact that the negative values of the correlations in the examined dataset when they become positive did not affect the number of values that were above the threshold value, as illustrated in the distribution plots of Figure 9. Additionally, the PCCS method produced a graph of 289 edges, as is depicted in Figure 8c.

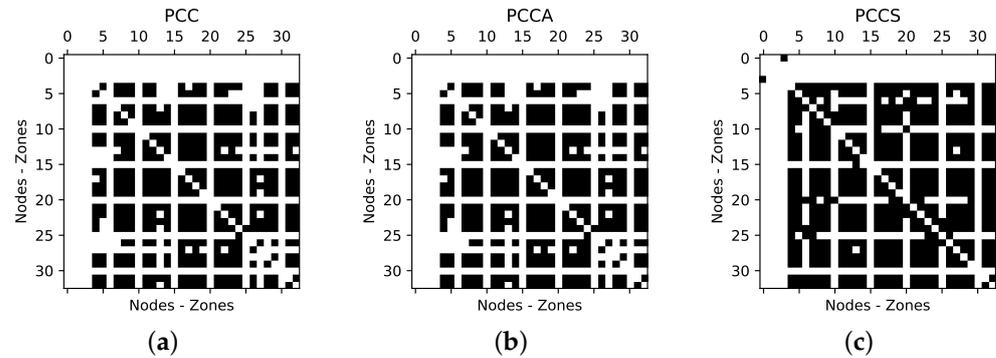


Figure 8. (a–c): The generated adjacency matrix for each computation method. The edges between the nodes and zones are presented with a black color, e.g., for the PCC method, the nodes with values lower than the threshold were not interconnected with the other nodes due to low or negative correlations.

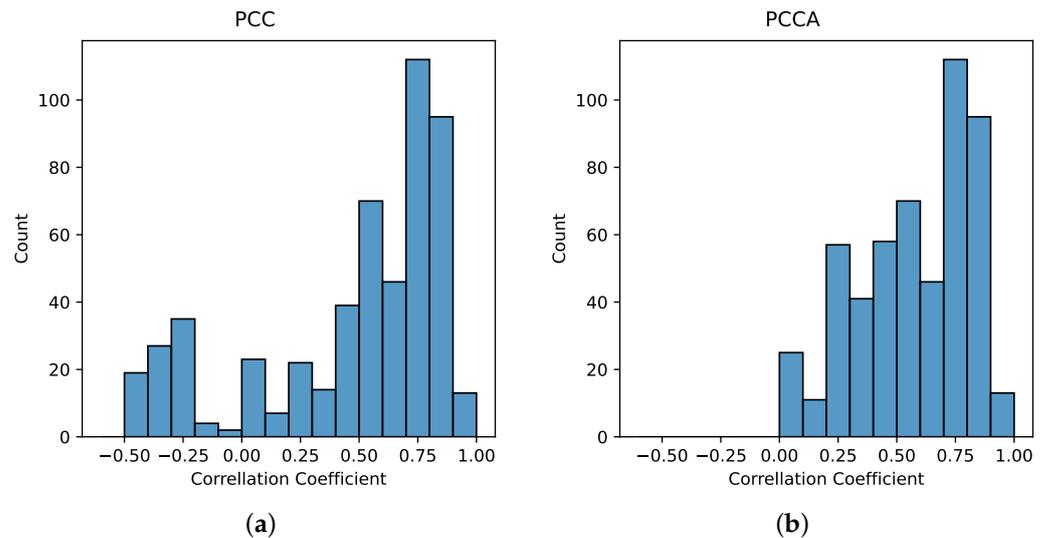


Figure 9. The correlation value distribution plots of the PCC (a) and PCCA (b) methods.

On the other hand, for the distance-based AMC methods, the value of threshold σ was chosen as the mean value of the Euclidean distances between the zones. Additionally, the θ threshold value was selected as 10. Hence, the aim was to connect only the nodes where the distances of the corresponding zones were smaller than the average distances. The adjacency matrices that were made by the EDS, EDT, and EDGK methods are shown in Figures 10a, 10b, and 10c, respectively.

