Journal of Building Engineering 77 (2023) 107464

Contents lists available at ScienceDirect

Journal of Building Engineering

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

Adaptive thermal load prediction in residential buildings using artificial neural networks

Mohammad Hossein Fouladfar [a,b,*], Anton Soppelsa [a], Himanshu Nagpal [a], Roberto Fedrizzi [a], Giuseppe Franchini [b]

[a] EURAC Research, Institute for Renewable Energy, Viale Druso 1, 39100, Bolzano, Italy
[b] Department of Engineering and Applied Sciences, University of Bergamo, 24044, Dalmine, Italy

ARTICLE INFO

Keywords:
Thermal load prediction
Residential buildings
Artificial neural network
Machine learning
Data-analysis
Multilayer perceptron

ABSTRACT

Accurate prediction of thermal load in buildings is essential for efficient energy planning. In this study, we investigate the application of Artificial Neural Networks (ANNs) to predict thermal load and indoor temperature evolution in residential buildings. We propose a flexible and adaptive model that is retrained on a daily basis to deliver updated hour-by-hour predictions for the following 24 h. To strike a balance between prediction accuracy and computational efficiency, we employ various design choices, including feature selection using Pearson's correlation coefficient (PCC), dynamic architecture, down-sampling, and specific state resetting and weight initialization. The results demonstrate the efficacy of our proposed model, as indicated by low root mean square error (RMSE) and mean bias error (MBE) values. For zone temperature, the average RMSE and MBE are 0.3 °C and 0.18 °C (summer) and 0.5 °C and 0.2 °C (winter), respectively. Furthermore, the average RMSE and MBE for thermal load predictions are 12 W/m^2 and -2.4 W/m^2 (winter) and 10 W/m^2 and -0.6 W/m^2 (summer), respectively. These performance metrics establish our model as a valuable tool for optimizing heating and cooling systems, resulting in energy savings and cost reductions. Our findings emphasize the potential of ANNs for precise thermal load and indoor temperature predictions, offering practical implications for building operators, engineers, and researchers involved in energy management.

Nomenclature

T	Temperature [°C]
t	Time step [5-min]
D	Day
S	Switch [0, 1]
q_x	Number of past samples of the state variables
q_u	Number of past samples of the input variables
W	Weight
$\boldsymbol{u}$	Input vector
$\boldsymbol{y}$	Output vector

* Corresponding author. EURAC Research, Institute for Renewable Energy, Viale Druso 1, 39100, Bolzano, Italy.
E-mail address: mohammadhossein.fouladfar@eurac.edu (M.H. Fouladfar).

https://doi.org/10.1016/j.jobe.2023.107464
Received 19 April 2023; Received in revised form 26 July 2023; Accepted 29 July 2023
Available online 29 July 2023

x	State vector
N	Number of samples
n	Vector size of the state
p	Vector size of the output
m	Vector size of the input

Acronyms

ANN	Artificial Neural Network
ARX	Autoregressive with exogenous input
CKPI	Computed key performance indicator
GU	Glorot uniform
KPI	Key performance indicator
LSTM	Long short-term memory
MBE	Mean bias error
ML	Machine learning
MSE	Mean squared error
MLP	Multilayer perceptron
MP	Mean period
NMBE	Normalized mean bias error
PCC	Pearson's correlation coefficient
RMSE	Root mean squared error
RNN	Recurrent neural network
SVM	Support vector machine
XGBoost	Extreme gradient boosting

Greek symbols

φ	Global horizontal irradiation [kJ/h/m^2]
ξ	Extended state vector
θ	Internal gain [W]
$\mathbb{E}$	Expected value
μ	Mean value
σ	Standard deviation
δ	Float number [0, 1]

Subscripts

a	Ambient temperature
z	Zone
w	Week
s	Solar
t	Time

1. Introduction

In 2021, the majority of energy consumed by households in the European Union was allocated to heating their homes, representing 64.4% of the total. Additionally, water heating accounted for 14.5% of energy usage, while space cooling constituted a mere 0.5% [1]. Thermal load refers to the amount of heat energy that needs to be either added or removed from a space or system to achieve the desired temperature conditions. It can represent both the heating load and cooling load. Given that buildings use significant amounts of energy and produce considerable carbon dioxide, accurate thermal load forecasting is crucial for the optimal planning and operation of the energy systems. It helps to reduce energy waste, enhance occupant comfort, and extend the lifespan of building equipment. This type of forecasting is widely utilized in the controls of heating, ventilating, and air conditioning (HVAC) systems [2–7], heat pumps (HPs) [8–10], and thermal energy storage systems [11–13]. Load forecasting is typically implemented through various modeling techniques [14]. The choice of modeling approach depends on the complexity, dynamics, and nonlinearity of building energy systems, whether in a single building or multiple buildings. The existing approaches in the field of building energy systems can be broadly classified into three main categories [15–17]:

Firstly, there are *White-box models,* which are solely based on physics. These models can be challenging and time-consuming to develop, as they necessitate a thorough understanding of the building's physical parameters and the equations to be specified. These models can be slow to execute in simulations due to their high level of complexity. However, they offer the advantage of simulating an as-yet unbuilt building, given that all its physical properties are known [18]. Examples of well-established software based on the white-box approach are EnergyPlus, ESP-r, TRNSYS, Ecotect, and Grasshopper, which have been extensively used to assess energy use and establish building control and operation plans [19,20].

Secondly, there are *Black-box models*, also known as (fully) data-driven models. Instead of physics, these models use historical data to establish a relationship between the input and output features. The essential data are already accessible, making the calibration straightforward and quick. In this case, the model generally uses statistical approaches to create a link between building energy usage and operating data [19,21]. For training these models, on-site observations over an extended period are necessary to anticipate building functioning under various scenarios. Artificial Neural Networks (ANN), Support Vector Machines (SVM) [22], and Statistical Regression are the commonly applied approaches in black-box models [20,23].

Lastly, we have *Gray-box models,* which blend qualitative knowledge with quantitative data. The optimum process model is selected using all the currently available information. The gray-box technique offers the advantage of leveraging prior knowledge and the data already accessible, as opposed to the black-box and white-box models [24–26].

Having a one-size-fits-all model for all types of buildings is unrealistic as every building is unique and requires its specific set of parameters. This is where the black-box model comes in handy, as it can be applied to a wide range of buildings, regardless of their specific characteristics, making it a generic solution [27]. Also, the widespread implementation of automated control systems and the Internet of Things in buildings has resulted in an abundance of data from measuring sensors, providing ample opportunities for developing black box models using Machine Learning (ML) algorithms, including linear regression, autoregressive exogenous (ARX), SVM, extreme gradient boosting (XGBoost), and ANN models [28,29].

Within the black-box category, choosing the appropriate algorithm is a challenging endeavor, with "*Is there a stronger method than others?*" as a frequently asked question. Linear regression [30–32], ARX model [33–36], and more complicated methods like SVM [37–39] and XGBoost [40–43] are only a few statistical models and techniques used to predict future building loads.

The other cutting-edge data-driven solution is the use of ANN. These networks are popular due to their striking features, such as adaptive learning, self-organization, real-time operations, generalization, stability and flexibility, and parallel processing. ANN has been widely recognized in the building energy domain as a promising approach due to its superior learning abilities and computational advantages. The Multilayer Perceptron (MLP) is a popular type of ANN that is made up of multiple layers of interconnected neurons. Unlike linear regression, which can only model linear relationships, MLP can handle both linear and nonlinear relationships by adding more hidden layers to the network. This allows MLPs to better capture the complex relationships in the data, leading to improved forecasting performance.

In addition to MLPs, the ANN domain includes various advanced ANN architectures such as Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) networks. While RNN is widely used and effective for time series forecasting, it is susceptible to the risk of exploding and vanishing gradients and may have a limited capacity for storing features for long sequences [44]. LSTM overcomes these issues and has a more sophisticated structure than traditional RNNs, which allows it to store and access information over an extended period of time, making it more suitable for time series forecasting tasks. Some studies have highlighted the superior performance of LSTM over simple ANN for predicting a building's energy consumption, but this advantage comes at the cost of the increased computational burden [45]. Also, LSTMs require a large amount of training data to produce accurate results. If the training data is limited or noisy, LSTMs may struggle to learn meaningful patterns. Therefore, the choice of ANN depends on the specific needs of the application [46,47].

Creating an appropriate ANN model is a complex process that requires careful consideration of several critical steps. The early step involves selecting the most suitable features for the model, and there are various extraction algorithms available to achieve this, including the filter approach, wrapping method, embedding method [48–50], and so on. Choosing the right inputs is also a crucial aspect of feature selection, as having too many inputs can make the model overly complex and result in increased training time [51]. On the other hand, having too few inputs may lead to a model with inadequate performance. Therefore, it is essential to find a balance between the number of inputs and the model's performance. Having access to the inputs is also another aspect of feature selection that should be considered. Temporal features that are easily accessible, such as time of day, day of the week, and holidays, can be utilized in data-driven model training to improve the model's accuracy [30]. For example, in a predictive model for building energy consumption, the time of day, day of the week, and holidays can be useful in predicting peak demand periods. While these temporal features can be easy to obtain and process, there may be cases where additional sensors or algorithms are necessary to extract essential features for the model. These sensors and algorithms may come at an additional cost and require more time and effort to implement. For example, utilizing occupancy as a feature for model training can be beneficial. Alternative algorithms, room occupancy detectors, passive infrared sensors [52], or CO_2 sensors [53–55] can also provide valuable information about occupancy. Nonetheless, these methods can be expensive for the end user. Ideally, information should come from sensors that would be installed anyway, such as indoor temperature measurement and fluid temperatures in heating and cooling system points.

This study aims to introduce a straightforward MLP model with a single hidden layer as an approach for predicting the thermal load of buildings. The objective was to develop a model that is fast, cost-effective, and adaptable, capable of accurate predictions even with limited data. To achieve this, complex data-driven models were deliberately avoided in favor of simpler alternatives. Instead, the study focused on selecting easy and straightforward inputs that could be universally applied to all building types without requiring detailed information about their physical characteristics. The chosen inputs consisted of commonly embedded sensors in buildings, eliminating the need for additional sensors. The MLP model was specifically designed for a controller based on model predictive control, emphasizing the necessity of precise predictions for the building's thermal load. Overall, the proposed approach prioritized simplicity and practicality while still achieving accurate predictions.

The key contributions of this paper can be summarized as follows:

- **Versatility**: The proposed model architecture is designed to be versatile and adaptable, making it suitable for various types and sizes of buildings. Unlike conventional approaches that rely on specific assumptions about a building's characteristics or location,

the presented model utilizes a data-driven approach that collects information from the building's performance over time. This makes the model highly versatile, allowing it to be applied to any construction and serving as a valuable tool for energy planning in the building sector.

- **Adaptivity**: The proposed model is adaptive, meaning it can be continuously trained and updated with newly available data. This enables ongoing improvement of forecasts and the delivery of more accurate results. Moreover, the model can provide continuous simulations and forecasts on a 24-h horizon, allowing it to dynamically respond to changes.
- **Advanced Techniques**: The model incorporates cutting-edge techniques to enhance its performance and efficiency, including dedicated internal states reset function and specialized weight initialization functions. These techniques enable the model to generate more precise predictions while minimizing computational resource requirements.

2. Methodology

In this section, the detailed procedure of constructing a self-adaptive ANN model for accurately predicting thermal loads in buildings will be explored. Subsequently, the model's effectiveness will be showcased using a simulated building, emphasizing the potential of this innovative approach.

Before diving into the basic details, we recall how the ANN training process works; ANNs are fed with several input and output data pairs corresponding to the input features and output targets. The ANN architecture has a set of trainable parameters which are continuously adjusted in the training phase to minimize/maximize a particular performance indicator, also known as the loss function. In our case, features and targets are snapshots of measured physical quantities observed over a certain period of time. This training process enables the ANN to learn and improve its predictions continuously. The end result is a highly adaptable and versatile model that can be used for energy planning and management in the building.

