Neural Networks 22 (2009) 1018–1024

Contents lists available at ScienceDirect

Neural Networks

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

A probabilistic neural network for earthquake magnitude prediction

Hojjat Adeli [a,*], Ashif Panakkat [b]

[a] Abba G. Lichtenstein Professor, Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, 470 Hitchcock Hall, 2070 Neil Avenue, Columbus, OH 43210, USA

[b] M. Engineering, Inc., 10 W Hubbard Street, Suite 3N, Chicago, IL 60610, USA

ARTICLE INFO

Article history:
Received 30 March 2008
Received in revised form 29 April 2009
Accepted 13 May 2009

Keywords:
Earthquake engineering
Probabilistic neural networks
Seismicity

ABSTRACT

A probabilistic neural network (PNN) is presented for predicting the magnitude of the largest earthquake in a pre-defined future time period in a seismic region using eight mathematically computed parameters known as seismicity indicators. The indicators considered are the time elapsed during a particular number (n) of significant seismic events before the month in question, the slope of the Gutenberg–Richter inverse power law curve for the n events, the mean square deviation about the regression line based on the Gutenberg–Richter inverse power law for the n events, the average magnitude of the last n events, the difference between the observed maximum magnitude among the last n events and that expected through the Gutenberg–Richter relationship known as the magnitude deficit, the rate of square root of seismic energy released during the n events, the mean time or period between characteristic events, and the coefficient of variation of the mean time. Prediction accuracies of the model are evaluated using three different statistical measures: the probability of detection, the false alarm ratio, and the true skill score or R score. The PNN model is trained and tested using data for the Southern California region. The model yields good prediction accuracies for earthquakes of magnitude between 4.5 and 6.0. The PNN model presented in this paper complements the recurrent neural network model developed by the authors previously, where good results were reported for predicting earthquakes with magnitude greater than 6.0.

1. Introduction

Earthquake prediction studies range from purely theoretical geophysics, to genetic mutations and biology, to statistical, mathematical, and computational modeling of earthquake parameter data recorded in historical catalogs of seismic regions. The most significant efforts in predicting the three main earthquake parameters, namely, the time of occurrence, the epicentral location and the magnitude of future earthquakes are reviewed in a recent article by Panakkat and Adeli (2008).

Neural networks have been used successfully to solve complicated pattern recognition and classification problems in different domains such as image and object recognition (Adeli & Hung, 1993, 1994; Bourbakis, Kakumanu, Makrogiannis, Bryll, & Panchanathan, 2007; Gopych, 2008; Hung & Adeli, 1993, 1994a; Khashman & Sekeroglu, 2008; Lian & Lu, 2007; Raftopoulos, Papadakis, Ntalianis, & Kollias, 2007; Rigatos & Tzafestas, 2007; Rodríguez-Sánchez, Simine, & Tsotsos, 2007; Vatsa, Singh, & Noore, 2007; Wersing et al., 2007), speech recognition (Yau, Kumar, & Arjunan, 2007), robotics and computer vision (Huang & Xu, 2007; Jorgensen, Haynes, & Norlund, 2008; Sabourin, Madani, & Bruneau, 2007; Taylor, Panchev, Hartley, Kasderidis, & Taylor, 2007; Villaverde, Grana, & d'Anjou, 2007), natural language and text processing (Ruiz-Pinales, Jaime-Rivas, Lecolinet, & Castro-Bleda, 2008; Terai & Nakagawa, 2007), signal processing (Cichocki & Zdunek, 2007), biomedical engineering and medical diagnosis (Adeli, Ghosh-Dastidar, & Dadmehr, 2005, 2008; Alexandre, Cuadra, Alvarez, Rosa-Zurera, & Lopez-Ferreras, 2008; Ghosh-Dastidar, Adeli, & Dadmehr, 2007; Gil-Pita & Yao, 2008; Huynh, Won, & Kim, 2008; Khashman, 2008; Kramer, Chang, Cohen, Hudson, & Szeri, 2007; Lan & Zhu, 2007; Lee, Cichocki, & Choi, 2007; Nemissi, Seridi, & Akdag, 2008; Osterhage, Mormann, Wagner, & Lehnertz, 2007), neuroscience (Bolle & Heylen, 2007; Chakravarthy, Sabesan, Tsakalis, & Iasemidis, 2007; Chen, Jiang, Wu, & Chen, 2007; Chiappalone, Vato, Berdondini, Koudelka, & Martinoia, 2007; Cutsuridis, 2007; Ghosh-Dastidar & Adeli, 2007; Guanella, Kiper, & Verschure, 2007; Iglesias & Villa, 2008; Mohan & Morasso, 2007; Montina, Mendoza, & Arecchi, 2007; Postnov, Ryazanova, Zhirin, Mosekilde, & Sosnovtseva, 2007), optimization and nonlinear programming (Chen & Young, 2007; Mayorga & Arriaga, 2007; Mayorga & Carrera, 2007; Mersch, Glasmachers, Meinicke, & Igel, 2007; Park & Adeli, 1995, 1997), construction engineering (Adeli & Wu, 1998; Senouci & Adeli, 2001), structural engineering (Adeli & Karim, 1997a, 1997b; Adeli & Kim, 2001; Adeli & Park, 1995a, 1995b, 1995c, 1996; Hung & Adeli, 1994b; Jiang & Adeli, 2005a, 2007, 2008a, 2008b; Sirca &

* Corresponding author.
E-mail address: adeli.1@osu.edu (H. Adeli).

