

## Research Article

# Use of Artificial Neural Networks to Predict Wind-Induced External Pressure Coefficients on a Low-Rise Building: A Comparative Study

**Josué U. Rodríguez-Alcántara** , **Adrián Pozos-Estrada** ,  
and **Roberto Gómez-Martínez** 

*Instituto de Ingeniería, UNAM, Mexico City 04510, Mexico*

Correspondence should be addressed to Adrián Pozos-Estrada; [apozose@iingen.unam.mx](mailto:apozose@iingen.unam.mx)

Received 2 March 2022; Accepted 12 August 2022; Published 5 September 2022

Academic Editor: Hossein Nassiraei

Copyright © 2022 Josué U. Rodríguez-Alcántara et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wind flow on a bluff body is a complex and nonlinear phenomenon that has been mainly studied experimentally or analytically. Several mathematical methods have been developed to predict the wind-induced pressure distribution on bluff bodies; however, most of them result impractical due to the mathematical complexity required. Long-short term memory artificial neural networks with deep learning have proven to be efficient tools in the solution of nonlinear phenomena, although the choice of a more efficient network model remains a topic of open discussion for researchers. The main objective of this study is to develop long-short term memory artificial neural network models to predict the external pressure distribution of a low-rise building. For the development of the artificial neural network models, the multilayer perceptron and the recurrent neural network were also employed for comparison purposes. To train the artificial neural networks, a database with the external pressure coefficients from boundary layer wind tunnel tests of a low-rise building is employed. The analysis results indicate that the long-short term memory artificial neural network model and the multilayer perceptron neural network outperform the recurrent neural network.

## 1. Introduction

The study of wind effects on low-rise buildings is carried out by using the wind-induced pressure distribution over the structure. This pressure distribution can be calculated from experimental wind tunnel tests [1] or by using computational wind engineering [2]. Based on the pressure distribution, international wind design codes and standards propose the use of external pressure coefficients (EPC) in order to calculate the wind-induced forces.

The prediction of EPC due to the incidence of wind on bluff bodies has been an important topic for researchers in the last decades. In particular, the study of the pressure distribution near the edges of the structure, since wind effects are often characterized as a nonlinear problem in these areas [3–5].

Database-assisted design (DAD) has been proposed for the design and revision of buildings under wind loads [6–9]. In

a DAD, aerodynamic information from experimental wind tunnel tests has been used; however, the scarce information on representative building models has been an obstacle to using it. Recently, researchers have developed and applied complex mathematical models to characterize or simulate the turbulent flow of wind within an environment with certain characteristics of natural roughness [10–14]. Other studies that include novel methodologies are those by Wan et al., [15] where a model based on support vector machine regression (SVR) and kernel ridge regression (KRR) was used to predict wind speed records; Wan et al. [16] developed a new universal power law based in the use of a wavelet multi-scale transform algorithm to predict wind speed; Li et al. [17] used a least-squares support-vector machine (LSSVM) model with parameter optimization to forecast wind speed; Pang et al. [18] propose a novel intelligence algorithm for airfoil design based on the combination of low wind field, considering the effects of

surface roughness on the lift coefficient and the lift-to-drag ratio of the airfoil, the instability of wind speed, and wind direction to ensure gentle stalling characteristics, and the stable power generation of wind turbines.

Recently, artificial neural networks (ANNs) offer a great advantage over complex mathematical models used in the prediction of nonlinear phenomena. The skill to simulate the role of a human brain gives ANN models the capability to solve problems from a database, combining and adapting to the conditions and changes of different input variables to find a generalized solution. Recent studies have shown the versatility and usefulness of ANN models, for example, Shaquid et al. [19] employed an ANN model to investigate a reliability model centered on the exponentiated Weibull distribution and the inverse power-law model. Moreover, Shaquid et al. [20] demonstrated that ANNs are an excellent engineering tool for predicting survival and mortality rates.

Some researchers have shown that the feed-forward neural network (FFNN) can be used to interpolate pressure coefficients for low-rise buildings [21, 22] or wind-induced pressure time series [23]. FFNNs have also been employed to predict wind-induced pressure on roofs of low-rise buildings in an efficient way [24, 25] and to estimate the dynamic along-wind response of tall buildings using ANNs as an alternative to wind tunnel tests [26, 27]. More recently, Çolak [28] employed an FFNN model to study the thermal conductivity of water-based zirconium oxide nanofluid and showed its ability to make predictions with a low margin of error.

Other studies have employed recurrent neural networks (RNN) for the study of dynamic problems, where the results obtained with the RNN outperform those obtained with the FFNN [29, 30]. It is noted that the use of RNNs in solutions of wind speed or pressure problems on structures is scarce in the literature. Likewise, in recent years, deep learning in ANNs has gained more attention for works with sequences and time series [31, 32], data for classification and regression tasks for language modeling [33, 34], speech recognition and video analysis [35, 36] with the employ of long-short term memory (LSTM) ANN; however, similar to the case of RNNs, the use of LSTM ANNs in the prediction of EPC is scarce in the literature.

In this study, a comparison of the prediction of EPC on a low-rise building by using the FFNN, RNN, and LSTM ANN models with different types of architectures is carried out. For the analyses, a database with EPC from boundary layer wind tunnel tests of a low-rise building is employed for training, validation, and testing of the ANN models. With the purpose of making this study self-contained, in the following section some basic concepts of ANNs are presented.

## 2. Basic Concepts of ANNs

An ANN is a tool designed primarily to mathematically model the internal architecture and operational characteristics of the human brain and nervous system, consisting of

three types of interconnected layers. The first and last layers are called input and output layers, respectively, and all other layers between the input and output layers are called hidden layers. Each layer has a certain number of artificial neurons, each connection neuron has a synaptic weight, and each layer has an activation function responsible for processing the data that the system must recognize. Moreover, each neuron has an activation value that is a function of the sum of the inputs received by other neurons, and that is multiplied by the corresponding synaptic weights of each connection.

The number of hidden layers and hidden neurons in each layer depends on several parameters, such as the complexity of the problem to be solved, the architecture of the network, the training algorithm, and the number of training cases. Throughout the development of artificial intelligence, rules have been proposed to choose the number of hidden layers and neurons [37, 38]; however, a unified procedure is still not available for all the possible cases considered. Up to date, one of the methods usually employed to identify the optimum ANN model is to vary the number of hidden layers and neurons, estimate the mean squared error (MSE), and choose the ANN with the minimum MSE as the optimal one, this method is called “Test and Error” [39] and has been extensively used in several ANN applications [40, 41].

ANN modeling usually involves three stages: training, validation, and testing. In the training stage, neurons are trained by a random input pattern to obtain a desired result. Training consists of optimizing the synaptic connection weights and modifying them after each iteration cycle until a minimum MSE is achieved. In the validation stage, optimized weights and biases are used to produce their associated output. Normally, the inputs used in the validation stage are associated with known outputs and additional comparisons of the MSE are carried out. In the testing stage, input scenarios are used to evaluate the prediction ability of the trained ANN models.

**2.1. The FFNN Model.** In the FFNN model, connections and data flow are unidirectional, from the input layer to the output layer, without transmission of information between neurons located in the same or previous layer. The FFNN model has been used to solve dynamic problems; however, this type of ANN does not offer any retention of information, making it less effective than the RNN to solve certain types of problems [42]. An FFNN has one of the simplest architectures for training and getting good results for static problems where the behavior of the values to be predicted does not depend on time.

If an ANN model with a single output neuron and two hidden layers are considered, the mathematical expression that relates the output neuron in the output layer with the neurons in the input and hidden layers is given by the following equation [43]:

