Functional Link Neural Network Learning for Response Prediction of Tall Shear Buildings With Respect to Earthquake Data

Deepti Moyi Sahoo and Snehashish Chakraverty

***Abstract*—This paper proposes the application of functional link neural networks (FLNNs) for structural response prediction of tall buildings due to seismic loads. The ground acceleration data are taken as input, and structural responses of different floors of multistorey shear buildings are considered as output. It is worth mentioning that handling of large earthquake data has become a great challenge in the design of tall structures viz., that of shear buildings. As such, here, a functional expansion block in FLNN has been used along with efficient Chebyshev and Legendre polynomials. Training is done with one earthquake data set, and testing is done with different intensities of other earthquake data sets; and it is seen that FLNN can very well predict the structural response of different floors of multistorey shear buildings subject to earthquake data. Results of the FLNN are compared with a multilayer neural network (MNN), and it is found that the FLNN gives better accuracy and takes less computation time compared to MNN, which shows the computational efficiency of FLNN over MNN. Numerical examples of two-, five-, and ten-storey buildings are considered, and corresponding results are presented in the form of tables and plots.**

***Index Terms*—Earthquake, functional link neural network (FLNN), multilayer neural network (MNN), multistorey shear buildings, response, structure.**

I. Introduction

One of the most frightening and terrible natural calamities after tsunami are the earthquakes which cause disaster to various infrastructure like monuments, bridges, dams, and multistorey buildings such as hospitals, schools, and offices. The strong ground motion due to the seismic wave disintegrates these structures, thereby causing loss to humans and society. Dynamic response of a structure subject to strong and complex earthquake ground motion data should be investigated to avoid accidents.

There exist various methods to study the characteristics of the dynamic response of a structure. Various authors have implemented different numerical methods for studying linear and nonlinear dynamic analysis. But use of these methods may be time consuming or problem dependent. In order to predict structural responses such as acceleration, velocity, and displacement, artificial neural networks (ANNs) may be an appropriate and fast learning method.

ANNs are a powerful technique used in various fields of engineering including civil engineering. ANNs are generally applied in prediction problem, pattern classification, speech recognition, data mining, and various other problems due to its high learning and mimicking characteristics. As such, some these problems may be found in [1]–[5]. Recent studies have demonstrated that ANNs are is also being applied to assess damage in structures. As regards, some work may be mentioned, e.g., Pandey and Barai [6], Zhao *et al.* [7], and Masri *et al.* [8]. ANN models are used for identifying dynamic characteristics of structures and also in predicting structural responses from seismic data [9]–[12]. Related publication may be found also in [13]–[16].

It reveals from above literature review that ANNs have been applied both for response prediction and damage detection. Previous authors have used traditional ANNs to solve these problems where they considered the data to be in crisp form. But always it is not possible to get the data in exact or crisp form. Mitra and Pal [17] used Kohonen's model of self-organization ANN in fuzzy classification problems. Very recently, Sahoo *et al.* [18] proposed identification methodologies for multistorey shear buildings using an interval neural network which can estimate the structural parameters. Chakraverty and Sahoo [19] also used a fuzzy neural network model to solve system identification problem of multistorey shear buildings using fuzzified response data.

A functional link neural network (FLNN) is a single-layer neural network where the hidden layer is eliminated by a functional expansion block for enhancement of the input vectors [20]. FLNNs are highly effectual, computationally more efficient, and faster learning than multilayer neural networks (MNNs). Few research works using FLANN in different fields have been discussed. Patra *et al.* [21] and Patra and Kot [22] solved dynamic nonlinear system identification problems using Chebyshev a functional neural network (ChNN). Purwar *et al.* [23] used a ChNN model for system identification of unknown dynamic nonlinear discrete time systems. Dehuri and Cho [24] developed a new learning scheme for Chebyshev FLNN. Xiuchun *et al.* [25] constructed a ChNN to obtain the weight-direct-determination method.

Manuscript received September 22, 2016; revised November 29, 2016; accepted April 20, 2017. This paper was recommended by Associate Editor F. Sun. *(Corresponding author: Snehashish Chakraverty.)*

The authors are with the Department of Mathematics, National Institute of Technology Rourkela, Rourkela 769008, India (e-mail: deeptisahoo.sahoo046@gmail.com; sne_chak@yahoo.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2700334

Patra *et al.* [26] used a Legendre neural network (LeNN) for equalization of nonlinear communication channels for wireless communication systems. Patra and Bornand [27] used LeNN for identification of nonlinear dynamic systems. The idea of designing a noise prediction model for opencast mining machineries using functional link ANN systems was introduced by Nanda and Tripathy [28]. Hassim and Ghazali [29] evaluated the FLNN using an artificial bee colony model for the task of pattern classification of two-class classification problems. A new methodology using LeNN has been investigated by Ali and Haweel [30] to enhance nonlinear multi-input multioutput signal processing. Works related to FLNN may also be found in [31]–[45].

Here, structural response of multistorey shear buildings subject to earthquake ground acceleration data has been estimated using an FLNN model. The polynomials used in the expansion block for enhancement of the input vectors are the Chebyshev and Legendre polynomials. It may be noted that traditional ANNs require more time to handle the complex earthquake data. The novelty of this paper is the use of FLNN for the first time to the best of our knowledge to handle the large and complex earthquake data set. For this paper, the Chamoli earthquake with magnitude 6.8 Mw and depth of about 21 km and Uttarkashi earthquake with magnitude 6.8 Mw and depth of about 10 km recorded at Barkot, Uttarakhand, India, in the northeast (NE) direction are considered. The Chamoli earthquake occurred on March 29, 1999 with peak acceleration value 19.58 cm/s^2, and the Uttarkashi earthquake occurred on October 20, 1991 with maximum ground acceleration of 93.1 cm/s^2. The ground acceleration data are considered as input which is expanded by the Chebyshev and Legendre polynomials, and the output is taken as the structural responses. First, the training is done with Chebyshev polynomials for different order polynomials with random weights, and after the training is complete the converged weights are stored. Then, the LeNN is trained with the stored converged weights of ChNN and also with random weights. It is seen that LeNN trained with stored converged weights shows better accuracy and takes less computation time than with random weights. Training with MNN is also done for different storey, and results are compared with the FLNN. Comparison of the results shows that the FLNN takes less computation time than the MNN. After training the network with the Chamoli earthquake data, the converged weights are stored both for FLNN and MNN. The stored converged weights are used in testing to predict storey responses for the different intensity of Uttarkashi earthquakes.