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.05.003

Adeli, 2001; Tashakori & Adeli, 2002), transportation engineering (Adeli & Jiang, 2003; Adeli & Samant, 2000; Cyganek, 2007, 2008; Ghosh-Dastidar & Adeli, 2003; Jiang & Adeli, 2005b; Karim & Adeli, 2003; Pande & Abdel-Aty, 2008; Samant & Adeli, 2001; Stathopoulos, Dimitriou, & Tsekeris, 2008; Vlahogianni, Karlaftis, & Golias, 2007, 2008), video and audio analysis (Fyfe, Barbakh, Ooi, & Ko, 2008; Tsapatsoulis, Rapantzikos, & Pattichis, 2007), computer networking (Kimura & Ikeguchi, 2007), control (Liu & Zhang, 2008; Rigatos, 2008), computer security (Neumann, Eckmiller, & Baruth, 2007), communication analysis (Chen, Goldberg, Magdon-Ismail, & Wallace, 2008), satellite data and GPS (Mosavi, 2007), water resources engineering (Adeli, 2001), reservoir engineering (Banchs, Klie, Rodriguez, Thomas, & Wheeler, 2007), air traffic control (Christodoulou & Kontogeorgou, 2008), solar activity (Ni & Yin, 2008), biology (Wang, Jiang, Lu, Sun, & Noe, 2007), and financial forecasting (Schneider & Graupe, 2008). But no journal article was published on the application of neural networks for earthquake prediction until recently.

Recently, the authors presented three neural network models, a backpropagation (BP) neural network (Hung & Adeli, 1993), a radial-basis function (RBF) neural network (Adeli & Karim, 2000; Liu, Wang, & Qiang, 2007; Mayorga & Arriaga, 2007), and a recurrent neural network (Schaefer & Zimmermann, 2007) for the categorical prediction of an earthquake of a pre-defined threshold magnitude or greater in a following pre-defined time period (Panakkat & Adeli, 2007). The historical earthquake records for a given region are divided into a number of pre-defined equal time periods such as one month or 15 days. The inputs to the neural networks are eight physical and mathematical earthquake parameters called seismicity indicators computed from a pre-defined number of significant seismic events that occurred prior to that time period. The indicators are selected based on the Gutenberg–Richter (Gutenberg & Richter, 1956) and characteristic earthquake magnitude distributions and also on the conclusions drawn by recent earthquake prediction studies. They are the time elapsed during a particular number (say n) of significant seismic events before the time period in question (T), the slope of the Gutenberg–Richter inverse power law curve for the n events (b), the mean square deviation about the regression line based on the Gutenberg–Richter inverse power law for the n events (η), the average magnitude of the last n events (M_{mean}), the difference between the observed maximum magnitude among the last n events and that expected using the Gutenberg–Richter relationship known as the magnitude deficit (ΔM), the rate of square root of seismic energy released during time T ($dE^{1/2}$), the mean time or period between characteristic events (μ value) and the coefficient of variation of the mean time (c).

In this paper, we investigate a class of neural networks used primarily in classification problems called probabilistic neural networks (PNN) (Specht, 1990) for earthquake magnitude prediction. Wasserman (1993) suggests a few advantages for the PNN over the conventional BP neural network as follows:

(1) The PNN does not require a separate training phase and is therefore computationally on average more than five times faster than the conventional BP network,
(2) Additional training instances can be incorporated easily into a PNN as they become available in the future, and
(3) The PNN provides classification robustness in domains with noisy data.

Probabilistic neural networks have been used successfully for solving complicated problems from classification of underwater sonar signals radiated by ships (Chen, Lee, & Lin, 2000) to classification of taxonomical and metabolic responses of the human body (Holmes, Nicholson, & Tranter, 2001) to text-independent speaker identification in natural language processing (Ganchev, Fakotakis, & Kokkinakis, 2002) to class prediction of leukemia and embryonic tumors (Huang & Liao, 2004), and the prediction of compressive strength of concrete samples (Kim, Lee, & Chang, 2005). The authors were able to locate only a single article in the published literature on the use of PNN in earthquake research. Goh (2002) uses a PNN model to evaluate the seismic liquefaction potential of soils. The input to the neural network is data obtained from cone penetration tests and recorded shear wave velocity in the soil. The output yields liquefiable or non-liquefiable soil.

In this research, earthquake magnitude prediction is treated as a classification problem where the magnitude range in a future time period is predicted as one of several output classes. An eight-element vector of seismicity indicators defined earlier is computed for each time period (such as one month or 15 days) in the historical record of a seismic region as reported in Panakkat and Adeli (2007). These vectors which form the training dataset are grouped into one of several input classes based on the magnitude of the largest seismic event that occurred during the time period for which the indicators are computed. Subsequently, when a new input vector is presented, the PNN places it in that input class which yields the maximum joint probability density function (PDF). The selected class yields the magnitude range of the largest earthquake in the corresponding time period.

2. A probabilistic neural network for earthquake magnitude prediction

2.1. Bayesian statistics and non-parametric density approximation

A PNN operates based on the concept of the Parzen windows classifier and its application to Bayesian statistics (Gelman, Carlin, Stern, & Rubin, 2003). In Bayesian statistics, probability of a given input belonging to a particular input class is estimated directly from a set of available historical data without calculating or assuming a set of model parameters such as mean and standard deviation. Bayes classifier is a classification technique based on Bayesian statistics where the posterior probability of an input class $\mathbf{C}$ is computed as

$$P(\mathbf{C}) = P(\mathbf{C}\,|\mathbf{x})P(\mathbf{x}) \tag{1}$$

where $P(\mathbf{C}\,|\mathbf{x})$ is the conditional probability of class $\mathbf{C}$ given a vector $\mathbf{x}$, and $P(\mathbf{x})$ is the probability of vector $\mathbf{x}$. The classification decision is made by computing the posterior probabilities of all classes and selecting that class which yields the maximum value. Since $P(\mathbf{x})$ remains the same for all classes it does not play a role in the classification process. Therefore, the most likely class for the given input vector is the one which yields the maximum $P(\mathbf{C}\,|\mathbf{x})$. Mathematically, $P(\mathbf{C}\,|\mathbf{x})$ is the integral of the joint PDF of $\mathbf{C}$ and $\mathbf{x}$ represented by $p(\mathbf{C}, \mathbf{x})$.