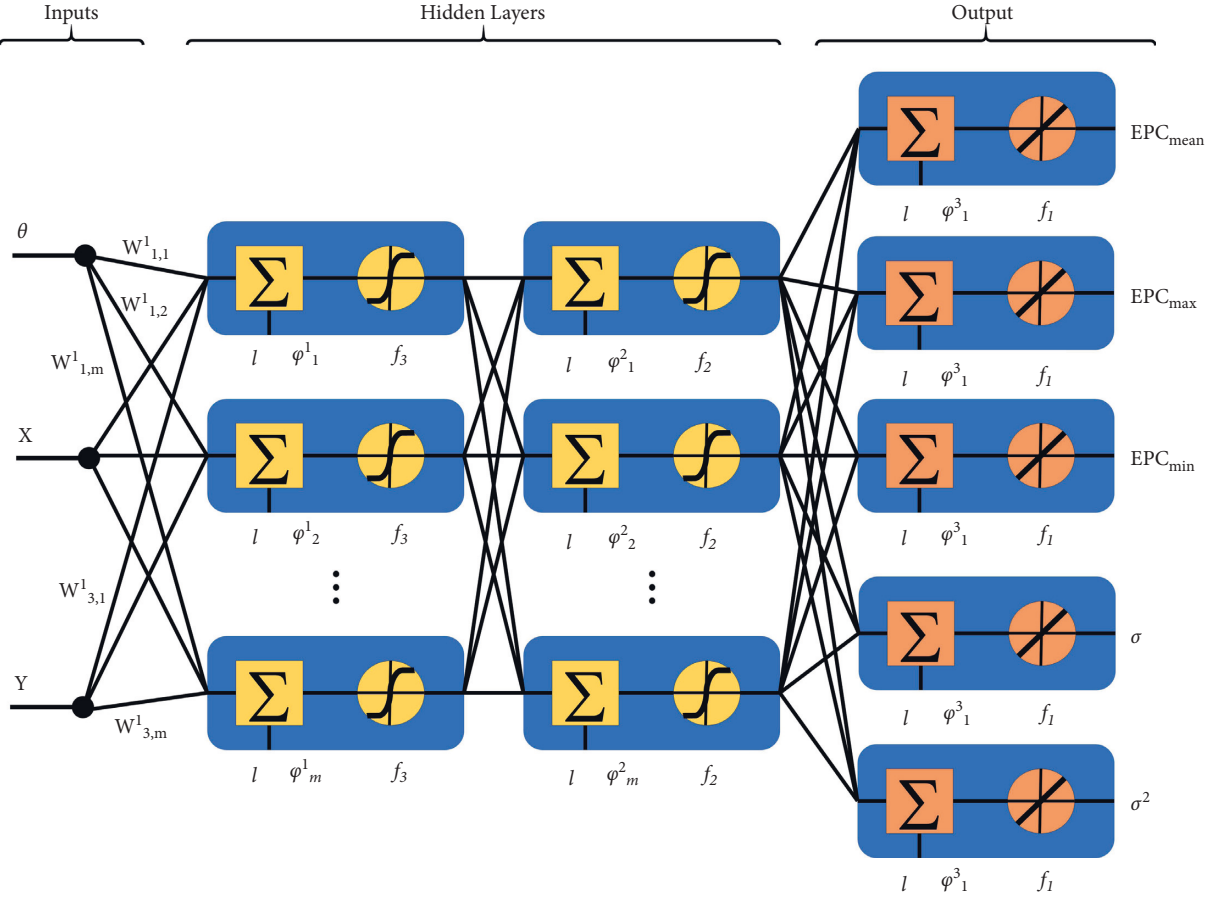


FIGURE 1: FFNN architecture with two hidden layers.

$$Y_{\text{Output}} = f_1 \left( \sum_{k=1}^m \left( [W_3]_{k,1} f_2 \left( \sum_{j=1}^m \left( [W_2]_{j,k} f_3 \left( \sum_{i=1}^n ([W_1]_{i,j} x_i) + (\varphi_1)_j \right) \right) + (\varphi_2)_k \right) \right) + (\varphi_3)_1 \right), \quad (1)$$

where  $n$  is the total number of input neurons;  $m$  is the total number of neurons in each hidden layer;  $x_i$  is the  $i$ -th neuron of the input layer;  $[W_1]_{i,j}$  is the matrix of synaptic weights that optimize the connection between the input layer and the first hidden layer;  $[W_2]_{j,k}$  is the matrix of synaptic weights that optimize the connection between the first and second hidden layers;  $[W_3]_{k,1}$  is the matrix of synaptic weights that optimize the connection between the second hidden layer and the output layer;  $(\varphi_1)_j$  is the vector associated with the results of the first hidden layer;  $(\varphi_2)_k$  is the vector associated with the results of the second hidden layer;  $(\varphi_3)_1$  is the vector associated with the results of the output layer;  $f_3(\bullet)$  and  $f_2(\bullet)$  are each a Hyperbolic Sigmoid Tangent activation function used between input and hidden layers and within the hidden layers; and  $f_1(\bullet)$  is a Linear activation function used between the second hidden layer and the output layer. Figure 1 presents the architecture of the FFNN model used in this study.

**2.2. The RNN Model.** RNNs employ feedforward and feedback connections, the latter refers to the information that can be interchanged between neurons that are in the

same or previous layers and subsequent layers, this type of connection gives them an advantage over the FFNNs when it comes to identifying and controlling a dynamic problem [44]. The RNN is mainly composed of three layers (Figure 2), the first one is the input layer with two types of neurons: the external input neurons ( $x_i$ ) responsible for feeding the RNN with external information, and the internal input neurons or context units ( $o_r$ ), which receive information from the neurons of the hidden layers (recurrent information). The function of the context units is to store knowledge generated by the network in each iteration, this knowledge will replace the external input to achieve the appropriate adjustment of the trained model; this ability provides the advantage of solving problems where the variable is changing over time. In the hidden layer, the combination of  $x_i$  and  $o_r$  is carried out by considering their corresponding synaptic weight matrix ( $w_{ij}$ ) and recurrent synaptic weight matrix ( $w_{rj}$ ). Finally, the output of the hidden layer ( $O_j$ ) is obtained by applying the activation function  $f(\bullet)$  to the combination. Equation (2) mathematically summarizes the calculation of  $O_j$  [43] as follows:

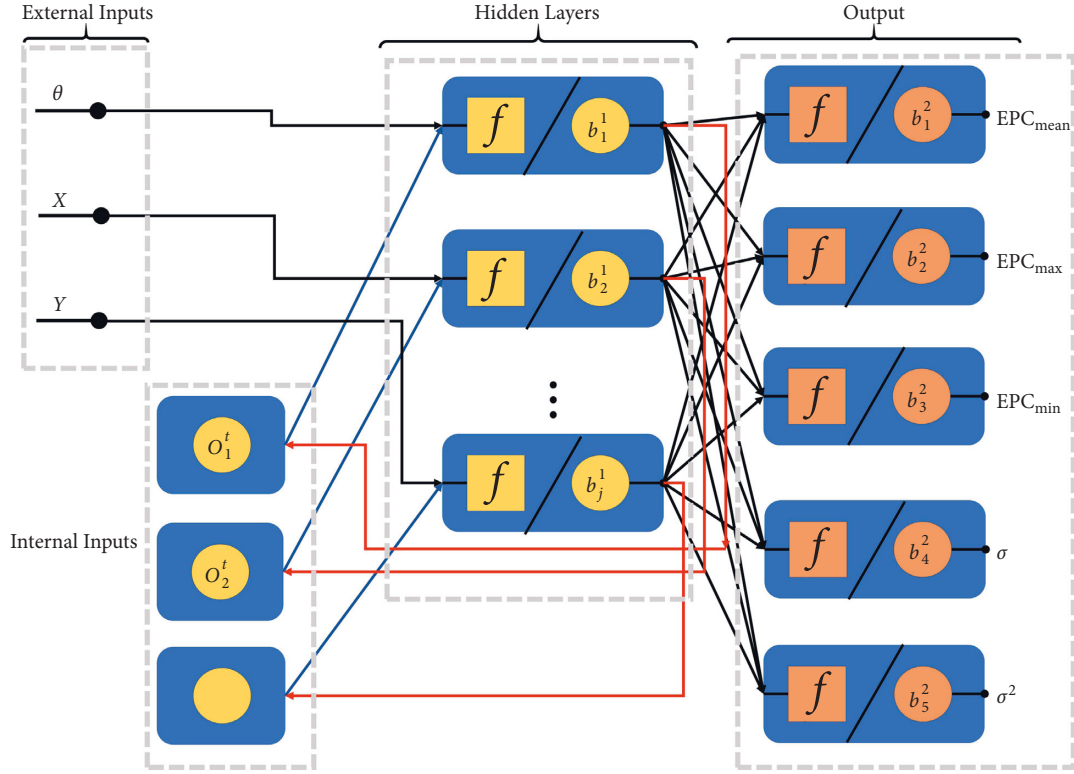


FIGURE 2: Recurrent neural network (RNN) architecture used. For simplicity, the synaptic weights between input layer (external and internal) and hidden layer have been omitted.

$$O_j = f\left(\sum_{i=1}^N x_i w_{ij} + \sum_{r=1}^R O_r w_{rj}\right), \quad (2)$$

where  $N$  is the total number of external neurons, and  $R$  is the total number of internal neurons.

The output of the hidden layer,  $O_j$ , becomes the internal input ( $O_r$ ) for the next time step  $t$ , according to the following equation [43]:

$$O_r^t = O_j^{t-1}. \quad (3)$$

Finally, the output or solution from the RNN is obtained by using the following equation [43]:

$$O_k = g\left(\sum_{j=1}^J O_r w_{jk}\right), \quad (4)$$

where  $_{jk}$  is a synaptic weight matrix,  $g$  is a linear activation function,  $J$  is the total number of hidden neurons, and  $k$  is the number of outputs.

**2.3. The LSTM ANN Model.** This kind of ANN model uses backpropagation through the time training algorithm [45] and deep learning in order to reduce the short-term dependencies that are generated due to the decrease in gradient, while the information from each step declines [46–48]. This network aims to reach a generalized solution to the problem. Such is done by overcoming the setbacks of the

declined gradients, selecting the information by filters or gates; thus, relevant information is retained, whereas irrelevant information is forgotten. Reduced vulnerability in time steps makes LSTM ANN better for data stream treatment compared to FFNN and conventional RNN models.

The general architecture of this kind of ANN is illustrated in Figure 3(a), where the main components of an LSTM ANN are shown. The input vectors that feed the LSTM ANN are also included in Figure 3(a). The  $h_t$  and  $c_t$  variables denote the hidden outputs or states, and the state of the cell in time  $t$ , respectively.

The network arrangement will have as many LSTM blocks as the number of time steps that need to be analyzed. Each block uses a predefined number of hidden units that will process the information within them.