II. Modeling for Response Analysis for a Multidegree-of-Freedom System

When a multistorey building is subjected to base excitation, then the governing equation of motion is written as

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = -[M]\{\chi\}\ddot{a} \tag{1}$$

where $[M]$ is a $n \times n$ mass matrix, $[K]$ is a $n \times n$ stiffness matrix of the structure, and $[C]$ represents $n \times n$ damping matrix and $\{\chi\}$ is the influence co-efficient vector. Here $\{x\}$ is the displacement relative to the ground, $\{\ddot{x}\}$ is the response acceleration, $\{\dot{x}\}$ is the response velocity, and $\ddot{a}$ is the earthquake ground acceleration. The global mass, stiffness and damping matrices are denoted as $[M]$, $[K]$, and $[C]$, respectively, and are given as

$$[M] = \begin{bmatrix} m_1 & 0 & \cdots & \cdots & 0 \\ 0 & m_2 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & 0 & m_{n-1} & 0 \\ 0 & \cdots & \cdots & 0 & m_n \end{bmatrix}$$

$$[K] = \begin{bmatrix} k_1+k_2 & -k_2 & 0 & \cdots & 0 \\ -k_2 & k_2+k_3 & -k_3 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & -k_{n-1} & k_{n-1}+k_n & -k_n \\ 0 & \cdots & \cdots & -k_n & k_n \end{bmatrix}$$

$$[C] = \begin{bmatrix} c_1+c_2 & -c_2 & 0 & \ldots & 0 \\ -c_2 & c_2+c_3 & -c_3 & \ldots & 0 \\ \ldots & \cdots & \cdots & \cdots & \ldots \\ 0 & \ldots & -c_{n-1} & c_{n-1}+c_n & -c_n \\ 0 & \cdots & \cdots & -c_n & c_n \end{bmatrix}.$$

Equation (1) is a set of n-coupled ordinary differential equations. Modal analysis technique can be used to solve (1) if the system is linear with proportional damping. The modal analysis technique becomes more efficient for earthquake response analysis. Hence, (1) can be reduced to n-modal equations as in [12]

$$\ddot{x}_r + 2\xi_r\omega_r\dot{x}_r + \omega_r^2 x_r = -\ddot{a}\gamma_r \quad r = 1, 2, \ldots, n \tag{2}$$

where n $(\leq N)$ is the number of significant modes, ξ_r is the damping ratio and the modal coordinate x_r is related to the displacement of the ith mass as

$$v_i = \sum_{r=1}^{n} \varphi_{ir} x_r \tag{3}$$

where φ_{ir} is the ith component of the rth mode-shape vector and γ_r is the modal participation factor. Equation (3) represents the equation of motion of n SDOF system and the response is obtained from Duhamel integral. The Duhamel integral is written as [46]

$$x_r = -\frac{\gamma_r}{\omega_{Dr}} \int_0^t \ddot{a}(\tau) \exp[-\xi_r\omega_r(t-\tau)] \sin[\omega_r(t-\tau)] d\tau. \tag{4}$$

Here $\omega_{Di} = \omega_i\sqrt{1-\xi_i^2}$, where ω_{Di}, ω_r, and ξ_r are damped frequency, free vibration frequency, and damping ratio, respectively. The time history response of the ith mass is then determined from (2) as

$$v_i(t) = \phi_{i1}\, x_1\,(t) + \phi_{i2}\, x_2\,(t) + \cdots \tag{5}$$

The floor responses of multistorey shear structures, viz. displacement, is obtained from (5). The FLNN architecture is constructed with Chebyshev and Legendre polynomials, considering the ground acceleration data as input and the storey response as output, obtained from the above solution for each time step.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

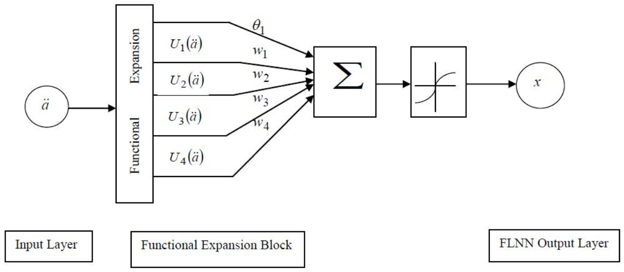


Fig. 1. Single layer FLNN.

III. Functional Link Neural Network

FLNN is a single-layer neural network in which the hidden layer is removed and the input vectors are expanded to higher dimensional vectors by some orthogonal polynomials. The FLNN model consists of mainly two parts, one is functional expansion part and the other is learning part. Due to the absence of hidden layer, FLNN is computationally more efficient and faster than MNN. Let us consider i input vector denoted as $Y = \{y_1, y_2, \ldots, y_i\}^T$. Thus, the enhanced input vector can be expressed as $Y^T = U(Y)$, where $U = [u_1(Y), u_2(Y), \ldots, u_N(Y)]$. Here $\{U\}_{k=1}^{N}$ are set of suitable functions. These functions may be considered as set of orthogonal polynomials, viz., Chebyshev and Legendre polynomials. Structure of single layer FLNN model with single input–output is considered here and their learning algorithms have been discussed below. The single input–output FLNN architecture is shown in Fig. 1.

A. Learning Algorithm of Functional Link Neural Network

In FLNN, the weights are updated to minimize a given cost function. Feed forward and error back propagation algorithm is used for learning. Error back propagation algorithm has been used to update the weights of FLNN. Inputs $Y_i = \ddot{a}_i$ are the ground acceleration and outputs $O_k = x_k$ are the structural responses of each floor of a multistorey building. The linear sum S_k can be calculated as

$$S_k = \sum_{k=1}^{N} w_k U_k(Y) + \theta_k \quad k = i = \text{number of input nodes} \tag{6}$$

where w_k are the weights, θ_k are the bias, and $U_k(Y)$ are the expanded input vectors. These $U_k(Y)$ are considered here as Chebyshev and Legendre polynomials.The net output is given as

$$O_k = f(S_k).$$

As the earthquake data are both positive and negative, therefore bipolar sigmoidal function has been used as the activation function. The bipolar activation function is defined as $f = [(1 - \exp(S_k))/(1 + \exp(S_k))]$.

Cost function used for minimization of error is defined as

$$E = \frac{1}{2}[d_k - O_k]^2 = \frac{1}{2}e_k^2 \quad k = i \tag{7}$$

where d_k is the desired output, O_k is the target output, and e_K is the error value. The error value is computed to obtain the desired accuracy. Weights are updated as follows:

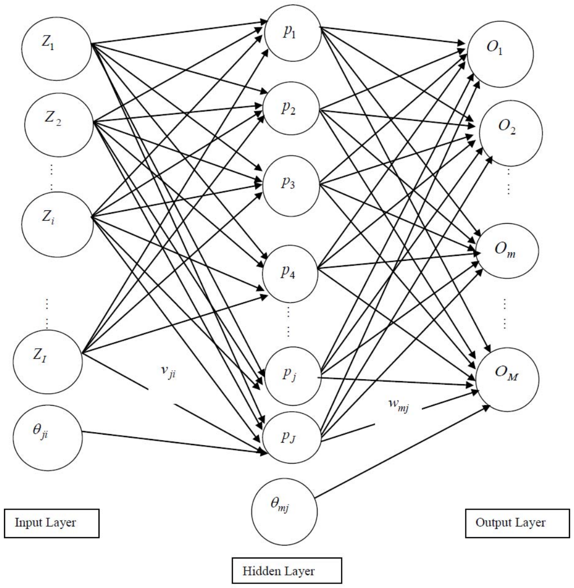


Fig. 2. MNN.

$$w_k(\text{New}) = w_k(\text{Old}) + \Delta w_k \tag{8}$$

where change in weights are calculated as

$$\Delta w_k = \left[-\eta \frac{\partial E}{\partial w_k}\right] = \left[-\eta(d_k - O_k)\left(1 - O_k^2\right) U_k(Y)\right] \tag{9}$$

where η is the learning parameter. The same procedure to update the bias θ_k.