There are several methods for estimating the joint PDF of a given input and a particular input class. One such method is the Parzen windows classifier, a non-parametric procedure where the joint PDF of a given input vector and a certain input class is approximated by the superposition of a number of (usually Gaussian) functions or *windows*. It has its basis in the fact that given some sample vectors in class $\mathbf{C}$, the joint PDF of a vector $\mathbf{x}$ and the class $\mathbf{C}$ is given by (Babich & Camps, 1996)

$$p(\mathbf{C}, \mathbf{x}) = \frac{k_c/n}{V_c} \tag{2}$$

provided the following three conditions are satisfied:

$$\lim_{n\to\infty} V_c = 0 \tag{3}$$

$$\lim_{n\to\infty} k_c = \infty \tag{4}$$

$$\lim_{n\to\infty} k_c/n = 0 \tag{5}$$

where k_c is a measure of the number of samples within class $\mathbf{C}$ that

are *similar* to **x**, n is the total number of samples (those belonging to class **C** as well as those not belonging to class **C**), and V_c is the volume of the space defined by the k_c samples. *Similarity* between the given vector **x** and a sample vector in class **C** is defined as the proximity between the two vectors (usually in the Euclidean sense). In the Parzen windows method of PDF approximation, the volume V_c is reduced to a finite volume (such as to a hypercube of volume equal to one) by defining a window function or kernel. For a predetermined V_c the value of k_c is determined analytically.

2.2. Parzen windows classifier

The use of the Parzen windows classifier for non-parametric PDF estimation is demonstrated through an example window function or kernel. Assume that the region defined by the k_c samples is a d-dimensional hypercube whose volume is given by

$$V_c = h_c^d \tag{6}$$

where h_c is the length of one edge of the hypercube. As an example, consider the following kernel or window function (Babich & Camps, 1996):

$$W(\mathbf{x}) = 1 \quad \text{when } \left|\mathbf{x}_j\right| \leq 1/2$$
$$= 0 \quad \text{otherwise (where } j = 1, 2, \ldots\ldots\ldots, d). \tag{7}$$

Graphically, W represents a step function whose value steps up from 0 to 1 at $x_j = -1/2$ and remains at 1 until $x_j = 1/2$ at which point the value steps down to 0 defining a square with a unit edge length centered at the origin. Therefore, for any d-element vector **x**, $W(\mathbf{x})$ defines a d-dimensional hypercube of edge length equal to one centered at the origin. It follows that for any vector $\mathbf{x}_i$ that belongs to class **C** and any given vector **x**, $W(\mathbf{x} - \mathbf{x}_i)$ equals unity when $\mathbf{x}_i$ falls within the unit hypercube centered at **x** and zero otherwise. Therefore, the number of samples falling within the unit hypercube centered at **x** is obtained by summing the values of $W(\mathbf{x} - \mathbf{x}_i)$ for all samples in class **C** as follows:

$$k_c = \sum_{i=1}^{N_c} W(\mathbf{x} - \mathbf{x}_i) \tag{8}$$

where N_c is the number of samples in class **C**. Substituting Eq. (8) into Eq. (2) and noting that the hypercube has a unit volume, we obtain an estimate of the joint PDF as

$$p(\mathbf{C}, \mathbf{x}) = \frac{1}{n} \sum_{i=1}^{N_c} W(\mathbf{x} - \mathbf{x}_i). \tag{9}$$

In this simple example and the function presented by Eq. (7), a value of one was pre-selected for the volume V_c. In real life classification problems which are often complicated and require a more sophisticated window function such as the Gaussian function, the volume V_c cannot be pre-selected. For the best classification results, the value of V_c can only be determined by trial-and-error or numerical experimentation. Therefore, Eq. (9) is generalized for any window function as follows:

$$p(\mathbf{C}, \mathbf{x}) = \frac{1}{nV_c} \sum_{i=1}^{N_c} W(\mathbf{x} - \mathbf{x}_i). \tag{10}$$

It should also be noted that in general k_c is not a whole number because the selected window function often yields a non-binary output (Eq. (8)). As such, k_c in general is merely a *representation* of the number of samples in class **C** that are similar to **x** and not necessarily the actual number. Essentially the window function is being used for interpolation where each sample contributes to the estimate of the PDF depending on its proximity to **x**.

The selected window function must satisfy the following conditions for $p(\mathbf{C}, \mathbf{x})$ in Eq. (9) to be a PDF (Bishop, 1995):

(i) $W(\mathbf{x})$ must be greater than or equal to 0 for all values of **x**,
(ii) $W(\mathbf{x})$ must be an integrable function and the value of its integral between the limits $-\infty$ and ∞ must equal nV_c.