The expression given in equation (5) is used for the calculation of the cell state ( $c_t$ ) in the time step  $t$  [49].

$$c_t = f_t c_{t-1} + i_t g_t, \quad (5)$$

where  $c_{t-1}$  is the initial cell state,  $g_t$  is the memory cell,  $i_t$  is the input gate, and  $f_t$  is the forget gate. The expressions to calculate  $g_t$ ,  $i_t$ , and  $f_t$  are given, respectively, by the following equations [49]:

$$g_t = \sigma_c(W_g x_t + R_g h_{t-1} + b_g), \quad (6)$$

$$i_t = \sigma_g(W_i x_t + R_i h_{t-1} + b_i), \quad (7)$$

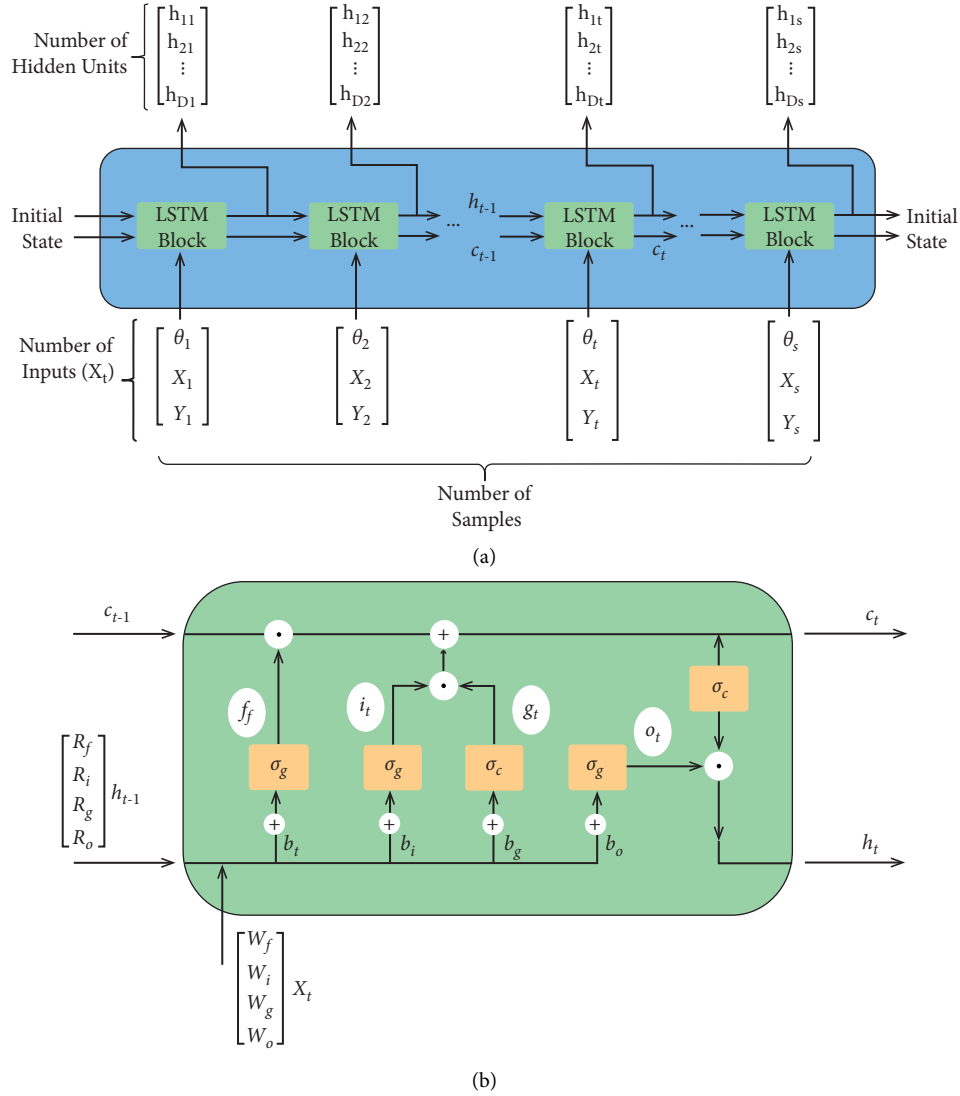


FIGURE 3: LSTM ANN: (a) architecture; (b) LSTM block.

$$f_t = \sigma_g(W_f x_t + R_f h_{t-1} + b_f), \quad (8)$$

where  $W_g, W_i$ , and  $W_f$  are the synaptic weights matrices by cell state, input gate, and forget gate, respectively;  $R_g, R_i$ , and  $R_f$  are the recurrent synaptic weights matrices by cell state, input gate, and forget gate, respectively;  $b_g, b_i$ , and  $b_f$  are the biases by cell state, input gate, and forget gate, respectively;  $x_t$  is the signal data associated with the database time step  $t$ , and  $h_{t-1}$  is the previous hidden output or state.

In the case of  $h_t$ , it will be updated according to the following equation (49):

$$h_t = o_t \sigma_c(c_t), \quad (9)$$

where  $o_t$  is the output gate given by the following equation [49]:

$$o_t = \sigma_g(W_o x_t + R_o h_{t-1} + b_o), \quad (10)$$

where  $W_o, R_o$ , and  $b_o$  are the synaptic weights matrix, the recurrent synaptic weights matrix, and biases of the output gate, respectively;  $h_{t-1}$  is the previous network status;  $x_t$  is the signal data associated with time step  $t$ . In equations (7), (8), and (10),  $\sigma_g$  denotes the gate activation function, and in equations (6) and (9),  $\sigma_c$  denotes the state activation function.

The status of each block depends on  $h_t$ , which contains the output of the LSTM block for the appropriate time step  $t$ , and the state of  $c_t$ , which contains the knowledge from previous time steps. Internally, at each block and at each time step, cell state information is added or removed by controlling gates (Figure 3(b)).

More specifically, the forget gate is used to control the volume of information that will be discarded from past time steps selecting the value of the vector  $f_t$ , if the result of the function is close to 1, the information stored by the state of the cell is retained; however, when the function value is close to 0, the cell state will discard the information.

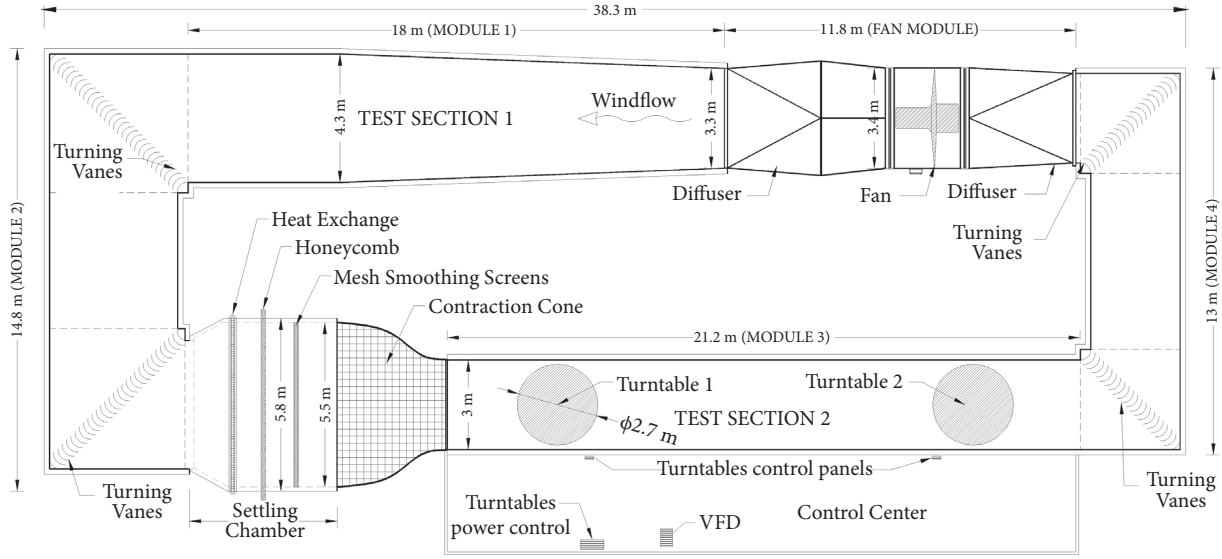


FIGURE 4: Sketch of the boundary layer wind tunnel of institute of engineering, UNAM.

In this way, the current memory  $g_t$  and the long-term memory  $c_{t-1}$  are combined in the LSTM, resulting in a new cellular state  $c_t$ . The information contained at a predefined time can be preserved by the control of the forget gate  $f_t$ , while the current inessential information is removed from memory by the control of the input gate  $i_t$ . The output gate  $o_t$  is configured to control the amount of memory information to update at the next time step, similar to the input gate calculation.

### 3. Analysis Procedure

**3.1. Experimental Input and Output Parameters.** To assemble the experimental database of wind-induced EPC of a generic low-rise building, wind tunnel tests at the boundary layer wind tunnel (BLWT) of the National Autonomous University of Mexico (UNAM for its acronym in Spanish) were carried out. The UNAM BLWT is of the closed-circuit type and is composed of four modules that form a rectangular configuration in plan (Figure 4). Modules 1 and 3 are approximately 38 m long, whereas modules 2 and 4 are 13.9 and 12.6 m long, respectively. Module 3 is the module for replicating the atmospheric boundary layer (ABL), it is 3 m wide, and the height ranges from 2 m at the contraction cone exit to 2.35 m behind the second turntable. The length of the test section permits the placement of roughness elements and turbulence generators to adequately characterize the ABL and turbulence intensity profile for a predefined terrain category [50]. All the experimental tests were carried out in Module 3. A complete description of the BLWT employed for the experimental tests can be found in Amaya-Gallardo et al. [51].