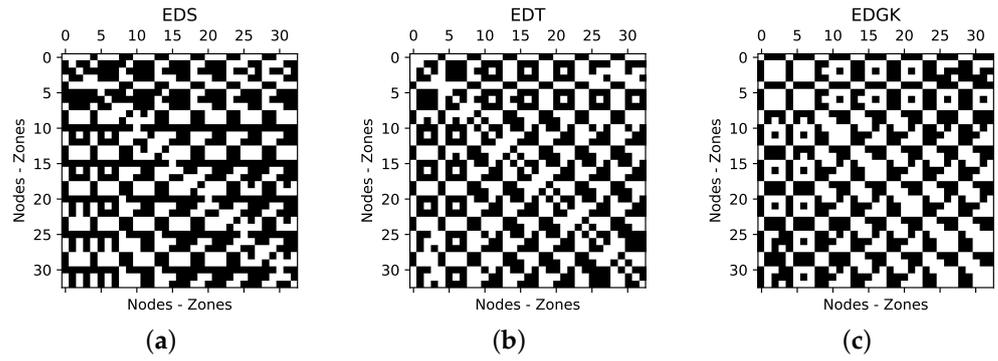


Figure 10. The generated adjacency matrices for the distance-based computation methods (a–c).

The above methods were evaluated at the zone and building levels for a prediction horizon of 5, 10, and 15 timesteps ahead. Table 1 presents the mean power prediction errors in the zones for the three prediction horizons across all of the forecasting methods. The metrics include the mean absolute error (MAE) and the mean squared error (MSE), which are measured in kilowatts (kW), as well as the coefficient of determination (R^2), which is expressed as a percentage. Furthermore, Figure 11 shows the alternation of the metrics across the prediction horizons on the three randomly selected zones, and Figure 12 presents a plot of a random zone for the predicted and the actual values.

Table 1. The zone-level mean prediction performance of the different models.

Models		MAE (kW)			MSE (kW)			$R^2\%$		
		5	10	15	Prediction Horizon			5	10	15
GCN-LSTM	PCCA	0.4410	0.6125	0.6180	5.0451	6.9455	7.0770	95.9628	91.5653	91.5658
	PCCS	0.5328	0.8042	0.8613	4.2162	8.5785	9.3781	94.8608	89.4263	88.6511
	EDS	0.3885	0.5087	0.5365	3.2260	4.9217	5.2070	97.0191	94.3589	94.1555
	EDT	0.3629	0.5027	0.5149	2.7790	4.2133	4.4246	98.0799	95.1549	95.0108
	EDGK	0.4417	0.5621	0.5926	4.8646	6.3391	6.6114	96.3522	93.3988	93.4145
Baselines	LSTM	1.3350	1.3766	1.7868	14.5464	15.5639	22.6170	84.4698	84.5264	75.6387
	CNN	1.4175	1.4626	1.6982	13.7222	16.1925	18.0752	82.2609	81.7772	78.0224
	MLP	0.8228	1.0712	1.5381	7.5929	10.0303	18.7174	93.9786	90.3119	82.1446
	HA	1.3698	1.8068	2.1499	16.6891	23.6742	30.7696	84.7380	79.9130	72.0788

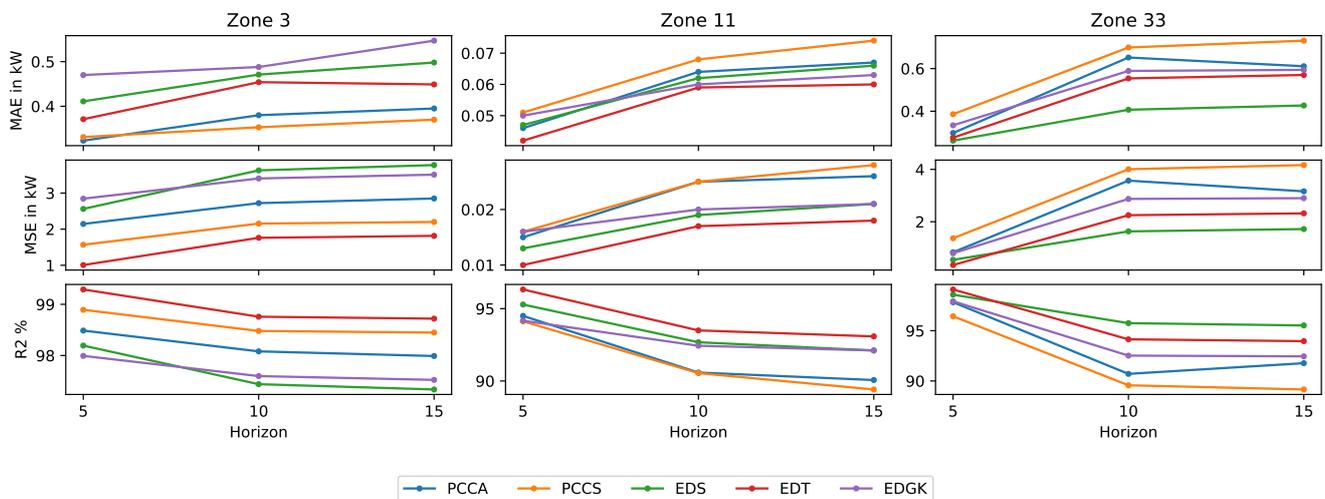


Figure 11. Prediction performance comparison of Zones 3, 11, and 33 for metrics MAE, MSE, and R^2 .