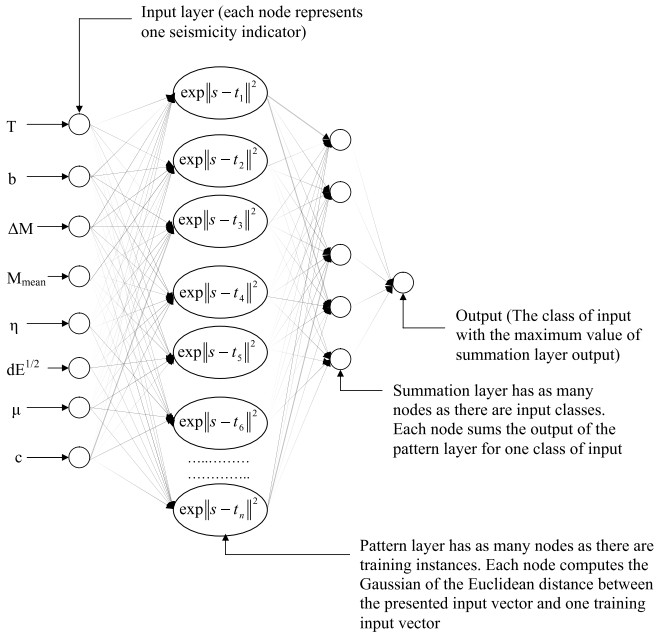

Fig. 1. Architecture of the PNN for predicting the magnitude range of the largest seismic event in a following time period.

It should be noted that, in the classification problem, the scalar 1/n is common for all input classes and therefore it is sufficient to compare the value of

$$nxp(\mathbf{C}, \mathbf{x}) = \frac{1}{V_c} \sum_{i=1}^{N_c} W(\mathbf{x} - \mathbf{x}_i) \tag{11}$$

for each class to arrive at the most likely class.

2.3. PNN Architecture and Operation

An eight-element vector of seismicity indicators is computed for each time period (such as one month). In the earthquake magnitude prediction problem, an output group is defined by upper and lower magnitude limits. For example, recorded earthquakes in a historical catalog with Richter magnitudes between 4.5 and 7.5 can be grouped into six output groups where each group is comprised of earthquakes of magnitude in a 0.5 Richter range. With this definition, an eight-element vector of seismicity indicators computed for a particular time period is classified into one of six input classes depending on the output group in which the largest earthquake that occurred during the time period falls under. In order to maintain the same numerical range for input parameters, they were normalized using their root mean square values.

The architecture of PNN for predicting the magnitude of the largest event in a following time period is shown in Fig. 1. It consists of an input layer, a hidden layer called the pattern layer, a second hidden layer called the summation layer, and a single-node output layer called the competition layer. The input layer has eight nodes where each node represents a seismicity indicator. The number of nodes in the pattern layer is equal to the number of available training input vectors, n. The summation layer has as many nodes as there are pre-defined input classes, N.

In the PNN model for earthquake magnitude prediction, a Gaussian function is used as the window function. Each node in the pattern layer computes the value of the Gaussian function of the Euclidean distance between the given input vector and one training input vector. The output of the node indicates how close the given input vector is to the training input vector. As such, the pattern layer operates similar to a hidden layer in an RBF network (Karim & Adeli, 2003). The output of the node in the pattern layer

corresponding to the ith training input vector in the cth input class is computed as

$$o_{pci} = \Phi(\mathbf{s}-\boldsymbol{t}_{ci}) = \frac{1}{\sigma_c\sqrt{2\pi}} \exp - \left(\frac{\|\mathbf{s}-\mathbf{t}_{ci}\|^2}{2\sigma_c^2}\right) \tag{12}$$

where $\mathbf{s}$ is the given input vector, $\mathbf{t}_{ci}$ is the ith training input vector in the cth input class, and σ_c is the width factor associated with the cth input class. The parameters $\mathbf{t}_{ci}$ and σ_c represent the center and spread (or volume) of the Gaussian bell curve, respectively. The value of o_{pci} is close to 1 when $\mathbf{s}$ is near $\mathbf{t}_{ci}$, and close to 0 when $\mathbf{s}$ is far from $\mathbf{t}_{ci}$ in a Euclidean sense.

Each node in the summation layer sums the outputs of all the nodes in the pattern layer belonging to a particular input class as follows:

$$o_{sc} = \sum_{i=1}^{N_c} o_{pci} = \frac{1}{\sigma_c\sqrt{2\pi}} \sum_{i=1}^{N_c} \exp - \left(\frac{\|\mathbf{s}-\mathbf{t}_{ci}\|^2}{2\sigma_c^2}\right) \tag{13}$$

where N_c is the number of available training input vectors in the cth input class. In Eq. (13) $\sigma_c\sqrt{2\pi}$ is equivalent to V_c in Eq. (11). The width factor σ_c associated with the cth class is determined by numerical experimentation.

Comparing Eq. (13) with Eq. (11), it can be concluded that the maximum PDF is obtained for the class which yields the maximum value of o_{sc}. The output (competition) node yields one for the input class, which has the maximum joint PDF for the given input and zero for all other classes.

In prediction problems, the PNN operates differently from conventional neural networks such as the simple BP neural network, the RBF neural network, and the recurrent neural network. In conventional neural networks a training or learning rule is used to obtain the weights of the links connecting various layers and to train the network. This is often achieved by solving an unconstrained optimization problem using a pre-defined performance function and convergence criteria. In contrast, in a PNN no learning rule is required, no weights are assigned to the links connecting the layers, and no pre-defined convergence criteria are needed. The drawback of the PNN in prediction problems is that it can be used only if the desired output is expressed as one of several pre-defined classes (Cowell, Lauritzen, Spiegelhater, & David, 2003). As such, the PNN is suitable for predicting earthquake magnitude ranges but not time or epicentral location.

The Parzen windows classifier technique provides an approximation for the joint PDF of a given input and a particular input class (an underlying population) using a limited number of sample inputs available in that class. The estimated PDF gets closer to the underlying PDF of the class as the size of the sample increases. In the earthquake prediction problem, the population corresponds to all historical earthquakes of a particular magnitude range that occurred in a seismic region. A particular sample of this population for which the joint PDF is estimated is comprised of those earthquakes in the same magnitude range that is available in the historical record. Hence it is expected that the prediction accuracy of the network will be superior for those magnitude ranges which have a large number of occurrences in the available historical record compared to those magnitude ranges with scarce recordings. In most seismically active regions such as Southern California, the former class is comprised of low to moderate earthquakes and the latter class is comprised of major earthquakes.