For the wind simulation, a wind speed scale of 1 : 2.3 was selected. The reference wind speed for testing was obtained from the Mexican wind standard for an urban terrain category. For the simulation of the mean wind profile and

turbulence characteristics, a passive turbulence generator was installed in Test Section 2, which consisted of roughness elements, a castellated barrier, and spines (Figure 5). A total of 27 measures of wind speed at different heights were obtained by using a digital hot-wire anemometer. These measurements were used to determine the mean wind speed profile and turbulence indicators (i.e., turbulence profile and power spectral density function (PSDF)). Figure 6(a) shows a comparison of the experimental mean wind velocity ( $U_x$ ) profile with the classical theoretical power law (POW) adopted by several international wind design codes. Also, in Figure 6(b), the turbulence intensity profile is shown. The mathematical expressions that defined the power-law mean wind velocity profile and longitudinal turbulence intensity are defined, respectively, as[52]:

$$U_x = \bar{U}_{\text{ref}} \left( \frac{z}{z_{\text{ref}}} \right)^{\alpha} I_{U_x} = \frac{\sigma_U}{U_x}, \quad (11)$$

where  $\alpha$  is the power law exponent,  $\bar{U}_{\text{ref}}$  is a reference mean wind velocity,  $z_{\text{ref}}$  a reference height, in this work considered as 0.2 m (which is equivalent to 10 m above ground in full scale), and  $\sigma_U$  is the standard deviation of the longitudinal turbulence wind component.

To determine the experimental power-law exponent  $\alpha$ , a fitting exercise based on the least square method was carried out, resulting in a value of  $\alpha = 0.295$  (urban terrain category). Furthermore, it is observed in Figure 6(b) that the longitudinal turbulence intensity values range from about 25% near the wind tunnel floor level, up to 8% in the upper part of the longitudinal turbulence profile. To further evaluate the turbulence simulated in the wind tunnel, Figure 7 presents the PSDF of the longitudinal turbulence component and its comparison with the Von Karman Spectrum, which is considered a suitable representation of the velocity spectrum [52] and is given by the following equation:



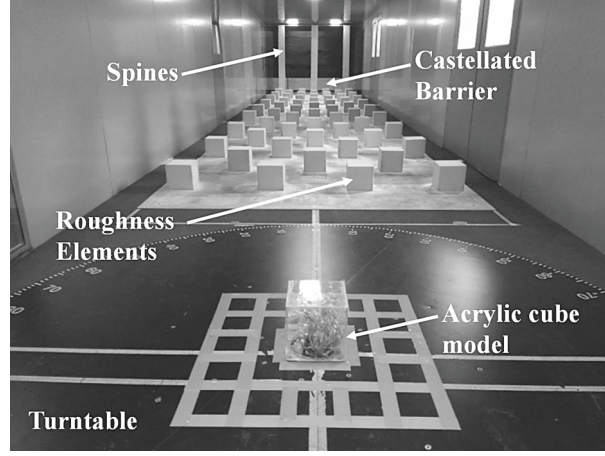


FIGURE 5: Wind tunnel set-up and passive turbulence generator.

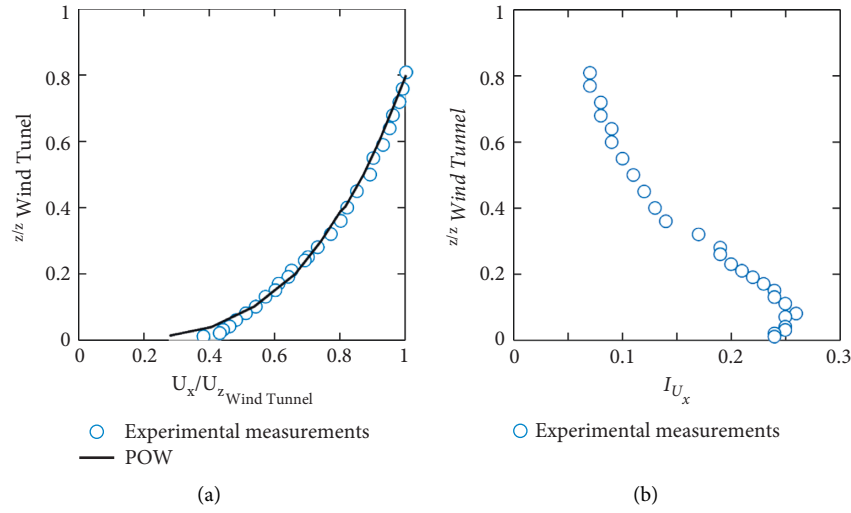


FIGURE 6: Longitudinal wind profiles: (a) mean wind speed; (b) turbulence intensity.

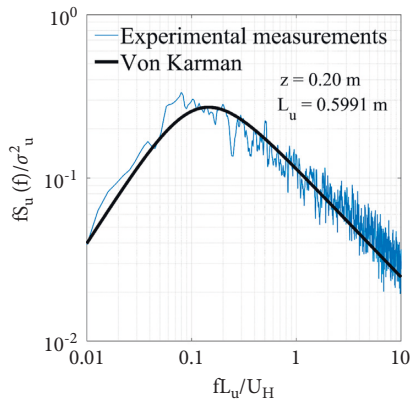


FIGURE 7: Comparison of experimental PSDF and von karman PSDF.

$$\frac{fS_u(f)}{\sigma_u^2} = \frac{4fL_u/U_x}{[1 + 70.8(fL_u/U_x)^2]^{5/6}}, \quad (12)$$

where  $f$  is frequency,  $L_U$  is the turbulence integral length scale in the longitudinal direction,  $S_U$  is the power spectral density of the fluctuating wind component. Figure 7 presents a comparison of the normalized PSDF obtained at  $z = 0.2 \text{ m}$  and the Von Karman Spectrum. It is observed in Figure 7 a good comparison of the experimental and theoretical PSDFs.

The parameters used as input neurons for the FFNN and RNN models were  $\theta$ , as well as the  $x$ - and  $y$ -coordinates of each tap. The output neurons were EPCmean, PCmax, EPCmin,  $\sigma$ , and  $\sigma^2$ . For the LSTM ANN model, input vectors containing the coordinates  $(x, y)$  of the taps and predefined  $\theta$  values were used, and the output neurons were the same as those for the FFNN and RNN models.

**3.2. ANN Data Sets and Training.** For the ANN training, a subset of the experimental database was used, where 75% of the data were randomly selected and employed to train, while the remaining 25% of the data were used as a first validation set to monitor the training process to avoid overtraining. A second validation set was used to identify the optimum ANN models. With the optimum ANN models,

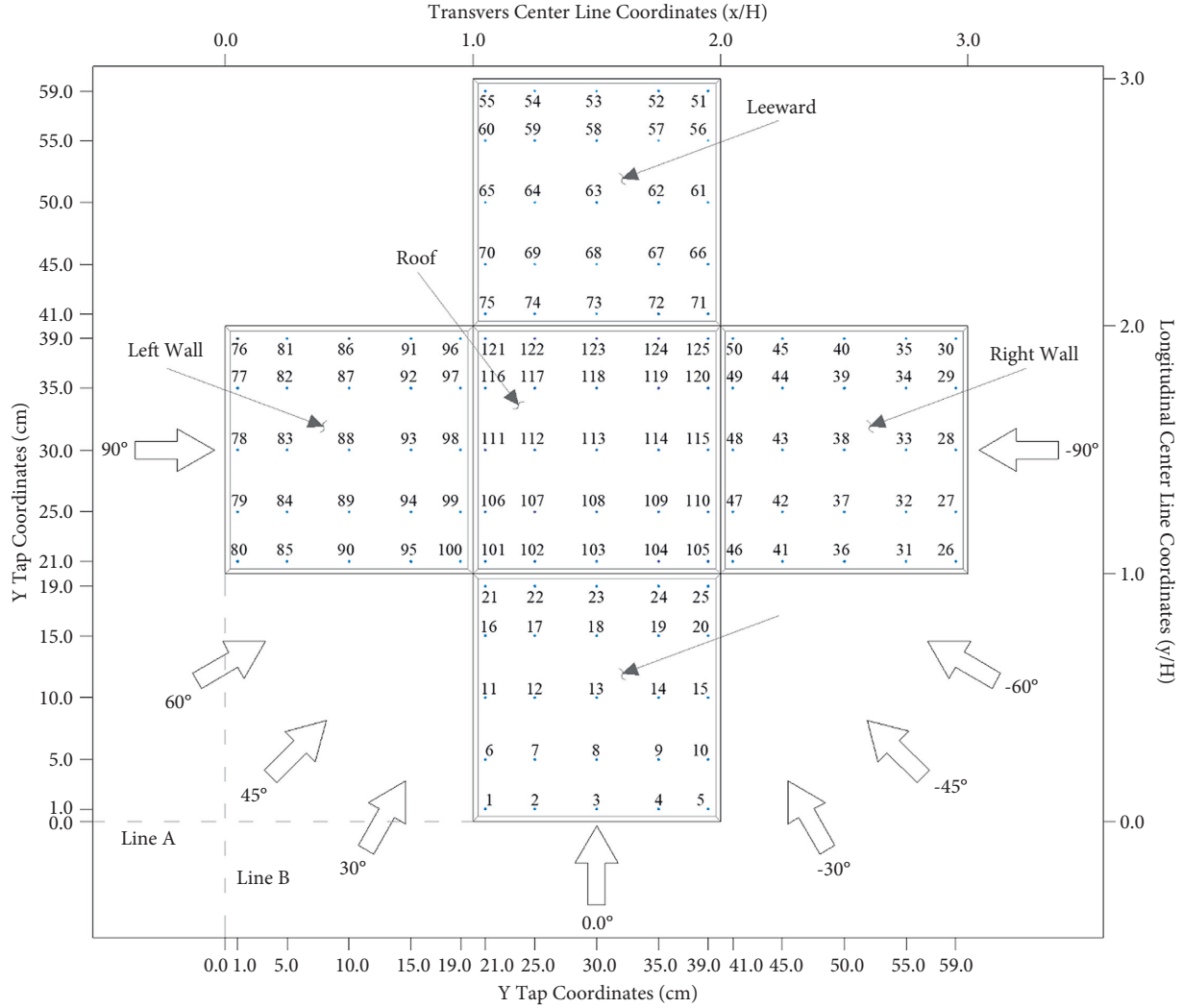


FIGURE 8: Coordinates of the taps located in the low-rise building tested in wind tunnel and wind directions.