B. Structure of Chebyshev Neural Network

The structure of the ChNN comprises of one input node, a functional expansion block consisting of Chebyshev polynomials and one output node. Each input vector is expanded to several terms of Chebyshev orthogonal polynomials so that they may be viewed as a new input vector in the functional expansion part. Chebyshev polynomial of kth order is denoted by $T_k(y)$, where $-1 < y < 1$. First five Chebyshev polynomials may be written as [47], [48]

$$\begin{aligned} T_1(y) &= 1 \\ T_2(y) &= y \\ T_3(y) &= 2y^2 - 1 \\ T_4(y) &= 4y^3 - 3y \\ T_5(y) &= 8y^4 - 8y^2 + 1. \end{aligned}$$

Higher order Chebyshev polynomials may be generated by the well-known recursive formula

$$T_{k+1}(y) = 2xT_k(y) - T_{k-1}(y). \tag{10}$$

C. Structure of Legendre Neural Network

In LeNN, the suitable functions are taken as Legendre orthogonal polynomials in the expansion block. In functional expansion, each input is expanded to several terms using Legendre polynomials to have a new input vector. Here we

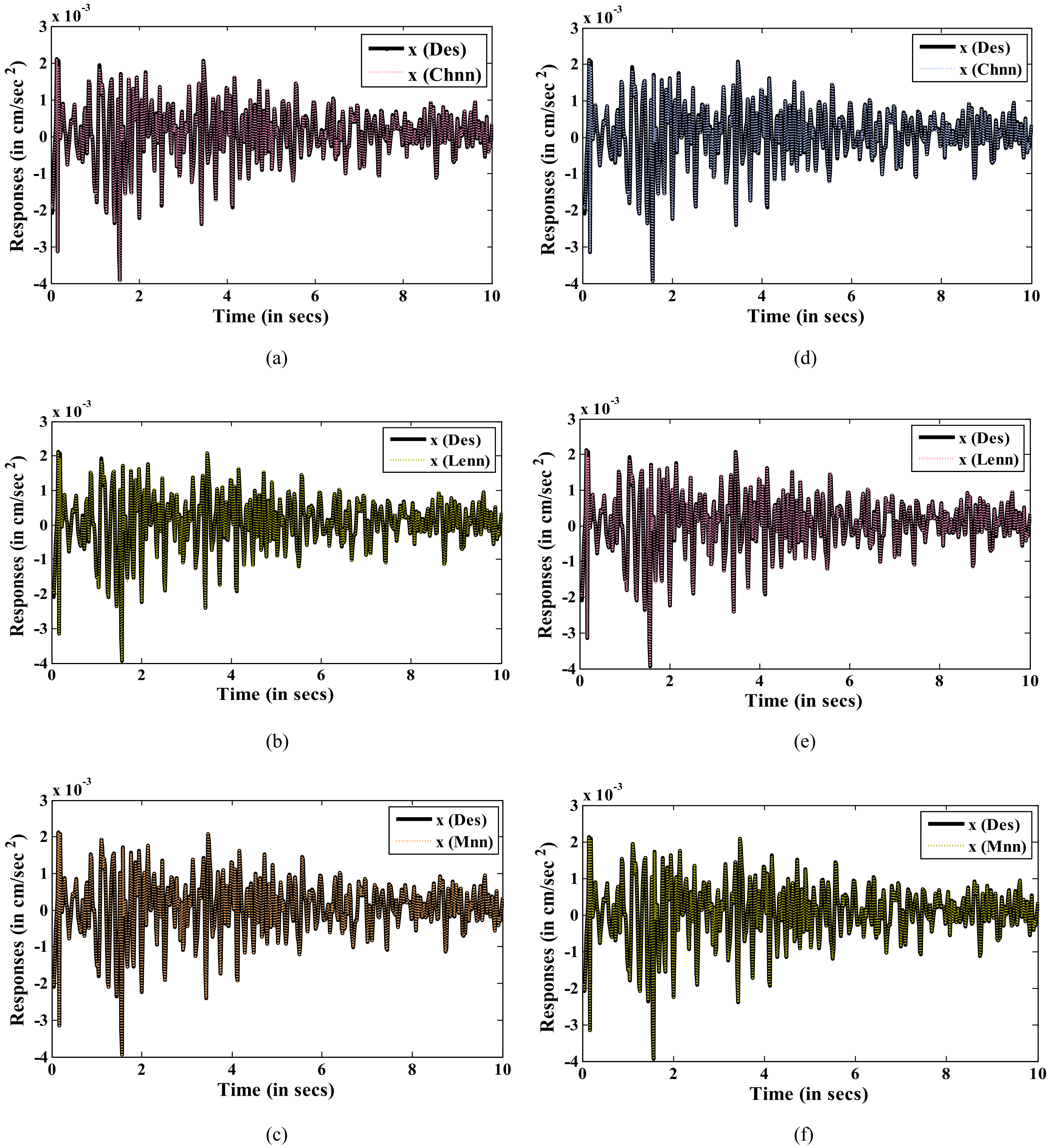


Fig. 3. Comparison between desired and (a) ChNN results for fourth floor, (b) LeNN results for fourth floor, (c) MNN results for fourth floor, (d) ChNN results for fifth floor, (e) LeNN results for fifth floor, and (f) MNN results for fifth floor.

have considered a single layer single input–output LeNN model to find structural response. The Legendre polynomial of kth order is denoted by $L_k(y)$, where $-1 < y < 1$. As such first five Legendre polynomials may be written as [47], [48]

$$L_1(y) = 1$$
$$L_2(y) = y$$
$$L_3(y) = \frac{1}{2}\left(3y^2 - 1\right)$$
$$L_4(y) = \frac{1}{2}\left(5y^3 - 3y\right)$$
$$L_5 y = \frac{1}{8}\left(35y^4 - 30y^2 + 3\right).$$

Legendre polynomials of higher order may be obtained by the well-known recursive rule

$$L_{k+1}(y) = \frac{1}{k+1}\left[(2k+1)\, y\, L_k\,(y) - k\, L_{k-1}\,(y)\right]. \quad (11)$$

TABLE I
COMPARISON BETWEEN ChNN (WITH DIFFERENT ORDER POLYNOMIAL) AND MNN FOR (a) FIRST STOREY AND (b) SECOND STOREY. COMPARISON BETWEEN DESIRED AND LeNN WITH STORED CONVERGED WEIGHTS AND RANDOM WEIGHTS FOR (c) FIRST STOREY AND (d) SECOND STOREY

(a)

Data Set	Desired (cm/sec^2)	$T_2(y)$ (cm/sec^2)	$T_3(y)$ (cm/sec^2)	$T_4(y)$ (cm/sec^2)	$T_5(y)$ (cm/sec^2)	MNN (cm/sec^2)
1	-0.00044194	-0.00049865	-0.00052216	-0.00044154	-0.00044113	-0.00042112
2	-0.0018653	-0.0018279	-0.0017278	-0.0018951	-0.0018643	-0.0017858
3	-0.0013531	-0.0014483	-0.0013881	-0.0013437	-0.0013564	-0.0013726
4	-0.00055874	-0.00063637	-0.00065088	-0.00055988	-0.00055866	-0.00054473
5	-0.00021603	-0.00022461	-0.00026186	-0.00021852	-0.00021655	-0.00019099
6	0.00019361	0.00028091	0.0002343	0.00019321	0.00019369	0.00018613
7	0.0017298	0.0017846	0.0018271	0.0017799	0.0017229	0.0017509
8	0.0026065	0.0022142	0.0022834	0.0026496	0.0026015	0.0025816
9	0.0016876	0.0017565	0.0017965	0.0016834	0.0016895	0.0016834
10	0.00030355	0.00041458	0.00036915	0.00030257	0.00030388	0.00027847

(b)

Data Set	Desired (cm/sec^2)	$T_2(y)$ (cm/sec^2)	$T_3(y)$ (cm/sec^2)	$T_4(y)$ (cm/sec^2)	$T_5(y)$ (cm/sec^2)	MNN (cm/sec^2)
1	-0.00048403	-0.00054613	-0.00057188	-0.00048073	-0.00048486	-0.00046126
2	-0.002043	-0.0020019	-0.0018923	-0.002055	-0.002046	-0.0019557
3	-0.001482	-0.0015862	-0.0015203	-0.001417	-0.001494	-0.0015034
4	-0.00061195	-0.00069697	-0.00071286	-0.00061938	-0.00061195	-0.00059664
5	-0.00023661	-0.000246	-0.0002868	-0.00023838	-0.00023698	-0.00020914
6	0.00021205	0.00030766	0.00025662	0.00021278	0.00021288	0.00020392
7	0.0018945	0.0019546	0.0020011	0.0018493	0.0018998	0.0019178
8	0.0028547	0.002425	0.0025008	0.0028829	0.0028516	0.0028275
9	0.0018483	0.0019238	0.0019676	0.0018991	0.0018413	0.0018436
10	0.00033246	0.00045406	0.0004043	0.00033662	0.00033244	0.00030502

(c)

Data Set	Desired (cm/sec^2)	With converged weights				With Random weights			
		$L_2(y)$ (cm/sec^2)	$L_3(y)$ (cm/sec^2)	$L_4(y)$ (cm/sec^2)	$L_5(y)$ (cm/sec^2)	$L_2(y)$ (cm/sec^2)	$L_3(y)$ (cm/sec^2)	$L_4(y)$ (cm/sec^2)	$L_5(y)$ (cm/sec^2)
1	-0.00044194	-0.00044865	-0.00044827	-0.00044145	-0.00044198	-0.00049867	-0.00052218	-0.00044156	-0.00044117
2	-0.0018653	-0.0018641	-0.0018576	-0.0018641	-0.0018651	-0.0018279	-0.0017279	-0.0018955	-0.0018648
3	-0.0013531	-0.0013852	-0.0013279	-0.0013497	-0.0013533	-0.0014485	-0.0013883	-0.0013439	-0.0013566
4	-0.00055874	-0.00055988	-0.00055831	-0.00055805	-0.00055877	-0.00063639	-0.00065085	-0.00055984	-0.00055869
5	-0.00021603	-0.00021705	-0.00021924	-0.00021655	-0.00021607	-0.00022463	-0.00026188	-0.00021855	-0.00021654
6	0.00019361	0.00019332	0.00019745	0.00019393	0.00019369	0.00028093	0.0002345	0.00019326	0.00019366
7	0.0017298	0.0017253	0.0017289	0.0017223	0.0017292	0.0017848	0.0018273	0.0017797	0.0017227
8	0.0026065	0.0026387	0.0026683	0.0026002	0.0026064	0.0022144	0.0022835	0.0026498	0.0026017
9	0.0016876	0.0016672	0.0016871	0.0016863	0.0016878	0.0017567	0.0017967	0.0016838	0.0016893
10	0.00030355	0.00030086	0.00030769	0.00030326	0.00030355	0.00041459	0.00036917	0.00030259	0.00030385

(d)

Data Set	Desired (cm/sec^2)	With converged weights				With Random weights			
		$L_2(y)$ (cm/sec^2)	$L_3(y)$ (cm/sec^2)	$L_4(y)$ (cm/sec^2)	$L_5(y)$ (cm/sec^2)	$L_2(y)$ (cm/sec^2)	$L_3(y)$ (cm/sec^2)	$L_4(y)$ (cm/sec^2)	$L_5(y)$ (cm/sec^2)
1	-0.00048403	-0.00048613	-0.00048572	-0.00048463	-0.00048401	-0.00054615	-0.00057185	-0.00048075	-0.00048488
2	-0.002043	-0.002019	-0.002054	-0.002056	-0.002048	-0.0020016	-0.0018925	-0.002057	-0.002044
3	-0.001482	-0.001486	-0.001484	-0.001488	-0.001484	-0.0015865	-0.0015205	-0.001419	-0.001498
4	-0.00061195	-0.00061697	-0.00061048	-0.00061157	-0.00061197	-0.00069699	-0.00071289	-0.00061934	-0.00061197
5	-0.00023661	-0.0002366	-0.00023697	-0.00023664	-0.00023665	-0.000247	-0.0002865	-0.00023835	-0.00023695
6	0.00021205	0.00021766	0.00021211	0.00021253	0.00021203	0.00030768	0.00025665	0.00021276	0.00021285
7	0.0018945	0.0018546	0.0018702	0.0018901	0.0018943	0.0019548	0.0020014	0.0018497	0.0018996
8	0.0028547	0.002825	0.0028843	0.0028545	0.0028548	0.002427	0.0025009	0.0028826	0.0028518
9	0.0018483	0.0018238	0.0018362	0.0018498	0.0018482	0.0019239	0.0019677	0.0018993	0.0018415
10	0.00033246	0.00033406	0.00033175	0.00033273	0.00033248	0.00045408	0.0004048	0.00033665	0.00033245

IV. Multilayer Neural Network

An MNN is a traditional or classical neural network which has three layers, viz., input, hidden, and the output layers. The MNN is trained with an error back propagation algorithm. The typical MNN architecture related to this paper is shown in Fig. 2 [11], [12], [49]. The inputs are the ground acceleration and the outputs are the storey response obtained from solution of (4) for each time step.

A. Learning Algorithm of Multilayer Neural Network

For MNN inputs $Z_I = \ddot{a}_i$ are the ground acceleration and outputs $O_M = x_m$ are the structural response. The weights between input and hidden layers are denoted by v_{ji} and the weights between hidden and output layers are denoted by w_{mj}. θ_{ji} is the bias between input and hidden layer and θ_{mj} is the bias between hidden and output layer. The error value is calculated as

$$E = \frac{1}{2}(d_m - O_m)^2 \tag{12}$$

where d_m is the desired output and O_m is the MNN output.

The error signal terms from output to hidden is written as

$$\delta_{O_m} = 0.5 * (d_m - O_m)\left(1 - O_m^2\right),\ m = 1, 2, \ldots, M. \tag{13}$$

Similarly, the error signal term from hidden to input is written as

$$\delta_{P_j} = 0.5 * \left(1 - P_j^2\right)\sum_{m=1}^{M}\delta_{O_m} w_{mj},\quad j = 1, 2, \ldots, J. \tag{14}$$

The output layer weights are adjusted as

$$w_{mj}(\text{New}) = w_{mj}(\text{Old}) + \eta\delta_{O_m}P_j,\ m = 1, 2, \ldots, M \text{ and } j = 1, 2, \ldots, J \tag{15}$$

whereas the hidden layer weights are adjusted as

$$v_{ji}(\text{New}) = v_{ji}(\text{Old}) + \eta\delta_{P_j}Z_i,\ j = 1, 2, \ldots, J \text{ and } i = 1, 2, \ldots, I. \tag{16}$$

Here, η is the learning constant and in the similar manner, bias weights are also updated.

V. Results and Discussion

The novel aim of the proposed method is to estimate the structural response of multidegree freedom system from earthquake ground acceleration data. FLNN is trained and tested with different order polynomials. On the other hand, MNN is also trained and tested with different number of nodes in the hidden layer. Training is done with random weights and converged weights are stored for FLNN and MNN. With these stored weights testing has been done in order to get an accuracy of 0.001. For present investigation, examples of two, five, and ten storey shear buildings have been considered for training with Chamoli earthquake data and then structural response for ninth floor is predicted (testing) subject to different intensities of Uttarkashi earthquake data.

TABLE II
Comparison Among the Desired, ChNN, LeNN, and MNN Peak Response Values (Testing Values)

Intensities (Uttarkashi at Barkot NE)	Desired (cm/sec^2)	ChNN (cm/sec^2)	LeNN (cm/sec^2)	MNN (cm/sec^2)
80%	0.0031215	0.0031213	0.0031225	0.003015
120%	0.004687	0.004682	0.004683	0.004685

Example 1 (Two Storey-Building): In this case, Chamoli earthquake data at Barkot (NE direction) has been considered for training with peak acceleration value as 19.58 cm/s^2. A 2DOF system with natural frequency parameters 19.544 and 51.167 and damping ratio as 5% critical in both natural modes has been taken. Two separate models for two floors are used in training with a total time range of 0–10 s (500 data set). Responses of first and second storey are obtained numerically by solving the Duhamel integral that is (4) considering the ground acceleration data of Chamoli earthquake at Barkot (NE). First the training is done by Chebyshev neural network taking earthquake ground acceleration as input and structural response as output with random weights. After training by ChNN the converged weights are stored. Using the stored converged weights of ChNN, LeNN is trained for different orders of polynomials. LeNN is also trained with random weights. It is found that LeNN trained with stored converged weights gives better accuracy and takes less computation time than when trained with random weights. It is also seen that as we increase the order of these two orthogonal polynomials we get better accuracy. The training is also done with MNN. Out of 500 data sets used for training, results for ten typical data sets are shown in all the tables below. Comparison of desired ChNN with different order polynomials and MNN values for ten data sets for the first floor of the two-storey structure shown in Table I(a), and Table I(b) shows the results for desired ChNN with different order polynomials and MNN values for ten data sets for the second floor of the two-storey. Table I(a) and (b) shows how an increase in the order of polynomials shows better accuracy for this case. The result comparison between desired and LeNN with stored converged weights and random weights (for first floor) has been incorporated in Table I(c). Similarly for second floor the result is given in Table I(d). Table I(c) and (d) shows how learning with converged weights take less computation time and better accuracy than learning with random weights. The CPU time for ChNN is 86.30 s, LeNN with converged weights is 75.49 s, random weights is 78.69 s, and MNN is 284.20 s. It is found that ChNN and LeNN take less computation time than MNN.

Example 2 (Five-Storey Building): Same earthquake data as mentioned in Example 1 has been taken in this example. Again damping is assumed as 5% critical for this case. The training is done for 500 data set with a total time range of 0–10 s. ChNN and LeNN are trained with different order polynomials.

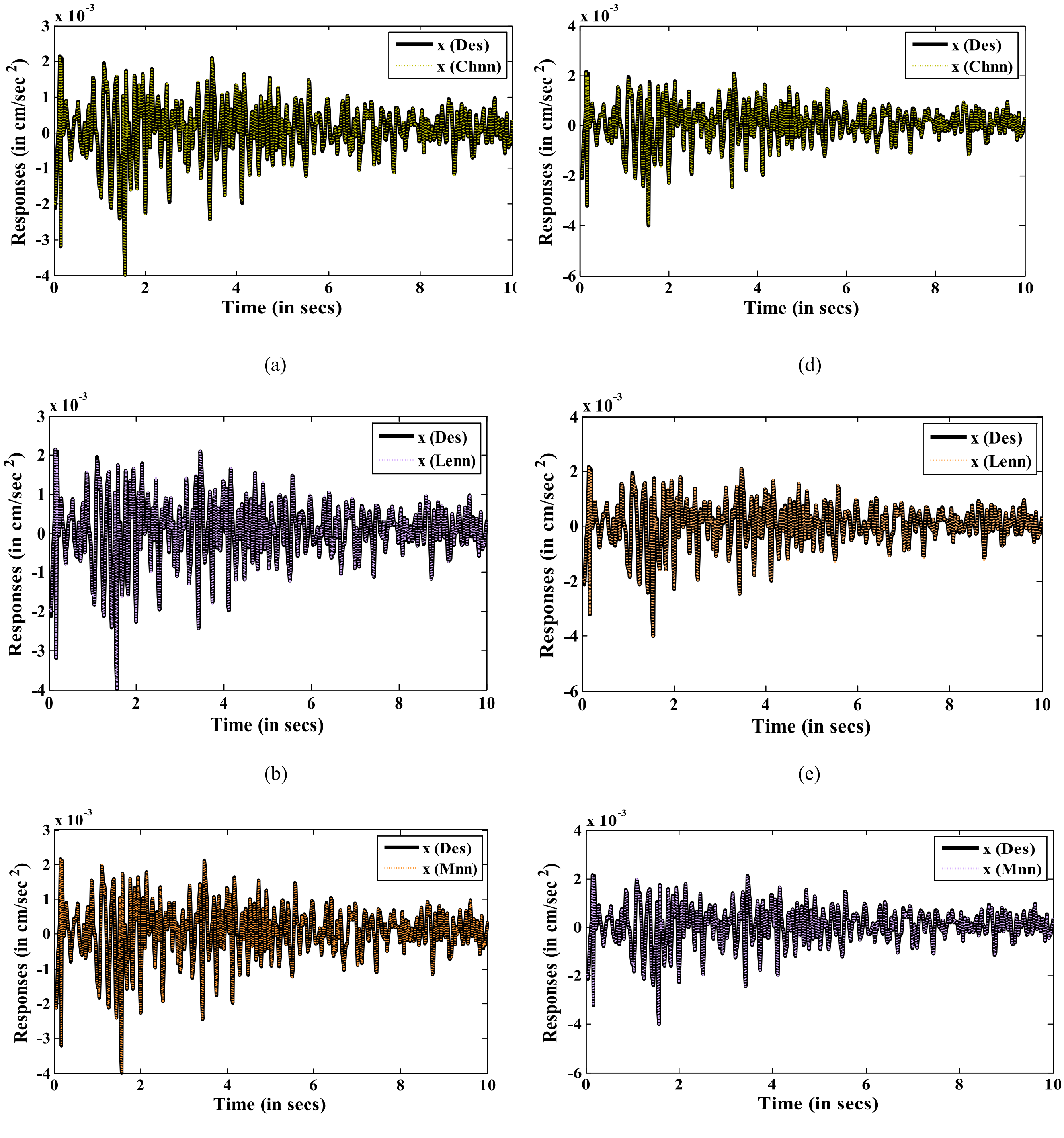
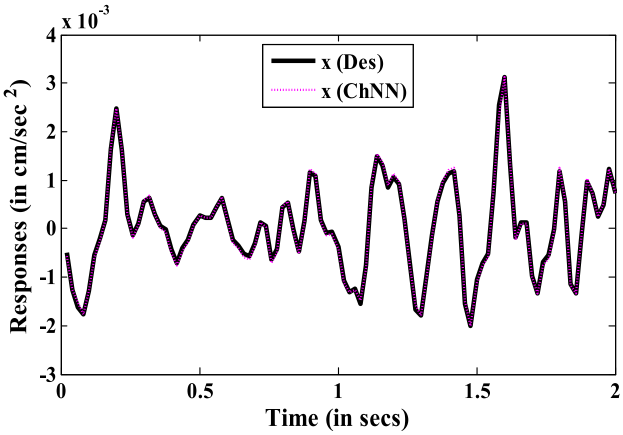


Fig. 4. Comparison between desired and (a) ChNN results for eighth floor, (b) LeNN results for eighth floor, (c) MNN results for eighth floor, (d) ChNN results for tenth floor, (e) LeNN results for tenth floor, and (f) MNN results for tenth floor.

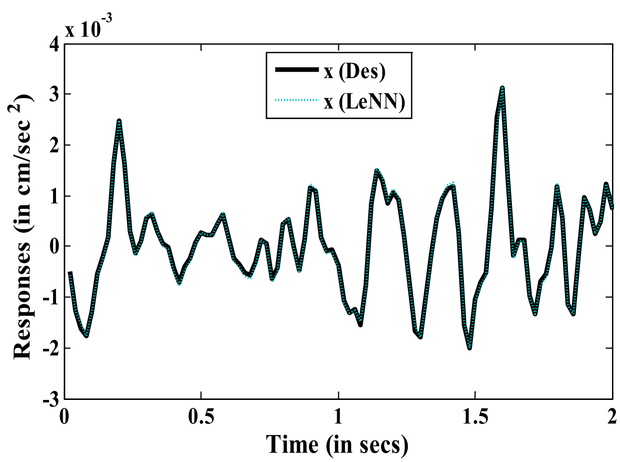
LeNN is trained with the stored converged weights of ChNN. MNN is trained with 15 hidden nodes in the hidden layer. Comparison between desired and ChNN of order 5 for fourth floor has been plotted in Fig. 3(a). Fig. 3(b) shows comparison between desired and LeNN (of order 5) for fourth floor. Similarly, comparison between desired and MNN for fourth floor has been shown in Fig. 3(c). Comparison between desired and ChNN (of order 5) for fifth floor is shown Fig. 3(d). The results of desired and LeNN (with order 5) for fifth floor is plotted in Fig. 3(e). Finally, Fig. 3(f) depicts result comparison between desired and MNN for fifth floor. The CPU time for this case for ChNN, LeNN and MNN are 3659.84, 3002.36, and 256393.21 s, respectively.

Example 3 (Ten Storey Building): In this case, the damping ratio is again assumed as 5% critical for all natural modes for the entire storey. In the similar manner using same earthquake

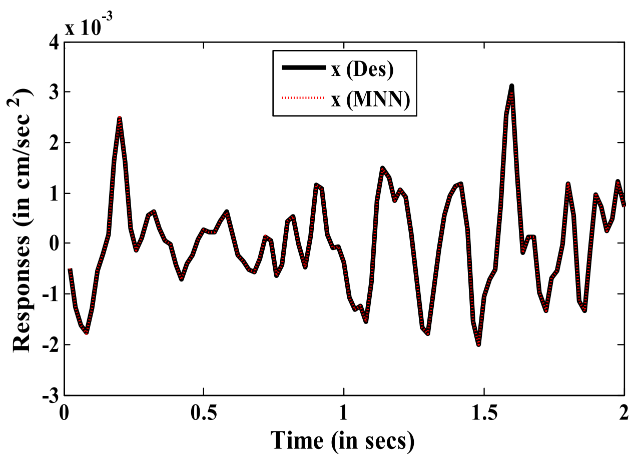
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a)

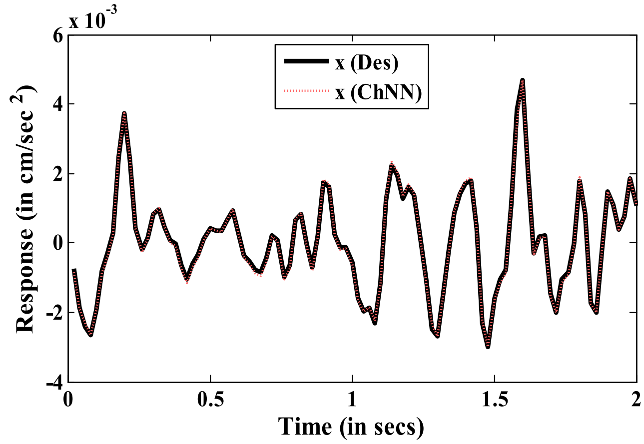


(b)

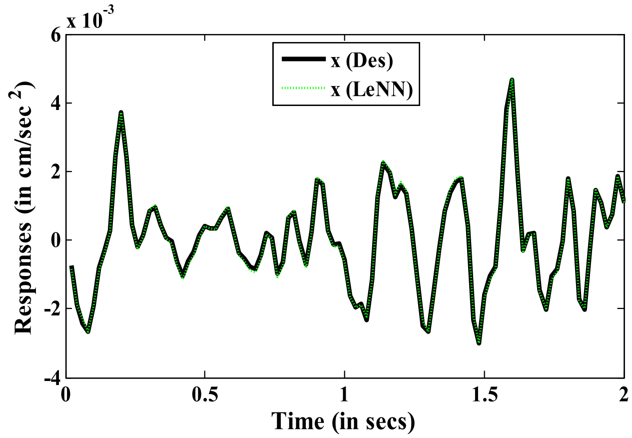


(c)

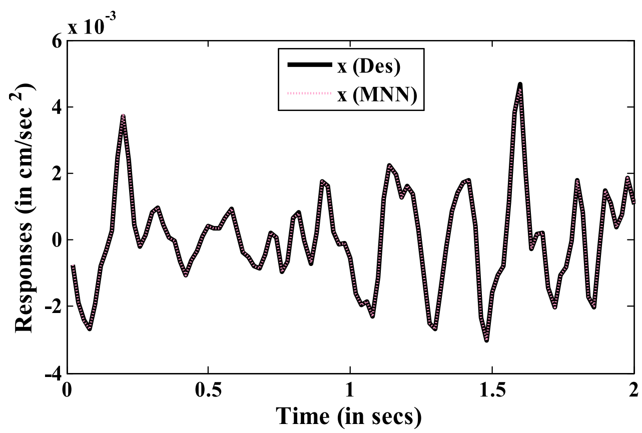
Fig. 5. Comparison between desired and (a) ChNN testing values (80%) for ninth floor, (b) LeNN testing values (80%) for ninth floor, and (c) MNN testing values (80%) for ninth floor.



(a)



(b)



(c)

Fig. 6. Comparison between desired and (a) ChNN testing values (120%) for ninth floor, (b) LeNN testing values (120%) for ninth floor, and (c) MNN testing values (120%) for ninth floor.

data as in cases 1 and 2, the training has been done for ten storey building for 500 data set with a total time of 0–10 s. Here ChNN and LeNN of order 5 have been considered and MNN has been trained with 15 hidden nodes. Comparison between desired, ChNN (of order 5), LeNN (with order 5), and MNN for eighth floor has been plotted separately in Fig. 4(a)–(c). Also results for tenth floor from ChNN, LeNN, and MNN are shown separately in Fig. 4(d)–(f), respectively.

The CPU time for this case for ChNN, LeNN, and MNN are 4412.06, 3012.16, and 328225.19 s, respectively.

A. Testing

Finally, testing is done with various intensities (80% and 120%) of Uttarkashi earthquake data. First training with FLNN and MNN is done with Chamoli earthquake data for ten storey shear buildings and weights were stored. These stored converged weights are then used to predict storey response for 80% and 120% of the Uttarkashi earthquake data. Training is done with 500 data sets with a time range of 0–10 s, but testing is done with 100 data sets with a total time of 0–2 s. Responses of ninth floor are found using the stored weights of FLNN and MNN. For testing, orthogonal polynomials of order 5 are considered for FLNN, and ten hidden nodes are taken for MNN. The peak response values (testing values) from desired, ChNN, LeNN, and MNN for 100 data sets with different intensities of Uttarkashi earthquake are given in Table II. Result comparisons among desired, ChNN, LeNN, and MNN for ninth floor with 80% intensity of Uttarkashi earthquake for 100 data sets (testing data) with time 0–2 s are shown in Fig. 5(a)–(c), respectively. Similarly the results of desired, ChNN, LeNN, and MNN for ninth floor with 120% intensity of Uttarkashi earthquake for 100 data set (testing data) with time 0–2 s are plotted separately in Fig. 6(a)–(c). From Figs. 5(a)–(c) and 6(a)–(c) and Table II, one may conclude that FLNN can very well be used for response prediction of multistorey shear buildings subject to various intensities of seismic loads. It is also seen that computation is faster in FLNN as compared to MNN and it can give better accuracy even with the absence of hidden layer.

VI. Conclusion

This paper estimates the structural response of multistorey shear buildings subject to seismic loads using FLNN. Training is done with ChNN, LeNN, and MNN taking the input as ground acceleration and output as structural response of each floor. ChNN is trained with random weights for different order of polynomials and it is seen that for higher order of polynomial better accuracy is achieved. LeNN is trained for different order of polynomials by two ways, one way is by using the stored converged weights of ChNN and the other way is by considering the random weights. After the training is complete it is found that LeNN with stored converged weights gives better accuracy than random weights. Training is done with Chamoli earthquake data but for testing different intensities of Uttarkashi earthquake data have been used. With the stored weights (of training), testing is done to show the powerfulness of the present model. In testing it is found that FLNN can predict structural response for different storey subject to different earthquake loads. It is worth mentioning that FLNN gives better accuracy in all the cases and also found to be computationally more efficient than MNN as it takes less computation time.

References

[1] P. C. Bressloff and J. Stark, "Analysis of associative reinforcement learning in neural networks using iterated function systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1348–1360, Nov./Dec. 1992.

[2] M. S. Mousavi and R. J. Schalkoff, "ANN implementation of stereo vision using a multi-layer feedback architecture," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 8, pp. 1220–1238, Aug. 1994.

[3] K.-C. Chan, C.-K. Koh, and C. S. G. Lee, "An automatic design of factors in a human-pose estimation system using neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 875–887, Jul. 2016.

[4] H. Ma *et al.*, "Neural-network-based distributed adaptive robust control for a class of nonlinear multiagent systems with time delays and external noises," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 750–758, Jun. 2016.

[5] Z. Wang, L. Liu, H. Zhang, and G. Xiao, "Fault-tolerant controller design for a class of nonlinear MIMO discrete-time systems via online reinforcement learning algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 5, pp. 611–622, May 2016.

[6] P. C. Pandey and S. V. Barai, "Multilayer perceptron in damage detection of bridge structures," *Comput. Struct.*, vol. 54, no. 4, pp. 597–608, 1995.

[7] J. Zhao, J. N. Ivan, and J. T. De Wolf, "Structural damage detection using artificial neural networks," *J. Infrastruct. Syst.*, vol. 4, no. 3, pp. 93–101, 1998.

[8] S. F. Masri, A. W. Smyth, A. G. Chassiakos, T. K. Caughey, and N. F. Hunter, "Application of neural networks for detection of changes in nonlinear systems," *J. Eng. Mech.*, vol. 126, no. 7, pp. 666–676, 2000.

[9] C. S. Huang, S. L. Hung, C. M. Wen, and T. T. Tu, "A neural network approach for structural identification and diagnosis of a building from seismic response data," *Earthquake Eng. Struct. Dyn.*, vol. 32, no. 2, pp. 187–206, 2003.

[10] V. K. Mathur, S. Chakraverty, and P. Gupta, "Response prediction of typical rural house subject to earthquake motions using artificial neural network," *J. Indian Build. Congr.*, vol. 11, no. 2, pp. 99–105, 2004.

[11] S. Chakraverty, T. Marwala, and P. Gupta, "Response prediction of structural system subject to earthquake motions using artificial neural network," *Asian J. Civil Eng.*, vol. 7, no. 3, pp. 301–308, 2006.

[12] S. Chakraverty, P. Gupta, and S. Sharma, "Neural network-based simulation for response identification of two-storey shear building subject to earthquake motion," *Neural Comput. Appl.*, vol. 19, no. 3, pp. 367–375, 2009.

[13] J. Reyes, A. Morales-Esteban, and F. Martínez-Álvarez, "Neural networks to predict earthquakes in Chile," *Appl. Soft Comput.*, vol. 13, no. 2, pp. 1314–1328, 2013.

[14] S. Niksarlioglu and F. Kulahci, "An artificial neural network model for earthquake prediction and relations between environmental parameters and earthquakes," *Int. J. Environ.*, vol. 7, no. 2, pp. 65–68, 2013.

[15] F. M. Alvarez, J. Reyes, A. M. Esteban, and C. R. Escudero, "Determining the best set of seismicity indicators to predict earthquakes two case studies: Chile and the Iberian Peninsula," *Knowl. Based Syst.*, vol. 50, pp. 198–210, Sep. 2013.

[16] A. Sriram, R. Shahryar, and F. Bourennani, "Artificial neural networks for earthquake anomaly detection," *J. Adv. Comput. Intell. Intell. Inf.*, vol. 18, no. 5, pp. 701–713, 2014.

[17] S. Mitra and S. K. Pal, "Self-organizing neural network as a fuzzy classifier," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 3, pp. 385–399, Mar. 1994.

[18] D. M. Sahoo, A. Das, and S. Chakraverty, "Interval data-based system identification of multistorey shear buildings by artificial neural network modelling," *Archit. Sci. Rev.*, vol. 58, no. 3, pp. 244–254, 2014.

[19] S. Chakraverty and D. M. Sahoo, "Fuzzy neural network-based system identification of multi-storey shear buildings," *Neural Comput. Appl.*, vol. 28, no. 3, pp. 597–612, 2017.

[20] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA, USA: Addison-Weseley, 1989.

[21] J. C. Patra, W. C. Chin, P. K. Meher, and G. Chakraborty, "Legendre-FLANN-based nonlinear channel equalization in wireless communication system," in *Proc. IEEE Int. Conf. Syst. Man and Cybern. (SMC)*, Singapore, 2008, pp. 1826–1831.

[22] J. C. Patra and A. C. Kot, "Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 4, pp. 505–511, Aug. 2002.

[23] S. Purwar, I. N. Kar, and A. N. Jha, "On-line system identification of complex systems using Chebyshev neural networks," *Appl. Soft Comput.*, vol. 7, no. 1, pp. 364–372, 2007.

[24] S. Dehuri and S. B. Cho, "A comprehensive survey on functional link neural networks and an adaptive PSO–BP learning for CFLNN," *Neural Comput. Appl.*, vol. 19, no. 2, pp. 187–205, 2010.

[25] X. Xiuchun, J. Xiohua, and Z. Yunong, "An algorithm for designing Chebyshev neural network," in *Proc. ISECS Int. Colloquium Comput. Commun. Control Manag.*, Sanya, China, 2009, pp. 206–209.

[26] J. C. Patra, P. K. Meher, and G. Chakraborty, "Nonlinear channel equalization for wireless communication systems using Legendre neural networks," *Signal Process.*, vol. 89, no. 11, pp. 2251–2262, 2009.

[27] J. C. Patra and C. Bornand, "Nonlinear dynamic system identification using Legendre neural network," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Barcelona, Spain, 2010, pp. 1–7.

[28] S. K. Nanda and D. P. Tripathy, "Application of functional link artificial neural network for prediction of machinery noise in opencast mines," *Adv. Fuzzy Syst.*, vol. 2011, Apr. 2011, Art. no. 831261, doi: 10.1155/2011/831261.

[29] Y. M. M. Hassim and R. Ghazali, "Using artificial bee colony to improve functional link neural network training," *Appl. Mech. Mater.*, vols. 263–266, pp. 2102–2108, Dec. 2012.

[30] H. H. Ali and M. T. Haweel, "Legendre neural networks with multi input multi output system equations," in *Proc. 7th Int. Conf. Comput. Eng. Syst. (ICCES)*, Cairo, Egypt, 2012, pp. 92–97.

[31] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 2, pp. 254–262, Apr. 1999.

[32] M. R. Mackenzie and A. K. Tieu, "Hermite neural network correlation and application," *IEEE Trans. Signal Process.*, vol. 51, no. 12, pp. 3210–3219, Dec. 2003.

[33] L. Ma and K. Khorasani, "Constructive feedforward neural networks using Hermite polynomial activation functions," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 821–833, Jul. 2005.

[34] B. B. Misra and S. Dehuri, "Functional link artificial neural network for classification task in data mining," *J. Comput. Sci.*, vol. 3, no. 12, pp. 948–955, 2007.

[35] S. K. Nanda, D. P. Tripathy, and S. S. Mahapatra, "Application of Legendre neural network for air quality prediction," in *Proc. 5th PSU-UNS Int. Conf. Eng. Technol. (ICET)*, Phuket, Thailand, May 2011, pp. 1–5.

[36] W. Li and C. Deng, "Fast and robust method for dynamic gesture recognition using Hermite neural network," *J. Comput.*, vol. 7, no. 5, pp. 1163–1168, 2012.

[37] S. Parija, P. K. Sahu, S. K. Nanda, and S. S. Singh, "A functional link artificial neural network for location management in cellular network," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Chennai, India, 2013, pp. 1160–1164.

[38] X. H. Yuan, Q. Y. Feng, T. J. Zeng, and H. B. Ma, "Digital predistortion for RF power amplifiers based on enhanced orthonormal Hermite polynomial basis neural network," in *Proc. Progr. Electromagn. Res. Symp. (PIERS)*, Taipei, Taiwan, Mar. 2013, pp. 1130–1133.

[39] H.-K. Chiang and C.-T. Chu, "Reference model with an adaptive Hermite fuzzy neural network controller for tracking a synchronous reluctance motor," *Elect. Power Compon. Syst.*, vol. 43, no. 7, pp. 770–780, 2015.

[40] S. Mall and S. Chakraverty, "Numerical solution of nonlinear singular initial value problems of Emden–Fowler type using Chebyshev neural network method," *Neurocomputing*, vol. 149, pp. 975–982, Feb. 2015.

[41] S. Mall and S. Chakraverty, "Chebyshev neural network based model for solving Lane–Emden type equations," *Appl. Math. Comput.*, vol. 247, pp. 100–114, Nov. 2015.

[42] A. Utkarsh, A. S. Kantha, J. Praveen, and J. R. Kumar, "Hybrid GA-PSO trained functional link artificial neural network based channel equalizer," in *Proc. 2nd Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Noida, India, 2015, pp. 285–290.

[43] D. K. Bebarta, B. Biswal, and P. K. Dash, "Polynomial based functional link artificial recurrent neural network adaptive system for predicting Indian stocks," *Int. J. Comput. Intell. Syst.*, vol. 8, no. 6, pp. 1004–1016, 2015.

[44] Y. M. M. Hassim and R. Ghazali, "Improving functional link neural network learning scheme for mammographic classification," in *Advances in Neural Networks* (Smart Innovation, Systems and Technologies), vol. 54. Cham, Switzerland: Springer, 2016, pp. 213–221.

[45] M. Kumar, S. K. Mishra, and S. S. Sahu, "Cat swarm optimization based functional link artificial neural network filter for Gaussian noise removal from computed tomography images," *Appl. Comput. Intell. Soft Comput.*, vol. 2016, Apr. 2016, Art. no. 6304915, doi: 10.1155/2016/6304915.

[46] S. Chakraverty, *Vibration of Plates*. Boca Raton, FL, USA: Taylor & Francis, 2009, pp. 1–411.

[47] C. F. Gerald, *Applied Numerical Analysis*. Boston, MA, USA: Pearson Educ., 2004, pp. 1–609.

[48] R. B. Bhat and S. Chakraverty, *Numerical Analysis in Engineering*. New Delhi, India: Narosa House, 2011, pp. 1–330.

[49] J. M. Zurada, *Introduction to Artificial Neural Systems*. Ahmedabad, India, Jaico House, 1994, pp. 1–679.

Deepti Moyi Sahoo is currently pursuing the Ph.D. degree with the Department of Mathematics, National Institute of Technology Rourkela, Rourkela, India.

Her current research interests include interval and fuzzy neural network, functional link neural network, and structural system identification.

Snehashish Chakraverty received the undergraduation degree from St. Columba's College Hazaribagh, Hazaribagh, India, the M.Sc. degree in mathematics and the M.Phil. degree in computer applications from the Indian Institute of Technology Roorkee (IIT Roorkee), Roorkee, India, and the Ph.D. degree from IIT Roorkee, in 1992.

He has experience of 27 years as a Researcher and a Teacher. He is currently a full Professor with the Department of Mathematics, National Institute of Technology Rourkela, Rourkela. He was with CSIR-Central Building Research Institute, Roorkee, India. He was a Post-Doctoral Research Fellow with the Institute of Sound and Vibration Research, University of Southampton, Southampton, U.K., and the Faculty of Engineering and Computer Science, Concordia University, Montreal, QC, Canada. He was also a Visiting Professor with Concordia University and McGill University, Montreal, from 1997 to 1999, and the University of Johannesburg, Johannesburg, South Africa, from 2011 to 2014. He has already guided 11 Ph.D. students and nine are ongoing. He has undertaken around 16 research projects as a Principle Investigator funded by international and national agencies totalling about Rs. 1.5 crores. His current research interests include soft computing and machine intelligence, artificial neural networks, fuzzy and interval computations, numerical analysis, differential equations, mathematical modeling, uncertainty modeling, and vibration and inverse vibration problems. He has authored ten books (of which two are in press) and published 259 research papers in journals and conferences and two more books are ongoing.

Dr. Chakraverty was a recipient of few prestigious awards, such as the Indian National Science Academy (INSA) nomination under International Collaboration/Bilateral Exchange Program (with Czech Republic), the Platinum Jubilee ISCA Lecture Award in 2014, the CSIR Young Scientist in 1997, the BOYSCAST (DST), the UCOST Young Scientist in 2007 and 2008, the Golden Jubilee Director's (CBRI) Award in 2001, the INSA International Bilateral Exchange Award in 2010 and 2011 (selected but could not undertake), and 2015 (selected), and the Roorkee University Gold Medals in 1987 and 1988. He is in the Editorial Boards of various international journals, book series, and conferences. He is the Chief Editor of the *International Journal of Fuzzy Computation and Modelling* (Switzerland: Inderscience). He is also a the Reviewer of around 30 national and international journals of repute. He was the Guest Editor for other few journals. He was the President of the Section of Mathematical sciences (including Statistics) of Indian Science Congress from 2015 to 2016, and the Vice President of the Orissa Mathematical Society from 2011 to 2013. He has organized a good number of international and national conferences, workshops, and training programmes.