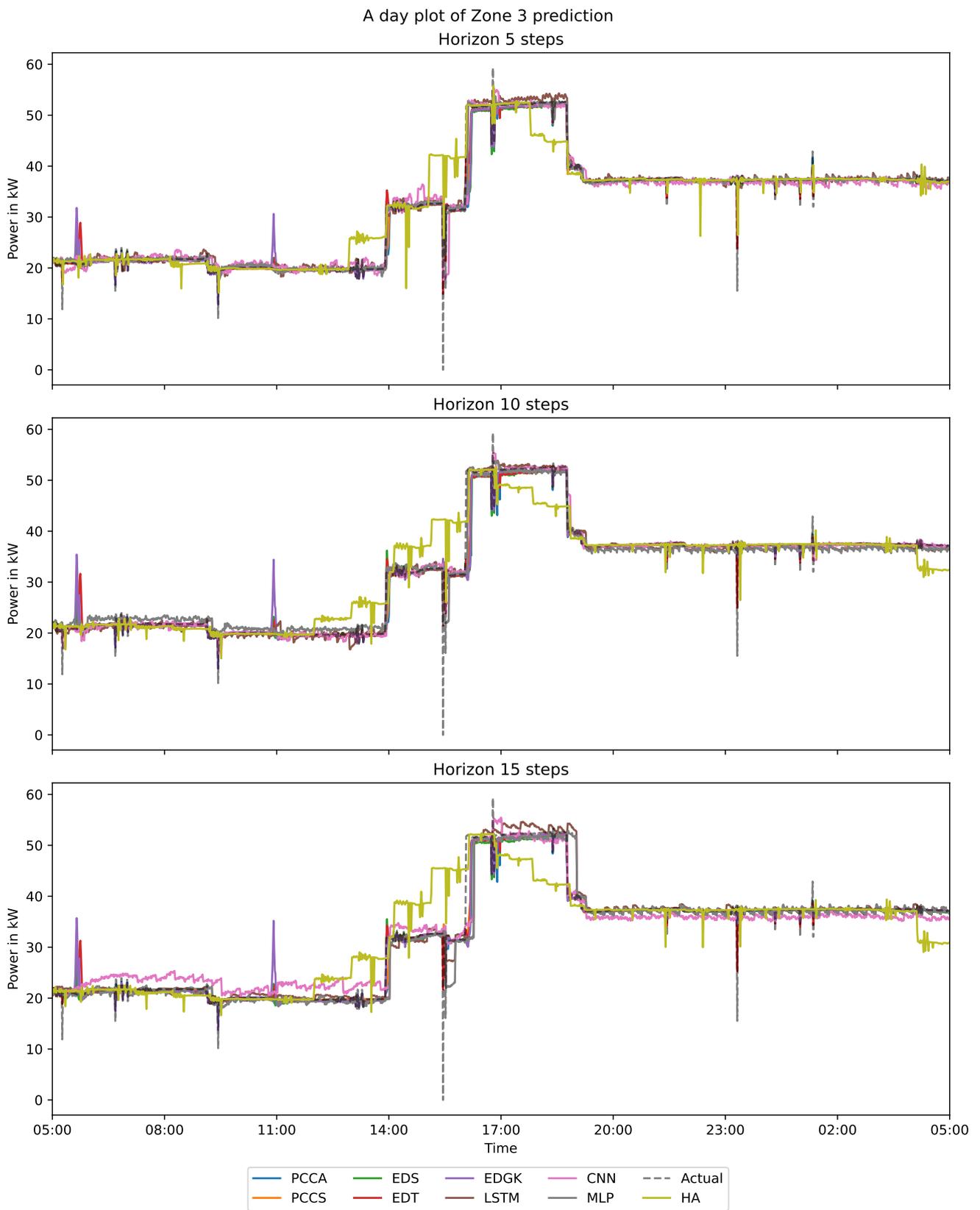


Figure 12. Plot of Zone 3’s actual and predicted power values grouped per prediction horizon.

In contrast, Table 2 presents the building-level power consumption forecasted performance metrics for the forecasting models across different prediction horizons (5, 10, and 15), and the metrics are plotted in Figure 13. The metrics include the mean absolute error (MAE), the mean squared error (MSE), and the coefficient of variation of root mean squared error (CV(RMSE)). Additionally, Figure 14 depicts the predicted and actual values of the building’s total consumed power.

Table 2. Building-level prediction performance of the different models.