3. Example application

As an example, it is attempted to predict the magnitude range (within 0.5 Richter) of the largest earthquake in Southern California in the following fifteen day time period. Southern California is defined by the Working Group on California Earthquake Probabilities (2004) as an area between geographic coordinates 114.75 W and 119.25 W longitude and 33.8 N and 35.4 N latitude. Historical seismic data recorded in southern California dating back to 1930 is archived by the Southern California Earthquake Data Center (SCEC) and is available for free download through the center's website at www.data.scec.org. Data obtained from the SCEC website is used to define input classes and test the PNN model developed in this research.

Table 1
Input classes in the training dataset for the southern California region, the output magnitude range, and the number of training instances available in each class.

Input class	Output magnitude range	Number of training instances available
Class 1	<4.5	441
Class 2	4.5–5.0	309
Class 3	5.0–5.5	137
Class 4	5.5–6.0	78
Class 5	6.0–6.5	18
Class 6	6.5–7.0	9
Class 7	7.0–7.5	5

The historical earthquake record of Southern California between 1st January 1950 and 13th December 1990 is divided into 997 fifteen day time periods. Seven output groups are defined based on upper and lower magnitude levels. The first group is comprised of earthquakes of magnitude less than 4.5 and the remaining six groups are comprised of earthquakes of magnitude between 4.5 Richter and 7.5 Richter where each group has a magnitude range of 0.5 Richter.

An eight-element vector of seismicity indicators is computed for each time period as explained in Panakkat and Adeli (2007) forming 997 training input vectors (training dataset). The training dataset is divided into seven input classes depending on the magnitude of the largest earthquake that occurred during each time period. Input classes of the training dataset, the corresponding output magnitude range, and the number of training input vectors available in each class are presented in Table 1. As explained earlier, the prediction success for a particular magnitude range is expected to improve as the number of training input vectors available in the corresponding input class increases. Therefore it can be concluded from Table 1 that the network is most accurate in classifying events into the $M < 4.5$, $M = 4.5$–5.0, $M = 5.0$–5.5, and $M = 5.5$–6.0 ranges and may not have much success in classifying events into the $M = 6.0$–6.5, $M = 6.5$–7.0, and $M = 7.0$–7.5 ranges. Samples of the training dataset showing the eight-element input vectors and the corresponding input class for the ten time periods between 1st January 1950 and 5th May 1950 are shown in Table 2.

For testing, the PNN is used to predict the magnitude range of the largest earthquake in each of the 383 fifteen day time periods between 1st January 1990 and 24th September 2005 by computing eight-element test input vectors for each time period (testing dataset). Network operation is repeated for each input vector and the magnitude range of the largest earthquake in the following fifteen day time period is determined for each of the 383 time periods. After each test run, the input vector is added to the training dataset. Therefore the number of training input vectors available for the first test iteration is 997 and increases by one with each iteration thereafter. This is done easily in a PNN model as opposed to a conventional neural network where the addition of any new training example requires the retraining of the network.

3.1. Prediction verification

For each time period in the testing dataset, the magnitude range obtained as neural network output is compared to the magnitude range of the largest earthquake recorded during that time period.

Table 2
Sample training datasets for the 10 time periods between 1st January 1950 and 25th May 1950 for southern California showing eight-element input vectors computed for each time period and the corresponding input class based on the magnitude of the largest earthquake that occurred during that time period.

Time period	Input vector								Input class
	T (Days)	b	η	ΔM	M_{mean}	$dE^{1/2}$ ($\times 10^{19}$ ergs)	μ (Days)	c	
1/1/50–1/14/50	5784	0.76	0.27	2.33	5.31	0.118	39	0.11	Class 2
1/15/50–1/28/50	5798	0.69	0.80	0.69	4.89	0.037	50	0.28	Class 1
1/29/50–2/11/50	5812	0.93	0.46	1.17	4.97	0.102	27	0.15	Class 3
2/12/50–2/26/50	5826	0.71	0.67	0.39	5.31	0.051	37	0.31	Class 1
2/27/50–3/12/50	5840	0.89	0.60	1.89	4.87	0.089	32	0.19	Class 3
3/13/50–2/27/50	5854	0.73	0.81	0.97	5.10	0.101	58	0.89	Class 2
3/28/50–4/10/50	5868	0.88	0.58	1.21	5.33	0.120	52	0.90	Class 1
4/11/50–4/25/50	5882	0.90	0.44	0.33	4.88	0.065	61	0.68	Class 4
4/26/50–5/10/50	5896	0.81	0.28	1.07	4.79	0.039	50	0.80	Class 4
5/11/50–5/25/50	5910	0.83	0.37	0.89	5.07	0.068	44	0.77	Class 1

Table 3
Computed values of POD, FAR, and R score for different magnitude ranges for the Southern California region.

Magnitude range (Richter)	p_0	POD	FAR	R score
4.5–5.0	0.39	0.80	0.14	0.66
5.0–5.5	0.21	0.88	0.16	0.62
5.5–6.0	0.11	0.83	0.05	0.78
6.0–6.5	0.06	0.50	0.00	0.50
6.5–7.0	0.009	0.00	0.00	0.00
7.0–7.5	0.003	0.50	0.00	0.50

Three statistical measures of success are computed for each predicted magnitude range from the number of correct and incorrect predictions. These are, the probability of detection (POD), the false alarm ratio (FAR), and the R score (Panakkat & Adeli, 2007). The parameters are computed using the following equations:

$$\text{POD} = \frac{N_{pc}}{N_{pc} + N_{ni}} \tag{14}$$

$$\text{FAR} = \frac{N_{pi}}{N_{pc} + N_{pi}} \tag{15}$$

$$R = \text{POD} - \text{FAR} \tag{16}$$

where N_{pc} (predicted-correct) is the number of time periods during which the magnitude of the largest recorded earthquake falls within the predicted magnitude range, N_{ni} (not predicted-incorrect) is the number of time periods during which the magnitude of the largest recorded earthquake is greater than the upper limit of the predicted magnitude range, and N_{pi} (predicted-incorrect) is the number of time periods during which the magnitude of the largest recorded earthquake is less than the lower limit of the predicted magnitude range.