2.1. Case study

The dataset used in our study was generated by simulating the real building in TRNSYS and saving the numerical data in a Comma-Separated Values (CSV) file. The building stratigraphy was already known from the EnergyMatching project (European Union's Horizon 2020 research and innovation program, grant agreement N° 768766). The TRNSYS model was utilized to generate a meticulously prepared dataset that accurately represents the behavior of an existing building subjected to climate conditions typical of

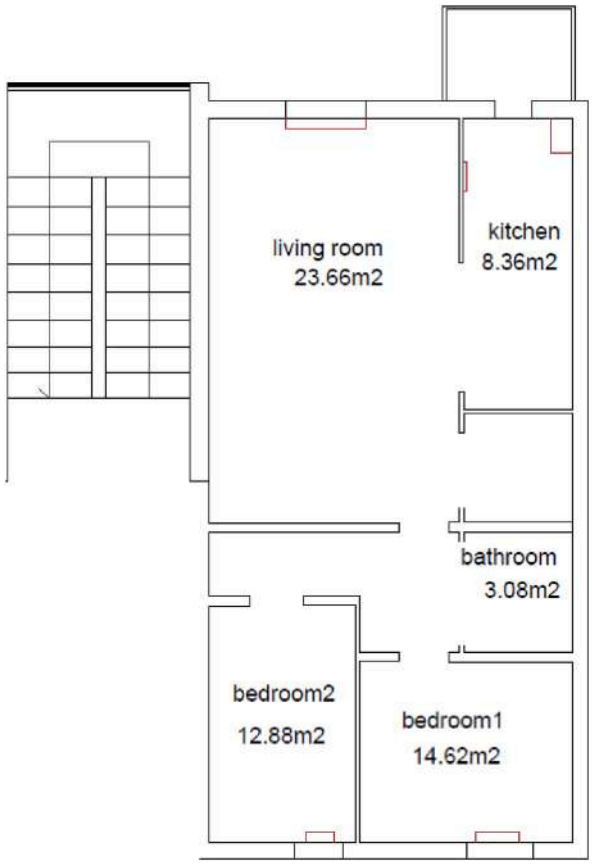

Fig. 1. The layout of the building in the case study.

Bolzano, Italy [56].

The dataset encompasses all aspects of the building, with a particular emphasis on the HVAC control system. It consists of 65 columns that represent the most crucial information related to the building's thermal behavior and energy consumption. In terms of size, the dataset contains a significant amount of information due to the one-year simulation duration and the 5-min sampling interval.

The building, as depicted in Fig. 1, is a small apartment with high insulation levels and a floor area of 62.6 m^2. The building's annual energy consumption stands at around 50 kWh/m^2. For space heating, all rooms except the bathroom and kitchen are equipped with a radiant floor heating and cooling system, each with its own dedicated thermostat for individual zones.

2.2. Feature selection

Selecting appropriate input features is fundamental to building an ANN. The process of choosing the right features is a challenging task and requires a basic knowledge of the parameters capable of influencing system. A general principle is to pick the number of features that can adequately model the system dynamics without putting excessive computational burden. To this end, some statistical-based feature selection methods evaluate the relation between input (features) and output (target) variables. These techniques attempt to concentrate on reducing redundant or unhelpful variables and selecting the most meaningful features. The relationship between the input and target variables depends on the variable types (numerical/categorical), leading to different statistical assessment (regression/classification) methods. In this case, the types of variables we encounter are numerical; therefore, we resort to calculating Pearson's correlation coefficient (PCC) to assess the input and target variable relationship. PCC measures the linear correlation ($\rho(X, Y)$) between two variables X and Y using the following expression:

$$\rho(X,Y) = \frac{\mathbb{E}[(X-\mu_X)(Y-\mu_Y)]}{\sqrt{\sigma_X \sigma_Y}} \tag{1}$$

Where $\mathbb{E}$ is the expected value, μ_X and μ_Y are the mean value of variables X and Y respectively, and σ_X and σ_Y are the standard deviation of variables X and Y, respectively. Based on the outputs of PCC, selected features are shown in Fig. 2.

2.3. Structure of neural network

In this study, we considered various ANN structures for our problem, with the goal of selecting the most appropriate one based on the specific needs of the task at hand. While we initially explored multiple structures, we ultimately found that MLP was the most effective option for our needs. While other structures may have had the potential to achieve higher accuracy or faster speeds with further optimization, the simplicity and speed of MLP were essential for meeting our specific requirements.

The Multilayer Perceptron (MLP) is one of the most used and versatile ANN architectures. MLPs are widely regarded as a powerful tool for solving complex nonlinear problems, including classification and prediction tasks, due to their ability to learn and represent complex nonlinear relationships between input and output data. One of the key advantages of MLPs is their flexibility and adaptability. They can be trained on a wide range of input and output data types, including continuous, discrete, and binary data, and can be used for both supervised and unsupervised learning [57]. Another advantage of MLPs is their ability to incorporate prior knowledge and expert input into the learning process. This makes it possible to tailor the network architecture and learning algorithm to specific problem domains and to optimize the network's performance based on domain-specific knowledge [58].

The MLP model that was chosen for the study has six input features, a hidden layer with 16 neurons, and only one output feature. In order to address the temporal dependence of the problem at hand, it was deemed necessary to include not only the parameters with the highest correlation but also the day of the week (D_w) and time of the day (t) as input features. The inclusion of D_w and t as input features in our MLP model are significant in capturing the cyclic and diurnal patterns in thermal load behavior. D_w captures differences in thermal load across weekdays and weekends, while t accounts for variations throughout the day due to factors like outdoor tem-

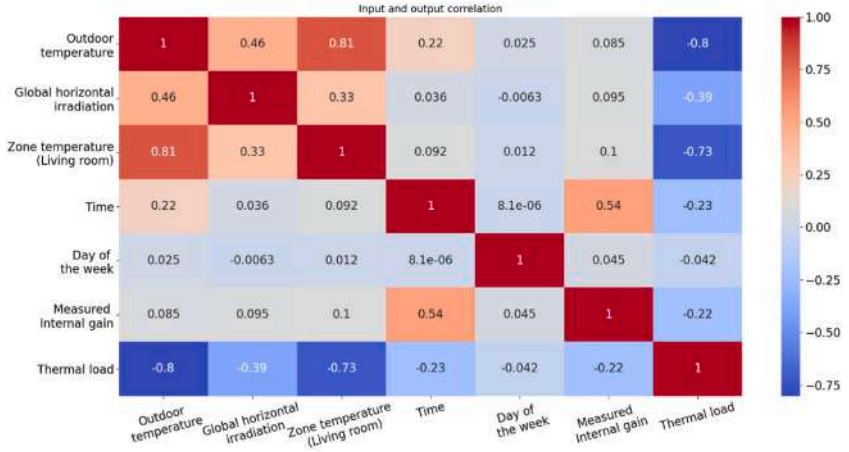

Fig. 2. Correlation Heatmap showing the Relationship between Inputs and Thermal Load.

perature and human activities. Including both features allows our model to account for weekly and daily variations in thermal load without requiring additional sensors. D_w and t can be easily derived from available data sources such as timestamps or system clocks.

In the process of developing the ANN model, experiments were conducted to determine the optimal number of neurons in the hidden layer. Several trials were performed with varying numbers of neurons, ranging from a small number to a larger one. Through these experiments, it was found that using 16 neurons in the hidden layer proved to be the most effective choice, as it yielded the best results. Besides, the choice of activation function for the hidden layer was considered. After evaluating different options, the Rectified Linear Unit (ReLU) activation function was selected as the optimal choice. This decision was made based on its ability to improve the model's performance and accuracy, as well as its computation efficiency compared to other activation functions. As the activation function for the output layer, a linear activation was considered in order to produce continuous output values that represent the predicted thermal load of the building. This choice was made to ensure that the output is directly proportional to the input, making it easier to interpret and integrate with the model predictive control (MPC) algorithm used in the controller. Fig. 3 illustrates the architecture of the proposed ANN model.

2.4. Technical challenges

In order for the ANN to effectively predict the thermal load, it must have access to the future values of its input features. Therefore, the future input feature values can be obtained through forecasting or made available through other means. In this work, the meteorological data (T_a and φ_s) and integral gains (θ) are sourced from TRNSYS, therefore, assuming to have a perfect prediction of outdoor conditions and internal gains. It should be noted that the term "internal gain" refers explicitly to the measured internal gains associated with lighting and electrical appliances that can be easily quantified using electricity meters. Measuring internal gain in the real world is challenging, often requiring models or calculations. To strike a balance between accuracy and practicality, we simplified the model and focused on electrical measurable sources.

As we can see in Fig. 3, the zone temperature is one of the input features for the thermal load model, implying that it must also be predicted. It turns out that the future values of the zone temperature can be forecasted with satisfying accuracy by making use of a second ANN model where the thermal load is an input feature as they are interdependent. The task of the second ANN model is to predict the zone temperature (T_z). This model requires similar features to those of the thermal load model with additional features representative of the thermal load and zone temperature at the previous time step. The zone temperature model can be formulated as equation (2).

$$T_z(t) = f(T_a(t), \varphi_s(t), t, D_w, \theta(t), P(t), T_z(t-1)) \tag{2}$$

The tie induced by the fact that each model output is also an input of the other is solved by connecting the two models in a feedback loop. Fig. 4 shows the final model structure.

2.5. Model refinement

This section will discuss the mathematical details of the proposed model's implementation. The model is designed to make hourly predictions over a 24-h horizon continuously. The model is based on a state-space model that can approximate the input-output

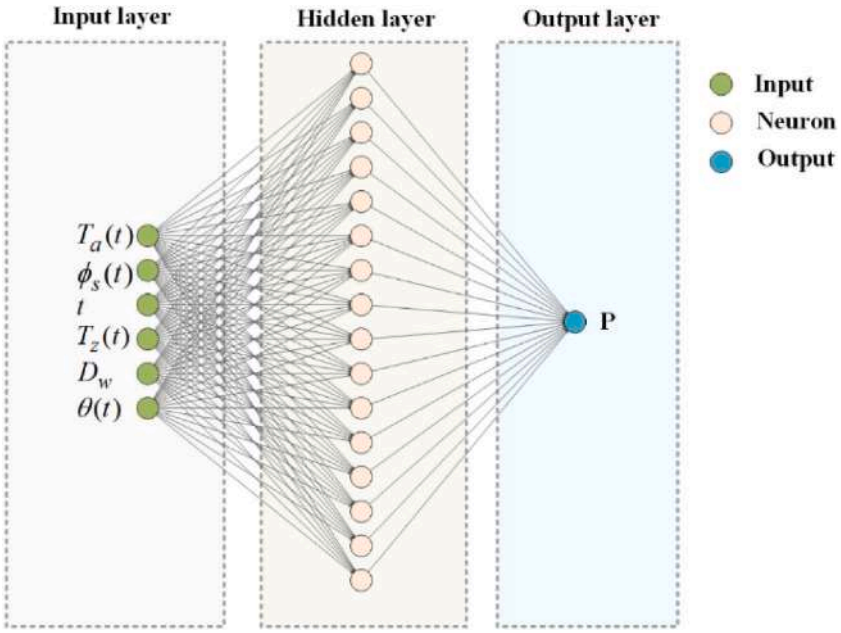

Fig. 3. Structure of proposed ANN model for load prediction.

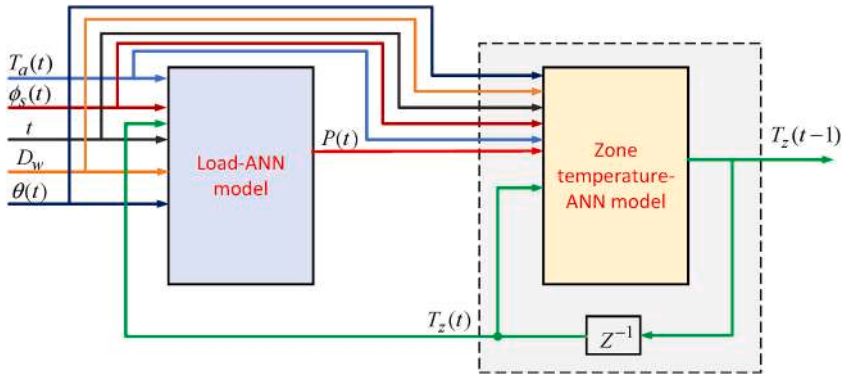

Fig. 4. Identified model for load (and zone temperature) prediction.

relation seen in the data, and this family of functions includes all networks arising from the combination of possible neuronal units, connections, activation functions, and weights. The extended state vector is introduced to include the recent past evolution of state and input variables, and a selector is used to reduce the network's complexity by selecting a subset of extended states to become features fed to the ANN core. We investigated the optimal size of training datasets for a 5-min time step model and designed a method for weight initialization. The model is trained every day at midnight using the most recent data and then used to make predictions for the next 24 h. To evaluate the accuracy of the proposed model, various performance metrics were utilized.

2.5.1. *Continuous training model*

Determining the behavior of a building in response to variations in its boundary and internal conditions can be seen as a (dynamic) system identification task. More specifically, since a real building can show both nonlinear and time-varying (NLTV) behaviors, we are dealing with identifying NLTV systems. Such systems can theoretically be formulated like equation (3) as NLTV state-space models.

$$\begin{cases} \boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \end{cases} \tag{3}$$

In this formula, $\boldsymbol{f}$ and $\boldsymbol{g}$ are vector values functions, while $\boldsymbol{u}(t) \in \mathbb{R}^m$, $\boldsymbol{x}(t) \in \mathbb{R}^n$ and $\boldsymbol{y}(t) \in \mathbb{R}^p$ represent the input, state, and output vectors at time t, respectively. In other words, $\boldsymbol{f}$ represents n scalar functions of $n+m+1$ (scalar) variables while $\boldsymbol{g}$ represents p scalar functions of $n+m+1$ variables. $\boldsymbol{f}$ is called the state update function, while $\boldsymbol{g}$ is called the output function. The system identification task consists in determining the state update and output function that best approximates the input-output relation seen in the data in a

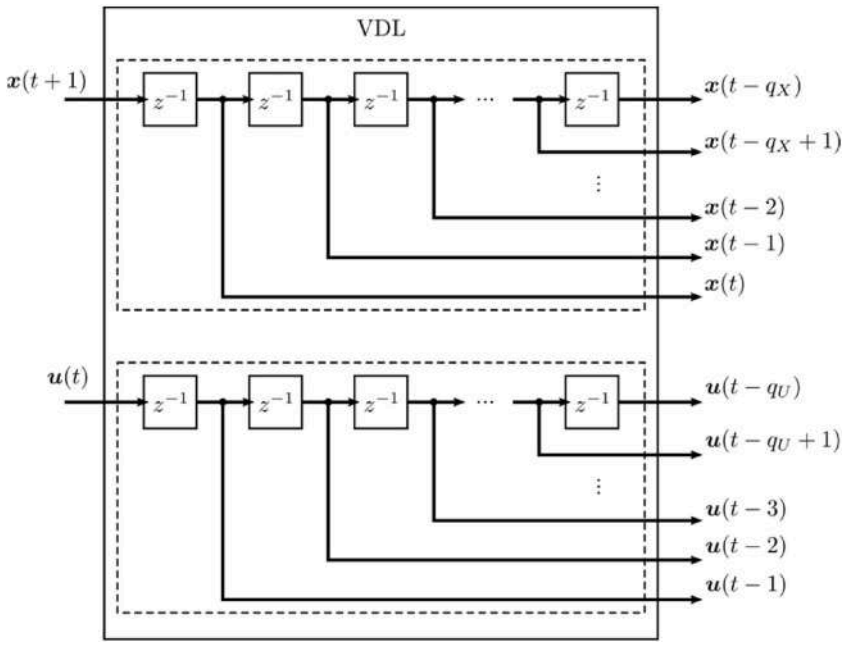

Fig. 5. Graphical representation of the construction of the extended state $\boldsymbol{\xi}(t)$ carried out by the VDL block.

certain family of functions. In this work, we performed the system identification task using ANNs, therefore the family of functions we are referring to is the family of all the networks arising from the combination of possible neuronal units, their connections, activation functions, and weights.

In this context, due to the use of ANN, we found it helpful to explicitly include the recent past evolution of state and input variables in the mathematical formulation, leading to a modified version of equation (3) to read as equation (4).

$$\begin{cases} \boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t-q_X), \ldots, \boldsymbol{x}(t-1), \boldsymbol{x}(t), \boldsymbol{u}(t-q_U), \ldots, \boldsymbol{u}(t-1), \boldsymbol{u}(t), t) \\ \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t-q_X), \ldots, \boldsymbol{x}(t-1), \boldsymbol{x}(t), \boldsymbol{u}(t-q_U), \ldots, \boldsymbol{u}(t-1), \boldsymbol{u}(t), t) \end{cases} \tag{4}$$

Here, $q_X \in \mathbb{R}^n$ is the number of past samples of the state variables, and $q_U \in \mathbb{R}^n$ is the number of past samples of the input variables. Note that although in equation (4) we used the same symbols as in equation (3), the symbols denote different state update and output functions. The total number of scalar arguments of the two functions is $n(q_X + 1) + m(q_U + 1) + 1$, including the time dependence. It is easy to show by introducing the extended state vector $\boldsymbol{\xi}(t)$,

$$\boldsymbol{\xi}(t) = [\boldsymbol{x}(t-q_X);\ \ldots;\ \boldsymbol{x}(t-1);\ \boldsymbol{x}(t); \boldsymbol{u}(t-q_U);\ \ldots;\ \boldsymbol{u}(t-1)] \tag{5}$$

that the modification does not change the structure of the formulation, as (4) can be cast in the same form as equation (5),

$$\begin{cases} \boldsymbol{\xi}(t+1) = \boldsymbol{\varphi}(\boldsymbol{\xi}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{y}(t) = \boldsymbol{\gamma}(\boldsymbol{\xi}(t), \boldsymbol{u}(t), t) \end{cases} \tag{6}$$

where $\boldsymbol{\varphi} : \mathbb{R}^{n(q_X+1)+mq_U} \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^{n(q_X+1)+mq_U}$ is the new (extended-)state update function and $\boldsymbol{\gamma} : \mathbb{R}^{n(q_X+1)+mq_U} \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^p$ is the new output function.

A diagram representing the relationship between extended state $\boldsymbol{\xi}(t)$, state at the next moment $\boldsymbol{x}(t+1)$ and input $\boldsymbol{u}(t)$ is given in Fig. 5, using two vector delay lines (VDL), one related to the state variables and the other to the input variables. The two VDLs consist of several vectorial one-step delay blocks (denoted by z^{-1}). As represented in Fig. 6, a vector delay block is composed of various scalar one-step delay elements (also denoted by z^{-1}).

Decimation, or down-sampling, is the process of reducing a signal sample rate. Decimation decreases the sampling rate of the inputs by an integer factor by periodically selecting one sample out of N samples. To ease the system identification with neural networks, we consider a different structural feature regarding the reduction of the extended states. The reasons why we introduced this feature in our model are reported in the following. In certain conditions, it may be helpful or even necessary to use relatively high values of the parameters q_X and q_U in order to capture "ancient" samples (like those of 24 h before). However, the fact that very old samples are needed does not mean that feeding the ANN element with the entire record of past samples (including the present) is the most appropriate choice. To provide the ANN element with a reduced number of samples, we introduce the selector $\Pi : \mathbb{R}^{n(q_X+1)+mq_U} \to \mathbb{R}^{n_f}$, allowing, in general, to select a subset of extended states to become features to be fed to the ANN core.

In this way, $\widetilde{\boldsymbol{f}}$ and $\widetilde{\boldsymbol{g}}$, the final state update and output functions are obtained. We remark that reducing the number of inputs to an ANN can reduce the network's complexity and consequently increase both the learning speed and the evaluation speed, especially in the case of shallow networks. In theoretical developments, Π can be assumed to be a full-rank matrix with $n_f : \mathbb{N}$ rows and $n(q_X+1)+mq_U$ columns, consisting of only numbers one and zero and having precisely one number 1 per row. The reduced (extended-)state $\Pi\boldsymbol{\xi}(t)$ and the current input $\boldsymbol{u}(t)$ concur to form the set of features fed to the neural network, which aims to produce certain output targets. Briefly, the structure of the model is written like equation (7).

$$\begin{cases} \boldsymbol{x}(t+1) &= \widetilde{\boldsymbol{f}}(\Pi\boldsymbol{\xi}(t), \boldsymbol{u}(t), t) \\ \boldsymbol{y}(t) &= \widetilde{\boldsymbol{g}}(\Pi\boldsymbol{\xi}(t), \boldsymbol{u}(t), t) \end{cases} \tag{7}$$

This description using mathematical symbols can be illustrated in Fig. 7. Fig. 7 highlights the fundamental structural elements of the functional block called ANNModel: the double VDL, the selector of the extended-state Π, the ANN block and the feedback loop

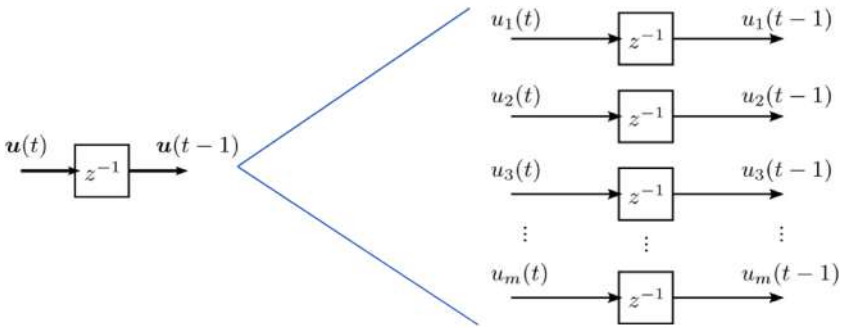

Fig. 6. Illustration of the vector representation in the case of the delay block.

($\boldsymbol{x}(t+1)$). Both the load model and the zone temperature model are instances of this ANN-model functional block. The load model has no state variables ($\boldsymbol{x}(t)$ is empty, $n = 0$) while the zone temperature model has no outputs apart from its state variable ($\boldsymbol{y}(t)$ is empty, $p = 0$). Both models however have an extended state because both make use of past inputs. Although the model allows for a flexible method to reduce the extended state to create the features, in this works we used only the decimation approach.

2.5.2. Length of the dataset for training and validating

The datasets used to train, validate, and test the proposed ANN model were generated by running yearly simulations of the TRNSYS model and will be referred to as TRNSYS datasets in the following. In the deployable implementation, as it is not possible to embed meaningful data for training and validating before the installation of the system, the model requires some time to gather experimental data before it can begin training and making predictions. The size of the dataset is of paramount importance, as having the right amount of data can reduce computational time, while not having enough data can result in the model not achieving acceptable performance. Empirical processes and experiments are a way of answering the question of how much data is needed for a well-performing model, as the answer depends on the type of problem. Therefore, based on our TRNSYS dataset, we investigated the size of the data that the model needs to be well-trained and proficient. It was discovered that the learning of the model is significantly impacted by days with frequent zero thermal loads. Consequently, it would be better not to use zero days for training. The size of the training dataset was also explored. As previously stated, a model that can be trained effectively with less data has a competitive advantage. The results of several experiments showed that on a 5-min time step model, 14 days of recorded data are sufficient to achieve acceptable performance. Utmost care has been taken to ensure that new data is consistently included in the training dataset while the oldest samples are left for validation. This approach allows for a more accurate and up-to-date model. Additionally, it is important to note that the size of the dataset may vary depending on the complexity of the problem and the type of model used. Thus, it is important to experiment and evaluate the performance of the model with different dataset sizes to find the optimal size for the specific problem.

2.5.3. Weights as a learnable term

Proper weight initialization is a crucial step in developing a neural network model. The optimization approach used in neural network modeling, stochastic gradient descent, gradually alters network weights to minimize a loss function, ultimately resulting in a set of weights that can generate accurate predictions. However, this optimization process needs a starting point, which is where weight initialization comes into play. Weight initialization is the process of assigning initial values to the weights of a neural network, often small random values. This serves as the starting point for the optimization process [59]. Choosing the appropriate weight initialization strategy is essential for the network to converge and reach a global minimum of the loss function. For example, it has been shown that using appropriate weight initialization techniques such as Xavier initialization [60] or He initialization [61] can lead to faster convergence and better performance of the model. The correct weight initialization can help to overcome local optima and help the model to converge faster. Therefore, weight initialization is a crucial step in developing a neural network model, and it should be carefully considered in the model design process.