TABLE 1: Training, validation, and test data.

Subsets	Wind direction (°)					
Training set <sup>1</sup>	-90.0	-45.0	0.0	45.0	90.0	—
Validation set <sup>2</sup>	-60.0	-30.0	30.0	60.0	—	—
Testing set	-80.0	-70.0	-50.0	-40.0	-20.0	-10.0
	80.0	70.0	50.0	40.0	20.0	10.0

Note. 1) For this set, 75% was used for training and 25% for the first validation; 2) Second validation set.

case scenarios were evaluated with a testing set. The training, first, and second validation, as well as test subsets, are summarized in Table 1.

For the development of FFNNs, models with one and two hidden layers with 3 to 50 hidden neurons per hidden layer, and the sigmoidal hyperbolic tangent and linear activation functions were considered. The variation of hidden layers and neurons is with the aim of avoiding the lack of learning and the inability of predicting outcomes for the scenarios that are not used in training (i.e., overfitting).

One of the algorithms used to train the FFNN is the back-propagation [54], where the error is propagated backward by adjusting the weights from the output to the

input layer. For training the FFNN models, the following steps were followed:

- (1) Provide the ANN model with sample inputs and known outputs;
- (2) Evaluate an error function in terms of the difference between the predicted and observed output;
- (3) Minimize the error function (MSE) by adjusting the weights and biases of all the layers from the output to the input layer.

In order to evaluate the impact of using different types of minimization algorithms on the prediction effectiveness of the



TABLE 2: Best FFNN models identified in the validation stage with 1 and 2 hidden layers.

Training algorithm	MSE		$\rho$	
	1HL	2HL	1HL	2HL
L-M	0.345 (30)	0.210 (20)	0.947	0.968
BR	0.246 (50)	0.227 (10)	0.967	0.960
BFGS	0.434 (50)	0.280 (50)	0.929	0.961
VLRGD	0.676 (15)	0.799 (25)	0.853	0.841
GDM	0.966 (5)	0.768 (40)	0.814	0.855
GD	0.867 (15)	0.799 (30)	0.817	0.853

1HL = 1 hidden layer; 2HL = 2 hidden layers. The values inside brackets indicate the number of hidden neurons.

TABLE 3: Best RNN models identified in the validation stage with 1, 2, and 3 delays.

Training algorithm	MSE			$\rho$		
	1D	2D	3D	1D	2D	3D
BFGS	0.607 (11)	0.662 (11)	0.674 (7)	0.341	0.320	0.333
BR	0.671 (11)	0.790 (11)	0.858 (7)	0.335	0.306	0.325
GD	0.699 (11)	0.894 (7)	0.941 (11)	0.309	0.302	0.296
VLRGD	0.765 (8)	0.791 (6)	0.850 (4)	0.321	0.308	0.310
GDM	0.774 (7)	0.815 (11)	0.804 (10)	0.701	0.692	0.319
L-M	0.777 (6)	0.727 (7)	0.847 (2)	0.334	0.325	0.302

1D = 1 delay; 2D = 2 delays; 3D = 3 delays. The values inside brackets indicate the number of hidden neurons.

FFNN models, the minimization of the MSE was carried out using the following algorithms [54]: Gradient Descent (GD), Gradient Descent with Momentum (GDM), Variable Learning Rate Gradient Descent (VLRGD), Levenberg-Marquardt (L-M), BFGS Quasi-Newton (BFGS), and Bayesian Regularization (BR).

In the case of the RNN, models with 1 to 3 delays, and with two groups of hidden neurons were considered. The first group includes 3, 5, 10, 15, 20, 25, 30, 40, and 50 hidden neurons, while the second group includes 2, 4, 6, 7, 8, 9, and 11 hidden neurons. The second group was included to evaluate with more precision the use of a small number of hidden neurons compared to the first group.

For the training of the RNN, the same steps followed for the training of the FFNN were adopted, except that the internal input neurons are also included in the minimization procedure by adjusting the recurrent synaptic weight matrix and biases of all the layers from the output to the input layer.

To develop the LSTM ANN models, a regression LSTM network with sequence output where the targets are the training sequences with values shifted over a time step was used. To create an LSTM regression network architecture, the next steps were followed:

- (1) Determine a sequence input layer that has an input size that matches the number of channels of the input data. In this case 3 inputs data.
- (2) Define the number of hidden units in each LSTM block. For the LSTM ANN models, each LSTM block considered 100, 200, 300, 400, and 500 hidden units.
- (3) Include a fully connected layer with an output size that matches the number of channels of the output targets.
- (4) Include a regression layer.

TABLE 4: Best LSTM NN models identified in the validation stage with ADAM training algorithm and a batch equal to 30.

HU	MSE	$\rho$
400	0.022	0.982
300	0.023	0.981
100	0.032	0.977
500	0.069	0.982
200	0.073	0.982

HU = Hidden unit.

For the training of the LSTM ANN, the algorithm used was the Adaptive Moment Estimation (ADAM) optimizer, which is an adaptive learning rate method. An ADAM optimizer is a stochastic gradient descent for use in Deep Learning on non-convex optimization problems.

## 4. Analysis Results and Discussion

**4.1. Identification of the Best ANN Models.** To evaluate the impact of record selection on the trained ANN models, a total of 100 trials were carried out. For each trial, a new set of randomly selected values (75% of the data used for training) were used. The second validation set was used to identify the best ANN models. Tables 2 and 3 summarize the results obtained from the best ANN models identified (i.e., the ANN models with the smallest MSE) for the FFNN and RNN, respectively, while Table 4 presents the best models identified for the LSTM ANN. Also, in Tables 2-4, the correlation coefficient ( $\rho$ ) between the predicted and the actual values are included.

It is observed in Table 2, that for the FFNN models, the optimum number of neurons and hidden layers that provide the lowest MSE for the trained model depends on the selected data. It is also observed that, in general, the optimum

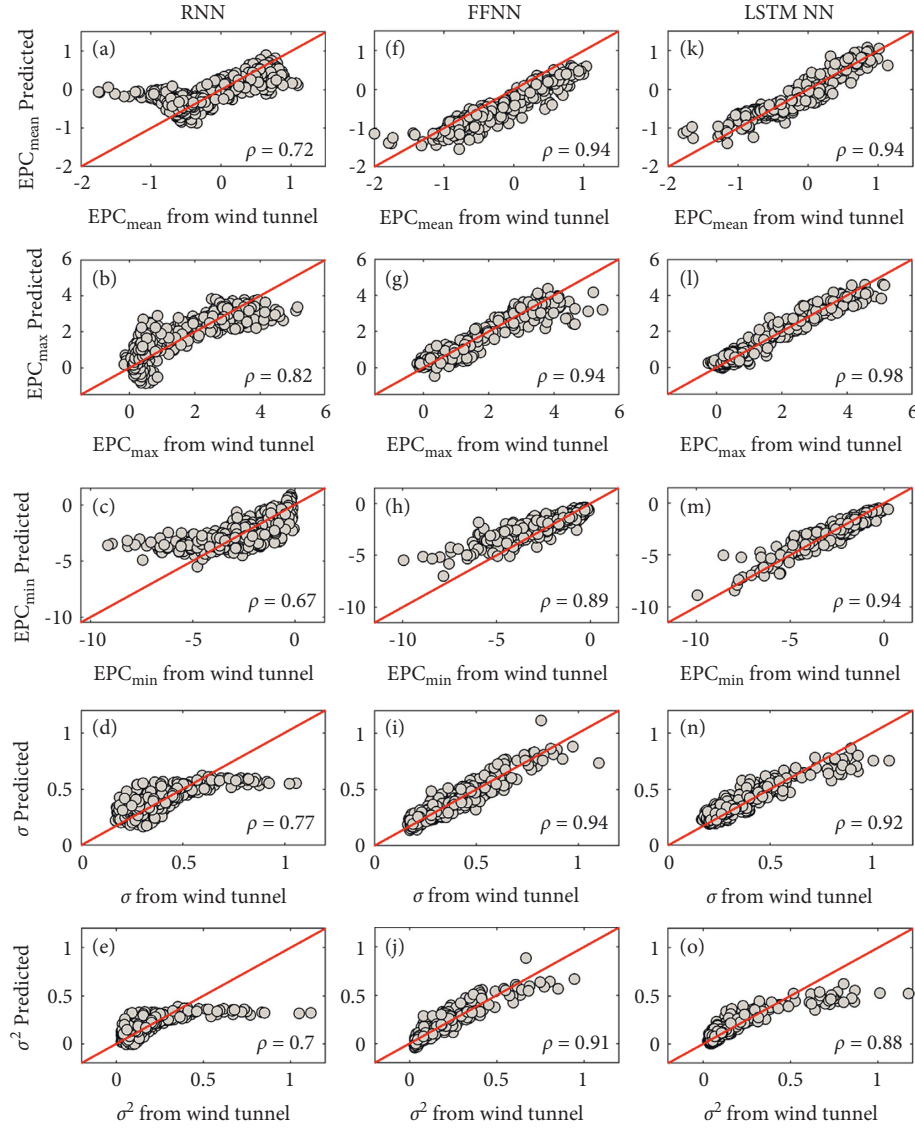


FIGURE 9: Comparison of the  $EPC_{mean}$ ,  $EPC_{max}$ ,  $EPC_{min}$ ,  $\sigma$ , and  $\sigma^2$  values predicted by the trained ANN models and the second validation set and those obtained from wind tunnel tests: (a), (b), (c), (d), (e) RNN; (f), (g), (h), (i), (j) FFNN; (k), (l), (m), (n), (o) LSTM NN. The red line indicates the ideal fit.

number of neurons is greater than 10, and that the 2-hidden-layers models outperform those of 1 hidden layer. The minimum MSE and maximum  $\rho$  are associated with the L-M training algorithm, although the BR and BFGS produce comparable results to those obtained with the L-M algorithm. Based on these observations, the use of 20 neurons and the ANN model with 2 hidden layers is selected for the prediction of  $EPC_{mean}$ ,  $EPC_{max}$ ,  $EPC_{min}$ ,  $\sigma$ , and  $\sigma^2$  values, with associated values of MSE and  $\rho$  equal to 0.210 and 0.968, respectively.

The results summarized in Table 3 for RNN models indicate that the optimum number of neurons and the optimum number of delays also depend on the selected data. The optimum number of neurons is between 2 and 11. The RNN models with 1 delay outperform those with 2 and 3 delays. The minimum MSE and maximum  $\rho$  are associated with the BFGS training algorithm. From the RNN

models presented in Table 3, the use of 11 neurons and 1 delay is selected for the prediction of  $EPC_{mean}$ ,  $EPC_{max}$ ,  $EPC_{min}$ ,  $\sigma$ , and  $\sigma^2$  values. The associated values of MSE and  $\rho$  for this RNN model are equal to 0.607 and 0.341, respectively.

The results presented in Table 4 for the LSTM ANN show that, in general, as the hidden units increase, the MSE reduces and  $\rho$  increases. Based on this observation, the best model selected for the prediction of  $EPC_{mean}$ ,  $EPC_{max}$ ,  $EPC_{min}$ ,  $\sigma$ , and  $\sigma^2$  values are with 400 hidden units and with 30 batches, with associated values of MSE and  $\rho$  equal to 0.0219 and 0.9824, respectively.

**4.2. Comparison of Trained ANN Models.** The comparison between the predicted  $EPC_{mean}$ ,  $EPC_{max}$ ,  $EPC_{min}$ ,  $\sigma$ , and  $\sigma^2$  by using the best FFNN, RNN, and LSTM trained models,

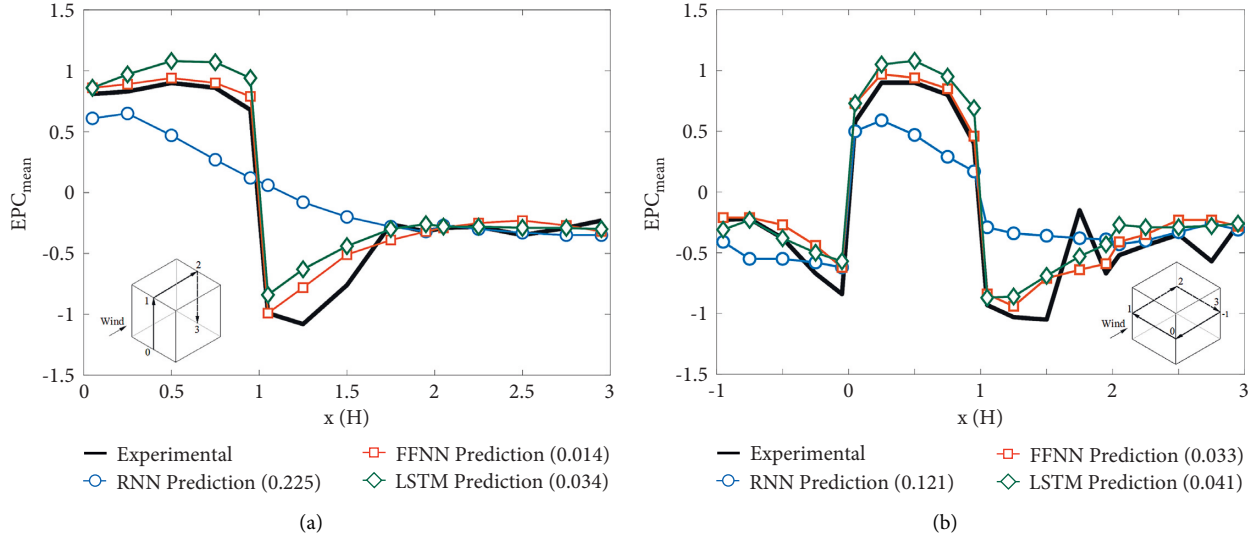


FIGURE 10: Comparison of the experimental  $EPC_{mean}$  and those predicted by the trained ANN models for  $\theta = -10^\circ$ : (a) along the longitudinal center line; (b) along the transverse center line. Numbers inside brackets indicate the MSE values.

obtained with the second validation set, and those obtained from the actual experimental database is shown in Figure 9. It is observed from Figure 9 that, in general, there is good agreement between the predicted and the observed values. The computed correlation coefficients range between 0.67 and 0.98. The best correlation is observed for the values predicted with the FFNN and LSTM models, while the worst correlation is obtained when the RNN model is employed. It is also observed in Figure 9 that there is no clear preference among the trained LSTM and FFNN models for providing the best estimates. Figure 9 also shows that the parameters best predicted by the LSTM ANN are  $EPC_{mean}$ ,  $EPC_{max}$ , and  $EPC_{min}$ , with correlation coefficients greater than 92%. On the other hand, the parameters best predicted by the FFNN models are  $EPC_{mean}$ ,  $EPC_{max}$ , and  $\sigma$ , with correlation coefficients greater than 91%. The previous observation indicates that, for preliminary design purposes, the use of the LSTM ANN model would be preferred, since the parameters  $EPC_{max}$  and  $EPC_{min}$  are generally used in the design of cladding, while  $EPC_{mean}$  is used for the design of the main structure. Moreover, the prediction made with the RNN models is not robust because they produce undesirable behavior for predicting  $EPC_{mean}$ ,  $EPC_{min}$ ,  $\sigma$ , and  $\sigma^2$ .

To further evaluate the ability of the ANN models developed, in the following, the discussion is focused on the  $EPC_{mean}$  since it is widely used for wind design in codes and standards. For the comparison between the results predicted by using the testing set and the experimental ones, a case scenario of wind direction equal to  $-10^\circ$  is considered. This wind direction was not included in the training set nor in the validation sets. By using the testing set, Figure 10 shows a comparison of predicted and experimental  $EPC_{mean}$  values along the longitudinal and transverse central lines for  $\theta = -10^\circ$ . It is observed in Figure 10 that the  $EPC_{mean}$  values predicted with the LSTM and the FFNN models follow closely those from the experimental test, and that the RNN model is the one with less predictive ability, with abrupt

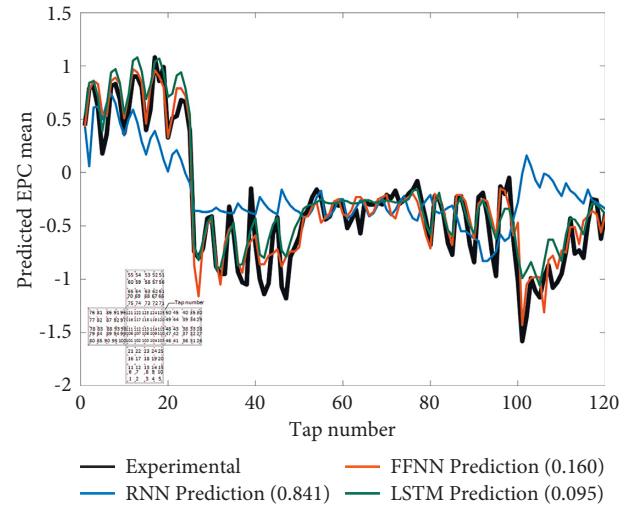


FIGURE 11: Comparison of the experimental  $EPC_{mean}$  and those predicted by the trained ANN models for all the taps for  $\theta = -10^\circ$ . Numbers inside brackets indicate the MSE values.

changes in the prediction of  $EPC_{mean}$  along the longitudinal and transverse central lines. The MSE obtained based on the experimental and predicted values for the three compared ANN models are included in Figure 10, where the minimum MSE is associated with the FFNN. It is further observed in Figure 10(a) that the predictions of  $EPC_{mean}$  made with both the FFNN and the LSTM ANN models for the windward wall (from point 0 to point 1) are the best, contrary to the predictions made with the RNN, which present a sudden drop compared to the experimental results. For the roof (from point 1 to point 2), the three ANN models considered do not reach the minimum  $EPC_{mean}$  values from experimental tests; however, the FFNN and LSTM ANN models are able to mimic the sudden change from pressure to suctions. The latter is not observed for the predictions made with the RNN model. For the leeward zone (from point 2 to point 3), the

TABLE 5: Variation of MSE on the faces of the low-rise building for different wind direction by using the trained ANN models.

$\theta$ (°)	Right wall			Windward wall			Left wall			Leeward wall			Roof		
	RNN	FFNN	LSTM	RNN	FFNN	LSTM	RNN	FFNN	LSTM	RNN	FFNN	LSTM	RNN	FFNN	LSTM
−80	0.212	0.029	0.018	0.141	0.025	0.022	0.014	0.015	0.031	0.023	0.007	0.039	0.199	0.026	0.149
−70	0.306	0.017	0.015	0.152	0.024	0.017	0.015	0.015	0.033	0.029	0.013	0.044	0.194	0.033	0.135
−50	0.340	0.023	0.061	0.205	0.033	0.032	0.059	0.023	0.044	0.033	0.010	0.022	0.305	0.081	0.073
−40	0.240	0.017	0.021	0.157	0.016	0.033	0.108	0.028	0.056	0.022	0.010	0.024	0.311	0.065	0.067
−20	0.141	0.027	0.018	0.141	0.018	0.018	0.193	0.034	0.062	0.010	0.015	0.015	0.456	0.039	0.083
−10	0.128	0.033	0.026	0.178	0.020	0.039	0.276	0.047	0.048	0.011	0.014	0.010	0.546	0.035	0.062
10	0.065	0.010	0.013	0.130	0.026	0.021	0.193	0.053	0.055	0.017	0.019	0.010	0.515	0.035	0.035
20	0.056	0.017	0.019	0.114	0.019	0.012	0.096	0.037	0.038	0.015	0.018	0.009	0.489	0.039	0.064
40	0.047	0.016	0.007	0.067	0.019	0.026	0.187	0.055	0.040	0.024	0.019	0.010	0.483	0.085	0.075
50	0.040	0.012	0.008	0.024	0.030	0.025	0.263	0.050	0.052	0.031	0.018	0.012	0.389	0.069	0.061
70	0.032	0.011	0.012	0.144	0.025	0.040	0.296	0.038	0.046	0.054	0.025	0.015	0.357	0.039	0.050
80	0.027	0.013	0.011	0.244	0.022	0.039	0.331	0.036	0.051	0.099	0.026	0.025	0.334	0.039	0.052
Mean	0.139	0.018	0.018	0.137	0.022	0.028	0.164	0.035	0.047	0.031	0.016	0.020	0.376	0.050	0.075

predictions of the three models (i.e., FFNN, LSTM, and RNN) are in good agreement with the experimental results. Similar observations to those drawn from Figure 10(a) are also applicable to Figure 10(b), except that the RNN model is not able to mimic the behavior observed from the experimental results for the lateral walls (from point -1 to 0 and from 1 to 2) and the windward wall (from point 0 to 1).

A generic low-rise building represented by a cube with a side dimension equal to 10 m was considered for the experimental tests. This type of low-rise building has been widely studied worldwide and is referred to as the Silsoe cube [53]. A length scale equal to 1:50 was selected for the modeling. The model was built with rigid acrylic plates with 4 mm thickness. The pressure tap layout consisted of a total of 125 pressure taps, with 25 pressure taps per face (Figure 8). Plastic tubes made of urethane with an internal diameter of  $8.6 \times 10^{-4}$  m and an exterior diameter of  $1.68 \times 10^{-3}$  m were used to connect the pressure taps to the pressure scanners. During the experimental tests, the temperature was controlled with a heat exchanger, the average temperature during the tests was 15°C. The pressure scanners were set with a sampling rate equal to 256 Hz and a sampling time of 21 s. Wind direction ( $\theta$ ) was varied within  $-90^\circ$  to  $90^\circ$  with increments of  $10^\circ$ ; additional angles of  $45^\circ$  and  $-45^\circ$  were also considered (Figure 8). From the experimental tests, a total of 1125 time-history records of EPC were registered and processed to calculate mean EPC (EPCmean), maximum EPC (EPCmax), minimum EPC (EPCmin), root mean square of EPC ( $\sigma$ ), and variance of EPC ( $\sigma^2$ ), which are typical measures to evaluate the wind-induced forces in structures [1]. With the information processed and organized, an experimental input and output database was assembled, which included EPCmean, EPCmax, EPCmin,  $\sigma$ , and  $\sigma^2$  values, which were associated with predefined  $\theta$  values and coordinates ( $x$ ,  $y$ ) for each tap (Figure 8). For simplicity, the instrumented faces of the cube were unfolded as shown in Figure 8, and the reference system to define the coordinates for each tap was located at the intersection of the lines A and B. The total number of elements in the experimental database was equal to 5625.

To further investigate whether the LSTM and FFNN models outperform the RNN model, Figure 11 presents a comparison of the experimental and predicted EPC<sub>mean</sub> values for all the taps and for wind direction  $-10^\circ$ . It is observed in Figure 11 that, in general, the LSTM and FFNN models are able to mimic the behavior of the EPC<sub>mean</sub> values, with very good predictions for the upper and lower bounds of EPC<sub>mean</sub> (i.e., the greatest positive values of EPC<sub>mean</sub> and the lowest negative values). It is also observed from Figure 11 that the RNN model provides predictions of the EPC<sub>mean</sub> values from taps 1 to 25 that follow a similar trend to that of the experimental values, but with lower magnitude; from taps 26 to 50 and from taps 101 to 125, the RNN predictions are considerably different to those from the experimental tests; and from taps 50 to 100, the RNN model provides EPC<sub>mean</sub> values that follow an average trend to that of the experimental values.

The comparison shown in Figure 11 was repeated, except that a wide range of wind directions (not included in the training set nor in the validation sets) and the prediction at each face of the low-rise building were considered, the results of this comparison in terms of the MSE are summarized in Table 5. Similar observations to those drawn for Figure 11 are applicable to the results presented in Table 5, indicating that the LSTM and FFNN models are the ones with better predictive ability. Among all the cases considered in Table 5, the predictions with the three ANN models at the leeward wall are associated with the minimum MSE values and those at the roof with the maximum. It is also observed from the results of Table 5 that, on average, the FFNN is the model associated with the minimum MSE values.

## 5. Conclusions

Numerical analyses were carried out to investigate whether the FFNN, RNN, and LSTM models could adequately predict the mean, maximum, minimum, root mean square, and variance of external pressure coefficients of a low-rise building. For the analysis, 1125 time-history records of external pressure coefficients calculated from wind tunnel tests were considered.

The main observations that can be drawn from the analysis results are:

- (1) The best-trained FFNN and RNN models require more than 10 hidden neurons, while the LSTM ANN requires 400 hidden units and 30 batches. The latter indicates that the developed ANN-based models need a considerable number of hidden neurons and units to provide predictions with low error values for the cases considered.
- (2) The input neurons employed in all the ANN models showed to be adequate and concordant with the physical phenomenon where the external pressure coefficients depend on the wind direction and location
- (3) For preliminary design purposes, the use of the LSTM ANN model would be preferred, since it provides the best predictions for the maximum, minimum, and mean external pressure coefficients, which are used in the design of cladding and the main structure.
- (4) The RNN models are not robust because they produce undesirable behavior for predicting  $EPC_{\text{mean}}$ ,  $EPC_{\text{min}}$ ,  $\sigma$ , and  $\sigma^2$ .

## Data Availability

The manuscript has no associated data.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

This work was supported by the Institute of Engineering of the National Autonomous University of Mexico (UNAM) and the National Council on Science and Technology of Mexico (CONACYT). We thank Osvaldo Martín del Campo Preciado for providing insightful comments and suggestions to improve this work.

## References

- [1] V. Guzmán-Solis, A. Pozos-Estrada, and R. Gómez-Martínez, "Experimental study of wind-induced shear, bending, and torsional loads on rectangular tall buildings," *Advances in Structural Engineering*, vol. 23, no. 14, 2020.
- [2] P. J. Richards and R. P. Hoxey, "Appropriate boundary conditions for computational wind engineering models using the k- $\epsilon$  turbulence model," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 46-47, pp. 145-153, 1993.
- [3] J. X. Lin, D. Surry, and H. W. Tieleman, "The distribution of pressure near roof corners of flat roof low buildings," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 56, no. 2-3, pp. 235-265, 1995.
- [4] J. X. Lin and D. Surry, "The variation of peak loads with tributary area near corners on flat low building roofs," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 77-78, pp. 185-196, 1998.
- [5] H. Kozmar, "Surface pressure on a cubic building exerted by conical Vortices," *Journal of Fluids and Structures*, vol. 92, Article ID 102801.
- [6] A. Rigato, P. Chang, and E. Simiu, "Database-assisted design, standardization, and wind direction effects," *Journal of Structural Engineering*, vol. 127, no. 8, pp. 855-860, 2001.
- [7] Y. Chen, G. A. Kopp, and D. Surry, "Interpolation of pressure time series in an aerodynamic database for low buildings," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 91, no. 6, pp. 737-765, 2003.
- [8] G. A. Kopp and Y. Chen, "Database-assisted design of low-rise buildings: aerodynamic considerations for a practical interpolation scheme," *Journal of Structural Engineering*, vol. 132, no. 6, pp. 909-917, 2006.
- [9] T. C. E. Ho, D. Surry, D. Morrish, and G. A. Kopp, "The UWO contribution to the NIST aerodynamic database for wind loads on low buildings: Part 1. Archiving format and basic aerodynamic data," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 93, no. 1, pp. 1-30, 2005.
- [10] K. R. Gurley, M. A. Tognarelli, and A. Kareem, "Analysis and simulation tools for wind engineering," *Probabilistic Engineering Mechanics*, vol. 12, no. 1, pp. 9-31, 1997.
- [11] R. Popescu, G. Deodatis, and J. H. Prevost, "Simulation of homogeneous non-Gaussian stochastic vector fields," *Probabilistic Engineering Mechanics*, vol. 13, no. 1, pp. 1-13, 1998.
- [12] M. Gioffre, V. Gusella, and M. Grigoriu, "Simulation of non-Gaussian field applied to wind pressure fluctuations," *Probabilistic Engineering Mechanics*, vol. 15, no. 4, pp. 339-345, 2000.
- [13] M. Gioffre, V. Gusella, and M. Grigoriu, "Non-Gaussian wind pressure on prismatic buildings II: numerical simulation," *Journal of Structural Engineering*, vol. 127, no. 9, pp. 990-995, 2001.
- [14] Y. Chen, *Time Series Simulation of Wind-Induced Pressures on Low Buildings*, Faculty of Graduate Studies the University of Western Ontario, London, UK, 2002.
- [15] J. Wan, G. Ren, J. Liu, Q. Hu, and D. Yu, "Ultra-short-term wind speed prediction based on multi-scale predictability analysis," *Cluster Computing*, vol. 19, no. 2, pp. 741-755, 2016.
- [16] J. Wan, J. Liu, G. Ren et al., "A universal power-law model for wind speed uncertainty," *Cluster Computing*, vol. 22, no. S4, Article ID 10347, 2019.
- [17] Y. Li, P. Yang, and H. Wang, "Short-term wind speed forecasting based on improved ant colony algorithm for LSSVM," *Cluster Computing*, vol. 22, no. S5, Article ID 11575, 2019.
- [18] X. Pang, H. Wang, and J. Chen, "Intelligence algorithm for optimization design of the low wind speed airfoil for wind turbine," *Cluster Computing*, vol. 22, no. S4, pp. 8119-8129, 2019.
- [19] A. Shafiq, A. B. Çolak, and T. N. Sindhu, "Reliability Investigation of Exponentiated Weibull Distribution Using IPL through Numerical and Artificial Neural Network Modeling," *Quality and Reliability Engineering International*, pp. 1-16, Wiley, Hoboken, NJ, USA, 2022.
- [20] A. Shafiq, A. Batur Çolak, T. Naz Sindhu, S. Ahmad Lone, A. Alsubie, and F. Jarad, "Comparative study of artificial neural network versus parametric method in COVID-19 data analysis," *Results in Physics*, vol. 38, Article ID 105613, 2022.
- [21] K. Varshney and K. Poddar, "Prediction of wind properties in urban environments using artificial neural network," *Theoretical and Applied Climatology*, vol. 107, no. 3-4, pp. 579-590, 2012.
- [22] X. Gavalda, J. Ferrer-Gener, G. A. Kopp, and F. Giralt, "Interpolation of pressure coefficients for low-rise buildings of

- different plan dimensions and roof slopes using artificial neural networks,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 99, no. 5, pp. 658–664, 2011.
- [23] Y. Chen, G. A. Kopp, and D. Surry, “Interpolation of wind-induced pressure time series with an artificial neural network,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 90, no. 6, pp. 589–615, 2002.
- [24] J. Y. Fu, S. G. Liang, and Q. S. Li, “Prediction of wind-induced pressures on a large gymnasium roof using artificial neural networks,” *Computers & Structures*, vol. 85, no. 3–4, pp. 179–192, 2007.
- [25] Y. Chen, G. A. Kopp, and D. Surry, “Prediction of pressure coefficients on roofs of low buildings using artificial neural networks,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 91, no. 3, pp. 423–441, 2003.
- [26] T. J. Nikose and R. S. Sonparote, “Dynamic along wind response of tall buildings using Artificial Neural Network,” *Cluster Computing*, vol. 22, no. S2, pp. 3231–3246, 2019.
- [27] T. J. Nikose and R. S. Sonparote, “Dynamic wind response of tall buildings using artificial neural network,” *The Structural Design of Tall and Special Buildings*, vol. 28, no. 13, 2019.
- [28] A. B. Çolak, “Experimental study for thermal conductivity of water-based zirconium oxide nanofluid: developing optimal artificial neural network and proposing new correlation,” *International Journal of Energy Research*, vol. 45, no. 2, pp. 2912–2930, 2021.
- [29] A. Thammano and P. Ruxpakawong, “Nonlinear dynamic system identification using recurrent neural network with multi-segment piecewise-linear connection weight,” *Memetic Computing*, vol. 2, no. 4, pp. 273–282, 2010.
- [30] Y. You and M. Nikolaou, “Dynamic process modeling with recurrent neural networks,” *AIChE Journal*, vol. 39, no. 10, pp. 1654–1667, 1993.
- [31] F. A. Gers, D. Eck, and J. Schmidhuber, “Applying LSTM to time series predictable through time-window approaches,” in *Artificial Neural Networks - ICANN 2001*. ICANN 2001, G. Dorffner, H. Bischof, and K. Hornik, Eds., Vol. vol 2130, Springer, Berlin, Germany, 2001.
- [32] C. J. Su and Y. Li, “Recurrent neural network based real-time failure detection of storage devices,” *Microsystem Technologies*, vol. 28, no. 2, pp. 621–633, 2019.
- [33] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, “Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis,” *Cognitive Computation*, vol. 10, no. 4, pp. 639–650, 2018.
- [34] W. Ying, L. Zhang, and H. Deng, “Sichuan dialect speech recognition with deep LSTM network,” *Frontiers of Computer Science*, vol. 14, no. 2, pp. 378–387, 2020.
- [35] A. Graves, D. Eck, N. Beringer, and J. Schmidhuber, “Biologically plausible speech recognition with LSTM neural nets,” in *Biologically Inspired Approaches to Advanced Information Technology*. BioADIT 2004, A. J. Ijspeert, M. Murata, and N. Wakamiya, Eds., Vol. 3141, Springer, Berlin, Germany, 2004.
- [36] J. Liu, A. Shahroudy, D. Xu, and G. Wang, “Spatio-temporal LSTM with trust gates for 3D human action recognition,” in *Computer Vision - ECCV 2016*. ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Vol. 9907, Springer, Cham, Switzerland, 2016.
- [37] S. Lawrence, C. L. Giles, and A. C. Tsoi, “Lessons in neural network training: overfitting may be harder than expected,” in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, RI, USA, July 1997.
- [38] S. Geman, E. Bienenstock, and R. Doursat, “Neural networks and the bias/variance dilemma,” *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.
- [39] D. R. Hush and B. G. Horne, “Progress in supervised neural networks,” *IEEE Signal Processing Magazine*, vol. 10, no. 1, pp. 8–39, 1993.
- [40] A. Pozos-Estrada, R. Gómez, and H. P. Hong, “Use of Neural network to predict the peak ground accelerations and pseudo spectral accelerations for Mexican Inslab and Interplate Earthquakes,” *Geofísica Internacional*, vol. 53, no. 1, pp. 39–57, 2014.
- [41] R. Flores-Mendoza, J. U. Rodríguez-Alcántara, A. Pozos-Estrada, and R. Gómez, “Use of artificial neural networks to predict strong ground motion duration of interplate and inslab Mexican earthquakes for soft and firm soils,” *Geofísica Internacional*, vol. 61, no. 3, pp. 153–179, 2022.
- [42] D. T. Pham and S. J. Oh, “A recurrent backpropagation neural network for dynamic system identification,” *Journal of Systems Engineering*, vol. 2, no. 4, pp. 213–223, 1992.
- [43] C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulation*, John Wiley & Sons, New York, NY, USA, 1999.
- [44] D. T. Pham and D. Karaboga, “Training Elman and Jordan networks for system identification using genetic algorithms,” *Artificial Intelligence in Engineering*, vol. 13, no. 2, pp. 107–117, 1999.
- [45] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [46] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.
- [47] J. F. Kolen and S. C. Kremer, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies,” in *A Field Guide to Dynamical Recurrent Networks*. IEEE Press, Piscataway, NJ, USA, 2001.
- [48] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-term residential load forecasting based on LSTM recurrent neural network,” *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.
- [49] A. Zemmari and J. Benois-Pineau, *Deep Learning in Mining of Visual Content*, Springer, Cham, Switzerland, 2020.
- [50] J. Counihan, “Adiabatic atmospheric boundary layers: a review and analysis of data from the period 1880–1972,” *Atmospheric Environment*, vol. 9, no. 10, pp. 871–905, 1975.
- [51] E. Amaya-Gallardo, A. Pozos-Estrada, and R. Gómez, “Characteristics and calibration of the Mexican boundary layer wind tunnel at UNAM,” *Ingeniería: Investigación y Tecnología*, vol. XXIII, no. 1, 2022.
- [52] E. Simiu and R. H. Scanlan, *Wind Effects on Structures*, John Wiley & Sons, New York, NY, USA, 1996.
- [53] P. J. Richards, R. Hoxey, B. D. Connell, and D. P. Lander, “Wind-tunnel modelling of the Silsoe cube,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 95, no. 9–11, pp. 1384–1399, 2007.
- [54] L. V. Fausset, *Fundamentals of Neural Networks: Architecture, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1994.