To test for statistical significance of the prediction results, the POD computed for each magnitude range is compared to the probability of occurrence of earthquakes in that magnitude range obtained using the Poisson's null hypothesis (p_0) (Panakkat & Adeli, 2007). Only those magnitude ranges for which the computed POD is larger than the computed p_0 are considered as significant predictions.

3.2. Prediction Results

The computed values of p_0, POD, FAR, and R score for each magnitude range based on the number of successful and unsuccessful predictions in the testing dataset is presented in Table 3. The suitability of the network in predicting small, moderate and large earthquakes is discussed in the following paragraphs:

(a) Prediction of large earthquakes (Magnitude 6.5 or greater but less than 7.5)

There were two periods in the testing dataset during which an earthquake of magnitude between 7.0 and 7.5 occurred (2nd June 2 – 16th June, 1992, and 9th October – 23rd October, 1999). The PNN correctly predicted one of these events and did not falsely predict any event in this magnitude range yielding a POD of 0.5 and FAR of 0.0. There was a single period in the testing dataset, during which an earthquake of magnitude between 6.5 and 7.0 occurred (27th December, 1993 – 10th January, 1994). The PNN did not correctly predict this event and did not falsely predict an event of magnitude between 6.5 and 7.0 for any period yielding a POD and FAR of 0.0.

(b) Prediction of moderate earthquakes (Magnitude 5.5 or greater but less than 6.5)

There were four periods in the testing dataset during which an earthquake of magnitude between 6.0 and 6.5 occurred. The network correctly predicted two of the four events yielding a POD of 0.50 and did not falsely predict any event of magnitude between 6.0 and 6.5 yielding an FAR of 0.0. There were 23 periods in the testing dataset during which an earthquake of magnitude between 5.5 and 6.0 occurred. The PNN correctly predicted nineteen of these events yielding a POD of 0.83 and falsely predicted an event of magnitude between 5.5 and 6.0 for only one period yielding an FAR of 0.05.

(c) Prediction of small earthquakes (Magnitude 4.5 or greater but less than 5.5)

There were 77 periods in the testing dataset during which an earthquake of magnitude between 5.0 and 5.5 occurred. The PNN correctly predicted 63 of these events yielding a POD of 0.88 and falsely predicted twelve events yielding an FAR of 0.16. There were 127 periods in the testing dataset during which an event of magnitude between 4.5 and 5.5 occurred. The PNN correctly predicted 102 of these events yielding a POD of 0.80 and falsely predicted seventeen events yielding an FAR of 0.14.

4. Final comments

The PNN model presented in this article yields good prediction accuracies for earthquakes of magnitude between 4.5 and 6.0 (yielding R score values between 0.62 and 0.78 presented in Table 3) but does not yield good results for the prediction of earthquakes of magnitudes greater than 6.0 (yielding R score values between 0.0 and 0.5). When used for predicting earthquake magnitudes in Southern California, the recurrent neural network model developed by the authors earlier was significantly more successful in predicting earthquakes of magnitude between 6.0 and 7.5 (yielding R score values between 0.50 and 1.00) compared to predicting earthquakes of magnitude less than 6.0 (yielding R score values between 0.36 and 0.51) (Panakkat & Adeli, 2007).

For the Southern California region, the input classes corresponding to the high magnitude ranges have few training input vectors because of the relative scarcity of major earthquakes in the training dataset. It can be expected that the prediction accuracy of the PNN will be higher for larger earthquakes also in regions with a

large number of major earthquakes in the historical record; for example, the East Pacific Rise Transform (EPRT) which has on average three earthquakes of magnitude 6.0 or greater, per year (McGuire, Boettcher, & Jordan, 2005).

For all magnitude ranges, except $M = 6.0$–6.5, the computed POD is larger than the computed p_0 indicating that overall the results are superior to those obtained by pure chance. The exception is due to the lack of data in that magnitude range. It should also be noted that the comparison is biased against the PNN model since the POD is computed based on the relatively small number of events in the testing dataset whereas the p_0 is computed from all events in that magnitude range in the historical record.

The model presented in this paper complements the recurrent neural network model for earthquake parameters' prediction reported earlier Panakkat and Adeli (2007). The PNN model can be used to predict earthquakes with magnitude less than 6.0 while the recurrent neural network model can be used to predict earthquakes with magnitude greater than 6.0.

References

Adeli, H. (2001). Neural networks in civil engineering: 1989–2000. *Computer-Aided Civil and Infrastructure Engineering*, *16*(2), 126–142.

Adeli, H., Ghosh-Dastidar, S, & Dadmehr, N. (2005). Alzheimer's disease and models of computation: Imaging, classification, and neural models. *Journal of Alzheimer's Disease*, *7*(3), 187–199.

Adeli, H., Ghosh-Dastidar, S., & Dadmehr, N. (2008). A spatio-temporal wavelet-chaos methodology for EEG-based diagnosis of Alzheimer's disease. *Neuroscience Letters*, *444*(2), 190–194.

Adeli, H., & Hung, S. L. (1993). A concurrent adaptive conjugate gradient learning algorithm on MIMD machines. *Journal of Supercomputer Applications*, *7*(2), 155–166.

