

Article

Prediction of Maximum Story Drift of MDOF Structures under Simulated Wind Loads Using Artificial Neural Networks

Omar Payán-Serrano, Edén Bojórquez *, Juan Bojórquez, Robespierre Chávez, Alfredo Reyes-Salazar, Manuel Barraza, Arturo López-Barraza, Héctor Rodríguez-Lozoya and Edgar Corona

Facultad de Ingeniería, Universidad Autónoma de Sinaloa, Culiacán, Sinaloa 80040, Mexico; omarijs@hotmail.com (O.P.-S.); jbm_squall_cloud@hotmail.com (J.B.); robespierre_chavez@hotmail.com (R.C.); reyes@uas.edu.mx (A.R.-S.); mb12_05@hotmail.com (M.B.); alopezb@uas.edu.mx (A.L.-B.); rolohe@uas.edu.mx (H.R.-L.); edgar_corvi@hotmail.com (E.C.)

* Correspondence: eden@uas.edu.mx; Tel.: +52-667-713-4053

Academic Editor: Giuseppe Lacidogna

Received: 3 April 2017; Accepted: 25 May 2017; Published: 30 May 2017

Abstract: The aim of this paper is to investigate the prediction of maximum story drift of Multi-Degree of Freedom (MDOF) structures subjected to dynamics wind load using Artificial Neural Networks (ANNs) through the combination of several structural and turbulent wind parameters. The maximum story drift of 1600 MDOF structures under 16 simulated wind conditions are computed with the purpose of generating the data set for the networks training with the Levenberg–Marquardt method. The Shinozuka and Newmark methods are used to simulate the turbulent wind and dynamic response, respectively. In order to optimize the computational time required for the dynamic analyses, an array format based on the Shinozuka method is presented to perform the parallel computing. Finally, it is observed that the already trained ANNs allow for predicting adequately the maximum story drift with a correlation close to 99%.

Keywords: dynamic analysis; wind simulation; artificial neural networks; parallel computing

PACS: 7.05.Tp

MSC: 7.05.Tp

JEL Classification: 7.05.Tp

1. Introduction

The maximum story drift of Multi-Degree of Freedom (MDOF) structures under wind loads is an important parameter for the structural design; however, for structural engineers, the estimation of this parameter by dynamic analysis could be impractical. The dynamic analysis of structures is complex and it requires real wind speed records, which are not available in most of the cases. Several studies suggest that the use of spectral density models are adequate to simulate the velocity field of the turbulent wind [1–3]. Due to the complexity of obtaining synthetic wind records and the dynamic structural response, the engineers have used non-complex methods such as the equivalent static analysis, which tries to consider the dynamic effects of the turbulent wind supported by the use of the well-known dynamic amplification factor [4]. Some modifications and improvements to the equivalent static method, originally presented by Davenport [5], have been proposed by different authors [6,7]. Although these methods are not complex, they require several calculation steps in order to estimate

variables such as power spectral density, background response, resonant response and others. The use of a prediction model that adequately represents a considerable amount of structures under wind loads could be useful. Nowadays, thanks to advances in computer hardware and software, Artificial Neural Networks (ANNs) have proven to be computational models capable of being applied in tasks as prediction, classification, data processing, robotic and engineering problems [8–11]. For this reason, the aim of this study is to evaluate the efficiency of ANNs as computational models for prediction of the maximum story drift of MDOF structures in terms of regional wind speeds, terrain categories and structural characteristics such as height, slenderness ratio, fundamental period, distributed mass and stiffness ratio. The Newmark method is employed to resolve the differential equation of motion. This method is frequently used in numerical evaluation of the dynamic response of structures [12–14]. The wind loads applied in the dynamic analyses are generated from synthetic records obtained by spectral representation with the well-known Shinozuka method [15], which has been shown to give the best results in terms of overall quality of the signal [16]. The dynamic responses of a considerable amount of MDOF structures are used to build a database about the maximum story drift related to structural characteristics and wind conditions. This database is employed to the training processes of ANNs with the Levenberg–Marquardt (LM) method, which is a standard technique for solving nonlinear least squares problems [17]. The LM method is an optimization procedure with features of stable and fast convergence [18]. The training process is applied in ANN models known as multilayer perceptrons that consist of neurons located in multiple layers and interconnected to form a network with forward propagation (Figure 1a). The multilayer perceptron approach resolve problems that are not linearly separable, which is the main limitation of the standard linear perceptron method. Hecht-Nielsen [8] proved that one hidden layer of neurons suffices to model any solution surface of practical interest. In this paper, some multilayer perceptron models with different amounts of neurons in a single hidden layer are evaluated to predict the maximum story drift of 25,600 case studies.

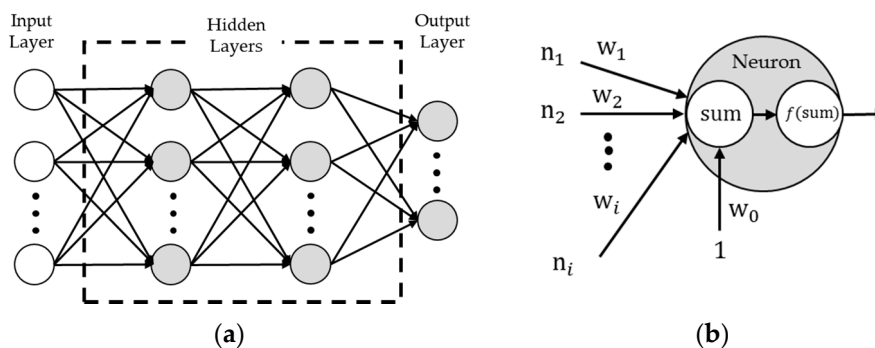


Figure 1. Graphic description of an Artificial Neural Network (ANN) (a) topology of multilayer perceptron (forward propagation); (b) internal process for each neuron.

Figure 1b shows the internal mathematical process of each neuron. The output is given by an activation function f which depends on the sum of the inputs $n_i w_i$:

$$\text{sum} = 1w_0 + n_1w_1 + n_2w_2 + \dots + n_iw_i, \tag{1}$$

where n_i is the output of other neuron and w_i is the synaptic weight (Figure 1b). The sigmoid function (values between 0 and 1) is frequently used like activation function and its mathematical form is:

$$f(\text{sum}) = n = \frac{1}{1 + e^{-\text{sum}}}. \tag{2}$$

An important property is that its derivative can be expressed by the function itself:

$$f'(\text{sum}) = d = n(1 - n). \tag{3}$$

The paper is organized in the following way: Section 2 exposes the detail of the theory employed in network training, wind simulation and dynamic response. Section 3 describes the set of structural characteristics and wind conditions; then, the results of the application of theoretical framework are shown. Finally, a brief discussion of the main results is presented in Section 4.

2. Theoretical Framework

2.1. Network Training

The Sum Square Error (SSE) is a performance index and it is used in order to evaluate the training of the network:

$$E = \frac{1}{2} \sum_{p=1}^{N_p} \sum_{\dot{o}=1}^{N_o} e_{p\dot{o}}^2 = \frac{1}{2} \sum_{p=1}^{N_p} \sum_{\dot{o}=1}^{N_o} (t_{p\dot{o}} - y_{p\dot{o}})^2, \tag{4}$$

where N_p is the number of patterns; N_o is the number of outputs; $e_{p\dot{o}}$ is the error value for the pattern p at the output \dot{o} ; $t_{p\dot{o}}$ is the target value for the pattern p at the output \dot{o} ; and $y_{p\dot{o}}$ is the output value of the network by the pattern p at the output \dot{o} .

The derivative of the error, as a function of the weights, defines the gradient, which gives information to find a minimum. The mathematical form is as follows:

$$\mathbf{G} = \frac{\partial \mathbf{E}(\mathbf{w})}{\partial \mathbf{w}}, \tag{5}$$

where \mathbf{G} is the gradient vector; and \mathbf{w} is the weight vector.

The use of the gradient as principal parameter for reducing the error is known as the descendent gradient method. The update of weights to minimize the error by descendent gradient can be written as:

$$\mathbf{w} + \Delta \mathbf{w} = \mathbf{w} - \alpha \mathbf{G}, \tag{6}$$

where α is the learning constant and $\Delta \mathbf{w}$ the update for the weights. This method is stable but could be very slow with a lot of interactions due to inefficient values of α . The Newton method allows for reducing the number of interactions using a square matrix of second-order partial derivatives. This matrix can be defined by Taylor series. According to Taylor, it is possible to represent a function from an infinite sum of terms that are calculated by derivatives and powers in a single point. The Taylor polynomial of second degree can be written in its compact form as:

$$T(x) \approx T(a) + \mathbf{G}(a)(x - a) + 1/2(x - a)^T \mathbf{H}(a)(x - a), \tag{7}$$

where a is the derivation point, $\mathbf{G}(a)$ is the gradient vector and $\mathbf{H}(a)$ is the Hessian matrix.

Using the Taylor approximation, Equation (7), to estimate the next possible value of the error function, i.e., $T(x) = E(\mathbf{w} + \Delta \mathbf{w})$, and considering the derivation point at $a = \mathbf{w}$, the expression would be:

$$E(\mathbf{w} + \Delta \mathbf{w}) \approx E(\mathbf{w}) + \mathbf{G}(\mathbf{w})\Delta \mathbf{w} + 1/2\Delta \mathbf{w}^T \mathbf{H}(\mathbf{w})\Delta \mathbf{w}. \tag{8}$$

Assuming that the value of error is zero $E(\mathbf{w} + \Delta \mathbf{w}) = 0$ with the aim to close the error to a minimum, it is possible to reduce the expression with the derivative with respect to $\Delta \mathbf{w}$:

$$0 = \mathbf{G}(\mathbf{w}) + \mathbf{H}(\mathbf{w})\Delta \mathbf{w}, \tag{9}$$

therefore:

$$\Delta \mathbf{w} = -\mathbf{H}(\mathbf{w})^{-1}\mathbf{G}(\mathbf{w}), \tag{10}$$

written in matrix form :

$$\begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_{N_w} \end{bmatrix} = - \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_{N_w}} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} & \cdots & \frac{\partial^2 E}{\partial w_2 \partial w_{N_w}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_{N_w} \partial w_1} & \frac{\partial^2 E}{\partial w_{N_w} \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_{N_w}^2} \end{bmatrix}^{-1} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{N_w} \end{bmatrix} \tag{11}$$

The updating of the weights by Newton method is obtained with the following expression:

$$\mathbf{w} + \Delta\mathbf{w} = \mathbf{w} - \mathbf{H}(\mathbf{w})^{-1}\mathbf{G}(\mathbf{w}). \tag{12}$$

The Hessian matrix \mathbf{H} of second-order partial derivatives of total error function have to be calculated and it could be very complicated. In order to simplify the calculating process, a matrix of first-order partial derivatives is presented as a Jacobian matrix, which has a mathematical relation with a Gradient and Hessian matrix. By combining the error of Equation (4) and the definition of gradient (5), it is possible to determinate that:

$$g_i = \frac{\partial \mathbf{E}}{\partial w_i} = \frac{\partial \left(\frac{1}{2} \sum_{p=1}^{N_p} \sum_{\dot{o}=1}^{N_o} e_{p\dot{o}}^2 \right)}{\partial w_i} = \sum_{p=1}^{N_p} \sum_{\dot{o}=1}^{N_o} \left(\frac{\partial e_{p\dot{o}}}{\partial w_i} e_{p\dot{o}} \right). \tag{13}$$

The error e expressed in vector form is:

$$\mathbf{e} = \left[e_{1,1} \quad \dots \quad e_{1,N_o} \quad e_{2,1} \quad \dots \quad e_{N_p,1} \quad \dots \quad e_{N_p,N_o} \right]^T, \tag{14}$$

and the Jacobian of the error e :

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \dots & \frac{\partial e_{1,N_o}}{\partial w_1} & \frac{\partial e_{2,1}}{\partial w_1} & \dots & \frac{\partial e_{N_p,1}}{\partial w_1} & \dots & \frac{\partial e_{N_p,N_o}}{\partial w_1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial e_{1,1}}{\partial w_{N_w}} & \dots & \frac{\partial e_{1,N_o}}{\partial w_{N_w}} & \frac{\partial e_{2,1}}{\partial w_{N_w}} & \dots & \frac{\partial e_{N_p,1}}{\partial w_{N_w}} & \dots & \frac{\partial e_{N_p,N_o}}{\partial w_{N_w}} \end{bmatrix}^T \tag{15}$$

Therefore, the gradient could be represented for the Jacobian matrix by the following expression:

$$\mathbf{G} = \mathbf{J}^T \mathbf{e}. \tag{16}$$

Each element of the Hessian matrix can be expressed in terms of error $e_{p\dot{o}}$ resolving the cross partial derivative of the sum square error:

$$h_{i,j} = \frac{\partial^2 \mathbf{E}}{\partial w_i \partial w_j} = \frac{\partial^2 \left(\frac{1}{2} \sum_{p=1}^{N_p} \sum_{\dot{o}=1}^{N_o} e_{p\dot{o}}^2 \right)}{\partial w_i \partial w_j} = \sum_{p=1}^{N_p} \sum_{\dot{o}=1}^{N_o} \left(\frac{\partial e_{p\dot{o}}}{\partial w_i} \frac{\partial e_{p\dot{o}}}{\partial w_j} \right) + \bar{S}_{i,j}, \tag{17}$$

where $\bar{S}_{i,j}$ is:

$$\bar{S}_{i,j} = \sum_{p=1}^{N_p} \sum_{\dot{o}=1}^{N_o} \left(\frac{\partial^2 e_{p\dot{o}}}{\partial w_i \partial w_j} e_{p\dot{o}} \right), \tag{18}$$

For the Gauss–Newton method, it is assumed that $\bar{S}_{i,j} \approx 0$; therefore, the Hessian matrix could be related with the Jacobian matrix by the following form:

$$\mathbf{H} \approx \mathbf{J}^T \mathbf{J}. \tag{19}$$

The updating of the weights by Gauss–Newton method is obtained as a function of the matrix of first-order partial derivatives:

$$\mathbf{w} + \Delta\mathbf{w} = \mathbf{w} - \left(\mathbf{J}^T \mathbf{J} \right)^{-1} \mathbf{J}^T \mathbf{e}. \tag{20}$$

The Gauss–Newton algorithm has better performance than the Newton method because it does not require the calculation of Hessian matrix. However, the Gauss–Newton algorithm could have problems of matrices not invertible. In order to resolve this problem, LM modified the Hessian matrix with an approximation:

$$\mathbf{H} \approx \mathbf{J}^T \mathbf{J} + \mu \mathbf{I}, \tag{21}$$

where μ is a value always positive such with combination to an identity matrix \mathbf{I} makes positive definite and therefore can be invertible. The updating of the weights by LM method can be presented as:

$$\mathbf{w} + \Delta \mathbf{w} = \mathbf{w} - \left(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I} \right)^{-1} \mathbf{J}^T \mathbf{e}. \tag{22}$$

Calculation of Jacobian Matrix

According to Equations (4) and (15), the elements of Jacobian matrix can be calculated as:

$$\frac{\partial e_{p\dot{o}}}{\partial w_i} = \frac{\partial (t_{p\dot{o}} - y_{p\dot{o}})}{\partial w_i} = - \frac{\partial y_{p\dot{o}}}{\partial w_i}. \tag{23}$$

Considering an ANN of one hidden layer (Figure 2), the backpropagation is used in order to resolve the Equation (23). As shown in Figure 2, the superscript denotes the position of the layer in the network while the subscript describes the position of the neuron in the layer. In the weights, the double subscript indicates the neural connection in backpropagation order. The number of input and hidden neurons are defined by N_x and N_2 , respectively.

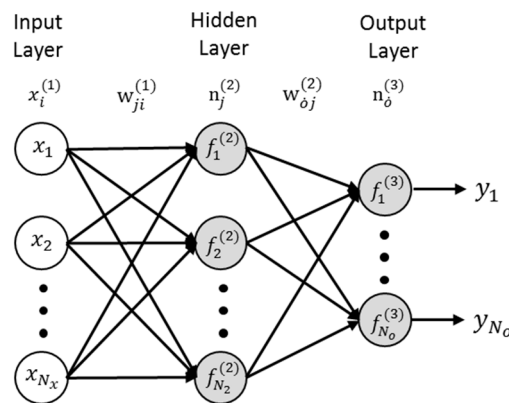


Figure 2. Perceptron with one hidden layer.

For a given pattern, the output $y_{p\dot{o}}$ could be calculated with backward computation:

$$y_{p\dot{o}} = f_o^{(3)} \left(\sum_{j=1}^{N_2} n_j^{(2)} w_{oj}^{(2)} \right), \tag{24}$$

$$y_{p\dot{o}} = f_o^{(3)} \left(\sum_{j=1}^{N_2} f_j^{(2)} \left(\sum_{i=1}^{N_x} n_i^{(1)} w_{ji}^{(1)} \right) w_{oj}^{(2)} \right), \tag{25}$$

deriving Equations (24) and (25) with respect to $w_{oj}^{(2)}$ and $w_{ji}^{(1)}$, respectively:

$$\frac{\partial y_{p\dot{o}}}{\partial w_{oj}^{(2)}} = d_o^{(3)} n_j^{(2)}, \tag{26}$$

$$\frac{\partial y_{p\dot{o}}}{\partial w_{ji}^{(1)}} = d_o^{(3)} d_j^{(2)} w_{oj}^{(2)} n_i^{(1)}. \tag{27}$$

For the problem with patterns and outputs, the Jacobian and error matrices can be organized as the pseudocode shown in Figure 3.

```

For all patterns
  % Forward computation
  ...
  % Backward computation
  For all outputs
    Calculate error  $e_{p\delta}$  Equation (4)
    Fill error array  $e$  Equation (14)
    For all layers
      %Calculate elements of Jacobian matrix
      Equations (26) and (27)
      Fill error array  $J$  Equation (15)
    End
  End
End
End
    
```

Figure 3. Pseudocode for Jacobian matrix.

Therefore, the training process using LM method can be planned as:

1. Calculate the forward computation with initial weights (randomly generated).
2. weights with Equation (22).
3. the current total error. If the current total error is increased, then reset the weights to the previous values and increase μ by a factor of 10. Then, go to Step 2. Otherwise, keep the new weights and decrease μ by a factor of 10.
4. Go to Step 2 until the current total error is smaller than the required value.

2.2. Wind Simulation

Among various simulation methods, the spectral representation methods appear to be the most widely used because they are fast and conceptually straightforward. Wind synthetic records can be generated from the spectral representation method proposed by Shinozuka and Jan [15]. Considering the case of $N_{\bar{p}}$ stationary stochastic processes $u_j(t)$, $j = 1, 2, 3, \dots, N_{\bar{p}}$ and discrete time $t = i\Delta t$, $i = 0, 1, 2, \dots, N_s$, the mathematical expression for the generation of synthetic records is:

$$u_j(t) = \sum_{k=1}^j \sum_{n=1}^{N_f} \left| \bar{H}_{jk}(\omega_n) \right| \sqrt{2\Delta\omega} \cos[\omega_n t + \theta_{kn}], \tag{28}$$

where $u_j(t)$ is a signal with zero mean of size $1 \times N_s$, representing the turbulent part of the longitudinal component at j th point in the space; $\bar{H}_{jk}(\omega_n)$ is an element of the lower triangular matrix $\bar{\mathbf{H}}(\omega_n)$ of size $N_{\bar{p}} \times N_{\bar{p}}$, which is defined by the Cholesky factorization process from the cross spectral density matrix; θ_{kn} is an element of the random phase angle matrix Θ of size $N_{\bar{p}} \times N_f$, with uniform distribution between $[0, 2\pi]$; ω_n is the discrete angular frequency (rad/s); $\Delta\omega$ is the increment of angular frequency; and N_f is the amount of values contained in the discrete spectral density function.

As it was shown in Equation (28), the calculation of the Shinozuka expression requires serial computing. In order to save computing time, an expression in matrix multiplication format based on the Shinozuka method is presented to perform the parallel computing taking into account the advantage of current multicore processors that can solve several vectors of the resulting matrix at the same time. The expression in array format is:

$$\mathbf{u} = \sqrt{2\Delta\omega} \hat{\mathbf{A}} \bar{\mathbf{X}}, \tag{29}$$

where \mathbf{u} is a matrix with size $N_{\bar{p}} \times N_s$, and represents $N_{\bar{p}}$ stochastic process; $\hat{\mathbf{A}}$ is a matrix with size $N_{\bar{p}} \times (N_{\bar{p}} N_f)$, defined by Equation (30); $\bar{\mathbf{X}}$ is a matrix with size $(N_{\bar{p}} N_f) \times N_s$, defined by Equation (31):

$$\hat{\mathbf{A}} = \begin{bmatrix} \bar{H}_{1,1}(\omega_1) & \bar{H}_{2,1}(\omega_1) & \dots & \bar{H}_{N_{\bar{p}},1}(\omega_1) \\ \bar{H}_{1,2}(\omega_1) & \bar{H}_{2,2}(\omega_1) & \dots & \bar{H}_{N_{\bar{p}},2}(\omega_1) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{H}_{1,N_{\bar{p}}}(\omega_1) & \bar{H}_{2,N_{\bar{p}}}(\omega_1) & \dots & \bar{H}_{N_{\bar{p}},N_{\bar{p}}}(\omega_1) \\ \bar{H}_{1,1}(\omega_2) & \bar{H}_{2,1}(\omega_2) & \dots & \bar{H}_{N_{\bar{p}},1}(\omega_2) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{H}_{1,N_{\bar{p}}}(\omega_2) & \bar{H}_{2,N_{\bar{p}}}(\omega_2) & \dots & \bar{H}_{N_{\bar{p}},N_{\bar{p}}}(\omega_2) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{H}_{1,1}(\omega_{N_f}) & \bar{H}_{2,1}(\omega_{N_f}) & \dots & \bar{H}_{N_{\bar{p}},1}(\omega_{N_f}) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{H}_{1,N_{\bar{p}}}(\omega_{N_f}) & \bar{H}_{2,N_{\bar{p}}}(\omega_{N_f}) & \dots & \bar{H}_{N_{\bar{p}},N_{\bar{p}}}(\omega_{N_f}) \end{bmatrix}^T \tag{30}$$

$$\bar{\mathbf{X}} = \begin{bmatrix} \cos(\omega_1 t_1 + \theta_{1,1}) & \cos(\omega_1 t_2 + \theta_{1,1}) & \dots & \cos(\omega_1 t_{N_s} + \theta_{1,1}) \\ \cos(\omega_1 t_1 + \theta_{1,2}) & \cos(\omega_1 t_2 + \theta_{1,2}) & \dots & \cos(\omega_1 t_{N_s} + \theta_{1,2}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\omega_1 t_1 + \theta_{1,N_{\bar{p}}}) & \cos(\omega_1 t_2 + \theta_{1,N_{\bar{p}}}) & \dots & \cos(\omega_1 t_{N_s} + \theta_{1,N_{\bar{p}}}) \\ \cos(\omega_2 t_1 + \theta_{2,1}) & \cos(\omega_2 t_2 + \theta_{2,1}) & \dots & \cos(\omega_2 t_{N_s} + \theta_{2,1}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\omega_2 t_1 + \theta_{2,N_{\bar{p}}}) & \cos(\omega_2 t_2 + \theta_{2,N_{\bar{p}}}) & \dots & \cos(\omega_2 t_{N_s} + \theta_{2,N_{\bar{p}}}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\omega_{N_f} t_1 + \theta_{N_f,1}) & \cos(\omega_{N_f} t_2 + \theta_{N_f,1}) & \dots & \cos(\omega_{N_f} t_{N_s} + \theta_{N_f,1}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\omega_{N_f} t_1 + \theta_{N_f,N_{\bar{p}}}) & \cos(\omega_{N_f} t_2 + \theta_{N_f,N_{\bar{p}}}) & \dots & \cos(\omega_{N_f} t_{N_s} + \theta_{N_f,N_{\bar{p}}}) \end{bmatrix} \tag{31}$$

Cross-Spectral Density Matrix

The cross-spectral density matrix describes the influence of the turbulence components at two points at a given frequency. This influence is due to the spatial dimension of the vortices in the wind field. Considering Cholesky, if the cross spectral density matrix $S^0(\omega)$ is symmetric positive definite, then $S^0(\omega)$ can be decomposed as [1]:

$$S^0(\omega) = \bar{\mathbf{H}}(\omega)\bar{\mathbf{H}}^T(\omega), \tag{32}$$

where $S^0(\omega)$ is the cross spectral density matrix for angular frequency ω ; and $\bar{\mathbf{H}}(\omega)$ is a lower triangular matrix.

The process of Cholesky factorization is as follows:

$$\begin{aligned}
 \bar{H}_{11}(\omega) &= S_{11}^0(\omega)^{1/2} \\
 \bar{H}_{21}(\omega) &= S_{21}^0(\omega)/\bar{H}_{11}(\omega) \\
 \bar{H}_{22}(\omega) &= (S_{22}^0(\omega) - \bar{H}_{21}(\omega)^2)^{1/2} \\
 \bar{H}_{31}(\omega) &= S_{31}^0(\omega)/\bar{H}_{11}(\omega) \\
 &\vdots \\
 \bar{H}_{jk}(\omega) &= \left(S_{jk}^0(\omega) - \sum_{l=1}^{k-1} \bar{H}_{jl}(\omega)\bar{H}_{kl}(\omega) \right) / \bar{H}_{kk}(\omega) \\
 \bar{H}_{kk}(\omega) &= (S_{kk}^0(\omega) - \sum_{l=1}^{k-1} \bar{H}_{kl}(\omega)^2)^{1/2}.
 \end{aligned}
 \tag{33}$$

For two points with vertical separation r_y , the $S_{ij}^0(\omega)$ corresponding elements are obtained by the cross-spectral density function $S_{uu}(z_i, z_j, n)$:

$$S_{uu}(z_i, z_j, n) = \sqrt{S_u(z_i, n)S_u(z_j, n)}\sqrt{\text{coh}(r_y, n)},
 \tag{34}$$

where $S_{uu}(z_i, z_j, n)$ is the cross spectral density function for two longitudinal turbulent components at space points z_1 and z_2 ; $S_u(z_i, n)$ is the single power density spectrum at z_i , defined in Equation (38); $\sqrt{\text{coh}(r_y, n)}$ is the root-coherence function, defined in Equation (35); n is the frequency in Hz. The root-coherence function defines the static dependence between two turbulent components at two different points. This function tends to zero when the separation r_y increases; in other words, the cross-influence of the wind turbulent is inversely proportional to the separation. Davenport [19] suggested an exponential expression as root-coherence function:

$$\sqrt{\text{coh}(r_y, n)} = e^{(-C_y r_y \frac{n}{\bar{U}})},
 \tag{35}$$

where \bar{U} is the average speed between two points, calculated as $1/2[\bar{U}(z_i) + \bar{U}(z_j)]$; C_y is a non-dimensional decay constant, whose typical value is 10.

The variation of mean wind speed to different levels of height can be calculated with an expression known as power law:

$$\bar{U}(z) = f_s \bar{b} \bar{U}_{10, \text{II}} \left(\frac{z}{10} \right)^{\bar{\alpha}},
 \tag{36}$$

where f_s is a time scale factor, according to Mackey [20] $f_s = 0.702$ in order to scale 3-s to 10-min of mean wind speeds; \bar{b} is the roughness factor, defined in Table 1; $\bar{U}_{10, \text{II}}$ is the basic wind speed, and it is a mean wind speed taking at reference height 10 m and terrain category II; $\bar{\alpha}$ is the exponent, defined in Equation (37).

According to Counihanm [21], the exponent $\bar{\alpha}$ depends on the characteristics of roughness, and can be calculated with the following equation:

$$\bar{\alpha} = 0.096 \log_{10} z_0 + 0.016 (\log_{10} z_0)^2 + 0.24.
 \tag{37}$$

The power spectral density function used in this paper is the von Karman–Harris model [23,24], which has shown a very good approach at high frequency with application to wind codes [25], and it is considering the most mathematically correct wind spectrum according to [26]. The mathematical form of von Karman-Harris model is:

$$S_u(z, n) = \frac{4(\sigma_u^2) L_u(z) / \bar{U}(z)}{\left[1 + 70.8 \left(\frac{n L_u(z)}{\bar{U}(z)} \right)^2 \right]^{5/6}},
 \tag{38}$$

where $L_u(z)$ is the integral length scale; and σ_u is standard deviations for the turbulence components.

Table 1. Terrain categories and terrain parameters based in Eurocode [22].

Terrain Category	z_0 (m)	\bar{b}
I—Lakes or flat and horizontal area with negligible vegetation and without obstacles	0.01	1.17
II—Area with low vegetation such as grass and isolated obstacles (trees, buildings) with separations of at least 20 obstacle heights	0.05	1.00
III—Area with regular cover of vegetation or buildings or with isolated obstacles with separations of maximum 20 obstacle heights (such as villages, suburban terrain, permanent forest)	0.30	0.77
IV—Area in which at least 15% of the surface is covered with buildings and their average height exceeds 15 m	1.00	0.55

The standard deviations of wind speed are related with the turbulence intensity. The turbulence intensity is the most basic measure of turbulence, and it is defined by the ratio of the standard deviation of the wind speed to the mean wind speed:

$$I_u(z) = \sigma_u / \bar{U}(z). \tag{39}$$

The turbulence intensity can be estimated by the following empirical expression [22]:

$$I_u(z) = \frac{1}{\ln(z/z_0)}. \tag{40}$$

The integral length scales are a measure of the sizes of the vortices in the wind or, in other words, the average size of a gust in a given direction [26]. In wind codes, a typical expression in order to estimate the integral length scales is [22]:

$$L_u(z) = 300 \left(\frac{z}{z_0} \right)^v, \tag{41}$$

where the exponent v depends on the roughness length z_0 , and it can be calculated by:

$$v = 0.67 + 0.05 \ln(z_0). \tag{42}$$

2.3. Dynamic Response

The Newmark method is one of the numerical integration methods most frequently used in structural dynamic analysis; software packages such as OpeenSees (v2002, CA, USA), Ruaumoko (v2004, Christchurch, New Zealand) and SAP2000 (v2012, CA, USA) [27–29] include routines of dynamic analysis by this method. This procedure is based on two types of assumptions in order to simplify and solve the differential equations of motion. The assumptions are focused on determining the acceleration between two time instants, which may be linear or a constant mean value. The idealization of a prismatic building as shear-frame or shear-building (Figure 4) can be represented by a Multi-Degree-of-Freedom (MDOF) system [30]. The governing equation of motion for MDOF systems subjected to external forces can be given in a matrix form as next:

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F}, \tag{43}$$

where \mathbf{M} , \mathbf{C} and \mathbf{K} are the mass, damping and stiffness matrices, respectively, \mathbf{d} , $\dot{\mathbf{d}}$ and $\ddot{\mathbf{d}}$ are the displacement, velocity and acceleration vectors; and \mathbf{F} is a matrix with the force vectors.

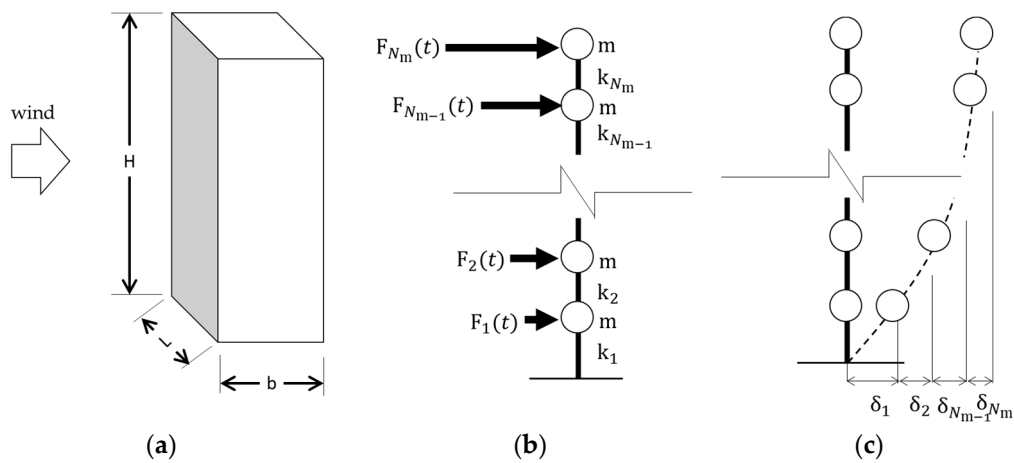


Figure 4. (a) Prismatic buildings with rectangular cross section; (b) multi-degree-of-freedom structure; (c) story drift.

The matrix form of the mass and stiffness matrices for a MDOF structure (Figure 4b) with shear resistant are:

$$\mathbf{M} = \begin{bmatrix} m & 0 & \dots & \dots & 0 \\ 0 & m & 0 & \dots & 0 \\ \dots & 0 & \dots & 0 & \dots \\ 0 & \dots & 0 & m & 0 \\ 0 & \dots & \dots & 0 & m \end{bmatrix}; \mathbf{K} = \begin{bmatrix} k_1 + k_2 & -k_2 & \dots & \dots & 0 \\ -k_2 & k_2 + k_3 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & k_{N_m-1} - k_{N_m} & -k_{N_m} \\ 0 & \dots & \dots & -k_{N_m} & k_{N_m} \end{bmatrix} \quad (44)$$

The form of the damping matrix \mathbf{C} depends on the damping characteristic of the structure, which can be classical or nonclassical damping. The classical damping is an appropriate idealization if similar damping mechanisms are distributed throughout the structure, e.g., a multistory building with a similar structural system and structural materials over its height [30]. In this work, the Rayleigh Damping method is used in order to build a classical matrix \mathbf{C} . The damping matrix can be approached by the following equation of Rayleigh:

$$\mathbf{c} = a_0 \mathbf{M} + a_1 \mathbf{K}, \quad (45)$$

where a_0 and a_1 Rayleigh coefficient and could be estimated by:

$$a_0 = \zeta \frac{2\omega_i \omega_j}{\omega_i + \omega_j}; a_1 = \zeta \frac{2}{\omega_i + \omega_j}, \quad (46)$$

where ζ is the damping ratio for the first mode; and ω_i and ω_j are two angular frequencies that correspond to two adjacent vibration modes.

Assuming a free vibration system without damping, the equation of motion is as follows:

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{0}. \quad (47)$$

Substituting $\mathbf{d} = \mathbf{\Phi}\mathbf{q}(t) = \mathbf{\Phi}\sin(\omega t)$ in Equation (47), the following expression is an eigenvalue problem:

$$\mathbf{K}\mathbf{\Phi} = \mathbf{M}\mathbf{\Phi}\omega^2, \quad (48)$$

where $\mathbf{\Phi}$ is modal matrix that represents the deflected shape for each vibration modes; \mathbf{q} is a vector for each instant time and represents the modal coordinates; and ω is a diagonal matrix with the angular frequency of vibration.

Rewriting the equation of motion to the modal form and multiplying all expressions by Φ^T :

$$\bar{M}\ddot{\mathbf{q}} + \bar{C}\dot{\mathbf{q}} + \bar{K}\mathbf{q} = \bar{F}, \tag{49}$$

where $\bar{M} = \Phi^T M \Phi$, $\bar{C} = \Phi^T C \Phi$, $\bar{K} = \Phi^T K \Phi$ and $\bar{F} = \Phi^T F$ are modal transformation matrices corresponding to the mass, damping, stiffness, and forces matrices, respectively.

The procedure of the Newmark method for MDOF structures is shown in Table 2, see Chopra [30], where i indicates the time step. The factors β and γ are constant values taking $\beta = 1/4$ and $\gamma = 1/2$ for the case of average acceleration and $\beta = 1/6$ y $\gamma = 1/2$ for the linear acceleration.

Table 2. Procedure of the Newmark method for multi-degrees of freedom systems.

Step-by-step procedure:
Initial calculations:
1.1 $\hat{K} = \bar{K} + \frac{\gamma}{\beta \Delta t} \bar{C} + \frac{1}{\beta(\Delta t)^2} \bar{M}$
1.2 $\mathbf{a} = \frac{1}{\beta \Delta t} \bar{M} + \frac{\gamma}{\beta} \bar{C}, \mathbf{b} = \frac{1}{2\beta} \bar{M} + \Delta t \left(\frac{\gamma}{2\beta} - 1 \right) \bar{C}$
Calculations for each time step i
2.1 $\bar{F}_i = \Phi^T F_i$
2.2 $\Delta \hat{F}_i = \Delta \bar{F}_i + \mathbf{a} \dot{\mathbf{q}}_i + \mathbf{b} \ddot{\mathbf{q}}_i$
2.3 $\Delta \mathbf{q}_i = \hat{K}^{-1} \Delta \hat{F}_i$
2.4 $\Delta \dot{\mathbf{q}}_i = \frac{\gamma}{\beta \Delta t} \Delta \mathbf{q}_i - \frac{\gamma}{\beta} \dot{\mathbf{q}}_i + \Delta t \left(1 - \frac{\gamma}{2\beta} \right) \ddot{\mathbf{q}}_i$
2.5 $\Delta \ddot{\mathbf{q}}_i = \frac{1}{\beta(\Delta t)^2} \Delta \mathbf{q}_i - \frac{1}{\beta \Delta t} \dot{\mathbf{q}}_i - \frac{1}{2\beta} \ddot{\mathbf{q}}_i$
2.6 $\mathbf{q}_{i+1} = \mathbf{q}_i + \Delta \mathbf{q}_i, \dot{\mathbf{q}}_{i+1} = \dot{\mathbf{q}}_i + \Delta \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_{i+1} = \ddot{\mathbf{q}}_i + \Delta \ddot{\mathbf{q}}_i$
2.7 $\mathbf{d}_{i+1} = \Phi \mathbf{q}_{i+1}$
Repetition for the next time step. Replace i by $i + 1$ and implement Steps 2.1 and 2.7 for the next time step.

3. Application of Theoretical Framework

3.1. Data Generation

In this paper, several steel shear-buildings discretized as MDOF structures with different structural vibration period T_0 were analyzed using wind loads from synthetic records of 10 min. The case studies are defined by the combinations of some parameters that consider diverse wind conditions and structural characteristics (see Figure 5). The range of values for each parameter was selected to cover a considerable number of sensitive structures to the dynamic effects of the wind. The values of basic wind speed or regional speed are presented in 3-s gust wind speeds because this is more common in this type of design; however, the time scale factor was considered for 10 min. The slenderness relation is defined as $R_e = H/b$, where H is the total height of the structure and b is the plane dimension parallel to the direction of the wind (see Figure 4a). The stiffness ratio corresponds to the reduction of the stiffness along the height between the first and the last story level, i.e., $R_k = k_{N_m} / k_1$ (see Figure 4b). The distributed mass represents the mass concentration per unit area for each floor, and this parameter defines the lumped mass m of each story level.

The dynamic loads are generated from synthetic records, where the turbulent parts of the longitudinal components are obtained by the Shinozuka method. According to Bernoulli, the wind is air mass with kinetic energy. It can be expressed in terms of pressure or force acting on a contact area; the mathematical form for a structure is [31]:

$$F_j(t) = \frac{1}{2} C_d \rho A_j \left(U_j(t) - \dot{d}_j(t) \right)^2 \tag{50}$$

where $F_j(t)$ is the wind force on the j -th floor; C_d is the drag coefficient, for rectangular prismatic $C_d \approx 1$ [31], which is used to quantify the drag or resistance of an object in a fluid environment; ρ is the density of the air, for this study $\rho = 1.25 \text{ kg/m}^3$; A_j is the contact area on the j -th floor; and

$U_j(t)$ is the longitudinal component of the wind. It is defined as the sum of the mean velocity and its corresponding turbulent part $U_j(t) = \bar{U}_j + u_j(t)$; $\dot{d}_j(t)$ is the velocity of the mass lumped on the j th floor.

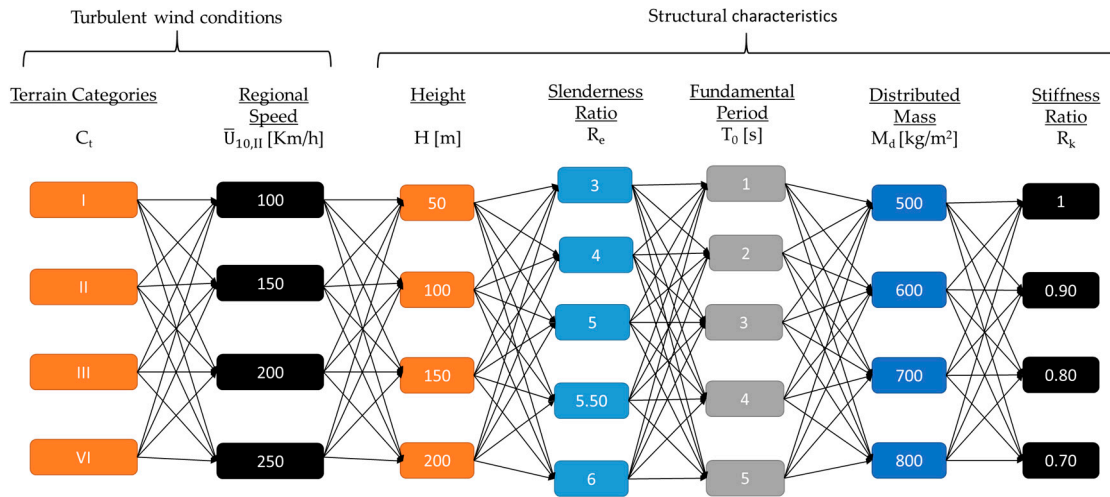


Figure 5. Parameters for definition of the structures and dynamic wind loads.

The stiffness matrix, required for the dynamic analysis, was calculated by an iterative process, where the all parameters of structure definition are involved. The pseudo-code used for the creation of the mass and stiffness matrices is shown in Figure 6.

```

For all heights H
  Define the story height h = 5.
  Define the size of the stiffness matrix with H/h.
  For all slenderness ratios Re
    For all distributed masses Md
      Create the mass matrix with current values of Re and Md.
      Save matrix M corresponding to current parameter values H, Re and Md.
      For all stiffness ratios Rk
        Create a differential stiffness matrix ΔK with values small enough considering current Rk.
        For all fundamental period T0
          Create a zero matrix K.
          Define a start value Ti = 0.
          While the difference between the current T0 and the estimated Ti is greater than expected.
            K = K + ΔK
            Resolve eigenvalue problem for  $\tilde{\mathbf{A}}\Phi = \Omega^2\Phi$  where  $\tilde{\mathbf{A}} = \mathbf{M}^{-1}\mathbf{K}$ .
            Estimate Ti from  $\Omega^2$ .
          End While
          Save matrix K corresponding to current parameter values H, Re, Md, Rk and T0.
        End For
      End For
    End For
  End For
End For

```

Figure 6. Pseudo-code in order to calculate mass and stiffness matrices.

With the loads calculated and the structures defined, the Newmark method is used to obtain the response of 1600 structures and each one is exposed to 16 scenarios of turbulent wind. Figure 7 shows the behavior of the maximum story drift for a certain group of structures and wind conditions. It is observed that the maximum story drift is directly proportional to the fundamental period and the wind speed and inversely proportional to the terrain roughness. The behavior of these surfaces is appropriate according to the variations of the parameters.

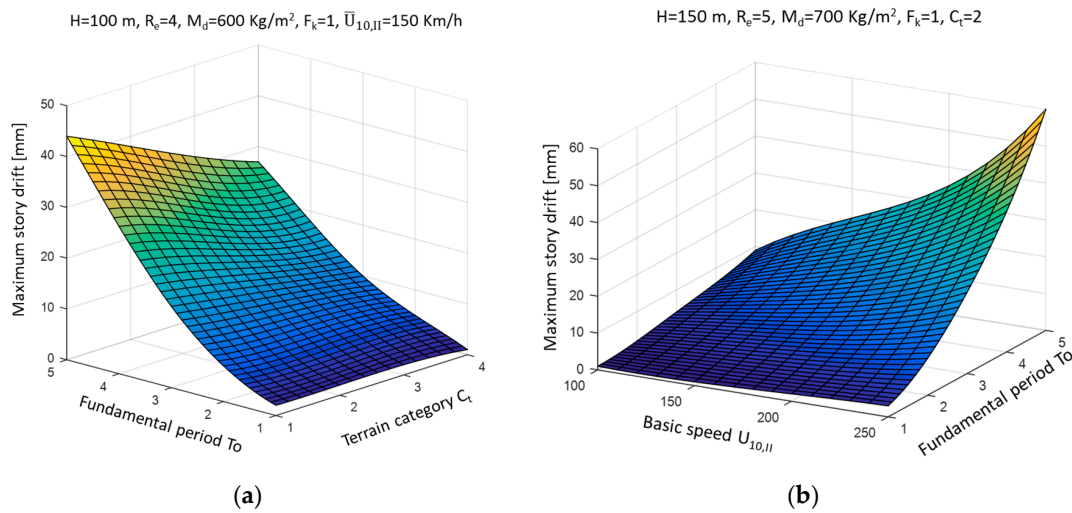


Figure 7. Maximum story drift for cases with: (a) $H = 100\text{ m}$, $R_e = 4$, $M_d = 600\text{ Kg/m}^2$, $F_k = 1$, $\bar{U}_{10,II} = 150\text{ Km/h}$; (b) $H = 150\text{ m}$, $R_e = 5$, $M_d = 700\text{ Kg/m}^2$, $F_k = 1$, $C_t = 2$.

3.2. Results of Network Trainings

The maximum story drifts obtained for each case study are used as the dataset for the network training process by the LM method. For the network architecture, the parameters of height, slenderness ratio, fundamental period, distributed mass, stiffness ratio, terrain categories and basic wind speed represent seven input values needed to estimate an output, which tends to predict the maximum story drifts. With the aim of evaluating the efficiency of the network against new data (not analyzed), a small percentage (15%) taken randomly of the dataset is excluded in the training process. Table 3 shows the mean square error (MSE) generated for the ANNs with different number of neurons in the hidden layer. It is observed that the error is reduced when the number of neurons in the hidden layer increases. A larger number of hidden neurons allows for modeling any variation in the data; however, the decrees of the error are not significant from 20 hidden neurons.

The cross-validation is a technique to evaluate the independence of the results of a prediction model with relation to the training and testing dataset selected. Table 4 shows the cross-validation of the ANN with 20 hidden neurons using, in each iteration, different training and evaluation dataset taken randomly that represent the 85 and 15 percent, respectively. Small variations of MSE between each dataset and iteration are observed. Therefore, the dataset selected generates in an independent way results very close to each other.

Table 3. Results of network trainings with different number of neurons in the hidden layer.

Hidden Neurons	Training Data	Evaluation Data	Training Iterations	MSE Training Data	MSE Evaluation Data
1	21,760	3840	1000	101.72	108.96
2	21,760	3840	1000	65.48	62.96
5	21,760	3840	1000	23.92	29.22
10	21,760	3840	1000	8.89	9.06
12	21,760	3840	1000	5.73	5.88
20	21,760	3840	1000	2.59	2.91
30	21,760	3840	1000	2.05	2.21

Figure 8 shows the relation between the target values and the output values of the ANNs with 1, 10 and 20 hidden neurons considering training and evaluation data. It is observed that the increase in the number of neurons in the hidden layer helps to improve the linear relation close to 99%.

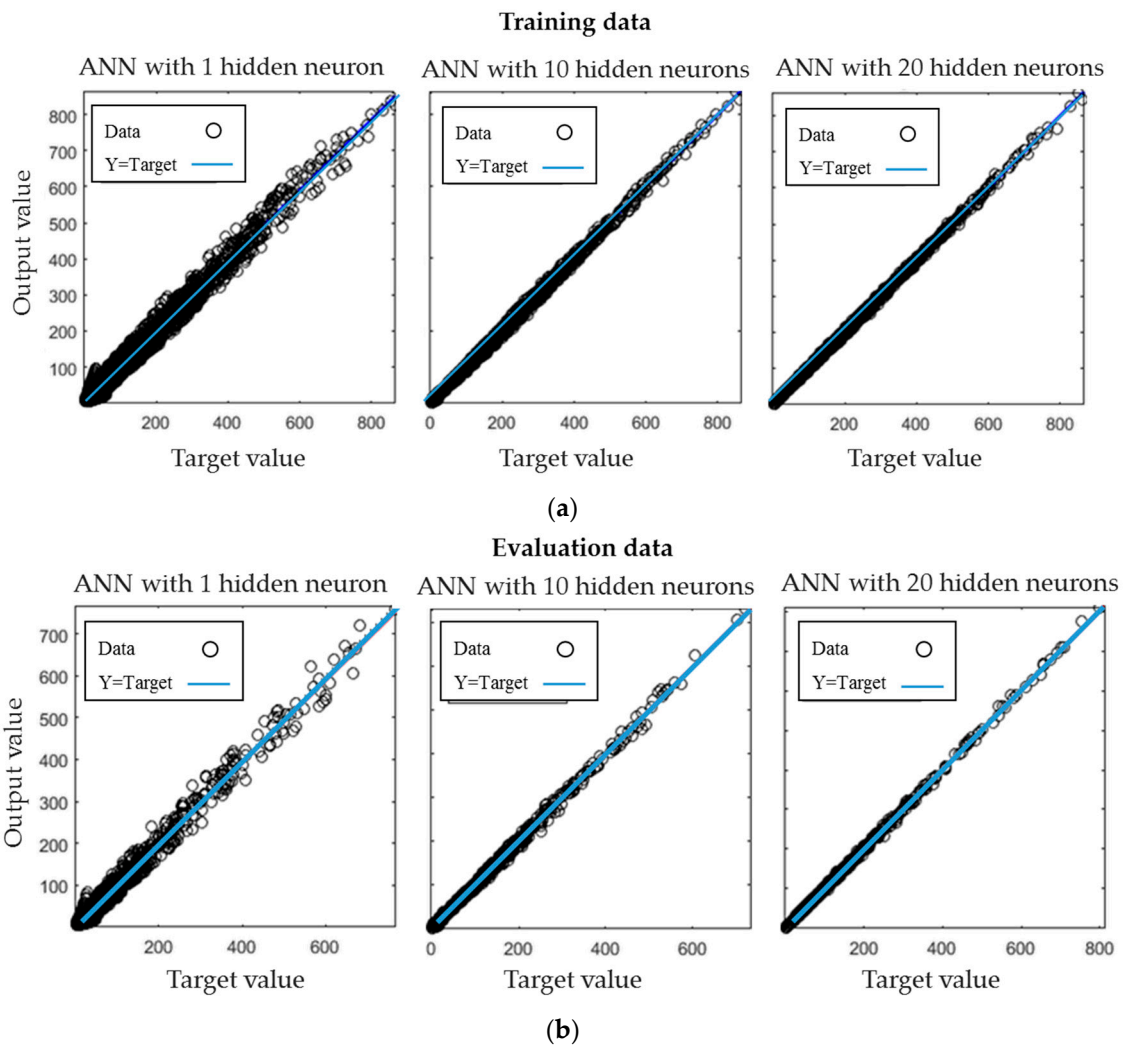


Figure 8. Relation between the target values and output values of the ANN with 1, 10 and 20 hidden neurons. (a) Using the training data; (b) using evaluation data.

Table 4. Cross-validation of the ANN with 20 hidden neurons.

Validation Iterations	MSE Training Data	MSE Evaluation Data
Iteration 1	3.51	3.71
Iteration 2	2.86	2.94
Iteration 3	2.24	3.01
Iteration 4	2.58	2.82
Iteration 5	3.99	3.65
Iteration 6	3.42	3.58
Iteration 7	2.79	2.74
Iteration 8	3.21	3.31
Iteration 9	2.49	2.58
Iteration 10	2.73	3.00

Table 5 shows the memory and computer time required in the tasks of computational processing for the case $H = 150$ m, $R_e = 5$, $M_d = 500$ Kg/m², $F_k = 1$, $T_0 = 3$ s, $C_t = 2$, $\bar{U}_{10,II} = 150$ Km/h. A greater computer time in the training network is observed than the sum of all phases of the standard procedure; however, once the network has been trained, it is prepared to respond to many possible cases of immediately and using little memory.

Table 5. Memory and computer time in order to process the tasks of the case: $H = 150$ m, $R_e = 5$, $M_d = 500$ Kg/m², $F_k = 1$, $T_0 = 3$ s, $C_t = 2$, $\bar{U}_{10,II} = 150$ Km/h.

Procedure	Phase	Memory (MB)	Computer Time (s)
Standard	Wind Simulation	6850	386.41
	Calculate stiffness matrix	13	5.93
	Dynamic analyse	29	24.07
ANN with 20 hidden neurons	Training	424	934.15
	Execute	0.01	0.001

4. Conclusions

The maximum story drift of 1600 shear-buildings discretized as MDOF structures were obtained. The Shinozuka and Newmark methods were employed to simulate the synthetic wind records and to obtain the dynamic response of the structures, respectively. The wind loads necessary to compute the dynamic response were developed from wind synthetic records considering 16 turbulent wind conditions. The Artificial Neural Networks (ANNs) were used to generate a computational model capable of estimating the maximum story drift of 25,600 case studies. The Levenberg–Marquardt (LM) method is applied in the process of network training. The parameters of height, slenderness ratio, fundamental period, distributed mass, stiffness ratio, terrain categories and basic wind speed represented the input values in the ANNs. Seven network architectures with 1 to 30 neurons in one hidden layer were studied to evaluate its predictive efficiency. It is observed that errors decrease when the number of neurons in the hidden layer increases. Nevertheless, the decreases of the errors are not significant from 20 hidden neurons. The cross-validation with 10 iterations suggested independence acceptable between the dataset selected for training and the results. The study of computer times required in the tasks of computational processing indicated a major processing time in training networks compared to a standard procedure for a specific case; nevertheless, once the network has been trained, it is prepared to respond to many possible cases of immediately and using little computational memory. Finally, it was demonstrated that with the use of ANNs and the LM method, it is possible to obtain a computational model capable of predicting with a good relationship the maximum story drift of MDOF structures.

Acknowledgments: The scholarship for PhD studies given by El Consejo Nacional de Ciencia y Tecnología to the PhD students and the support under grant CB-01-167419 to the second author is appreciated. Financial support also was received from the Universidad Autónoma de Sinaloa under grant PROFAPI 2015.

Author Contributions: Omar Payán-Serrano, Edén Bojórquez and Juan Bojórquez designed the Artificial Neural Networks; Robespierre Chávez, Alfredo Reyes and Manuel Barraza defined the models of wind and structures. Arturo López, Héctor Rodríguez and Edgar Corona analyzed the models under dynamics loads. Omar Payán-Serrano and Edén Bojórquez wrote the paper.

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

References

1. Veers, P.S. *Three-Dimensional Wind Simulation*; Sandia National Labs: Albuquerque, NM, USA, 1988.
2. Chay, M.T.; Albermani, F.; Wilson, R. Numerical and analytical simulation of downburst wind loads. *Eng. Struct.* **2006**, *28*, 240–254. [[CrossRef](#)]
3. Bojórquez, E.; Payán-Serrano, O.; Reyes-Salazar, A.; Pozos, A. Comparison of spectral density models to simulate wind records. *KSCE J. Civ. Eng.* **2016**, *1*, 1–8. [[CrossRef](#)]
4. Davenport, A.G. Gust loading factors. *J. Struct. Div.* **1967**, *93*, 11–34. [[CrossRef](#)]
5. Davenport, A.G. The representation of the dynamic effects of turbulent wind by equivalent static wind loads. *Proceedings of International Symposium on Structural Steel*, Chicago, IL, USA, 1985; pp. 1–13.
6. Chen, X.; Kareem, A. Equivalent static wind loads on buildings: New model. *J. Struct. Eng.* **2004**, *130*, 1425–1435. [[CrossRef](#)]

7. Repetto, M.P.; Solari, G. Equivalent static wind actions on vertical structures. *J. Wind Eng. Ind. Aerodyn.* **2004**, *92*, 335–357. [[CrossRef](#)]
8. Hecht-Nielsen, R. Neurocomputer applications. *Neural Comput.* **1989**, *41*, 445–453. [[CrossRef](#)]
9. Conte, J.P.; Durrani, A.J. Seismic Response modeling of multi-story buildings using neural networks. *J. Intell. Mater. Syst. Struct.* **1994**, *5*, 392–402. [[CrossRef](#)]
10. Chakraverty, S.; Gupta, P.; Sharma, S. Neural network-based simulation for response identification of two-storey shear building subject to earthquake motion. *Neural Comput. Appl.* **2010**, *19*, 367–375. [[CrossRef](#)]
11. Bojórquez, J.; Ruiz, S.; Bojórquez, E.; Reyes-Salazar, A. Probabilistic seismic response transformation factors between SDOF and MDOF systems using artificial neural networks. *J. Vibroeng.* **2016**, *18*, 2248–2262. [[CrossRef](#)]
12. Roy, D.; Dash, M.K. Explorations of a family of stochastic Newmark methods in engineering dynamics. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 4758–4796. [[CrossRef](#)]
13. Krenk, S. Energy conservation in Newmark based time integration algorithms. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 6110–6124. [[CrossRef](#)]
14. Bourihane, O.; Braikat, B. Dynamic analysis of a thin-walled beam with open cross section subjected to dynamic loads using a high-order implicit algorithm. *Eng. Struct.* **2016**, *120*, 133–146. [[CrossRef](#)]
15. Shinozuka, M.; Jam, C.M. Digital simulation of random processes and its applications. *J. Sound Vib.* **1972**, *25*, 111–128. [[CrossRef](#)]
16. Rossi, R.; Lazzari, M.; Vitaliani, R. Wind field simulation for structural engineering purposes. *Int. J. Num. Method Eng.* **2004**, *61*, 738–763. [[CrossRef](#)]
17. Levenberg, K. A method for the solution of certain problems in least squares. *Q. Appl. Math.* **1944**, *5*, 164–168. [[CrossRef](#)]
18. Suratgar, A.A.; Tavakoli, M.B.; Hoseinabadi, A. Modified Levenberg–Marquardt method for neural networks training. *World Acad. Sci. Eng. Technol.* **2005**, *6*, 46–48.
19. Davenport, A.G. The spectrum of horizontal gustiness near the ground in high winds. *Q. J. R. Meteorol. Soc.* **1961**, *87*, 194–211. [[CrossRef](#)]
20. Mackey, S. Gust Factors. In *Proceedings of the Seminar USA-Japan Research Seminar: Wind Loads on Structures*; University of Hawaii: Honolulu, HI, USA, 1970; pp. 191–202.
21. Counihan, J. Adiabatic atmospheric boundary layers: A review and analysis of data from the period from 1880–1972. *Atmos. Environ.* **1967**, *9*, 871–905. [[CrossRef](#)]
22. Hirt, M.A. Eurocode 1. Basis of design and action on structures Part 1–4. *Schweizer Ingenieur Architekt* **1993**, *16–17*, 273–275.
23. Von Karman, T. Progress in the statistical theory of turbulence. *Proc. Natl. Acad. Sci. USA* **1948**, *34*, 530–539. [[CrossRef](#)] [[PubMed](#)]
24. Harris, R.I. *On the Spectrum and Auto-correlation Function of Gustiness in High Winds*; Electrical Research Association: Leatherhead, UK, 1968.
25. Lungu, D.; van Gelder, P. Characteristics of wind turbulence with applications to wind codes. In *Proceedings of the 2nd European & African Conference on Wind Engineering*, Genova, Italy, 22–26 June 1997; pp. 1271–1277.
26. Xu, Y.L. *Wind Effects on Cable-Supported Bridges*; John Wiley and Sons: New York, NY, USA, 2013.
27. McKenna, F.; Fenves, G.L.; Scott, M. *Open System for Earthquake Engineering Simulation (OpenSees)*; University of California: Berkeley, CA, USA, 2002.
28. Carr, A.J. *RUAUMOKO-2D Inelastic Time-History Analysis of Two Dimensional Framed Structures*; Department of Civil Engineering, University of Canterbury: Christchurch, New Zealand, 2004.
29. Computer and Structure, Inc. *SAP2000 Documentation, Release 15.1.0*; University Ave: Berkeley, CA, USA, 2012.
30. Chopra, A. *Dynamics of Structures: Theory and Applications to Earthquake Engineering*; Prentice Hall: Englewood Cliffs, NJ, USA, 2014. [[CrossRef](#)]
31. Holmes, J.D. *Wind Loading of Structures*; CRC Press: Boca Raton, FL, USA, 2015.