Models		MAE (kW)			MSE (kW)			CV(RMSE) %		
		5	10	15	Prediction Horizon			5	10	15
GCN-LSTM	PCCA	9.2108	11.4480	11.6217	462.3184	856.3270	908.8390	9.9176	13.4975	13.9052
	PCCS	12.7969	20.2977	22.3180	743.0862	2050.3345	2402.7141	12.5734	20.8856	22.6092
	EDS	9.0959	12.3476	13.0835	292.8690	517.3485	543.3596	7.8935	10.4912	10.7517
	EDT	8.2379	11.1851	11.6887	157.2238	293.3612	312.0810	5.7835	7.9002	8.1483
	EDGK	10.0197	12.7862	13.6889	269.7044	438.5116	468.4237	7.5749	9.6588	9.9828
Baselines	LSTM	22.6126	24.1234	31.0320	1590.3106	1810.7129	2977.6661	18.3940	19.6272	25.1694
	CNN	19.0131	22.1651	24.6825	821.2873	1243.6567	1739.2184	13.2185	16.2661	19.2358
	MLP	12.7255	15.5074	25.6344	356.3368	545.5808	1919.0386	8.7069	10.7737	20.2058
	HA	31.8453	39.3830	48.2358	3711.3302	4006.8004	6630.6853	28.0995	29.1967	37.5590

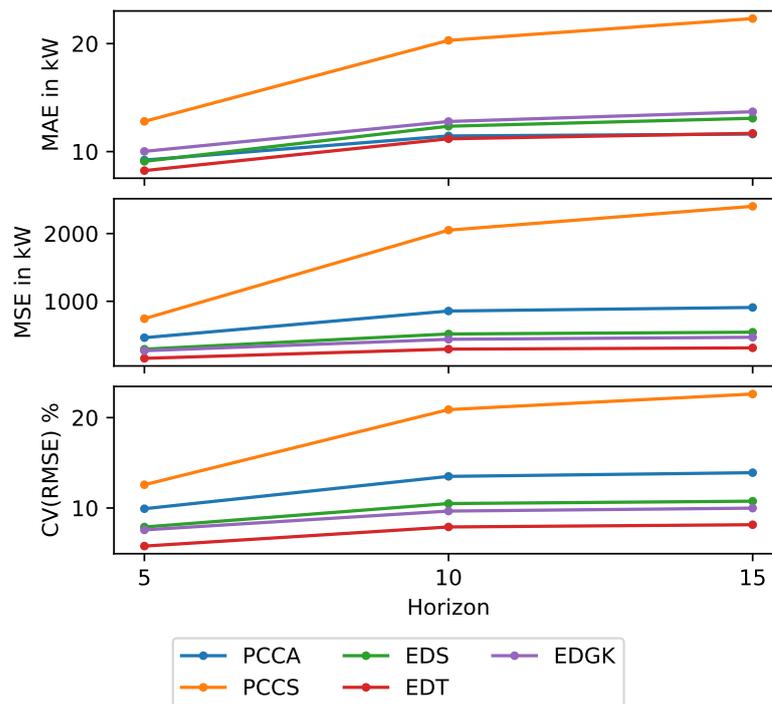


Figure 13. Prediction performance comparison of the total power for the MAE, MSE, and CV(RMSE) metrics.

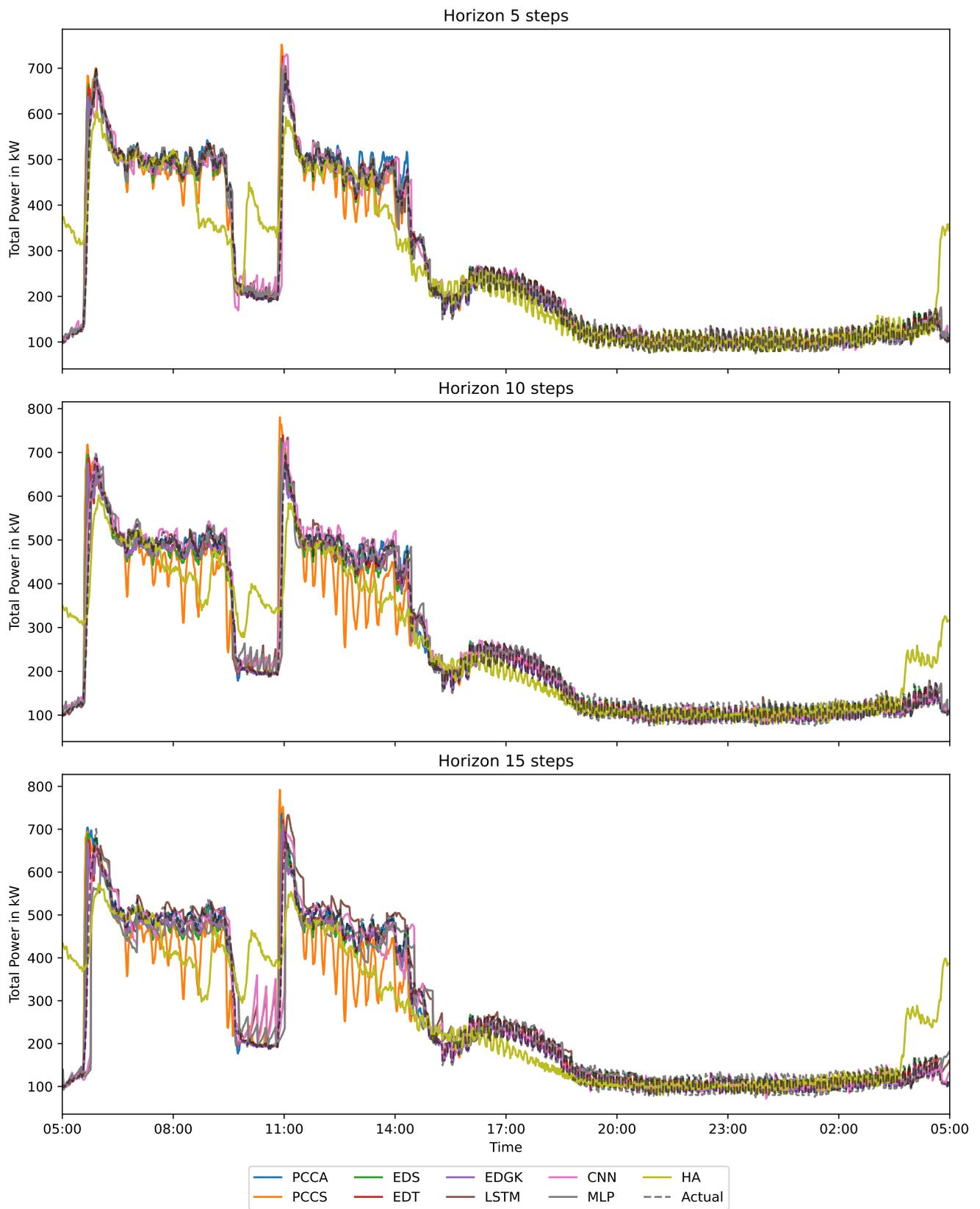


Figure 14. Plot of the building’s total actual and predicted power values grouped per time step horizon.

4. Discussion

In this study, the impact of several graph computation techniques that were applied over the GCN-LSTM model was investigated to predict the energy usage of an educational building. Among the computation models utilized, at the zone and building level, EDT consistently achieved the lowest MAE and MSE values on all horizons, thus indicating a superior prediction accuracy. PCCA and EDS also demonstrated competitive performance, with relatively low MAE and MSE values. In contrast, the LSTM, CNN, and MLP models exhibited higher MAE and MSE values, thereby suggesting less accurate predictions compared to the other models. For LSTM and CNN in particular, this prediction behavior can be explained due to the lack of exogenous features for the training of the models.

In terms of the building-level prediction metric CV(RMSE), all methods, except for HA, produced prediction values below 25%, which shows a good model fit with a more than satisfactory total power forecasting according to ASHRAE Guideline 14. For the mean zone-level estimation metric R^2 , the EDT method presented the confounding model fitting effect for the three prediction horizons in comparison with the other computation methods. The HA model performed the worst across all metrics and prediction horizons, with significantly higher MAE and MSE values, thus indicating poor predictive performance.

Additionally, as shown in Figures 12 and 14, the predictive accuracy of the various models and time horizons for power forecasting at both the zone and building levels was evident. It was apparent that the proposed models exhibited a high level of consistency with the actual curve when compared.

However, several limitations warrant consideration. First, the present study examined the computation of binary adjacency matrices over a simplified GCN-LSTM model for spatio-temporal building power prediction. This simplified model does not take into account the weights of the edges. Second, exogenous parameters, such as weather conditions and building usage in terms of occupancy, were not taken into consideration during model training. Third, the finer sampling resolution (one minute) of the utilized dataset and the educational use of the building, which presents a repetitive load shape, might lead to better prediction results.

These findings suggest certain recommendations for future research, such as the application of the proposed methodologies over a variation of the GNN forecasting models proposed by other academics for time series estimation, like graph attention neural networks, GraphSAGE, etc. It may be of scholarly interest to explore the application of these methods on datasets characterized by varying levels of granularity. Moreover, future studies should explore the analysis of how the choice of threshold value affects the prediction outcomes. Additionally, the proposed methodologies can be implemented on a dataset representing a composite building type, such as residential apartments, or on a group of buildings forming a microgrid.

In summary, this study contributes valuable insights into the efficacy of graph neural networks and the several graph representation methodologies that were used for the building energy prediction.

5. Conclusions

In this paper, a comprehensive study was carried out to predict the energy consumption of a multistory, multizone building for multiple prediction horizons. For this purpose, several adjacency matrix computation methods were proposed over a GNN-RNN deep learning model, and they were compared with the baseline machine and deep learning models such as LSTM, MLP, CNN, and a statistical model. The proposed adjacency matrix computation methods were divided into correlation-based and distance-based methods. More specifically, the methods that are computed with the Pearson correlation matrix as a basis are the Pearson Correlation Coefficient (PCC), the Absolute Pearson Correlation Coefficient (PCCA), and the Pearson Correlation Coefficient Scaled (PCCS). Furthermore, the methods that have Euclidean distance among zones as their basis are the Euclidean

Distance Scaled (EDS), the Euclidean Distance with Threshold (EDT), and the Euclidean Distance with Gaussian Kernel (EDGK).

Based on the results, all of the proposed computation methods yielded very high performance, good model fit, and improved energy consumption prediction in contrast with the baseline models. It is worth noting that the EDT method outperformed the other computation methods in predicting the power at the zone and building level, and the EDS, PCCA, EDGK, and PCCS methods followed in ascending order in terms of prediction accuracy.

Overall, the present employed work will be useful for building managers and engineers in terms of energy efficiency, financial savings, and occupant comfort. Also, this study will be helpful for electrical grid managers for resource planning and grid stability. The findings of this study could be applied to other building sectors, such as industrial and residential, for obtaining more accurate conclusions.

Author Contributions: Conceptualization, G.V. and A.C.; methodology, G.V.; software, G.V. and V.L.; validation, G.V., V.L., A.C., D.B. and T.E.K.; formal analysis, G.V., V.L. and A.C.; investigation, G.V.; resources, G.V., V.L. and A.C.; data curation, G.V.; writing—original draft preparation, G.V., V.L. and A.C.; writing—review and editing, G.V., V.L., A.C., D.B. and T.E.K.; visualization, G.V.; supervision, D.B. and T.E.K.; project administration, D.B. and T.E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available in a publicly accessible repository, available online <https://doi.org/10.1038/s41597-020-00582-3> (accessed on 8 January 2024), as was mentioned in Section 3.1. The dataset is licensed under a Creative Commons Attribution 4.0 International License.

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

Abbreviations

The following abbreviations are used in this manuscript:

A	Adjacency matrix
b	Biases
CNN	Convolutional neural network
E	A set of edges
ED	Euclidean distance
EDGK	Euclidean Distance with Gaussian Kernel
EDS	Euclidean Distance Scaled
EDT	Euclidean Distance with Threshold
G	Graph
GCN	Graph convolutional network
GNN	Graph neural network
HA	Historical average
LSTM	Long short-term memory
MAE	Mean absolute error
MLP	Multilayer perceptron
MSE	Mean squared error
PCC	Pearson correlation coefficient
PCCA	Absolute Pearson Correlation Coefficient
PCCS	Pearson Correlation Coefficient Scaled
R^2	R-squared
RNN	Recurrent neural network
V	A set of nodes
W	Weights

References

1. International Energy Agency. Energy System—Buildings. Available online: <https://www.iea.org/energy-system/buildings> (accessed on 18 January 2024).
2. Commission, E. Energy Performance of Buildings Directive. Available online: https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficient-buildings/energy-performance-buildings-directive_en (accessed on 18 January 2024).
3. Shahcheraghian, A.; Madani, H.; Ilinca, A. From White to Black-Box Models: A Review of Simulation Tools for Building Energy Management and Their Application in Consulting Practices. *Energies* **2024**, *17*, 376. [[CrossRef](#)]
4. Dong, B.; Li, Z.; Rahman, S.M.; Vega, R. A hybrid model approach for forecasting future residential electricity consumption. *Energy Build.* **2016**, *117*, 341–351. [[CrossRef](#)]
5. Amasyali, K.; El-Gohary, N.M. A review of data-driven building energy consumption prediction studies. *Renew. Sustain. Energy Rev.* **2018**, *81*, 1192–1205. [[CrossRef](#)]
6. Sun, Y.; Haghighat, F.; Fung, B.C. A review of the-state-of-the-art in data-driven approaches for building energy prediction. *Energy Build.* **2020**, *221*, 110022. [[CrossRef](#)]
7. Vontzos, G.; Laitzos, V.; Bargiotas, D. Data-Driven Airport Multi-Step Very Short-Term Load Forecasting. In Proceedings of the 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), Volos, Greece, 10–12 July 2023; pp. 1–6. [[CrossRef](#)]
8. Ahn, Y.; Kim, B.S. Prediction of building power consumption using transfer learning-based reference building and simulation dataset. *Energy Build.* **2022**, *258*, 111717. [[CrossRef](#)]
9. Skomski, E.; Lee, J.Y.; Kim, W.; Chandan, V.; Katipamula, S.; Hutchinson, B. Sequence-to-sequence neural networks for short-term electrical load forecasting in commercial office buildings. *Energy Build.* **2020**, *226*, 110350. [[CrossRef](#)]
10. Gao, Y.; Ruan, Y.; Fang, C.; Yin, S. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. *Energy Build.* **2020**, *223*, 110156. [[CrossRef](#)]
11. Lei, L.; Chen, W.; Wu, B.; Chen, C.; Liu, W. A building energy consumption prediction model based on rough set theory and deep learning algorithms. *Energy Build.* **2021**, *240*, 110886. [[CrossRef](#)]
12. Somu, N.; MR, G.R.; Ramamritham, K. A deep learning framework for building energy consumption forecast. *Renew. Sustain. Energy Rev.* **2021**, *137*, 110591. [[CrossRef](#)]
13. Fan, C.; Wang, J.; Gang, W.; Li, S. Assessment of deep recurrent neural network-based strategies for short-term building energy predictions. *Appl. Energy* **2019**, *236*, 700–710. [[CrossRef](#)]
14. Irankhah, A.; Yaghmaee, M.H.; Ershadi-Nasab, S. Optimized short-term load forecasting in residential buildings based on deep learning methods for different time horizons. *J. Build. Eng.* **2024**, *84*, 108505. [[CrossRef](#)]
15. Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A. Minutely active power forecasting models using neural networks. *Sustainability* **2020**, *12*, 3177. [[CrossRef](#)]
16. Stergiou, K.; Karakasidis, T.E. Application of deep learning and chaos theory for load forecasting in Greece. *Neural Comput. Appl.* **2021**, *33*, 16713–16731. [[CrossRef](#)]
17. Laitzos, V.; Vontzos, G.; Bargiotas, D.; Daskalopulu, A.; Tsoukalas, L.H. Enhanced Automated Deep Learning Application for Short-Term Load Forecasting. *Mathematics* **2023**, *11*, 2912. [[CrossRef](#)]
18. Sajjad, M.; Khan, Z.A.; Ullah, A.; Hussain, T.; Ullah, W.; Lee, M.Y.; Baik, S.W. A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. *IEEE Access* **2020**, *8*, 143759–143768. [[CrossRef](#)]
19. Li, G.; Zhao, X.; Fan, C.; Fang, X.; Li, F.; Wu, Y. Assessment of long short-term memory and its modifications for enhanced short-term building energy predictions. *J. Build. Eng.* **2021**, *43*, 103182. [[CrossRef](#)]
20. Liang, F.; Qian, C.; Yu, W.; Griffith, D.; Golmie, N. Survey of graph neural networks and applications. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9261537. [[CrossRef](#)]
21. Reiser, P.; Neubert, M.; Eberhard, A.; Torresi, L.; Zhou, C.; Shao, C.; Metni, H.; van Hoesel, C.; Schopmans, H.; Sommer, T.; et al. Graph neural networks for materials science and chemistry. *Commun. Mater.* **2022**, *3*, 93. [[CrossRef](#)] [[PubMed](#)]
22. Khemani, B.; Patil, S.; Kotecha, K.; Tanwar, S. A review of graph neural networks: Concepts, architectures, techniques, challenges, datasets, applications, and future directions. *J. Big Data* **2024**, *11*, 18. [[CrossRef](#)]
23. Hu, Y.; Cheng, X.; Wang, S.; Chen, J.; Zhao, T.; Dai, E. Times series forecasting for urban building energy consumption based on graph convolutional network. *Appl. Energy* **2022**, *307*, 118231. [[CrossRef](#)]
24. Guo, J.; Han, M.; Zhan, G.; Liu, S. A Spatio-Temporal Deep Learning Network for the Short-Term Energy Consumption Prediction of Multiple Nodes in Manufacturing Systems. *Processes* **2022**, *10*, 476. [[CrossRef](#)]
25. Lu, J.; Zhang, C.; Li, J.; Zhao, Y.; Qiu, W.; Li, T.; Zhou, K.; He, J. Graph convolutional networks-based method for estimating design loads of complex buildings in the preliminary design stage. *Appl. Energy* **2022**, *322*, 119478. [[CrossRef](#)]
26. Jia, Y.; Wang, J.; Hosseini, M.R.; Shou, W.; Wu, P.; Chao, M. Temporal Graph Attention Network for Building Thermal Load Prediction. *Energy Build.* **2023**, 113507. [[CrossRef](#)]
27. Zhang, J.; Xiao, F.; Li, A.; Ma, T.; Xu, K.; Zhang, H.; Yan, R.; Fang, X.; Li, Y.; Wang, D. Graph neural network-based spatio-temporal indoor environment prediction and optimal control for central air-conditioning systems. *Build. Environ.* **2023**, *242*, 110600. [[CrossRef](#)]
28. Xie, Y.; Stravrovavdis, S. Generating Occupancy Profiles for Building Simulations Using a Hybrid GNN and LSTM Framework. *Energies* **2023**, *16*, 4638. [[CrossRef](#)]

29. Li, A.; Fan, C.; Xiao, F.; Chen, Z. Distance measures in building informatics: An in-depth assessment through typical tasks in building energy management. *Energy Build.* **2022**, *258*, 111817. [CrossRef]
30. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing. Springer Topics in Signal Processing*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 258, pp. 1–4. [CrossRef]
31. Shuman, D.I.; Narang, S.K.; Frossard, P.; Ortega, A.; Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.* **2013**, *30*, 83–98. [CrossRef]
32. Hamilton, W.L. *Graph Representation Learning*; Synthesis Lectures on Artificial Intelligence and Machine Learning; Springer: Cham, Switzerland, 2020; pp. 1–141. [CrossRef]
33. Kipf, T. Graph Convolutional Networks. 2016. Available online: <https://tkipf.github.io/graph-convolutional-networks/> (accessed on 18 January 2024).
34. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; Volume 9, pp. 249–256.
35. Lindemann, B.; Müller, T.; Vietz, H.; Jazdi, N.; Weyrich, M. A survey on long short-term memory networks for time series prediction. *Procedia CIRP* **2021**, *99*, 650–655. [CrossRef]
36. Xu, Z.; Zeng, W.; Chu, X.; Cao, P. Multi-aircraft trajectory collaborative prediction based on social long short-term memory network. *Aerospace* **2021**, *8*, 115. [CrossRef]
37. Banoula, M. An Overview on Multilayer Perceptron (MLP). 2023. Available online: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/multilayer-perceptron> (accessed on 18 January 2024).
38. Amidi, A.; Amidi, S. Convolutional Neural Networks Cheatsheet. 2019. Available online: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks> (accessed on 18 January 2024).
39. Pipattanasomporn, M.; Chitalia, G.; Songsiri, J.; Aswakul, C.; Pora, W.; Suwankawin, S.; Audomvongseree, K.; Hoonchareon, N. CU-BEMS, smart building electricity consumption and indoor environmental sensor datasets. *Sci. Data* **2020**, *7*, 241. [CrossRef]
40. Wang, X.; Wang, X.; Yin, X.; Li, K.; Wang, L.; Wang, R.; Song, R. Distributed LSTM-GCN based spatial-temporal indoor temperature prediction in multi-zone buildings. *IEEE Trans. Ind. Inform.* **2024**, *20*, 482–491. [CrossRef]
41. Ma, Z.; Wang, J.; Ye, S.; Wang, R.; Dong, F.; Feng, Y. Real-time indoor thermal comfort prediction in campus buildings driven by deep learning algorithms. *J. Build. Eng.* **2023**, *78*, 107603. [CrossRef]
42. Sari, M.; Berawi, M.A.; Zagloel, T.Y.; Madyaningarum, N.; Miraj, P.; Pranoto, A.R.; Susantono, B.; Woodhead, R. Machine learning-based energy use prediction for the smart building energy management system. *J. Inf. Technol. Constr. (ITcon)* **2023**, *28*, 622–645. [CrossRef]
43. Hofmann, H.; Wickham, H.; Kafadar, K. Letter-Value Plots: Boxplots for Large Data. *J. Comput. Graph. Stat.* **2017**, *26*, 469–477. [CrossRef]
44. Sammut, C.; Webb, G.I. *Encyclopedia of Machine Learning*; Springer: New York, NY, USA, 2010; pp. 652–653. [CrossRef]
45. Chicco, D.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [CrossRef]
46. Ashrae, A.G. *Guideline 14-2014: Measurement of Energy, Demand, and Water Savings*; American Society of Heating, Refrigerating, and Air Conditioning Engineers: Atlanta, GA, USA, 2014.

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.