Adeli, H., & Hung, S. L. (1994). An adaptive conjugate gradient learning algorithm for effective training of multilayer neural networks. *Applied Mathematics and Computation*, *62*(1), 81–102.

Adeli, H., & Karim, A. (1997a). Neural dynamics model for optimization of cold-formed steel beams. *Journal of Structural Engineering, ASCE*, *123*(11), 1535–1543.

Adeli, H., & Karim, A. (1997b). Scheduling/cost optimization and neural dynamics model for construction. *Journal of Construction Management and Engineering, ASCE*, *123*(4), 450–458.

Adeli, H., & Karim, A. (2000). Fuzzy-wavelet RBFNN model for freeway incident detection. *Journal of Transportation Engineering*, *126*(6), 464–471.

Adeli, H., & Kim, H. (2001). Cost optimization of composite floors using the neural dynamics model. *Communications in Numerical Methods in Engineering*, *17*, 771–787.

Adeli, H., & Jiang, X. (2003). Neuro-fuzzy logic model for freeway work zone capacity estimation. *Journal of Transportation Engineering, ASCE*, *129*(5), 484–493.

Adeli, H., & Park, H. S. (1995a). Counter propagation neural network in structural engineering. *Journal of Structural Engineering, ASCE*, *121*(8), 1205–1212.

Adeli, H., & Park, H. S. (1995b). A neural dynamics model for structural optimization — Theory. *Computers and Structures*, *57*(3), 383–390.

Adeli, H., & Park, H. S. (1995c). Optimization of space structures by neural dynamics. *Neural Networks*, *8*(5), 769–781.

Adeli, H., & Park, H. S. (1996). Fully automated design of superhighrise building structure by a hybrid AI model on a massively parallel machine. *AI Magazine*, *17*(3), 87–93.

Adeli, H., & Samant, A. (2000). An adaptive conjugate gradient neural network — Wavelet model for traffic incident detection. *Computer-Aided Civil and Infrastructure Engineering*, *15*(4), 251–260.

Adeli, H., & Wu, M. (1998). Regularization neural network for construction cost estimation. *Journal of Construction Engineering and Management, ASCE*, *124*(1), 18–24.

Alexandre, A., Cuadra, L., Alvarez, L., Rosa-Zurera, M., & Lopez-Ferreras, F. (2008). Two-layer automatic sound classification for conversation enhancement in hearing aids. *Integrated Computer-Aided Engineering*, *15*(1), 85–94.

Babich, G., & Camps, O. (1996). Weighted Parzen windows for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(5), 567–570.

Banchs, R., Klie, H., Rodriguez, A., Thomas, S. G., & Wheeler, M. F. (2007). A neural stochastic multiscale optimization framework for sensor-based parameter estimation. *Integrated Computer-Aided Engineering*, *14*(3), 213–223.

Bishop, C. (1995). *Neural network for pattern recognition*. Oxford, UK: Oxford University Press.

Bolle, D., & Heylen, R. (2007). Adaptive thresholds for neural networks with synaptic noise. *International Journal of Neural Systems*, *17*(4), 241–252.

Bourbakis, N., Kakumanu, P., Makrogiannis, S., Bryll, R., & Panchanathan, S. (2007). Neural network approach for image chromatic adaptation for skin color detection. *International Journal of Neural Systems*, *17*(1), 1–12.

Chakravarthy, N., Sabesan, S., Tsakalis, K., & Iasemidis, L. (2007). Controling synchronization in a neural-level population model. *International Journal of Neural Systems*, *17*(2), 123–138.

Chen, C., Lee, J., & Lin, M. (2000). Classification of underwater signals using neural networks. *Tamkang Journal of Science and Engineering*, *3*(1), 31–48.

Chen, H. C., Goldberg, M., Magdon-Ismail, M., & Wallace, W. A. (2008). Reverse engineering a social agent-based hidden Markov Model. *International Journal of Neural Systems*, *18*(6), 491–526.

Chen, M., Jiang, C. S., Wu, Q. X., & Chen, W. H. (2007). Synchronization in arrays of uncertain delay neural networks by decentralized feedback control. *International Journal of Neural Systems*, *17*(2), 115–122.

Chen, Y. Y., & Young, K. Y. (2007). An SOM-based algorithm for optimization with dynamic weight updating. *International Journal of Neural Systems*, *17*(3), 171–181.

Chiappalone, M., Vato, A., Berdondini, L., Koudelka, M., & Martinoia, S. (2007). Network dynamics and synchronous activity in cultured cortical neurons. *International Journal of Neural Systems*, *17*(2), 87–103.

Christodoulou, M. A., & Kontogeorgou, C. (2008). Collision Avoidance in commercial aircraft free fight, via neural networks and non-linear programming. *International Journal of Neural Systems*, *18*(5), 371–387.

Cichocki, A., & Zdunek, R. (2007). Multilayer nonnegative matrix factorization using projected gradient approaches. *International Journal of Neural Systems*, *17*(6), 431–446.

Cowell, R., Lauritzen, S., Spiegelhater, D., & David, P. (2003). *Probabilistic networks and expert systems*. New York, NY: Springer.

Cutsuridis, V. (2007). Does abnormal spinal reciprocal inhibition lead to co-contraction of antagonist motor units? Preliminary simulation results. *International Journal of Neural Systems*, *17*(4), 319–327.

Cyganek, B. (2007). Circular road signs recognition with soft classifiers. *Integrated Computer-Aided Engineering*, *14*(4), 323–343.

Cyganek, B. (2008). Color image segmentation with support vector machines: Applications to road signs detection. *International Journal of Neural Systems*, *18*(4), 339–345.

Fyfe, C., Barbakh, W., Ooi, W. C., & Ko, H. (2008). Topological mappings of video and audio data. *International Journal of Neural Systems*, *18*(6), 481–489.

Ganchev, T., Fakotakis, N., & Kokkinakis, G. (2002). Text-independent speaker verification based on probabilistic neural networks. In *Proceedings of the conference on acoustics*.

Gelman, A., Carlin, J., Stern, H., & Rubin, D. (2003). *Bayesian data analysis*. Boca Raton, FL: CRC Press.

Ghosh-Dastidar, S., & Adeli, H. (2003). Wavelet-clustering-neural network model for freeway incident detection. *Computer-Aided Civil and Infrastructure Engineering*, *18*(5), 325–338.

Ghosh-Dastidar, S., & Adeli, H. (2007). Improved spiking neural networks for EEG classification and epilepsy and seizure detection. *Integrated Computer-Aided Engineering*, *14*(3), 187–212.

Ghosh-Dastidar, S., Adeli, H., & Dadmehr, N. (2007). Mixed-band wavelet-chaos-neural network methodology for epilepsy and epileptic seizure detection. *IEEE Transactions on Biomedical Engineering*, *54*(9), 1545–1551.

Gil-Pita, R., & Yao, X. (2008). Evolving edited k-nearest neighbour classifiers. *International Journal of Neural Systems*, *18*(6), 459–467.

Goh, A. (2002). Probabilistic neural network for evaluating seismic liquefaction potential. *Canadian Geotechnical Journal*, *39*(1), 219–232.

Gopych, P. (2008). Biologically plausible BSDT recognition of complex images: The case of human faces. *International Journal of Neural Systems*, *18*(6), 527–545.

Guanella, A., Kiper, D., & Verschure, P. (2007). A model of grid cells based on a twisted torus topology. *International Journal of Neural Systems*, *17*(4), 231–240.

Gutenberg, B., & Richter, C. F. (1956). Earthquake magnitude, intensity, energy and acceleration. *Bulletin of the Seismological Society of America*, *46*(1), 105–146.

Holmes, E., Nicholson, J., & Tranter, G. (2001). Metabonomic characterization of genetic variations in toxicological and metabolic responses using probabilistic neural networks. *Chemical Research in Toxicology*, *14*(2), 182–191.

Huang, C., & Liao, W. (2004). Application of probabilistic neural networks to the class prediction of leukemia and embryonal tumor of the central nervous system. *Neural Processing Letters*, *19*(3), 211–226.

Huang, P., & Xu, Y. (2007). SVM-based learning control of space robotic capturing operation. *International Journal of Neural Systems*, *17*(6), 467–477.

Hung, S. L, & Adeli, H. (1993). Parallel backpropagation learning algorithms on cray Y-MP8/864 supercomputer. *Neurocomputing*, *5*(6), 287–302.

Hung, S. L., & Adeli, H. (1994a). A parallel genetic/neural network learning algorithm for MIMD shared memory machines. *IEEE Transactions on Neural Networks*, *5*(6), 900–909.

Hung, S. L., & Adeli, H. (1994b). Object-oriented back propagation and its application to structural design. *Neurocomputing*, *6*(1), 45–55.

Huynh, H. T., Won, Y., & Kim, J. J. (2008). An improvement of extreme learning machine for compact single- hidden-layer feedforward neural networks. *International Journal of Neural Systems*, *18*(5), 433–441.

Iglesias, J., & Villa, A. E. P. (2008). Emergence of preferred firing sequences in large spiking neural networks during simulated neuronal development. *International Journal of Neural Systems*, *18*(4), 267–277.

Jiang, X., & Adeli, H. (2005a). Dynamic wavelet neural network for nonlinear identification of highrise buildings. *Computer-Aided Civil and Infrastructure Engineering*, *20*(5), 316–330.

Jiang, X., & Adeli, H. (2005b). Dynamic wavelet neural network model for traffic flow forecasting. *Journal of Transportation Engineering, ASCE*, *131*(10), 771–779.

Jiang, X., & Adeli, H. (2007). Pseudospectra, MUSIC, and dynamic wavelet neural network for damage detection of highrise buildings. *International Journal for Numerical Methods in Engineering*, *71*(5), 606–629.

Jiang, X., & Adeli, H. (2008a). Dynamic fuzzy wavelet neuroemulator for nonlinear control of irregular highrise building structures. *International Journal for Numerical Methods in Engineering*, *74*(7), 1045–1066.

Jiang, X., & Adeli, H. (2008b). Neuro-genetic algorithm for nonlinear active control of highrise buildings. *International Journal for Numerical Methods in Engineering*, *75*(8), 770–786.

Jorgensen, T. D., Haynes, B. P., & Norlund, C. C. F. (2008). Pruning Artificial Neural Networks using neural complexity measures. *International Journal of Neural Systems*, *18*(5), 389–403.

Karim, A., & Adeli, H. (2003). Radial basis function neural network for work zone capacity and queue estimation. *Journal of Transportation Engineering, ASCE*, *129*(5), 494–502.

Khashman, A. (2008). Blood cell identification using a simple neural network. *International Journal of Neural Systems*, *18*(5), 453–458.

Khashman, A., & Sekeroglu, B. (2008). Document image binarisation using a supervised neural network. *International Journal of Neural Systems*, *18*(5), 405–418.

Kim, D., Lee, J., & Chang, S. (2005). Application of probabilistic neural networks for prediction of concrete strength. *Journal of Materials in Civil Engineering, ASCE*, *17*(3), 353–362.

Kimura, T., & Ikeguchi, T. (2007). An optimum strategy for dynamic and stochastic packet routing problems by chaotic neurodynamics. *Integrated Computer-Aided Engineering*, *14*(4), 307–322.

Kramer, M. A., Chang, F. L., Cohen, M. E., Hudson, D., & