processes

MDPI

Article

Mean Radiant Temperature Calculation Using Neural Networks in Python

Daniel Fernando Espejel-Blanco, José Antonio Hoyo-Montaño *, Rosalia del Carmen Gutiérrez-Urquidez, Fredy Alberto Hernández-Aguirre, Carlos Alberto Pereyda-Pierre, Jorge Oswaldo Rivera-Nieblas and Eduardo Chávez-Mendiola

Tecnológico Nacional de México, Instituto Tecnológico de Hermosillo, Av. Tecnológico S/N, col. El Sahuaro, Hermosillo 83170, Mexico; daniel.espejelb@hermosillo.tecnm.mx (D.F.E.-B.); rosalia.gutierrezu@hermosillo.tecnm.mx (R.d.C.G.-U.); fredy.hernandeza@hermosillo.tecnm.mx (F.A.H.-A.); carlos.pereydap@hermosillo.tecnm.mx (C.A.P.-P.); jorge.riveran@hermosillo.tecnm.mx (J.O.R.-N.); eduardo.chavezm@hermosillo.tecnm.mx (E.C.-M.)

* Correspondence: j.hoyo@ieee.org; Tel.: +52-662-260-6500 (ext. 231)

Abstract

Reducing energy consumption is a very important topic for engineers and researchers. The need for this reduction, combined with reductions in energy generation using fossil sources, with the aim of mitigating climate change, has led to research and the proposal of new ways to make the equipment responsible for energy consumption more efficient. An academic or office building installed in a hot climate exhibits very inefficient use of electricity for air conditioning; generally, buildings are overcooled, and this causes high energy consumption due to the set points used. Conventional control methods for air conditioning equipment use hysteresis temperature control and only work with the temperature in the space to calculate the operating set point. The mean radiant temperature (*MRT*) of a specific area is the average temperature of the objects surrounding a person and influences the person's heat loss or gain due to the exchange of thermal radiation. To directly measure the *MRT*, a meter called a radiometer is used, which is very expensive and sophisticated; its measurement is carried out using a black balloon thermometer, and empirical formulas are used for its calculation, such as the method of Fanger. In order to generate savings in the use of electrical energy in an academic or office building, the proposed *MRT* calculation algorithm can help to reduce electrical energy consumption through the use of the estimated predicted mean value (PMV) method proposed by the American Society of Heat, Refrigeration, and Air Conditioning Engineers (ASHRAE). This method uses the ambient temperature, black balloon temperature, and air speed to calculate the *MRT*. By calculating the set points of air conditioning equipment using the PMV index, it will be possible to reduce electrical energy consumption without sacrificing user comfort. The general objective is to develop an algorithm for calculating the value of the *MRT* using neural networks in Python. The neural network model developed in Python uses 20 data input points, such as the air temperature, to predict the *MRT* for single-hidden-layer and double-hidden-layer models, and it obtains R^2 values of 0.58 and 0.65, respectively. The three-layer neural network model, using normalized data and 104 input data points, obtains an R^2 value of 0.71. Using a second input variable (occupancy) in the developed neural network model increases the R^2 coefficient to 0.89.

Keywords: mean radiant temperature; neural networks; energy efficiency

Check for updates

Academic Editor: Federica Raganati

Received: 22 September 2026
Revised: 16 October 2026
Accepted: 19 October 2026
Published: 25 October 2026

Copyright: © 2026 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.

1. Introduction

The imminent depletion of fossil resources used for electrical energy production, and the large amount of greenhouse gases (GHG) emitted to the atmosphere when this energy is produced, has forced governments, politicians, businessmen, and researchers to look for new means to produce energy through renewable sources, because they do not emit GHG to the atmosphere [1]. For the above, it is important to design energy management systems (EMSs) that involve the most efficient use of energy and are supported by energy generation systems through renewables [2]. Up to 40% of energy is being consumed in buildings, and this energy consumption is responsible for 30% of CO_2 emissions [2]. On the other hand, 69% of consumption in buildings is caused by commercial buildings and office buildings [2]. For this reason, more attention is required regarding the control of electrical energy consumption in non-residential buildings.

In order to manage electricity consumption in a building, it is necessary to take into account all factors involved in energy generation or consumption. It should therefore be noted that, by monitoring and controlling the temperature, relative humidity, artificial light, and airflow, among other factors, a substantial percentage of electricity consumption can be saved.

For an energy consumption analysis, it is very important to assess all loads involved in energy consumption in the entire building. This analysis provides important information about which equipment has the highest energy consumption, allowing researchers to find ways to reduce or optimize this consumption and to explore the outcomes [3]. In buildings, the first step is to implement traditional options for reducing electricity use, such as improvements to the thermal and cooling internal conditions of the building, through the modernization of the existing energy system or by replacing the existing system with new, more efficient equipment [4].

An energy management system (EMS) is a system supported by information and communication technologies (ICTs), utilized to monitor, control, and optimize the generation, transmission, and consumption of electrical energy [5,6]. Meanwhile, a building energy management system (BEMS) is a system that uses computers, sensors, actuators, and the internet to monitor, control, and optimize a building's self-generation and energy consumption; in other words, a BEMS system is an EMS system applied to a building. A BEMS system usually utilizes a wireless network of sensors and actuators, which communicate with each other and with a central wired network server, and a computer or processor linked to the Internet, used to monitor, store, and process the data obtained from the sensor network [6]. Statistical results show that the use of BEMSs increased savings from 11.39% to 16.22% annually from 1976 to 2014 [7].

The rapid growth of technology has enabled "closed" systems with SCADA-type processing systems with PLCs and/or PCs with human–machine interfaces (HMIs) with inflexible control panels to evolve into user-friendly and flexible internet-based systems. This is accelerating even further with the arrival of the Internet of Things (IoT) and cloud-based systems and the generation of Big Data and its analysis and processing for decision-making [8]. Moreover, the concept of the Industrial Internet of Things (IIoT), also known as Industry 4.0, has provided a way to achieve better visibility within industrial operations through the integration of machinery with sensors, software, cloud computing, and data storage systems [9].

Nowadays, the appropriate management of electrical energy consumption in buildings is becoming increasingly important, because buildings are responsible for 40% of the worldwide energy consumed—mainly electrical energy [3,10–13]. Therefore, this research aims to reduce the energy consumption of the HVAC system, which is one of the systems that consumes the most energy in buildings.

This project will contribute to saving energy through a reduction in the electrical energy consumption in a building. This reduction is to be obtained from the HVAC system, and it is intended to mitigate or reduce the effects of climate change by indirectly reducing the generation of electricity from conventional fuels. Since less energy will be required, less will be generated, resulting in a reduction in GHG emissions to the atmosphere. Mexico has a very strong commitment in this regard. The General Law on Climate Change (LGCC) [14] established that, by 2020, GHG emissions into the atmosphere had to be reduced by 20%, and they should be reduced by 50% by 2050, compared to a baseline for emissions in the year 2000.

For the HVAC system, it is proposed to develop a control system that includes air quality measurements, particularly carbon dioxide, using the thermal comfort zones designed by ASHRAE. These comfort zones are based on the PMV index. This index is calculated using four environmental variables: the ambient temperature, relative humidity, mean radiant temperature (*MRT*), and air velocity [15]. Moreover, the PMV index calculation uses two qualitative variables: the metabolic rate and clothing insulation factor. The environmental variables of a space can be measured using sensors. Qualitative variables must be assumed depending on the conditions of the environment, the occupants using the space, and the space in which the HVAC equipment is located. The selection of HVAC equipment set points should be performed without sacrificing occupant comfort.

The authors in [16] present an analysis of the problem involved in identifying black-box prediction models that can be used for the predictive control of the indoor climate in buildings. The authors conducted a study in two buildings, applying two different neural network models to each: one using artificial neural network (ANN) models and the other using linear regression ARX models. The study demonstrated that the nonlinear artificial neural network (ANN) models provided more accurate predictions of the temperature than the linear regression ARX models.

The authors in [17] present an approach using deep neural networks with nonlinear regression to simulate virtual temperature sensors at specific points on a nuclear reactor vessel, training the neural network with real operational data. The objective of the project was to estimate the values of the temperatures to which the reactor's structures, systems, and components were exposed during their service life, with the aim of evaluating temperature-induced accelerated aging.

The objective of the study shown in [18] was to use artificial neural networks (ANNs) to obtain a prediction of the dew point temperature using historical meteorological data as input, seeking to obtain this prediction with a horizon of 1 to 12 h. The study uses a three-layer backpropagation ANN and three years of meteorological data from 20 different locations in the state of Georgia, located in the United States, with the goal of developing general models for predicting the dew point temperature anywhere in the state of Georgia. The final developed models predicted the dew point temperature accurately using previous meteorological data. These models can predict extreme frost conditions and heat stress, which are considered very difficult to predict. These predictions are very important because they can be used for decision-making in agriculture, as the dew point temperature, combined with the air temperature, modifies the severity of frosts and heat waves, which can cause significant damage to crops, equipment, and structures and can cause injury or death to animals and people.

The authors in [19] present a series of experiments to predict the operating temperature in two buildings using artificial neural networks with linear and nonlinear estimation models. In the buildings mentioned in the study, they demonstrated that the operating temperature could be estimated quite accurately using easily measurable variables, such as indoor and outdoor temperatures, electricity consumption, wall temperatures, ventilation

flows, and the time of day. They also demonstrated that nonlinear artificial neural network models generally provide better estimates than linear models.

In [20], the authors analyzed the design of neural network models to predict the indoor air temperature in an educational building. The predicted air temperature was used to develop predictive control for the air conditioning systems. To develop the neural network models, climatic and environmental data were collected via a remote data acquisition system from a secondary school located in Portugal. The authors demonstrated, through simulations, that the use of long-range predictive models to control air conditioning systems achieves effective temperature regulation with significant energy savings.

The authors in [21] present a field demonstration of a control strategy for HVAC systems based on the PMV index in a typical residential building located in Kuwait, under hot and dry climatic conditions. The control strategy is based on an advanced thermal comfort-based controller (TCC) for PMV-based HVAC control. The TCC continuously monitors the indoor and outdoor temperature conditions and adjusts the set point temperature of the HVAC system to optimize energy consumption while ensuring indoor thermal comfort. To estimate the *MRT* for the PMV-based HVAC control system, the authors use three machine learning models: linear regression, regression trees, and artificial neural networks. The results show that the proposed TCC, when installed in an HVAC system, performs better in terms of thermal comfort compared to an HVAC system without the TCC; furthermore, an improvement of up to 60% in the PMV index is achieved. The proposed TCC, which is controlled using machine learning methods, achieves energy savings of more than 20% in the HVAC system while also maintaining the desired thermal comfort levels in the experimental building.

Evaluating indoor thermal environments is essential for achieving thermal comfort and energy efficiency; however, the parameters used to calculate thermal comfort exhibit nonlinear variations, which complicates engineering design. In this context, in [22], the authors present the design of an artificial neural network (ANN) model to overcome the limitations of the nonlinear relationships between thermal parameters and the predicted mean vote (PMV) values. By using thermal parameters as input data for the ANN model, the PMV index could be predicted. Subsequently, the authors employed a genetic algorithm, the harmony search algorithm, the gravitational search algorithm, and the whale optimization algorithm to optimize the neural network structures. The authors used 5000 datasets to train the ANN. The ANN-GA neural network model demonstrated high prediction accuracy, with a mean relative error of less than 1.38, a mean squared error of less than 1.34, and a regression coefficient of approximately 1.

Furthermore, in [23], the authors present a predictive model leveraging machine learning (ML) algorithms. The model presented seeks to predict thermal comfort levels and optimize energy consumption in HVAC systems. The authors employ four distinct ML algorithms: support vector machine (SVM), an artificial neural network (ANN), random forest (RF), and extreme gradient boosting (XGBOOST). The model uses Raspberry Pi boards equipped with multiple sensors to collect data. The evaluation of the performance of each algorithm is based on four statistical error metrics, namely the root mean square error (RMSE), mean squared error (*MSE*), mean absolute error (MAE), and coefficient of determination (R^2). The results of this study show that the RF and XGBOOST algorithms perform better, achieving an R^2 coefficient of 96.7% and 96.4%, respectively. In contrast, the SVM algorithm performs less well, with an R^2 of 81.1%. The ANN algorithm presents an R^2 coefficient of 92.2%.

In [24], the authors present a model to predict the distribution of the *MRT* around buildings based on a multilayer neural network (MLNN) optimized using genetic algorithms (GAs) and backpropagation (BP). To train the proposed neural network, the authors

used meteorological data from 2014 to 2018 as input parameters to predict the *MRT* distribution around buildings in 2019. The input parameters used to train the neural network and predict the *MRT* were the solar altitude, solar orientation, midday air temperature, minimum air temperature, and global solar radiation. The study yielded prediction results with very high accuracy; these can be combined with sensitivity analysis methods to analyze and identify the most important input parameters affecting the *MRT* on hot days.

All of the alternatives reviewed above involve the use of advanced learning models and neural networks that require significant computing power to run. Although they achieve significant reductions in the R^2 coefficient and other operational parameters, implementing them in real-time intelligent control systems can be very complicated. The goal is to reduce the neural network's training time so that it can be applied in a real-time control system on a Raspberry Pi or ESP32 microprocessor.

In different studies, the *MRT* value used for the HVAC's set point calculation has been set equal to the ambient temperature value [11,25–32]. Therefore, the main goal of this work lies in the development of an algorithm for calculating the *MRT* value using neural networks in Python. This algorithm can be installed on an embedded system to control the set point of HVAC equipment, with the goal of reducing electricity consumption without sacrificing the user's thermal comfort.

2. Materials and Methods

Regarding humidity and temperature, Mexico has a standard that addresses both variables, namely Mexican standard NMX-C-7730-ONNCCE-2018 [33], Construction industry—Ergonomics of the thermal environment—Analytical determination and interpretation of thermal comfort through the calculation of the PMV (predicted mean vote) and PPD (predicted percentage dissatisfied) indices and local thermal comfort criteria. This standard aligns with UNE-EN ISO 7730 (ISO 7730:2005) [34], published by the International Organization for Standardization (ISO).

Similarly, ASHRAE Standard 55-2010 [35], published by the American Society of Heating, Refrigerating, and Air Conditioning Engineers (ASHRAE), uses the same indices (PMV and PPD) to graphically determine the comfort zone for typical indoor environments, with one for cooling systems and the other for heating systems.

2.1. Thermal Comfort Zones

The thermal comfort zones are calculated based on the indoor ambient temperature, indoor relative humidity, indoor air velocity, metabolic rate (according to the activities carried out in the space), and clothing insulation (depending on the clothes that people wear in the space). The comfort zones utilize the values of the indoor operating temperature that provide comfortable thermal indoor environmental conditions for its occupants, or they are calculated using a combination of the indoor air temperature and the mean radiant temperature (*MRT*) values, with the same goal: to ensure that occupants perceive the indoor conditions as thermally acceptable.

2.2. PMV Index Calculation

The ASHRAE-55-2010 standard uses a graph showing the winter and summer thermal comfort zones, derived from the PMV and PPD indices, and applies it to indoor spaces where occupants perform activities with metabolic rates between 1.0 and 1.3 met (1 metabolic unit = 1 met = 58.2 W/m^2) and wear clothing providing between 0.5 and 1.0 clo (1 clothing unit = 1 clo = 0.155 °C/W) of personal thermal insulation. For reference, a winter business suit provides about 1.0 clo of insulation, and a short-sleeved shirt and trousers provide about 0.5 clo. Most office and educational spaces have these characteristics.

The metabolic rate for sedentary activities in offices, schools, homes, and laboratories is 1.2 met (70 W/m^2) [34].

Figure 1 shows the thermal comfort graph based on the PMV and PPD methods. The graph is constructed within an operating temperature range with 80% of acceptability among occupants of the space, based on 10% dissatisfaction with general behavior, and is used to obtain the respective indices, considering an additional 10% average dissatisfaction caused by local thermal discomfort.

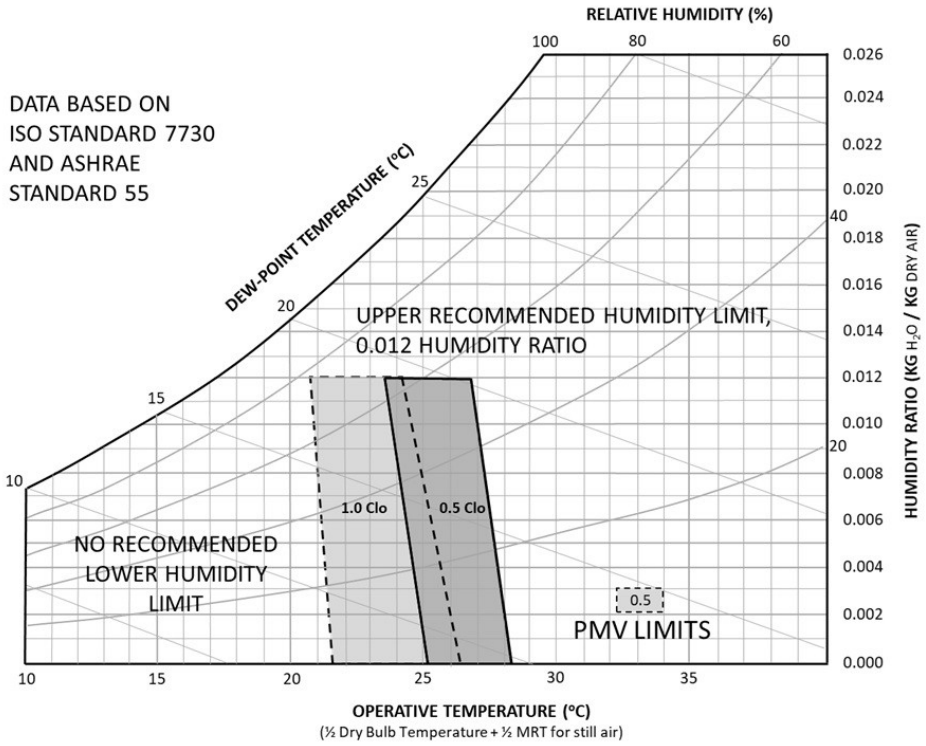


Figure 1. Graphical method for summer and winter comfort zones based on ASHRAE Standard 55-2010 [35]. Acceptable ranges of operative temperature and humidity at air speeds $\leq$ 0.2 m/s for people wearing 1.0 and 0.5 clo clothing during primarily sedentary activity ($\leq$1.1 met).

This method is used to predict the thermal sensation and degree of thermal dissatisfaction for people exposed to moderate indoor thermal environments. The PMV and PPD indices are used to calculate and interpret thermal comfort. This method is called the "Fanger method", referring to its developer, P.O. Fanger, who developed these analytical calculation indices for thermal comfort by considering various factors that contribute to the evaluation of the thermal indoor environment [36].

The PMV index shows the average value of the thermal sensation assigned by a large group of people on the ASHRAE thermal comfort scale, which consists of seven levels [35], based on the thermal equilibrium of the human body. Table 1 presents these seven levels. Thermal balance is achieved when the amount of internal body heat production is equal to the amount of heat loss expended to the environment. In an environment with a moderate thermal balance, the system that regulates the body's temperature (thermoregulatory system) will automatically try to modify the temperature of the skin and activate the secretion of sweat, seeking to maintain thermal equilibrium or balance.

Table 1. ASHRAE thermal sensation scale [35].

Score	Thermal Sensation
+3	Hot
+2	Warm
+1	Slightly warm
0	Neutral
−1	Slightly cool
−2	Cool
−3	Cold

When using the PMV index method, it is essential to estimate the amount of insulation provided by the clothing worn by the occupants of the space. It is difficult to know exactly how much insulation clothing provides, so this parameter is estimated using the tables predefined in ISO 7730; for example, 0 clo insulation is for a naked user, 0.5 clo insulation is for a short-sleeved shirt and trousers, 1.0 clo insulation is for a winter business suit, and 1.5 clo insulation is for heavy winter clothes.

The units used to measure the thermal insulation of clothing are clo or square meters kelvin per watt ($m^2 \cdot K/W$). To calculate the PMV index, the insulation value of the clothing worn by the occupants of the space, in $m^2 \cdot K/W$, is required. The conversion between the two units considers that 1 clo is equal to 0.155 $m^2 \cdot K/W$.

To meet the thermal insulation requirements for clothing, the metabolic rate is needed to calculate the PMV. The metabolic rate measures the amount of muscular energy required by the occupant of the space when performing a task in that space. When the occupant performs an activity, approximately 25% of the energy is used to perform the activity, and the rest of the energy expenditure is converted into heat.

The metabolic rate is expressed in met or watts per square meter (W/m^2). If the value of the metabolic rate is only available in met units, the following conversion is applied: 1 met is equal to 58.15 W/m^2.

For example, the metabolic rate calculated for sedentary work, such as that performed in offices or schools, is calculated at 1.2 met or 70 W/m^2 [34].

Other parameters needed to calculate the PMV index are the indoor air temperature in degrees Celsius, indoor mean radiant temperature (*MRT*) in degrees Celsius, indoor relative humidity in percent, and indoor air velocity in meters per second (m/s).

The *MRT* parameter describes the exchange of heat emitted by radiation between the human body and all surfaces around it. The *MRT* parameter is obtained from the measured values of the indoor dry-bulb temperature, indoor globe temperature, and indoor air velocity.

2.3. Procedure

The experimental procedure consists of the following steps: data collection and variable selection, *MRT* calculation, neural network architecture construction, neural network training, performance tests, and experimentation.

2.3.1. Data Collection and Variable Selection

At this stage, the process for measuring the variables to be used and the measurement equipment to be employed are defined. For the development of the Python 3.13 neural network algorithm used to calculate the mean radiant temperature (*MRT*), it is necessary to collect measurements of the dry-bulb temperature, globe temperature, and relative air velocity in different spaces; in this case, the measurements were taken in different classrooms at different hours and with different numbers of students. To take the measurements,

a PEAKMETER PM6252B digital anemometer, manufactured by Shenzhen Peakmeter Technology Co., Ltd., Shenzhen, China, was used for wind speed measurement, and an EXTECH HT200 thermal stress meter, manufactured by Extech Instruments, Nashua, NH, United States, was used for air temperature and globe temperature measurements.

2.3.2. *MRT* Calculation

At this stage, the method for performing the mathematical calculation of the *MRT* is defined. The mathematical formula specified in ISO 7726 will be used [37]. According to ISO 7726, the following variables are required to calculate the *MRT*: dry-bulb temperature, globe temperature, and relative air velocity.

2.3.3. Neural Network Architecture

The architecture of the neural network to be developed is basic: single-layer, two-layer, and three-layer, with three hidden layers. In the design of the neural network, it will be possible to modify the number of neurons in each hidden layer. It will also be possible to modify the activation function of the output layer.

2.3.4. Neural Network Training

Two different combinations of input data will be used to train the neural network. The neural network models will be tested using 20 and 104 input data points, respectively. Training will also be conducted using normalized data and different activation functions for the output layer. A total of 1000 epochs will be used for training.

2.3.5. Performance Tests

The next step is to evaluate the performance of the developed *MRT* neural network algorithm and compare the *MRT* obtained with the *MRT* calculated based on the in situ measurements with the aforementioned measurement equipment. Two performance tests will be conducted on the models to verify the predicted *MRT* values: the mean squared error (*MSE*) test and the coefficient of determination or R^2 (R-squared) test.

2.3.6. Experimentation

At this stage of the project, tests will be conducted on the different neural network models developed with different activation functions for the output layer; the neural network will be trained using a dataset expanded from 20 to 104; and additional tests will be performed on the three-layer neural network using two input variables.

3. Results and Discussion

3.1. Data Collection

The first step in achieving the project's proposed results was to develop the algorithm for calculating the mean radiant temperature (*MRT*). To do this, 104 measurements were taken in different classrooms at TecNM IT in Hermosillo over two weeks at different hours, with the objective of creating a database containing the indoor air temperature, indoor globe temperature, indoor wind speed, date, and hour. All measurements were taken in 2026.

The measurement instruments used were a PEAKMETER PM6252B digital anemometer for the wind speed and an EXTECH HT200 thermal stress meter for measuring the air temperature and globe temperature. Table 2 shows the collected data, including the date and hour.

Table 2. Measurements of environmental variables obtained.

Number	Globe Temperature (*GT*)	Air Velocity (V_a)	Air Temperature (T_a)	Occupation	Hour	Day
1	25.8	0.12	28.0	2	9:23	2 April
2	25.6	0.14	26.9	9	8:25	3 April
3	28.2	0.23	29.1	14	8:21	31 March
4	25.8	0.16	26.1	2	8:22	2 April
5	26.4	0.12	26.5	7	8:20	30 March
6	23.8	0.23	25.5	7	9:12	31 March
7	21.7	0.17	21.8	13	9:12	1 April
8	21.3	0.10	23.2	13	9:13	3 April
9	22.5	0.16	25.2	6	9:18	2 April
10	22.5	0.09	24.5	27	9:21	31 March
11	25.6	0.14	25.4	13	9:23	1 April
12	25.6	0.22	25.3	37	9:24	3 April
13	27.3	0.14	26.4	28	9:28	2 April
14	27.2	0.23	26.4	13	9:33	31 March
15	26.8	0.15	26.5	13	9:50	31 March
16	26.4	0.14	24.8	36	10:17	2 April
17	21.3	0.27	23.7	28	10:09	31 March
18	27.1	0.30	25.6	37	10:10	1 April
19	25.3	0.26	25.4	9	10:11	3 April
20	26.1	0.30	25.0	8	10:13	30 March
21	33.3	0.44	33.8	34	8:15	1 April
22	29.7	0.26	31.4	21	10:16	31 March
23	28.3	0.29	30.5	21	8:29	1 April
24	26.5	0.22	27.9	12	10:18	1 April
25	24.8	0.17	26.8	12	10:18	3 April
26	23.7	0.29	26.5	12	10:20	31 March
27	24.2	0.21	25.9	36	10:22	2 April
28	25	0.26	25.9	12	10:22	3 April
29	26.3	0.18	27.4	29	10.04	30 March
30	25.6	0.33	26.0	4	10:28	31 March
31	25.4	0.17	26.0	3	10:30	1 April
32	24.1	0.19	26.0	21	10:35	31 March
33	23.8	0.19	25.4	35	10:36	2 April
34	24.4	0.11	25.3	3	10:37	3 April
35	20	0.17	20.6	13	10:37	30 March
36	23.5	0.18	25.3	13	10:38	1 April
37	25	0.20	25.5	17	7:35	30 March
38	28.9	0.20	27.8	9	10:45	1 April
39	26.8	0.23	26.5	32	10:42	2 April
40	25	0.14	25.7	36	10:42	3 April
41	22.8	0.11	24.7	16	10:57	1 April
42	24.3	0.12	24.7	3	11:05	1 April
43	25.9	0.11	25.0	33	11:01	30 March
44	27.6	0.17	27.1	21	10:44	30 March
45	26.3	0.20	26.1	21	10:55	3 April
46	27.1	0.12	26.1	14	11:17	1 April
47	25.9	0.10	25.5	17	10:51	2 April
48	23.1	0.10	24.1	13	11:04	2 April
49	24.1	0.12	26.4	13	10:52	30 March
50	23.8	0.16	25.0	29	11:09	30 March
51	24.6	0.22	25.5	14	11:03	3 April
52	26.9	0.10	26.6	21	11:17	30 March

Table 2. *Cont.*

Number	Globe Temperature (*GT*)	Air Velocity (V_a)	Air Temperature (T_a)	Occupation	Hour	Day
53	26.1	0.23	26.6	19	11:12	3 April
54	25.9	0.14	26.6	19	15:01	6 April
55	25.5	0.11	26.5	14	15:18	6 April
56	25	0.14	25.8	42	15:01	8 April
57	27.1	0.12	27.5	21	14:51	10 April
58	25.9	0.21	25.5	42	14:59	7 April
59	26.6	0.13	26.0	27	15:02	9 April
60	26	0.10	26.3	10	15:11	9 April
61	24.6	0.21	26.5	23	15:29	6 April
62	25	3.23	23.9	27	15:10	7 April
63	26	0.12	25.6	21	15:19	9 April
64	25.4	0.13	26.0	38	15:02	10 April
65	25	0.14	25.8	38	15:38	6 April
66	25.7	0.10	25.5	9	15:14	8 April
67	25.9	0.11	25.6	7	15:45	6 April
68	26.2	0.14	25.8	16	15:15	10 April
69	26.3	0.14	26.4	2	15:22	8 April
70	26.4	0.14	25.9	28	15:23	9 April
71	27.6	0.17	27.1	43	15:35	8 April
72	28.4	0.10	28.0	32	15:56	6 April
73	26.4	0.10	26.3	31	15:22	7 April
74	26.3	0.29	26.1	19	15:27	9 April
75	27.1	0.12	27.5	19	15:42	8 April
76	28.1	0.12	27.7	14	15:29	10 April
77	29.1	0.14	28.4	9	16:04	6 April
78	24.3	0.12	25.1	27	15:49	8 April
79	25.2	0.19	26.0	5	15:41	10 April
80	25.1	0.12	26.5	33	16:21	6 April
81	27.2	0.10	28.0	9	15:36	9 April
82	26	0.12	27.7	38	15:39	7 April
83	24.1	0.19	25.4	28	15:44	9 April
84	24.9	0.10	25.1	30	16:33	6 April
85	25.1	0.14	25.2	32	16:41	6 April
86	25	0.23	25.9	27	15:59	8 April
87	25.6	0.12	25.6	27	15:44	7 April
88	25.4	0.23	26.0	13	15:52	7 April
89	26.7	0.14	25.8	28	16:07	8 April
90	24.1	0.10	25.5	28	15:49	10 April
91	27.1	0.12	27.5	28	16:00	10 April
92	26.2	0.20	24.1	27	16:19	8 April
93	26.6	0.13	24.5	27	15:58	7 April
94	26	0.15	26.3	33	16:12	10 April
95	24.6	0.21	26.5	28	16:25	8 April
96	25	0.31	23.9	23	16:09	7 April
97	26	0.12	25.6	6	15:53	9 April
98	27.1	0.30	25.6	28	16:26	10 April
99	25.3	0.26	25.4	28	16:24	7 April
100	26.1	0.30	25.0	21	16:35	9 April
101	27	0.14	25.9	21	16:03	9 April
102	27.6	0.17	26.3	38	16:42	10 April
103	29.1	0.10	27.9	39	16:41	7 April
104	26.4	0.10	26.3	42	16:49	6 April

3.2. MRT Calculation Procedure

After the data were collected, the equation for calculating the *MRT* was applied in accordance with ISO 7726 [37]. Equation (1) was used to calculate the *MRT* under forced convection, i.e., in environments with air circulation [15]:

$$MRT = \left[(GT + 273.15)^4 + \frac{1.1 \times 10^8 \times V_a{}^{0.6}}{\varepsilon \times D^{0.4}} \times (GT - T_a) \right]^{1/4} - 273.15 \quad (1)$$

where *MRT* is the mean radiant temperature in degrees Celsius (°C), *GT* is the globe temperature in degrees Celsius (°C), V_a is the indoor air velocity at the globe level in meters per second (m/s), ε is the emissivity of the globe (dimensionless), *D* is the diameter of the globe in meters (m), and T_a is the air temperature in degrees Celsius (°C).

The parameters ε and *D* are listed in the manufacturer's data sheet for the thermal stress meter EXTECH HT200; the values 0.95 and 0.05 are used, respectively. Table 3 shows the calculated MRTs obtained using Equation (1).

Table 3. Calculated *MRT* values using Equation (1) for measurements in Table 1.

Number	*MRT*	Number	*MRT*	Number	*MRT*
1	23.56	36	21.11	71	28.21
2	24.15	37	24.31	72	28.75
3	26.88	38	30.35	73	26.49
4	25.44	39	27.24	74	26.64
5	26.30	40	24.22	75	26.70
6	21.19	41	20.91	76	28.49
7	21.57	42	23.89	77	29.84
8	19.49	43	26.76	78	23.48
9	19.10	44	28.21	79	24.13
10	20.73	45	26.57	80	23.67
11	25.82	46	28.09	81	26.49
12	26.03	47	26.26	82	24.28
13	28.27	48	22.17	83	22.33
14	28.37	49	21.72	84	24.72
15	27.14	50	22.32	85	24.99
16	28.14	51	23.27	86	23.64
17	17.10	52	27.17	87	25.60
18	29.65	53	25.36	88	24.50
19	25.14	54	25.13	89	27.68
20	27.99	55	24.54	90	22.81
21	32.28	56	24.11	91	26.70
22	27.05	57	26.70	92	29.02
23	24.56	58	26.46	93	28.78
24	24.47	59	27.23	94	25.65
25	22.26	60	25.73	95	21.85
26	18.69	61	21.85	96	26.95
27	21.74	62	32.74	97	26.40
28	23.54	63	26.40	98	29.65
29	24.89	64	24.76	99	25.14
30	24.86	65	24.11	100	27.99
31	24.65	66	25.88	101	28.19
32	21.50	67	26.16	102	29.17
33	21.61	68	26.64	103	30.14
34	23.52	69	26.19	104	26.49
35	19.21	70	26.95		

Figure 2 shows the behavior of the air temperature (T_a) against that of the mean radiant temperature (*MRT*).

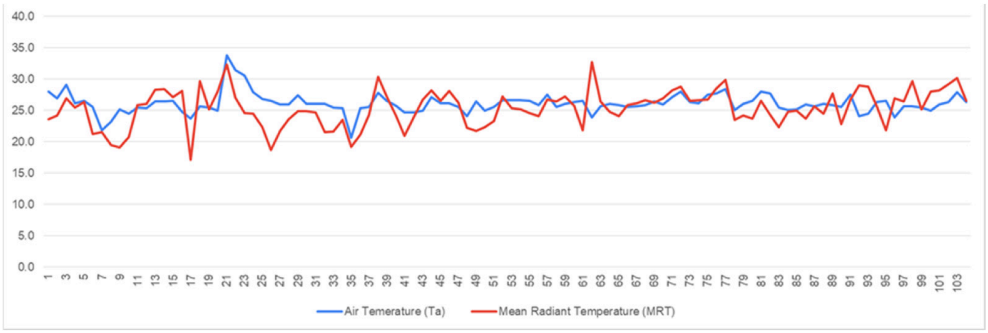


Figure 2. Air temperature and *MRT* behavior graph.

3.3. Development of Neural Network Architecture

The neural network architecture to be used is a basic single-layer, double-layer, and triple-layer configuration. A single-layer neural network is the simplest type of artificial neural network. This type of neural network maps an input vector directly to an output using a set of weights and a bias. The neural networks were programmed in Python using TensorFlow/Keras; additionally, the NumPy and Matplotlib libraries were used for calculations and graphical results, respectively. In the neural network models developed, the number of neurons to be used can be adjusted.

The developed neural network architecture consists of an input layer, weights (w), bias (b), hidden layers, and an activation function. The input layer receives the raw input data—in this case, the air temperature. The weights (w) are the scalable values assigned or selected for each input to determine its importance. The bias (b) is an adjustable constant added to the weighted sum to shift the decision boundary. The hidden layer is a vertical column between the input layer and the output layer in a neural network. The activation function (f) is the mathematical function (such as a step function, sigmoid, or ReLU) that transforms the weighted sum into a final output.

The weights and the bias are automatically selected by the TensorFlow library (because the program uses the tf.keras.layers.Dense instruction), and this creates and manages them behind the scenes for each layer.

The activation functions used to generate the output layer were ReLU, Mish, and Swish.

3.4. Neural Network Training Procedure

According to the neural network architecture, the next step was to create a Python program and train it with the available air temperature and *MRT* data, with the aim of having the neural network perform *MRT* calculations based on the air temperature value. Twenty data points were selected from Table 2 to train the neural network. The selected data contained values covering the lowest value, the highest value, and intermediate data for the calculated *MRT*. Table 4 shows the data selected for training the neural network.

Table 4. Selected data from Table 2 used to train the developed Python neural network.

Sample	Air Temperature (T_a)	Mean Radiant Temperature (*MRT*)
2	26.9	24.15
3	29.1	26.88
7	21.8	21.57

Table 4. *Cont.*

Sample	Air Temperature (T_a)	Mean Radiant Temperature (*MRT*)
8	23.2	19.49
12	25.3	26.03
17	23.7	17.10
21	33.8	32.28
22	31.4	27.05
24	27.9	24.47
24	30.5	24.56
35	20.6	19.21
38	27.8	30.35
40	25.7	24.22
44	27.1	28.21
60	26.3	25.73
65	25.8	24.11
67	25.6	26.16
68	26.1	26.64
84	25.1	24.72
85	25.2	24.99

Two neural networks were developed in Python to test the algorithm: a single-layer network and a two-layer network. Figure 3a shows the program developed for the single-layer neural network, and Figure 3b shows the program for the double-layer network.

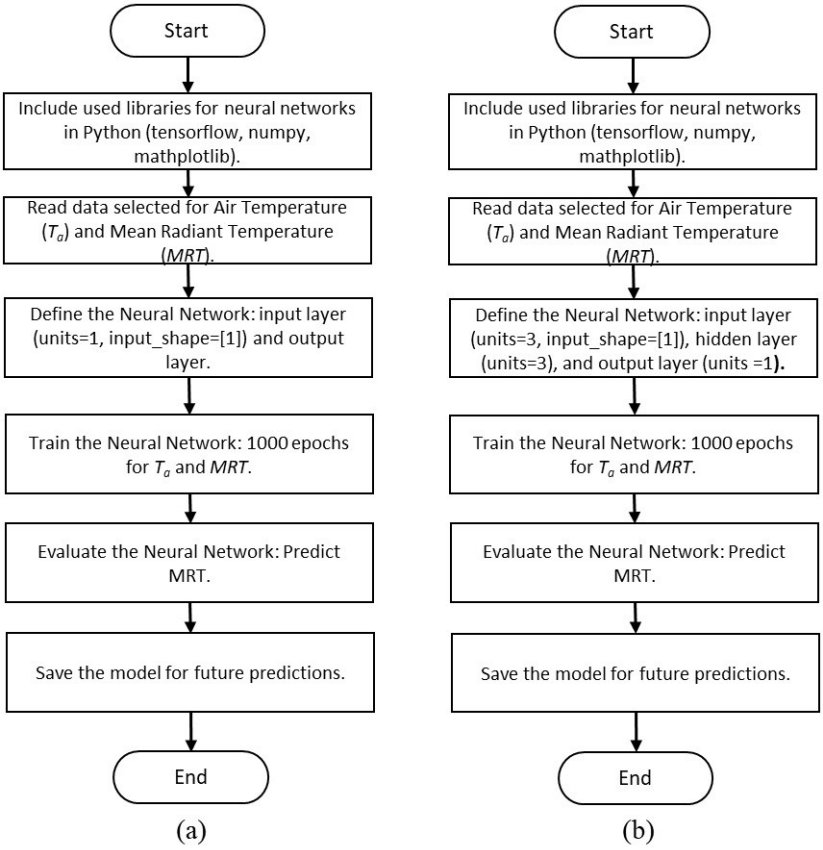


Figure 3. (**a**) Program developed for the single-layer neural network and (**b**) program developed for the double-layer network.

The algorithm was tested for functionality. Figure 4 shows the output of the Python program for the single-layer neural network. Figure 5 shows the magnitude-of-loss graph of the neural network for 1000 epochs, which shows that the neural network reaches zero loss after epoch 50.

```
Trained Model
Start Predictions ....
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 253ms/step
For Ta = 25.0 °C, Predicted MRT [[23.580835]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 169ms/step
For Ta = 26.0 °C, Predicted MRT [[24.483772]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 66ms/step
For Ta = 26.8 °C, Predicted MRT [[25.206121]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 69ms/step
For Ta = 27.5 °C, Predicted [[25.838179]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 66ms/step
For Ta = 28.0 °C, Predicted [[26.289646]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 65ms/step
For Ta = 33.8 °C, Predicted [[31.52668]] °C
>>>
```

Figure 4. Output of Python IDLE for the single-layer neural network for *MRT* prediction.

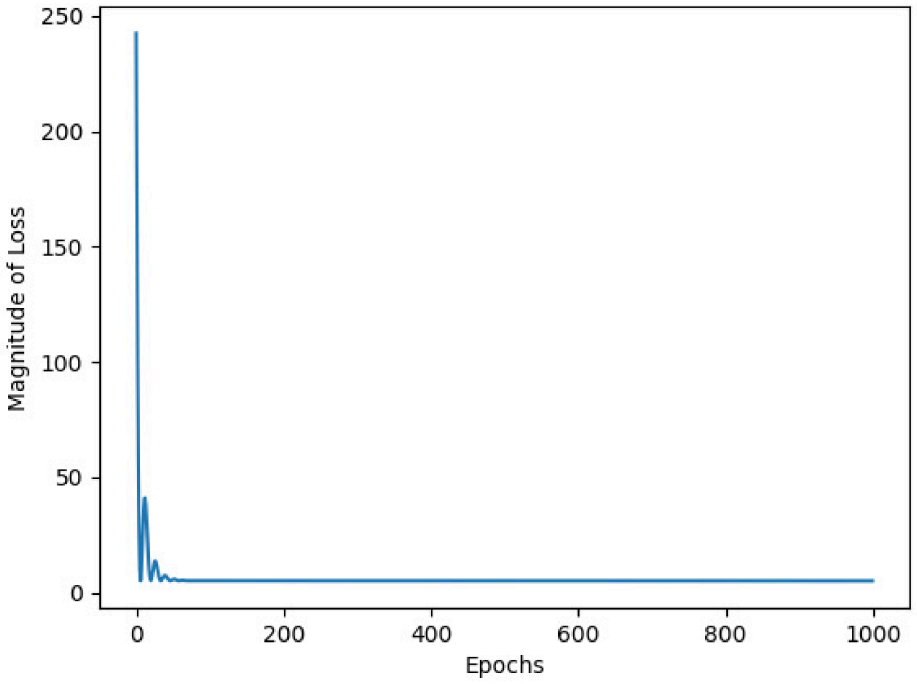


Figure 5. Python graph for the magnitude of loss of the single-layer neural network model.

Table 5 shows the *MRT* predictions obtained from the single-layer neural network developed in Python. It also shows the error (in percentage) obtained when comparing the calculated *MRT* (using the measurement equipment) with the predicted *MRT*. A maximum 13% error was observed; this error value represents 3 degrees Celsius. This value represents a problem with the PMV index calculation, because a 1 degree Celsius change in the *MRT* affects the PMV to virtually the same extent as a 1 degree Celsius change in the air

temperature. For this reason, it is necessary to run different configurations and models to try to reduce the error.

Table 5. Predicted *MRT* values for each air temperature value using the single-layer neural network in Python.

Number	Air Temperature (T_a)	Predicted *MRT*	Calculated *MRT*	Error (%)
1	25.0	23.58	26.76	11.88
2	26.0	24.48	24.65	0.69
3	26.8	25.21	22.26	13.25
4	27.5	25.84	26.70	3.22
5	28.0	26.29	26.49	0.76
6	33.8	31.52	32.88	4.14

For comparative purposes, a second experiment has also been performed using a model with a double-layer neural network. Figure 6 shows the output of the Python program execution for the double-layer neural network model. Figure 7 shows the magnitude-of-loss graph for the neural network with 1000 epochs, showing that the neural network reaches zero loss by epoch 35.

```
Trained Model
Start Predictions ....
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 261ms/step
For Ta = 25.0 °C, Predicted MRT [[23.63551]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 268ms/step
For Ta = 26.0 °C, Predicted MRT [[24.507833]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 67ms/step
For Ta = 26.8.0 °C, Predicted MRT [[25.205692]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 57ms/step
For Ta = 27.5 °C, Predicted MRT [[25.816319]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 66ms/step
For Ta = 28.0 °C, Predicted MRT [[26.252483]] °C
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 61ms/step
For Ta = 33.8 °C, Predicted MRT [[31.311962]] °C
>>>
```

Figure 6. Python IDLE output for the double-layer neural network for *MRT* prediction.

Table 6 shows the *MRT* predictions obtained from the single-layer neural network developed in Python. It also reports the error (in percentage) obtained when comparing the calculated *MRT* (using the measurement equipment) with the predicted *MRT*. In this model configuration, a maximum 13% error was observed, which is also a problematic error value, and more different models and configurations are needed.

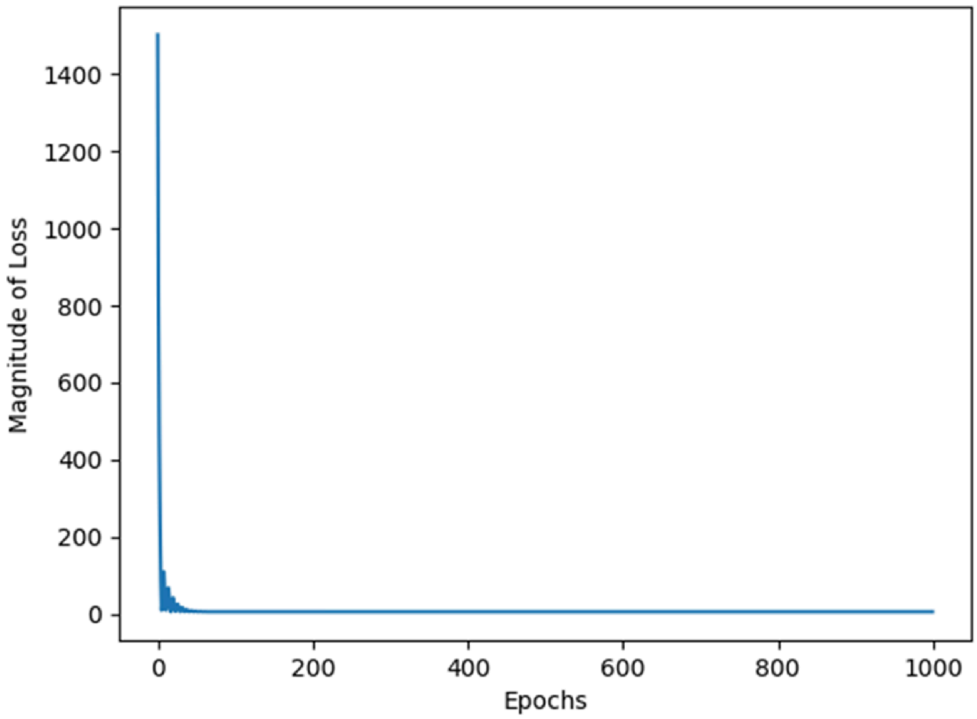


Figure 7. Python graph for the magnitude of loss of the double-layer neural network model.

Table 6. Predicted *MRT* values for each air temperature value using the double-layer neural network in Python.

Number	Air Temperature (T_a)	Predicted *MRT*	Calculated *MRT*	Error (%)
1	25.0	23.64	26.76	11.69
2	26.0	24.51	24.65	0.57
3	26.8	25.21	22.26	13.25
4	27.5	25.82	26.70	3.29
5	28.0	26.25	26.49	0.91
6	33.8	31.31	32.88	4.77

Tables 5 and 6 show that the differences obtained in the *MRT* predictions of the single-layer neural network and the double-layer neural network are practically the same, with differences of tenths or hundredths. Therefore, using either of the two models when predicting the *MRT* does not have a significant impact.

In order to ensure that the network training did not depend on the number of program executions (of the neural network model), ten executions of each Python program were performed, i.e., ten for the single-layer neural network model and five for the double-layer neural network model. Tables 7 and 8 show the results, respectively.

Table 7. Results obtained from ten runs of the predicted *MRT* value for each air temperature value using the single-layer neural network in Python.

Air Temperature (T_a)	1	2	3	4	5	6	7	8	9	10
25.0	23.5626	23.6701	23.6730	23.6635	23.6584	23.6355	23.6014	23.6583	23.6164	23.6228
26.0	24.4757	24.5230	24.5243	24.5202	24.5179	24.5078	24.4928	24.5178	24.4994	24.5022
26.8	25.2063	25.2054	25.2054	25.2055	25.2055	25.2057	25.2060	25.2055	25.2058	25.2058

Table 7. *Cont.*

Air Temperature (T_a)	1	2	3	4	5	6	7	8	9	10
27.5	25.8455	25.8025	25.8013	25.8051	25.8071	25.8164	25.8300	25.8072	25.8239	25.8214
28.0	26.3021	26.2290	26.2270	26.2334	26.2369	26.2325	26.2757	26.2370	26.2655	26.2611
33.8	31.5985	31.1761	31.1647	31.2019	31.2219	31.3122	31.4459	31.2225	31.3869	31.3618

Table 8. Results obtained from ten runs of the predicted *MRT* value for each air temperature value using the double-layer neural network in Python.

Air Temperature (T_a)	1	2	3	4	5	6	7	8	9	10
25.0	23.6318	23.6347	23.6250	23.6353	23.6237	23.6332	23.6221	23.6336	23.6341	23.6353
26.0	24.5062	24.5075	24.5032	24.5077	24.5026	24.5068	24.5019	24.5070	24.5072	24.5077
26.8	25.2057	25.2057	25.2058	25.2057	25.2058	25.2057	25.2058	25.2057	25.2057	25.2057
27.5	25.8177	25.8166	25.8205	25.8164	25.8211	25.8172	25.8217	25.8171	25.8169	25.8164
28.0	26.2549	26.2530	26.2596	26.2526	26.2606	26.2540	26.2616	26.2537	26.2534	26.2526
33.8	31.3262	31.3151	31.3532	31.3129	31.3585	31.3209	31.3647	31.3193	31.3175	31.3128

3.5. Performance Tests Evaluated

In order to test the predicted *MRT* values, a mean squared error (*MSE*) test was performed. The *MSE* test calculates the average of the squared errors—that is, the difference between the calculated *MRT* values and the *MRT* neural network's predictions. The *MSE* formula is defined as

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

where y_i is the calculated *MRT* value, $\hat{y}_i$ is the predicted *MRT* value, and n is the total number of samples evaluated. The *MSE* calculation process consists of the following: subtracting the predicted *MRT* value from the calculated *MRT* value for each data point (calculating the error); squaring each result (this eliminates negative signs and penalizes large errors more heavily); summing all these squared values; and dividing the result by the total number of data points. The *MSE* result is expressed in "squared units"; for the error in the original units, a simple square root of the result is needed to obtain the root mean squared error (RMSE).

Table 9 shows the *MSE* and RMSE calculations for the single-layer neural network and the double-layer neural network using the ReLU activation function.

Table 9. Mean squared error (*MSE*) for the single-layer neural and double-layer neural networks.

Number	Calculated *MRT*	Predicted *MRT* (Single-Layer NN)	Predicted *MRT* (Double-Layer NN)
1	26.76	23.58	23.64
2	24.65	24.48	24.51
3	22.26	25.21	25.21
4	26.70	25.84	25.82
5	26.49	26.29	26.25
6	32.88	31.52	31.31
	Mean Squared Error (*MSE*) Value	4.27	3.62
	Root Mean Squared Error (RMSE) Value	2.06	1.90

In Table 9, it is possible to observe that the single-layer neural network has an RMSE of 2.06 degrees Celsius, while the two-layer neural network has an RMSE of 1.90 degrees

Celsius, which indicates that the two-layer neural network performs better, with a lower RMSE. Figures 8 and 9 show the residual plots for the single-layer neural network in Python and the double-layer neural network, respectively.

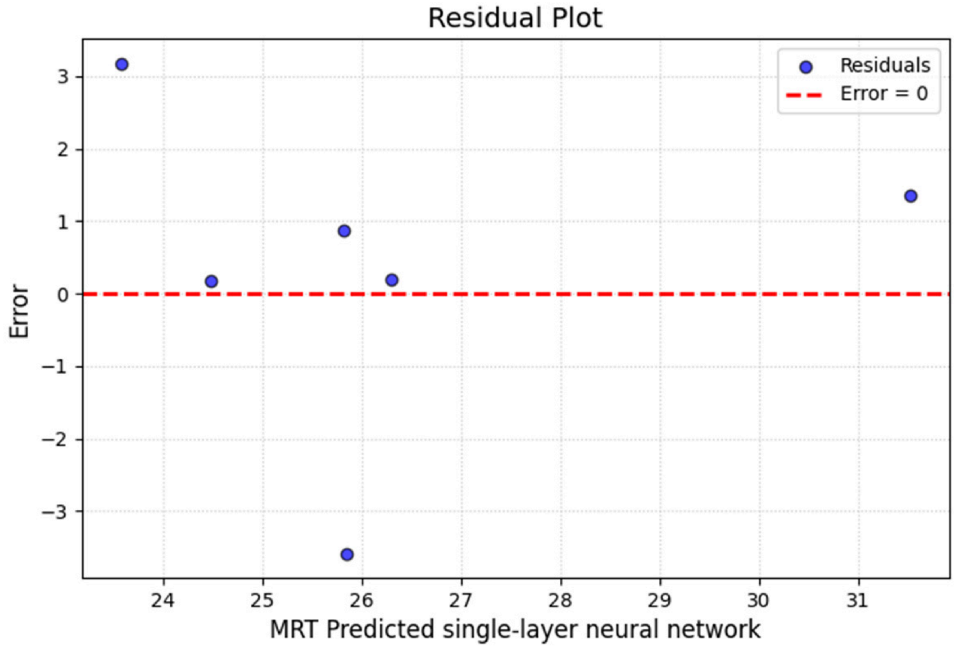


Figure 8. Python residual plot for the single-layer neural network model.

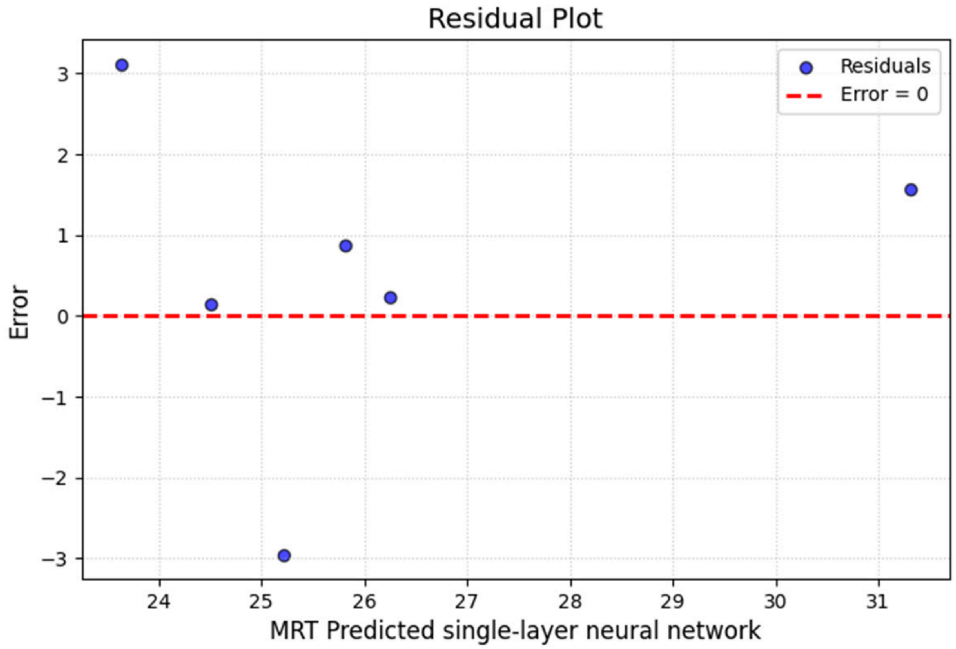


Figure 9. Python residual plot for the double-layer neural network model.

Another numerical test for evaluating neural networks in regression tasks is the coefficient of determination, or R^2 (R-squared), test. The R^2 is a standardized metric that measures the percentage of variation in the actual data that the neural network can explain and predict. The results of the R^2 test are easy to interpret: $R^2 = 1$ means that the model is perfect and explains 100% of the variability in the data; $R^2 = 0$ means that the model is no better than simply guessing the average of the data every time; $R^2 < 0$ (negative

values) means that the model is extremely poor, and its predictions are worse than using the average.

Table 10 shows the R^2 values for the single-layer and double-layer neural network models.

Table 10. R-squared (R^2) coefficients for the single-layer neural network and double-layer neural network.

Number	Calculated *MRT*	Predicted *MRT* (Single-Layer NN)	Predicted *MRT* (Double-Layer NN)
1	26.76	23.58	23.64
2	24.65	24.48	24.51
3	22.26	25.21	25.21
4	26.70	25.84	25.82
5	26.49	26.29	26.25
6	32.88	31.52	31.31
	Coefficient of determination (R^2)	0.58	0.65

The single-layer neural network model (R^2 = 0.58) has only one hidden layer, so its behavior is very similar to that of a simple linear or mathematical regression. An R^2 of 58% indicates that the model identifies a clear trend but lacks the "power" to capture the details. On the other hand, the double-layer neural network (R^2 = 0.65) has a second hidden layer, and this characteristic allows for more in-depth representations.

The increase in the R^2 value from 0.58 to 0.65 when moving from the single-layer to the double-layer configuration demonstrates that the data have complex nonlinear relationships that a single-layer network cannot fully capture. By adding the second layer, the model has the ability to combine features and better understand the data's behavior.

3.6. Testing with Different Activation Functions

The two neural networks were developed using the Rectified Linear Unit (ReLU) activation function for the hidden layers and a linear activation for the output layer. The hidden layer was developed with three neurons in each layer. In an effort to increase the R^2 coefficient, various neural network models were tested. The first involved adding a hidden layer (triple layer); then, in the triple-layer model, different activation functions were tested, along with normalized data and varying numbers of neurons in the hidden layers. The activation functions used were ReLU, Mish, and Swish. The results are shown in Table 11.

Table 11. Predicted *MRT* values for different triple-layer neural network models.

	ReLU (64-32-16)	Mish (64-32-16)	Normalized Data ReLU (64-32-16)	Normalized Data Mish (64-32-16)	Normalized Data ReLU (16-8-4)	Normalized Data Swish (16-8-4)
	23.64	23.63	24.24	23.39	23.33	23.81
	24.51	24.52	25.64	25.22	26.65	25.56
	25.21	25.22	26.40	26.19	26.54	26.37
	25.81	25.84	27.06	27.32	26.40	26.52
	26.25	26.28	27.17	27.42	26.29	26.47
	31.28	31.31	32.20	32.39	32.25	30.77
R^2	0.6480	0.6487	0.5891	0.5393	0.4429	0.5025

According to the results shown in Table 11, the R^2 coefficient does not change when the model is modified from two to three hidden layers or when the number of neurons is

increased from 3-3 in a double hidden layer to 64-32-16 in a triple hidden layer; in other words, the double-layer model provides a very good approximation of the *MRT* value using the air temperature parameter. Similarly, there is no difference in the R^2 coefficient when changing the activation function from ReLU to Mish. If the input data are normalized, the R^2 coefficient drops below 60%, which indicates that the *MRT* predictions are less reliable.

The neural network model with two or three hidden layers is the one that provides the most reliable data based on the R^2 coefficient; therefore, this model is best suited for predicting the *MRT* based on the air temperature.

3.7. Three-Layer Neural Network with Additional Data

To increase the R^2 value, a more robust neural network was tested. The amount of input data was increased from the 20 selected data points to the total dataset—that is, from 20 data points to 104 data points. The 104 data points used are shown in Table 2 in the "Air Temperature (T_a)" column and in Table 3 in the "MRT" column. A three-hidden-layer model with 16-8-4 neurons and a Swish activation function was programmed. Table 12 shows the results obtained with this model.

Table 12. Predicted *MRT* values in a triple-layer neural network with 104 data points.

Number	Calculated *MRT*	Predicted *MRT* (Single-Layer NN)
1	26.76	25.07
2	24.65	25.37
3	22.26	25.96
4	26.70	26.78
5	26.49	27.34
6	32.88	32.31
	Coefficient of determination (R^2)	0.7083

According to the results shown in Table 12, increasing the number of input data points from 20 to 104 allowed us to achieve an R^2 of 0.7083; this result mathematically confirms that the neural network needed more data to learn and improve its reliability. Explaining nearly 71% of the variability in the mean radiant temperature (*MRT*) using only the air temperature (T_a) is a very good result, and the neural network model can be used to predict the *MRT*.

3.8. Three-Layer Neural Network with Two Inputs

In a final attempt to increase the R^2 value, a more robust neural network was tested. The neural network developed had two input values, the air temperature and occupation. The amount of input data was again increased from the 20 selected data points to the total dataset—that is, from 20 data points to 104 data points. The 104 data points used are shown in Table 2 in the "Air Temperature (T_a)" column and the "Occupation (O_c)" column and in Table 3 in the "MRT" column. A three-hidden-layer model with 16-8-4 neurons and a Swish activation function was programmed. A total of 3000 epochs were used to train the model. Table 13 shows the results obtained with this model.

Table 13. Predicted *MRT* values in a triple-layer neural network with 104 data points with "Air Temperature" and "Occupation" inputs.

Number	Calculated *MRT*	Occupation	Predicted *MRT* (Single-Layer NN)
1	26.76	13	25.31
2	24.65	19	22.53
3	22.26	30	22.49
4	26.70	28	26.32
5	26.49	28	26.67
6	32.88	42	32.22
	Coefficient of determination (R^2)		0.8918

According to the results shown in Table 12, increasing the number of input data points from 20 to 104 and including a second input variable (air temperature and occupation) allowed us to achieve an R^2 of 0.8918; this result mathematically confirms that the neural network needed more data to learn and increase its reliability, and adding additional inputs helps to estimate the *MRT* more accurately. Explaining nearly 90% of the variability in the mean radiant temperature (*MRT*) using the air temperature (T_a) and occupation (O_c) leads to better results, but it involves adding sensors to measure the number of occupants.

3.9. Embedded System for Testing the Neural Network Algorithm

The developed neural network algorithm is tested in a control system implemented on a Raspberry Pi. The control system consists of a sensor node (to sense the air temperature and occupancy), an electrical consumption meter, and an actuator node. The control system is the same as that used by the authors in [38]. The control system developed and installed on the Raspberry Pi calculates the PMV index and sends turn-on/turn-off activation signals to the HVAC compressor. The Python program calculates the PMV index according to ASHRAE and ISO 7730.

In the control system program, the ambient temperature and occupancy are measured through the node sensor, and mean radiant temperature values are assigned using the value corresponding to the neural network algorithm prediction. According to ISO 7730, the external work variable W is typically assumed to be zero, so W = 0. The relative humidity (RH) is obtained from the sensor node. Then, only three values are required to calculate the indices: the clothing insulation value, the metabolic rate, and the air velocity.

4. Conclusions

The principal goal of energy efficiency improvement in buildings is to save electrical energy used in building operations by controlling energy consumption. One strategy to achieve energy savings in buildings is to optimize the HVAC system. It is very important to optimize HVAC systems for energy savings, because these systems represent a higher consumption load. A balance in the amount of energy consumption of the HVAC system is obtained through the operating conditions of the building relative to the cooling and heating needs of the building and the amount of energy needed to reach these operating conditions.

The use of neural networks to estimate parameters that influence set point calculation—particularly those that are difficult to measure—in HVAC systems can help to achieve energy savings. Developing algorithms that use neural networks to calculate the PMV index, as recommended by ASHRAE for HVAC systems, can greatly contribute to achieving energy savings while maintaining occupant comfort.

During the development of the neural network model to calculate the *MRT*, it was observed that the single-hidden-layer and double-hidden-layer models produced R^2 values of 0.58 and 0.65, respectively. The increase from 0.58 to 0.65 when moving from one layer to two clearly demonstrates that the data have complex nonlinear relationships that a single-layer network cannot fully capture. By adding the second layer, the model was able to combine features and better understand the behavior of the data. Regarding the single-layer model (R^2 = 0.58), a single hidden layer typically behaves very similarly to a simple linear or mathematical regression. The value of 58% indicates that there is a clear trend that the model identifies, but it lacks the "power" to capture the details. Regarding the two-layer model (R^2 = 0.65), the second layer allows for deeper representations. A seven-point increase in the R^2 is very good progress and confirms that the two-layer model delivers better results.

The three-layer neural network model, with normalized data and using 104 input data points, yields an R-squared value of 0.71. Explaining nearly 71% of the variability in the mean radiant temperature (*MRT*) using only the air temperature (T_a) is a very good result. Therefore, it is possible to use this neural network model to estimate the *MRT* value based on the air temperature, which requires installing only one sensor in the space where the HVAC equipment is used to calculate the set point and reduce the energy consumption.

Using a second input variable (occupancy) in the developed neural network model increased the R^2 coefficient to 0.89. The increase in the coefficient of determination (R^2) from 0.71 to 0.89 represents a significant improvement in the neural network's predictive power and accuracy. However, this model would require the addition of sensors to measure the other variable, which would complicate the design of the control system needed to implement this model.

Developing neural networks in Python is relatively straightforward and can be used for various applications aimed at estimating variables or parameters that are difficult to measure or require extremely expensive measurement equipment; this is achieved by performing such calculations through another associated variable that is easier to obtain.

A reduction in energy consumption for HVAC equipment can be achieved if the proposed *MRT* calculation algorithm is used instead of the air temperature value in the control system using the ASHRAE PMV index.

Future detailed analysis of the proposed neural network algorithm will be performed in the control system to measure energy consumption and probe the control strategy used. The project is still in progress, and more information is being gathered to perform statistical analyses and validate the energy consumption behavior.

Author Contributions: Conceptualization, D.F.E.-B. and J.A.H.-M.; methodology, D.F.E.-B.; software, D.F.E.-B., C.A.P.-P., R.d.C.G.-U. and F.A.H.-A.; validation, D.F.E.-B., J.A.H.-M., R.d.C.G.-U., F.A.H.-A., C.A.P.-P., J.O.R.-N. and E.C.-M.; formal analysis, D.F.E.-B., J.A.H.-M., R.d.C.G.-U., F.A.H.-A., C.A.P.-P., J.O.R.-N. and E.C.-M.; investigation, D.F.E.-B. and J.A.H.-M.; resources, D.F.E.-B., F.A.H.-A. and J.A.H.-M.; data curation, D.F.E.-B. and J.A.H.-M.; writing—original draft preparation, D.F.E.-B., J.A.H.-M., R.d.C.G.-U., F.A.H.-A., C.A.P.-P., J.O.R.-N. and E.C.-M.; writing—review and editing, D.F.E.-B., J.A.H.-M., R.d.C.G.-U., F.A.H.-A., C.A.P.-P., J.O.R.-N. and E.C.-M.; visualization, D.F.E.-B. and J.A.H.-M.; supervision, D.F.E.-B.; project administration, D.F.E.-B. and J.A.H.-M.; funding acquisition, D.F.E.-B. All authors have read and agreed to the published version of the manuscript.

Funding: The first author was funded by Tecnológico Nacional de México under commission 17610.23-P.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors thank Tecnológico Nacional de México (TecNM) for their support through the Call for Scientific and Technological Research Proposals 2026 program. Additional

support was received from the Program for the Professional Development of Professors for Higher Education (PRODEP) of the Mexican Ministry of Education (SEP).

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

References

1. Pérez-Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [CrossRef]
2. Shaikh, P.H.; Nor, N.B.M.; Nallagownden, P.; Elamvazuthi, I.; Ibrahim, T. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renew. Sustain. Energy Rev.* **2014**, *34*, 409–429. [CrossRef]
3. Ma, L.; Liu, N.; Wang, L.; Zhang, J.; Lei, J.; Zeng, Z.; Wang, C.; Cheng, M. Multi-party energy management for smart building cluster with PV systems using automatic demand response. *Energy Build.* **2016**, *121*, 11–21. [CrossRef]
4. Mujeebu, M.A.; Alshamrani, O.S. Prospects of energy conservation and management in buildings—The Saudi Arabian scenario versus global trends. *Renew. Sustain. Energy Rev.* **2016**, *58*, 1647–1663. [CrossRef]
5. Lee, Y.-C.; Chuang, C.-Y.; Chen, Y.-C.; Hsueh, C.-W. An efficient scheduling mechanism for building energy management systems. In Proceedings of the 2015 IEEE International Conference on Building Efficiency and Sustainable Technologies, Singapore, 17–22 September 2015. [CrossRef]
6. Ghorbanpanah, H.; Moghaddam, M.H.Y.; Saeedi, A.; Alishahi, S. Design and implementation of building energy monitoring system using wireless sensor networks. *IET Conf. Publ.* **2013**, *2013*, 230–233. [CrossRef]
7. Lee, D.; Cheng, C.-C. Energy savings by energy management systems: A review. *Renew. Sustain. Energy Rev.* **2016**, *56*, 760–777. [CrossRef]
8. Chen, C.-N.; Cho, M.-Y.; Lee, C.-H. Design and Implementation of Building Energy Management System. In *Proceedings of the 2016 3rd International Conference on Green Technology and Sustainable Development (GTSD), Kaohsiung, Taiwan, 24–25 November 2016*; IEEE: New York, NY, USA; pp. 106–111. [CrossRef]
9. Gilchrist, A. *Industry 4.0: The Industrial Internet of Things*, 1st ed.; Apress: Bangkok, Thailand, 2016. [CrossRef]
10. Oldewurtel, F.; Sturzenegger, D.; Morari, M. Importance of occupancy information for building climate control. *Appl. Energy* **2013**, *101*, 521–532. [CrossRef]
11. Park, J.; Kim, T.; Lee, C.S. Development of thermal comfort-based controller and potential reduction of the cooling energy consumption of a residential building in Kuwait. *Energies* **2019**, *12*, 3348. [CrossRef]
12. Cao, X.; Dai, X.; Liu, J. Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy Build.* **2016**, *128*, 198–213. [CrossRef]
13. Hassan, J.S.; Zin, R.M.; Majid, M.Z.A.; Balubaid, S.; Hainin, M.R. Building energy consumption in Malaysia: An overview. *J. Teknol.* **2014**, *70*, 33–38. [CrossRef]
14. Cámara de Diputados del H. Congreso de la Unión. *Ley General de Cambio Climático*; Cámara de Diputados del H. Congreso de la Unión, Distrito Federal: Ciudad de Mexico, Mexico, 2012.
15. Fundamentals (SI). In *2025 ASHRAE Handbook*; ASHRAE: Peachtree Corners, GA, USA, 2025; pp. 1–1178, ISBN 9781964173115. Available online: https://store.accuristech.com/standards/2025-ashrae-handbook-fundamentals-si?product_id=3009593 (accessed on 21 September 2026).
16. Thomas, B.; Soleimani-Mohseni, M. Artificial neural network models for indoor temperature prediction: Investigations in two buildings. *Neural Comput. Appl.* **2007**, *16*, 81–89. [CrossRef]
17. De Mello, A.S.; Miranda, A.; De Lima, M. Deep neural networks for estimation of temperature values for thermal ageing evaluation of nuclear power plant equipment. *Prog. Nucl. Energy* **2023**, *156*, 1–7. [CrossRef]
18. Shank, D.B.; Hoogenboom, G.; McClendon, R.W. Dewpoint Temperature Prediction Using Artificial Neural Networks. *J. Appl. Meteorol. Climatol.* **2008**, *47*, 1757–1769. [CrossRef]
19. Soleimani-Mohseni, M.; Thomas, B.; Fahlén, P. Estimation of operative temperature in buildings using artificial neural networks. *Energy Build.* **2006**, *38*, 635–640. [CrossRef]
20. Ruano, A.E.; Crispim, E.M.; Conceição, E.Z.E.; Lúcio, M.M.J.R. Prediction of building ' s temperature using neural networks models. *Energy Build.* **2006**, *38*, 682–694. [CrossRef]
21. Park, J.; Kim, T.; Kim, D.; Alghimlas, F.; AlRagom, F.; Choi, H.; Cho, H. Field test of machine-learning based mean radiant temperature estimation methods for thermal comfort-integrated air-conditioning control improvement and energy savings. *Energy Rep.* **2024**, *11*, 5682–5702. [CrossRef]
22. Duan, X.; Yu, S.; Chu, J.; Chen, D.; Chen, Y. Evaluation of Indoor Thermal Environments Using a Novel Predicted Mean Vote Model Based on Artificial Neural Networks. *Buildings* **2022**, *12*, 1880. [CrossRef]
23. Boutahri, Y.; Tilioua, A. Machine learning-based predictive model for thermal comfort and energy optimization in smart buildings. *Results Eng.* **2024**, *22*. [CrossRef]

24. Xie, Y.; Hu, W.; Zhou, X.; Yan, S.; Li, C. Artificial Neural Network Modeling for Predicting and Evaluating the Mean Radiant Temperature around Buildings on Hot Summer Days. *Buildings* **2022**, *12*, 513. [CrossRef]
25. Donaisky, E.; Oliveira, G.H.C.; Freire, R.Z.; Mendes, N. PMV-Based Predictive Algorithms for Controlling Thermal Comfort in Building Plants. In Proceedings of the 16th IEEE International Conference on Control Applications, Singapore, 1–3 October 2007; pp. 182–187. [CrossRef]
26. Freire, R.Z.; Oliveira, G.H.C.; Mendes, N. Predictive controllers for thermal comfort optimization and energy savings. *Energy Build.* **2008**, *40*, 1353–1365. [CrossRef]
27. Chang, Y.Y.; Lin, Y.P. PMV-based genetic algorithms for indoor temperature control system. In Proceedings of the 2016 IEEE International Symposium on Computer, Consumer and Control, IS3C 2016, Xi'an, China, 4–6 July 2016; pp. 295–298. [CrossRef]
28. Ku, K.L.; Liaw, J.S.; Tsai, M.Y.; Liu, T.S. Automatic control system for thermal comfort based on predicted mean vote and energy saving. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 378–383. [CrossRef]
29. Farzaneh, Y.; Tootoonchi, A.A. Controlling automobile thermal comfort using optimized fuzzy controller. *Appl. Therm. Eng.* **2008**, *28*, 1906–1917. [CrossRef]
30. Atmaca, I.; Yigit, A. Predicting the effect of relative humidity on skin temperature and skin wettedness. *J. Therm. Biol.* **2006**, *31*, 442–452. [CrossRef]
31. Toftum, J.; Jørgensen, A.S.; Fanger, P.O. Upper limits for indoor air humidity to avoid uncomfortably humid skin. *Energy Build.* **1998**, *28*, 1–13. [CrossRef]
32. Jang, M.S.; Koh, C.D.; Moon, I.S. Review of thermal comfort design based on PMV/PPD in cabins of Korean maritime patrol vessels. *Build. Environ.* **2007**, *42*, 55–61. [CrossRef]
33. *NMX-C-7730-ONNCCE-2018*; Industria de la Construcción—Ergonomía del Ambiente Térmico—Determinación Analítica e Interpretación del Confort Térmico Mediante el Cálculo De Los Índices vme y Pei y los Criterios De Confort Térmico Local. S. C. ONNCCE—Organismo Nacional de Normalización y Certificación de la Construcción y Edificación: Ciudad de Mexico, Mexico, 2019.
34. *UNE-EN ISO 7730*; Ergonimía del Ambiente Térmico. 2006. Asociación Española de Normalización y Certificación (AENOR): Madrid, Spain, 2016.
35. *ANSI/ASHRAE Standard 55-2010*; Thermal Environmental Conditions for Human Occupancy. Inc. American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE): Peachtree Corners, GA, USA, 2010.
36. Fanger, P.O. *Thermal Comfort: Analysis and Applications in Enviromental Engineering*; McGraw-Hill: New York, NY, USA, 1972.
37. *ISO 7726:2001*; Ergonomics of the Thermal Environment—Instruments for Measuring Physical Quantities. International Standars Organization: Geneva, Switzerland, 1998.
38. Espejel-Blanco, D.F.; Hoyo-Montaño, J.A.; Arau, J.; Valencia-Palomo, G.; García-Barrientos, A.; Hernández-De-León, H.R.; Camas-Anzueto, J.L. HVAC Control System Using Predicted Mean Vote Index for Energy Savings in Buildings. *Buildings* **2022**, *12*, 38. [CrossRef]

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.