In this study, the proposed model is designed in a way to be retrained every 24 h and make predictions hourly over a horizon of 24 h. This process will be repeated daily (e.g., at midnight). Each training starts with an initial set of weights. The weights can be set at random or with the previous best gains. Starting from the steps of the previous weight rather than starting at random can improve convergence speed. It might be interesting in the future to combine the two options, allowing for "blur" only a portion of your previous gains. In addition to that, the applied weights for the next 24 h after the retraining are a convex combination of the most recent optimal gains and the previous optimal gains. Equation (8) shows precisely how the weights are manipulated. The factor δ is a floating number in the [0,1] range. One entirely discards the previous weights, whereas zero dismisses the new ones (rendering the model totally non-adaptive). Fig. 8 represents the structure of weight initialization and weight selection in the proposed ANN model. Also, the role of a switch is formulated in equation (9).

$$W_i = (1-\delta)W_{i-1} + \delta W_i,\ \delta \in [0,1] \tag{8}$$

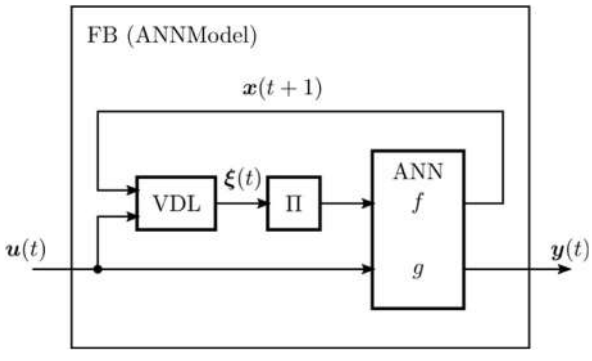

Fig. 7. Graphical representation of a functional block (FB) implemented by the neural network.

$$\begin{cases} \delta = 1, & s = 0 \\ \delta \in [0, 1], & s = 1 \end{cases} \tag{9}$$

2.5.4. Resetting state

In this study, the proposed ANN model was designed to predict thermal load and indoor temperature for the following 24 h. Typically, models can predict the first few hours with high accuracy, but their accuracy decreases over time, leading to diverging predictions. However, since in real-world scenarios, continuous access to sensor data is available, the proposed model is able to update its predictions more frequently, every hour. This is achieved by updating the internal state of the model with the most recent information and then running a 24-h simulation. It is important to note that the process of training the model and making predictions are distinct. The model is trained every day at midnight, using the most recent data available. After the training process is complete, the trained model is used to make predictions every hour for the next 24 h.

2.6. Key performance indicators (KPIs)

To evaluate the accuracy of the proposed model, several key performance indicators (KPIs) were used. One of the most widely used error metrics in the regression problem is the mean square error (MSE). MSE measures the average squared discrepancies between a dataset's predicted and actual target values. Root Mean Square Error (RMSE) is an extension of the mean squared error [62], and it is a commonly used measure of the difference between values predicted by a model or an estimator and the true values. An important feature of RMSE is that the units of the RMSE are the same as the original units of the target value that is being predicted. Another commonly used metric is mean bias error (MBE). MBE is a measure of the long-term performance of a model, as it indicates the average

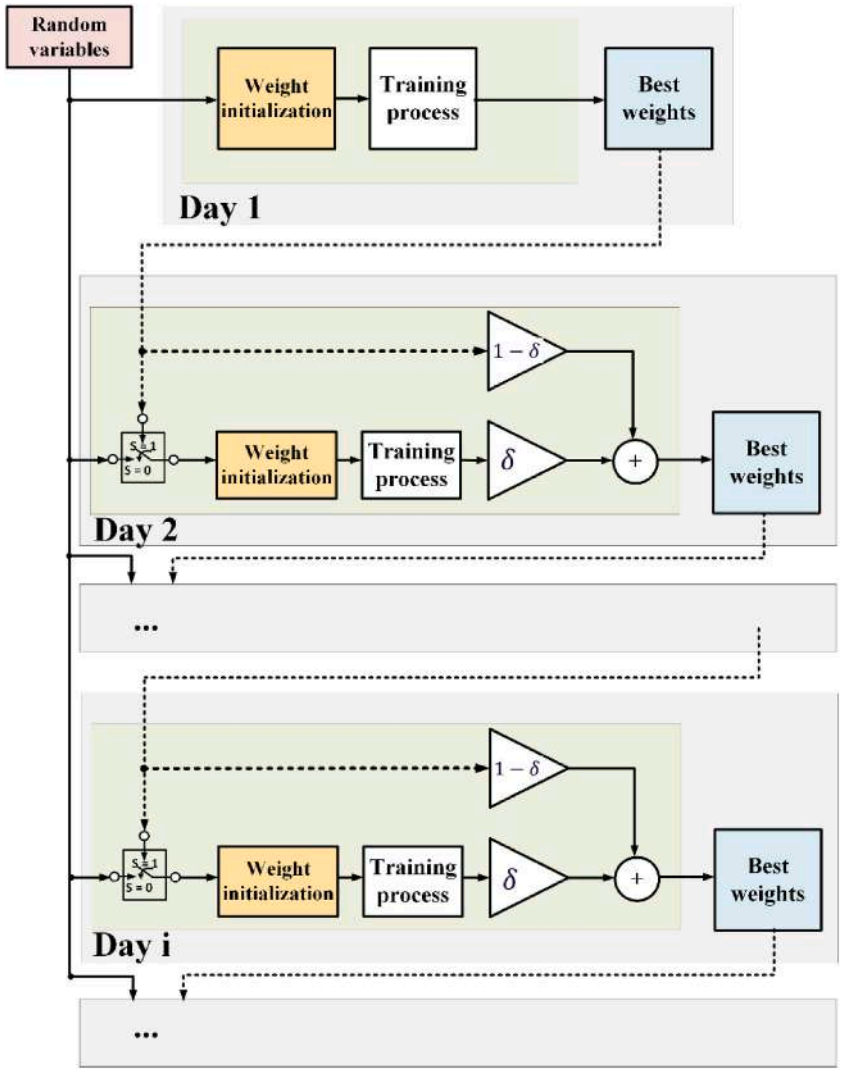

Fig. 8. The structure of the proposed idea regarding weights in ANN.

difference between the predicted and actual values. A positive number indicates that the estimated values are typically overestimated and vice versa [63]. To make the MBE findings comparable, the normalized mean bias error (NMBE) index is used, which normalizes the MBE index by dividing it by the mean of the measured values [64]. This allows for a more accurate comparison of the performance of the model across different datasets and scenarios.

$$MSE = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N} \tag{10}$$

$$RMSE = \sqrt{MSE} \tag{11}$$

$$MBE = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)}{N} \tag{12}$$

$$NMBE = \frac{\sum_{i=1}^{N} \left(\hat{y}_i - y_i \right)}{(N-1)\overline{y}} \times 100 \tag{13}$$

In equations (10)–(13), N is the number of samples, $\hat{y}_i$ is the predicted value, y_i is the actual value and $\overline{y}$ is the average of actual values.

3. Result and discussion

This section presents the simulation results that evaluate the effectiveness of the suggested model for thermal load prediction. The proposed ANN model was implemented in Python on a laptop with an i7 2.3 GHz CPU and 32 GB memory. It is a versatile model capable of predicting the thermal load of a building zone, such as an apartment, throughout the year. Regardless of the building's size and location, the model requires two weeks of performance data to be collected. Since it is a black-box model, it does not have pre-existing data in its repository, highlighting the importance of data collection as a crucial step. Once the data is gathered, the model can generate predictions for control purposes. It produces hourly predictions and self-trains every 24 h, replacing old data with newly recorded data. The training and validation period for the model spans two weeks. In the prediction model, for each time step, the model examines the data from the past 6 h and generates a prediction.

Fig. 9 provides a detailed illustration of the model performance during the winter season. In the beginning, the model is trained on data spanning from the 17th to the 30th of November. It was deployed for prediction for the next seven days. During the winter, the thermostat hysteresis is set between 20.75 and 21.25 $^{\circ}C$. The figure is divided into three sections. The initial section gives a summary of the ambient temperature (T_a) and the relative global horizontal irradiance (φ_s). In the following section, the temperature of the zone is presented, showcasing both the actual values and the model's predicted values. The actual temperature is represented by a black line, while the model's hourly predictions are distinguished using various colors. Lastly, the third section displays the thermal load of the living room, depicting both the ground truth and the predicted values in a similar format.

Fig. 10(a) and (b) show the RMSE and MBE, which are used to assess the accuracy of the zone-temperature ANN model. Typically,

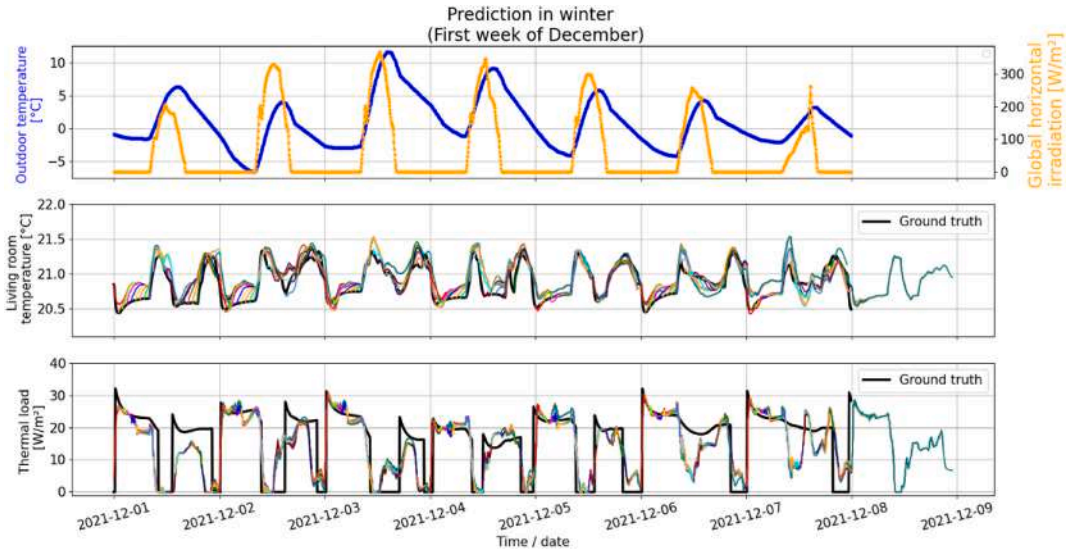

Fig. 9. Living room's temperature and thermal load prediction for the first week of December.

KPIs are calculated in the literature based on the prediction horizon, which means that if the prediction horizon is 24 h, all points of ground truth and forecast are considered, and the relative KPI is generated. As previously stated, models can generally accurately anticipate the first several hours. Nonetheless, their precision will deteriorate over time, and it is critical to know the accuracy at various intervals. To that purpose, computed KPI (CKPI) was established to calculate error hourly and over 3, 6, 12, and 24 h. Fig. 10(a) depicts the RMSE for temperature in different CKPIs where the mean of all errors is around 0.3 °*C*. As can be seen, when the CKPI is computed at a 24-h interval, the error range is narrow, but when the CKPI is computed at a 1-h interval, the error range widens. Fig. 10 (b) shows the range of the temperature bias error. Like RMSE, the error distribution varies depending on the data aggregation; for a 24-h period, the error range is lower than for a shorter period. The mean of MBE for all CKPIs is 0.18 °*C*. To enhance the clarity of the data visualization, the authors of this paper removed outliers from all the box plots. This was done to provide a clearer view of the box and improve the overall visual representation of the data.

Fig. 11(a) and (b) show the RMSE and MBE of the thermal load model. The average of RMSE and MBE is about 12 and -2.4 W/m^2, respectively.

The NMBE of the thermal load is shown in Fig. 12. According to equation (11), we are dealing with $\bar{y}$ in the denominator and as it observed from Fig. 12 in some intervals $\bar{y}$ is zero. While the amount of CKPI is less than 6, $\bar{y}$ sometimes will be zero and, in that, the amount of error will be infinite. To address this problem, the mean period (MP) is introduced, which considers the average of the next 6, 12, and 24 h. The range of the NMBE will be reduced concurrent with increasing CKPI.

Fig. 13 represents the results of the model during summer. The model has been trained by data from the 17th of June to the 30th of June and started predicting the 1st to the 7th of July. During the summer, the thermostat hysteresis is set between 24. 5 and 25. 5 °*C*. The cooling load is considered here as a negative value.

Fig. 14(a) shows the RMSE of the temperature model in summer. In all CKPIs, the average and median of RMSE is less than 1 °*C*. Also, as presented in Fig. 14(b), the average MBE in summer is around 0.2 °*C*.

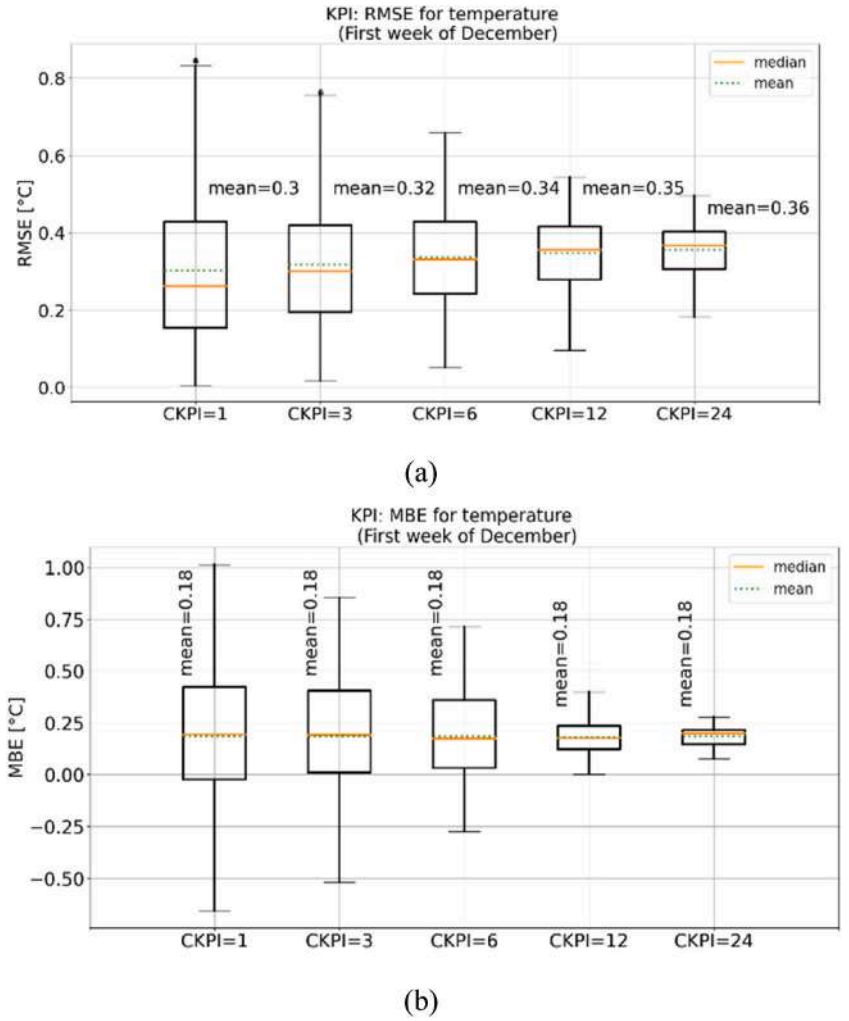

Fig. 10. RMSE (a) and MBE (b) for temperature in the winter.

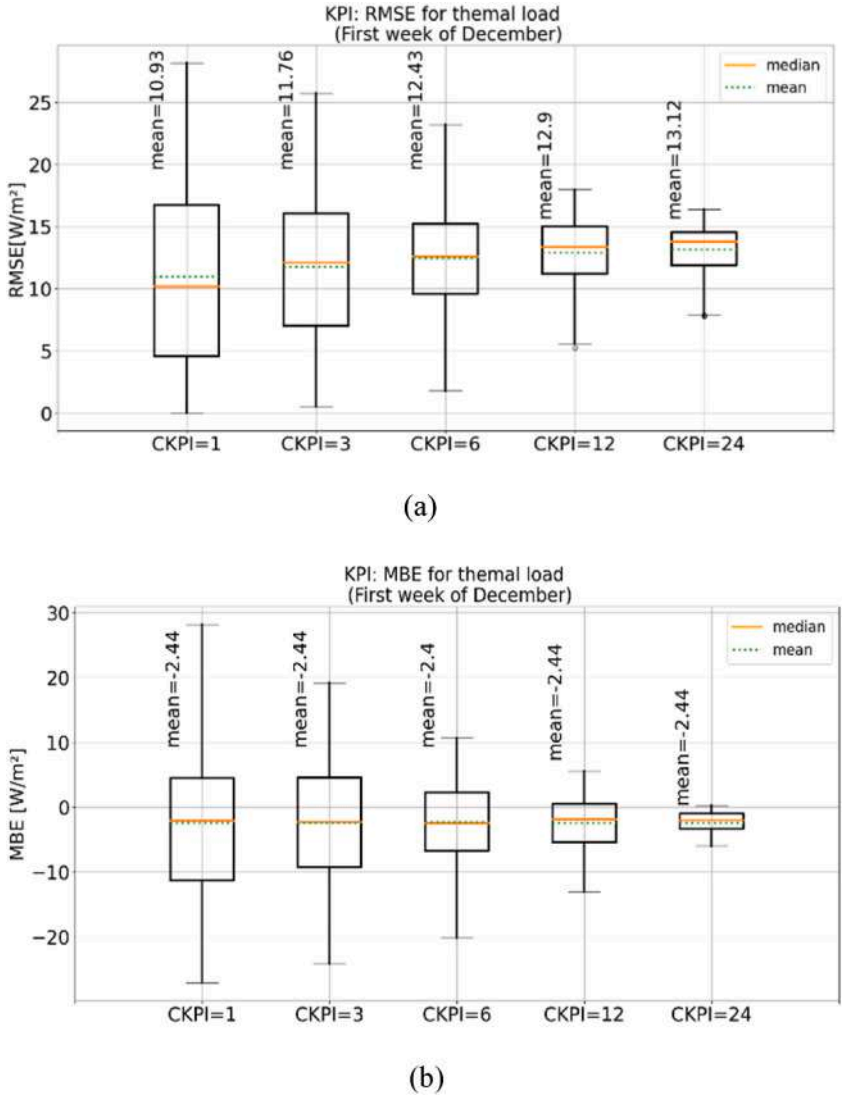

Fig. 11. RMSE (a) and MSE (b) of thermal load in the winter.

In Fig. 15(a) and (b), the average of RMSE and MBE for thermal load is about 10 and -0.6 W/m^2, respectively. Based on Fig. 12, same as winter, the average of ground truth in some intervals is zero, which can result in a high amount of NMBE. Especially once MP is 6, since the value of $\bar{y}$ is close to zero, as it observed from Fig. 16 in some period of time (between 5th to 7th of July), the range of error increased substantially.

In the proposed model, training load-ANN and zone temperature models are trained sequentially, with the training load model being trained first and then the zone temperature model being trained on the output of the load model. In this regard, the total training time for both models is presented in Table 1. The total training time for both models is between 1.3 and 2.9 min, which is reasonable for a model that provides hourly predictions for an entire year. The training time is shorter during the summer season, which can be attributed to the lower thermal load and the relatively stable weather conditions during this period.

Also, As the results are presented for a one-week period, and one prediction is made every hour, it is not feasible to present all the simulation times. However, the model's minimum and maximum simulation times were 4.1 and 8.6 s, respectively.

To optimize training time, one can enhance the computational power of the training system or employ parallel processing methods. The duration required to train a machine learning model depends significantly on the specifications of the computer system utilized. In general, a more robust system yields faster training times. For instance, a high-end graphics processing unit (GPU) can significantly expedite the training process by leveraging parallel processing techniques. Similarly, a system equipped with ample random access memory (RAM) can reduce data loading time during training. Consequently, the reported training times were influenced by the specific system specifications employed, and using a different system with distinct specifications may lead to varying training times.

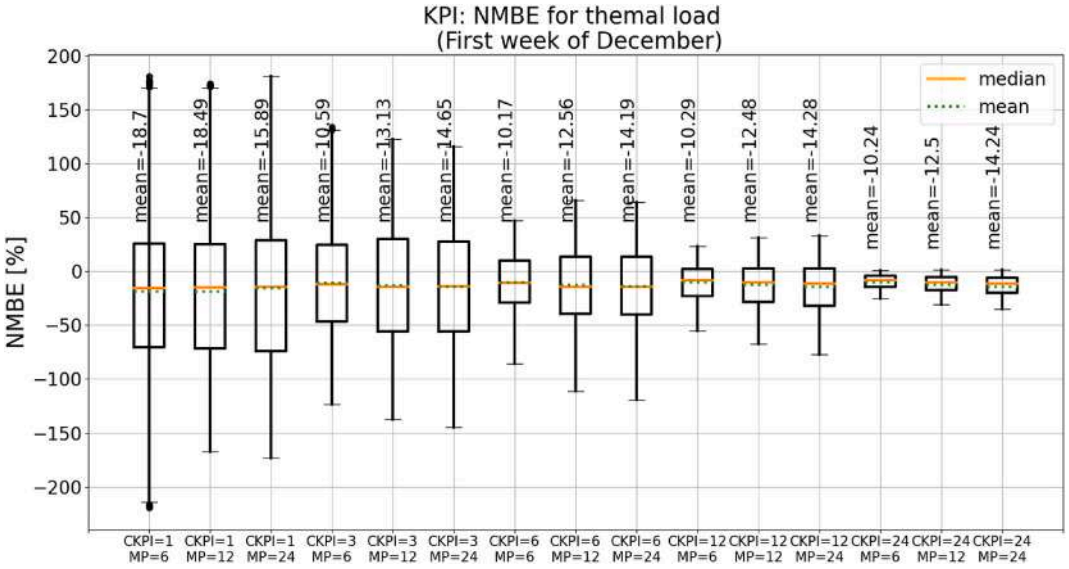

Fig. 12. NMBE for thermal load.

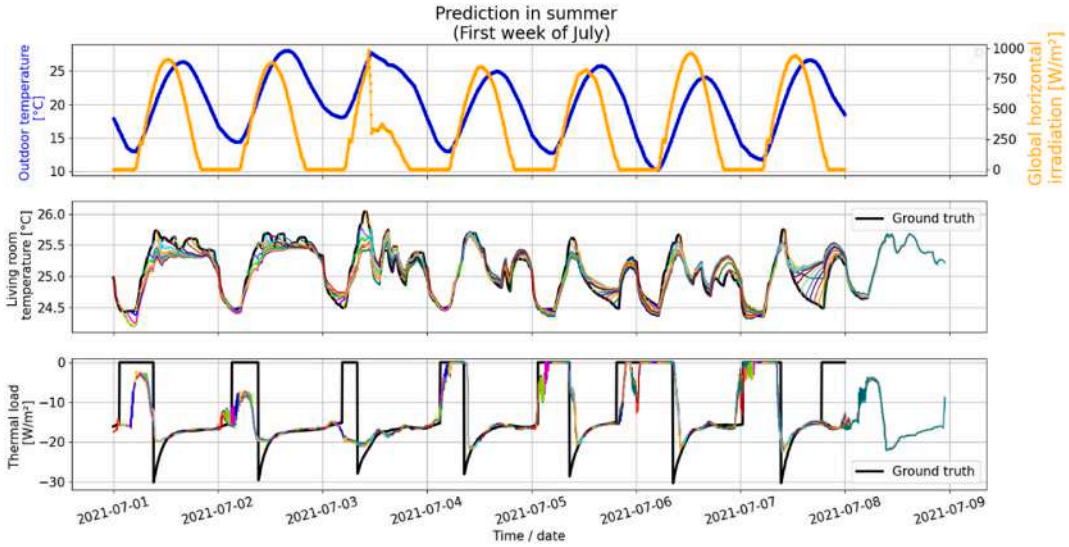

Fig. 13. Living room's temperature and thermal load prediction for the first week of July.

3.1. Comparison of typical and proposed weight initialization

The Keras library offers various pre-existing methods for weight initialization. However, based on our observations, these commonly used methods in the Keras library did not consistently yield optimal results for our specific use case. Consequently, we developed a novel initialization method that demonstrated improved outcomes and faster convergence compared to the traditional approaches. Notably, our method enables the partial retention of previous gains, which is advantageous when the physical system undergoes structural changes. Furthermore, our proposed method includes a switch that empowers users to either discard previous weights entirely or render the model non-adaptive, enhancing flexibility in model development. For the purpose of comparison, the results presented in Fig. 17 were obtained using Glorot uniform (GU) initialization for winter. It is evident from the results that GU initialization has led to chaotic predictions and produced unsatisfactory results. This chaos is particularly noticeable in the temperature section, which is concerning as temperature prediction is used as an input for load modeling and can significantly affect thermal load prediction.

To compare the performance of the proposed weight initialization method with the GU initialization, we considered only RMSE as a quantitative measure. Fig. 18(a) shows the RMSE of thermal load prediction in winter with GU initialization, which is the same as the graph in Fig. 10(a) for the proposed initialization method. Interestingly, both the mean and range of RMSE values in Fig. 18(a) are

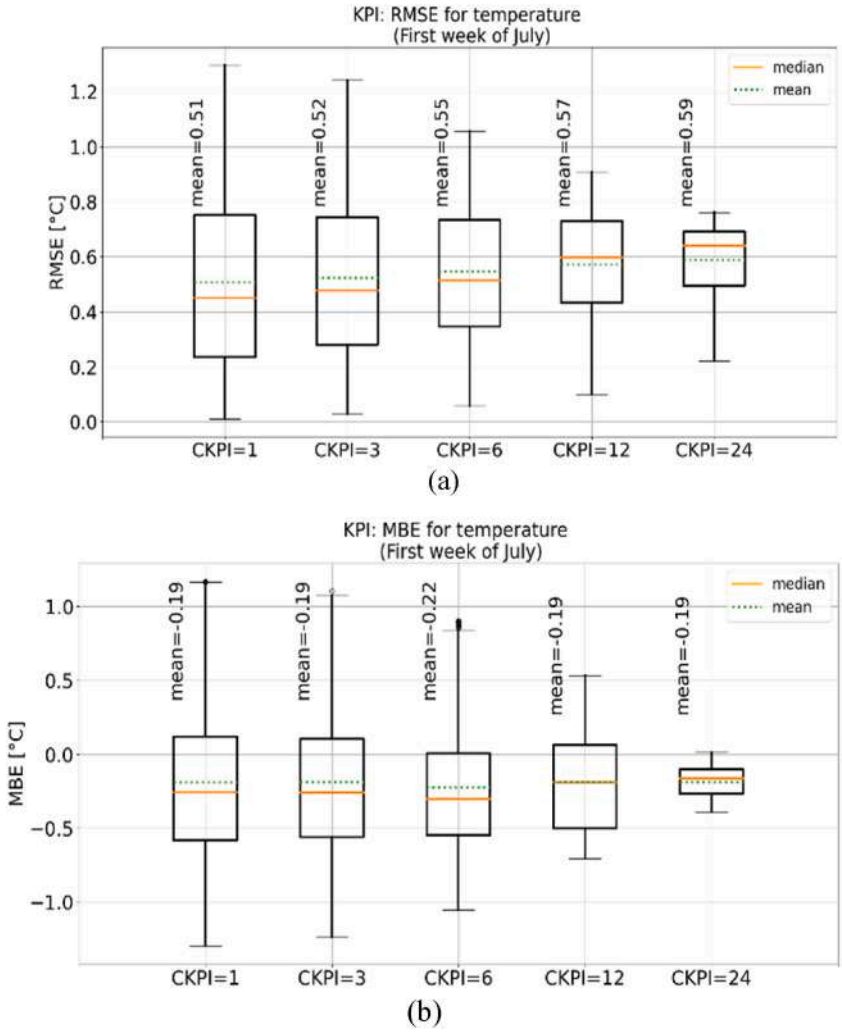

Fig. 14. RMSE (a) and MBE (b) for temperature in the summer.

higher than those in Fig. 10(a). Moreover, Fig. 18(b) shows the RMSE of living room temperature prediction with GU initialization, while Fig. 10(a) presents the RMSE for the proposed initialization method. Again, we observe that the mean RMSE is higher, and the error range has increased when using GU initialization.

Our findings indicate that utilizing the GU initialization method leads to both an increased average error and a broader range of errors when compared to our proposed initialization method. The good performance of our proposed initialization method and the satisfactory outcomes it produced for the control purpose of the model suggest that it can be used effectively in practical applications.

Table .2 provides a comparison of the training times for the two initialization methods over a period of seven days. The minimum and maximum times for the proposed method are 1.3 and 2.9 min, respectively, while for the GU initialization method, they are 1 and 7 min, respectively. The proposed method generally shows more consistent and faster training times than the GU initialization method.

While other initialization methods may potentially yield better results with more investigation and experimentation, our current focus was on developing an effective initialization method for this specific project and not exhaustively comparing all possible methods. Thus, our proposed method is well-suited for the purpose of this paper and our project, and we believe it can be a valuable contribution to the field.

3.2. Discussion

The major findings of this work highlight several advantages of our proposed ANN model in the context of previous studies and existing methods. These advantages can be summarized as follows:

1. Versatility and Adaptability: Unlike traditional approaches that rely on specific assumptions about building characteristics or location, our model adopts a black-box data-driven approach, making it highly versatile and adaptable to different building types

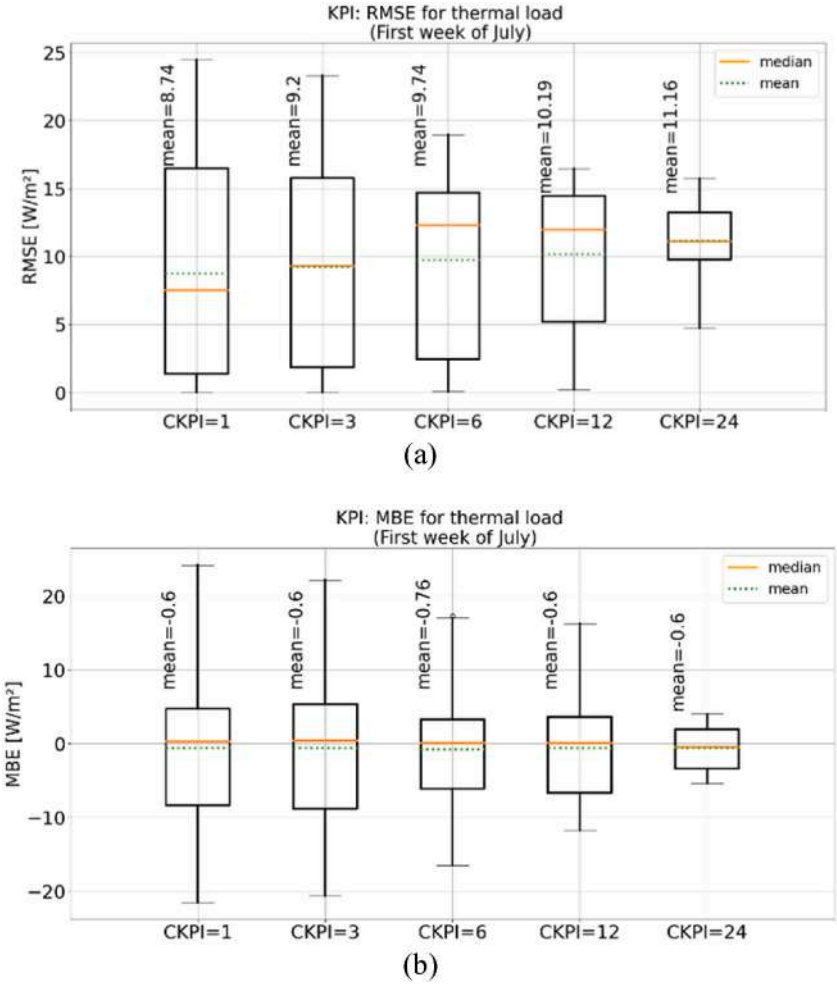

Fig. 15. RMSE (a) and MBE (b) for thermal load in the summer.

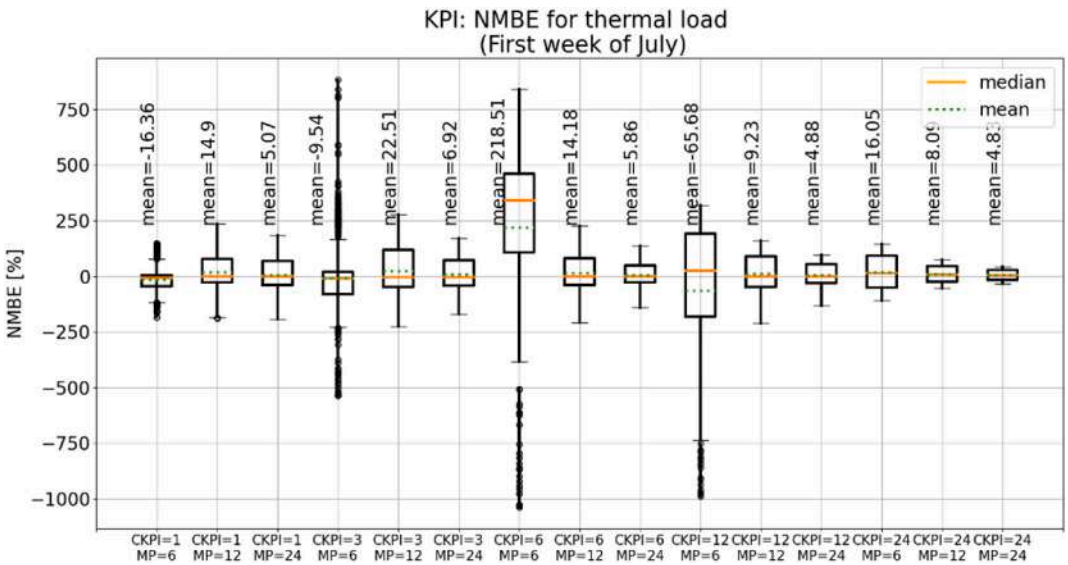

Fig. 16. NMBE for thermal load.

Table 1
Time of training for winter and summer [minutes].

Days	1st day	2nd day	3rd day	4th day	5th day	6th day	7th day
Winter	2.9	2.7	1.5	1.3	2.4	2.6	1.6
Summer	2.7	1.4	2.1	1.4	1.3	1.8	1.8

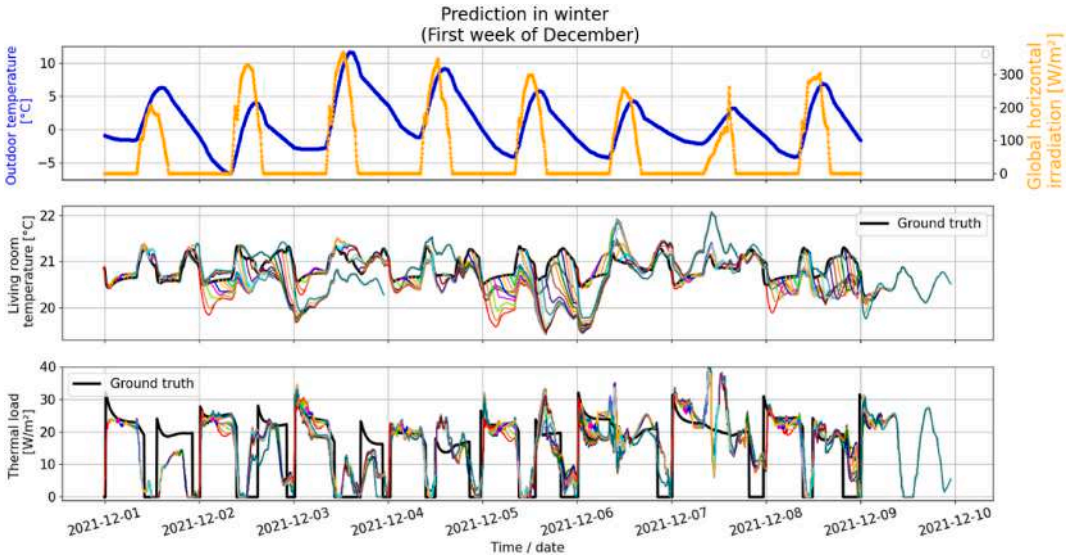

Fig. 17. Living room temperature and thermal load prediction for the first week of December. Weight initialization: Glorot uniform.

and sizes. It learns underlying patterns and relationships by collecting data from the building's performance over time, without explicitly requiring detailed information about the specific characteristics of each building.

2. Practicality and Cost-effectiveness: Our model utilizes simple and commonly available inputs, such as outdoor temperature, global horizontal irradiation, zone temperature, internal gain, day of the week, and time. By focusing on easily accessible inputs, it can be applied to a wide range of buildings without requiring complex or specialized data collection. This practical and cost-effective solution sets it apart from other methods that may require extensive building-specific information.

In comparing our approach to previous studies, our proposed model offers notable advantages. It is designed to be adaptive and continually trained with newly available data, enabling it to improve its forecasts over time and accommodate changes in building characteristics and behavior. Furthermore, our model achieves accurate predictions with minimal data by employing a simple MLP architecture with advanced techniques like specific internal state reset and specialized weight initialization functions. This simplicity not only allows for efficient implementation across various building types but also showcases the effectiveness and potential of our method in achieving superior outcomes.

While acknowledging that each building may possess unique characteristics, our approach provides a practical and cost-effective solution for accurate thermal load predictions. The model's adaptability, simplicity, and the use of common inputs make it suitable for general applications across a wide range of building types and characteristics.

However, it is important to address potential limitations and conflicting results. Further investigation is warranted to assess the model's performance in specific scenarios with unusual building characteristics or extreme climatic conditions. Additionally, future studies could explore potential refinements to enhance the model's robustness and accuracy.

In summary, our work advances the field by offering a versatile, practical, and cost-effective ANN model that surpasses previous approaches. It demonstrates the advantages of adaptability, simplicity, and accuracy while acknowledging the need for continued research to address potential limitations and explore further refinements.

4. Conclusion

In conclusion, the importance of accurate thermal load prediction in buildings for effective energy planning in HVAC control is emphasized. Thermal load prediction can be achieved through various methods, including data-driven black-box models, physics-based white-box models, and reduced-order gray-box models. In this study, a data-driven approach using an ANN was employed to forecast the thermal load of a residential building. The ANN model consists of two components: one for predicting thermal load and the other for predicting zone temperature. The novel model architecture is characterized by versatility, adaptability, and scalability, enabling its application to different building types and sizes. Additionally, the model continuously updates with new data and incorporates advanced techniques, such as dedicated internal states reset and specialized weight initialization, to enhance its

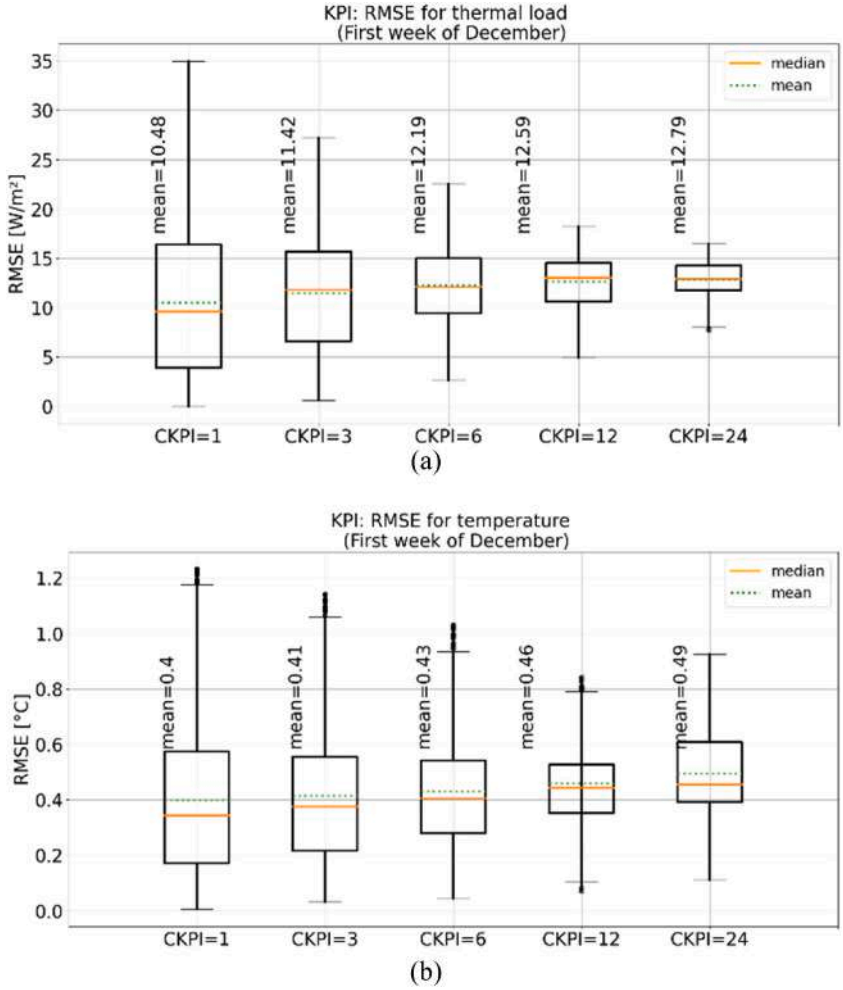

Fig. 18. RMSE of thermal load (a) and temperature (b). Weight initialization: Glorot uniform.

Table 2
Comparison of training time for proposed and GU initialization methods for winter [minutes].

Days	1st day	2nd day	3rd day	4th day	5th day	6th day	7th day
Proposed weight initialization	2.9	2.7	1.5	1.3	2.4	2.6	1.6
GU initialization	2.6	7	3	1.5	1.2	1.2	1

performance. The results of our case study, conducted with the TRNSYS simulation platform, demonstrate that the proposed ANN model achieves a high level of accuracy in forecasting both thermal load and indoor temperature. For zone temperature predictions, the RMSE is 0.3 °C with an MBE of 0.18 °C in the winter and an average RMSE of 0.5 °C with an MBE of 0.2 °C in the summer. Moreover, the average RMSE and MBE for thermal load predictions are 12 W/m^2 and -2.4 W/m^2 in the winter, and 10 W/m^2 and -0.6 W/m^2 in the summer, respectively. The training time for the model ranges from 1.3 to 2.9 min, which is reasonable considering it produces hourly predictions for an entire year. The training time is shorter during the summer due to lower thermal loads and more stable weather conditions. Additionally, the simulation times for the model range from 4.1 to 8.6 s, as the results cover a one-week period with hourly predictions. Overall, the proposed data-driven ANN model can be readily applied in real-world scenarios for efficient energy planning and HVAC control in residential buildings.

Authorship contributions

Mohammad Hossein Fouladfar: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization. **Anton Soppelsa**: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Review & Editing, Visualization. **Himanshu Nagpal**: Conceptualization,

Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Review & Editing. **Roberto Fedrizzi**: Conceptualization, Methodology, Validation, Formal analysis, Supervision, Project administration, Funding acquisition. Giuseppe Franchini: Writing - Review & Editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:Roberto Fedrizzi reports financial support was provided by Horizon 2020.

Data availability

No data was used for the research described in the article.

Acknowledgment

This work is a part of the research activities of the project REWARDHeat funded by the European Union Horizon 2020 research and innovation program under grant agreement No 857811.

References

[1] Energy consumption in households." https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_consumption_in_households (accessed July. 3, 2023).
[2] R. Lv, Z. Yuan, B. Lei, J. Zheng, X. Luo, Building thermal load prediction using deep learning method considering time-shifting correlation in feature variables, J. Build. Eng. 61 (2022), 105316.
[3] S. Buffa, A. Soppelsa, M. Pipiciello, G. Henze, R. Fedrizzi, Fifth-generation district heating and cooling substations: demand response with artificial neural network-based model predictive control, Energies 13 (17) (2020) 4339.
[4] F. Bünning, et al., Physics-informed linear regression is competitive with two Machine Learning methods in residential building MPC, Appl. Energy 310 (Mar. 2022), 118491, https://doi.org/10.1016/j.apenergy.2021.118491.
[5] M. Elnour, et al., Neural network-based model predictive control system for optimizing building automation and management systems of sports facilities, Appl. Energy 318 (2022), 119153.
[6] S.J. Cox, D. Kim, H. Cho, P. Mago, Real time optimal control of district cooling system with thermal energy storage using neural networks, Appl. Energy 238 (Mar. 2019) 466–480, https://doi.org/10.1016/j.apenergy.2019.01.093.
[7] A. Maiorino, M.G. Del Duca, C. Aprea, ART.I.CO. (ARTificial Intelligence for COoling): an innovative method for optimizing the control of refrigeration systems based on Artificial Neural Networks, Appl. Energy 306 (Jan. 2022), 118072, https://doi.org/10.1016/j.apenergy.2021.118072.
[8] W. Gang, J. Wang, Predictive ANN models of ground heat exchanger for the control of hybrid ground source heat pump systems, Appl. Energy 112 (Dec. 2013) 1146–1153, https://doi.org/10.1016/j.apenergy.2012.12.031.
[9] W. Gang, J. Wang, S. Wang, Performance analysis of hybrid ground source heat pump systems based on ANN predictive control, Appl. Energy 136 (Dec. 2014) 1138–1144, https://doi.org/10.1016/j.apenergy.2014.04.005.
[10] C. Vering, L. Maier, K. Breuer, H. Krützfeldt, R. Streblow, D. Müller, Evaluating heat pump system design methods towards a sustainable heat supply in residential buildings, Appl. Energy 308 (Feb. 2022), 118204, https://doi.org/10.1016/j.apenergy.2021.118204.
[11] L. Jia, J. Liu, A. Chong, X. Dai, Deep learning and physics-based modeling for the optimization of ice-based thermal energy systems in cooling plants, Appl. Energy 322 (Sep. 2022), 119443, https://doi.org/10.1016/j.apenergy.2022.119443.
[12] J. Lange, M. Kaltschmitt, Probabilistic day-ahead forecast of available thermal storage capacities in residential households, Appl. Energy 306 (Jan. 2022), 117957, https://doi.org/10.1016/j.apenergy.2021.117957.
[13] L. Scapino, H.A. Zondag, J. Diriken, C.C.M. Rindt, J. Van Bael, A. Sciacovelli, Modeling the performance of a sorption thermal energy storage reactor using artificial neural networks, Appl. Energy 253 (Nov. 2019), 113525, https://doi.org/10.1016/j.apenergy.2019.113525.
[14] Why modelling and model use matter: J. Oper. Res. Soc.: Vol 61, No 1." https://www.tandfonline.com/doi/abs/10.1057/jors.2009.141?journalCode=tjor20 (accessed Nov. 16, 2022).
[15] N. Abdou, Y. El Mghouchi, K. Jraida, S. Hamdaoui, A. Hajou, M. Mouqallid, Prediction and optimization of heating and cooling loads for low energy buildings in Morocco: an application of hybrid machine learning methods, J. Build. Eng. 61 (2022), 105332.
[16] O. Loyola-González, Black-box vs. White-box: understanding their advantages and weaknesses from a practical point of view, IEEE Access 7 (Oct. 2019) 154096–154113, https://doi.org/10.1109/ACCESS.2019.2949286.
[17] L. Zhang, et al., A review of machine learning in building load prediction, Appl. Energy 285 (2021), 116452.
[18] M. Velez, P. Jamshidi, N. Siegmund, S. Apel, C. Kästner, White-box analysis over machine learning: modeling performance of configurable systems, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), May 2021, pp. 1072–1084, https://doi.org/10.1109/ICSE43902.2021.00100.
[19] I. Adilkhanova, J. Ngarambe, G.Y. Yun, Recent advances in black box and white-box models for urban heat island prediction: implications of fusing the two methods, Renew. Sustain. Energy Rev. 165 (Sep. 2022), 112520, https://doi.org/10.1016/j.rser.2022.112520.
[20] Y. Chen, M. Guo, Z. Chen, Z. Chen, Y. Ji, Physical energy and data-driven models in building energy prediction: a review, Energy Rep. 8 (Nov. 2022) 2656–2671, https://doi.org/10.1016/j.egyr.2022.01.162.
[21] X. Li, J. Wen, Review of building energy modeling for control and operation, Renew. Sustain. Energy Rev. 37 (2014) 517–537.
[22] D. Dehestani, S. Su, H. Nguyen, Y. Guo, Robust fault tolerant application for HVAC system based on combination of online SVM and ANN black box model, in: 2013 European Control Conference (ECC), Jul. 2013, pp. 2976–2981, https://doi.org/10.23919/ECC.2013.6669140.
[23] M. Emami Javanmard, S.F. Ghaderi, M. Hoseinzadeh, Data mining with 12 machine learning algorithms for predict costs and carbon dioxide emission in integrated energy-water optimization model in buildings, Energy Convers. Manag. 238 (Jun. 2021), 114153, https://doi.org/10.1016/j.enconman.2021.114153.
[24] N. Asprion, et al., Gray-box modeling for the optimization of chemical processes, Chem. Ing. Tech. 91 (3) (2019) 305–313, https://doi.org/10.1002/cite.201800086.
[25] Y. Li, Z. O'Neill, L. Zhang, J. Chen, P. Im, J. DeGraw, Grey-box modeling and application for building energy simulations-A critical review, Renew. Sustain. Energy Rev. 146 (2021), 111174.
[26] H. Nagpal, I.-I. Avramidis, F. Capitanescu, P. Heiselberg, Optimal energy management in smart sustainable buildings – a chance-constrained model predictive control approach, Energy Build. 248 (Oct. 2021), 111163, https://doi.org/10.1016/j.enbuild.2021.111163.
[27] J. Ling, N. Dai, J. Xing, H. Tong, An improved input variable selection method of the data-driven model for building heating load prediction, J. Build. Eng. 44 (2021), 103255.
[28] M. Emami Javanmard, S.F. Ghaderi, Energy demand forecasting in seven sectors by an optimization model based on machine learning algorithms, Sustain. Cities Soc. 95 (Aug. 2023), 104623, https://doi.org/10.1016/j.scs.2023.104623.

[29] M. Emami Javanmard, Y. Tang, Z. Wang, P. Tontiwachwuthikul, Forecast energy demand, CO2 emissions and energy resource impacts for the transportation sector, Appl. Energy 338 (May 2023), 120830, https://doi.org/10.1016/j.apenergy.2023.120830.
[30] Z. Wang, T. Hong, M.A. Piette, Building thermal load prediction through shallow machine learning and deep learning, Appl. Energy 263 (Apr. 2020), 114683, https://doi.org/10.1016/j.apenergy.2020.114683.
[31] G. Ciulla, A. D'Amico, Building energy performance forecasting: a multiple linear regression approach, Appl. Energy 253 (Nov. 2019), 113500, https://doi.org/10.1016/j.apenergy.2019.113500.
[32] Z. Ma, J. Wang, F. Dong, R. Wang, H. Deng, Y. Feng, A decomposition-ensemble prediction method of building thermal load with enhanced electrical load information, J. Build. Eng. 61 (Dec. 2022), 105330, https://doi.org/10.1016/j.jobe.2022.105330.
[33] R. Sarwar, H. Cho, S.J. Cox, P.J. Mago, R. Luck, Field validation study of a time and temperature indexed autoregressive with exogenous (ARX) model for building thermal load prediction, Energy 119 (Jan. 2017) 483–496, https://doi.org/10.1016/j.energy.2016.12.083.
[34] A. Sandberg, F. Wallin, H. Li, M. Azaza, An analyze of long-term hourly district heat demand forecasting of a commercial building using neural networks, Energy Proc. 105 (May 2017) 3784–3790, https://doi.org/10.1016/j.egypro.2017.03.884.
[35] Y. Guo, E. Nazarian, J. Ko, K. Rajurkar, Hourly cooling load forecasting using time-indexed ARX models with two-stage weighted least squares regression, Energy Convers. Manag. 80 (Apr. 2014) 46–53, https://doi.org/10.1016/j.enconman.2013.12.060.
[36] K.M. Powell, A. Sriprasad, W.J. Cole, T.F. Edgar, Heating, cooling, and electrical load forecasting for a large-scale district energy system, Energy 74 (Sep. 2014) 877–885, https://doi.org/10.1016/j.energy.2014.07.064.
[37] P. Lusis, K.R. Khalilpour, L. Andrew, A. Liebman, Short-term residential load forecasting: impact of calendar effects and forecast granularity, Appl. Energy 205 (Nov. 2017) 654–669, https://doi.org/10.1016/j.apenergy.2017.07.114.
[38] D. Koschwitz, J. Frisch, C. van Treeck, Data-driven heating and cooling load predictions for non-residential buildings based on support vector machine regression and NARX Recurrent Neural Network: a comparative study on district scale, Energy 165 (Dec. 2018) 134–142, https://doi.org/10.1016/j.energy.2018.09.068.
[39] K. Liu, T. Liu, P. Fang, J. Zhao, Thermal load prediction by various hybrid models based on different artificial intelligence techniques, in: 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC), Aug. 2016, pp. 464–467, https://doi.org/10.1109/IHMSC.2016.60.
[40] H. Yan, K. Yan, G. Ji, Optimization and prediction in the early design stage of office buildings using genetic and XGBoost algorithms, Build. Environ. 218 (Jun. 2022), 109081, https://doi.org/10.1016/j.buildenv.2022.109081.
[41] Z. Wei, et al., Prediction of residential district heating load based on machine learning: a case study, Energy 231 (Sep. 2021), 120950, https://doi.org/10.1016/j.energy.2021.120950.
[42] P. Xue, Y. Jiang, Z. Zhou, X. Chen, X. Fang, J. Liu, Multi-step ahead forecasting of heat load in district heating systems using machine learning algorithms, Energy 188 (Dec. 2019), 116085, https://doi.org/10.1016/j.energy.2019.116085.
[43] S. Seyedzadeh, F. Pour Rahimian, P. Rastogi, I. Glesk, Tuning machine learning models for prediction of building energy loads, Sustain. Cities Soc. 47 (May 2019), 101484, https://doi.org/10.1016/j.scs.2019.101484.
[44] K. ArunKumar, D.V. Kalaga, C.M.S. Kumar, M. Kawaji, T.M. Brenza, Forecasting of COVID-19 using deep layer recurrent neural networks (RNNs) with gated recurrent units (GRUs) and long short-term memory (LSTM) cells, Chaos, Solit. Fractals 146 (2021), 110861.
[45] W. Wang, T. Hong, X. Xu, J. Chen, Z. Liu, N. Xu, Forecasting district-scale energy dynamics through integrating building network and long short-term memory learning algorithm, Appl. Energy 248 (Aug. 2019) 217–230, https://doi.org/10.1016/j.apenergy.2019.04.085.
[46] Y. Xu, F. Li, A. Asgari, Prediction and optimization of heating and cooling loads in a residential building based on multi-layer perceptron neural network and different optimization algorithms, Energy 240 (2022), 122692.
[47] Z. Yan, et al., A multi-energy load prediction of a building using the multi-layer perceptron neural network method with different optimization algorithms, Energy Explor. Exploit. 41 (1) (2023) 273–305.
[48] G. Suryanarayana, J. Lago, D. Geysen, P. Aleksiejuk, C. Johansson, Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods, Energy 157 (Aug. 2018) 141–149, https://doi.org/10.1016/j.energy.2018.05.111.
[49] J. Sun, G. Liu, B. Sun, G. Xiao, Light-stacking strengthened fusion based building energy consumption prediction framework via variable weight feature selection, Appl. Energy 303 (2021), 117694.
[50] C. Fan, F. Xiao, Y. Zhao, A short-term building cooling load prediction method using deep learning algorithms, Appl. Energy 195 (2017) 222–233.
[51] D.-S. Kapetanakis, E. Mangina, D.P. Finn, Input variable selection for thermal load predictive models of commercial buildings, Energy Build. 137 (2017) 13–26.
[52] Y.P. Raykov, E. Ozer, G. Dasika, A. Boukouvalas, M.A. Little, Predicting room occupancy with a single passive infrared (PIR) sensor through behavior extraction, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, in UbiComp '16, Association for Computing Machinery, New York, NY, USA, Sep. 2016, pp. 1016–1027, https://doi.org/10.1145/2971648.2971746.
[53] M.S. Zuraimi, A. Pantazaras, K.A. Chaturvedi, J.J. Yang, K.W. Tham, S.E. Lee, Predicting occupancy counts using physical and statistical Co2-based modeling methodologies, Build. Environ. 123 (Oct. 2017) 517–528, https://doi.org/10.1016/j.buildenv.2017.07.027.
[54] L.M. Candanedo, V. Feldheim, Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models, Energy Build. 112 (Jan. 2016) 28–39, https://doi.org/10.1016/j.enbuild.2015.11.071.
[55] I.B. Arief-Ang, M. Hamilton, F.D. Salim, A scalable room occupancy prediction with transferable time series decomposition of CO2 sensor data, ACM Trans. Sens. Netw. 14 (3–4) (Nov. 2018), https://doi.org/10.1145/3217214, p. 21:1-21:28.
[56] Paolo Bonato, Anton Soppelsa, Marta Avantaggiato, Roberto Fedrizzi, Assessment of Demand-Side Management on the Performance of a Single-Dwelling Mechanical Ventilation Plus Radiant Floor System," Presented at the Building Simulation Applications BSA 2022, bu,press, Bolzano, Italy, 2022, pp. 281–288, https://doi.org/10.13124/9788860461919.
[57] H. Taud, J.F. Mas, Multilayer perceptron (MLP), in: M.T. Camacho Olmedo, M. Paegelow, J.-F. Mas, F. Escobar (Eds.), Geomatic Approaches for Modeling Land Change Scenarios, Springer International Publishing, Cham, 2018, pp. 451–455, https://doi.org/10.1007/978-3-319-60801-3_27, in Lecture Notes in Geoinformation and Cartography.
[58] S.S. Mousavi, M. Schukat, E. Howley, Deep reinforcement learning: an overview, in: Y. Bi, S. Kapoor, R. Bhatia (Eds.), Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016, Springer International Publishing, Cham, 2018, pp. 426–440, https://doi.org/10.1007/978-3-319-56991-8_32, in Lecture Notes in Networks and Systems.
[59] J. Brownlee, Why initialize a neural network with random weights? Machine Learning Mastery (Aug. 14, 2022). https://machinelearningmastery.com/why-initialize-a-neural-network-with-random-weights/ (accessed Oct. 05, 2022).
[60] X. Glorot, Y. Bengio, 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, Mar. 2010, pp. 249–256. Accessed: Feb. 16, 2023. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html.
[61] K. He, X. Zhang, S. Ren, J. Sun, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, 2015, https://doi.org/10.48550/arXiv.1502.01852 arXiv, Feb. 06.
[62] X. Wen, M. Jaxa-Rozen, E. Trutnevyte, Accuracy indicators for evaluating retrospective performance of energy system models, Appl. Energy 325 (2022), 119906.
[63] C. Jacovides, H. Kontoyiannis, Statistical procedures for the evaluation of evapotranspiration computing models, Agric. Water Manag. 27 (3–4) (1995) 365–371.
[64] G.R. Ruiz, C.F. Bandera, Validation of calibrated energy models: common errors, Energies 10 (10) (Oct. 2017), https://doi.org/10.3390/en10101587. Art. no. 10.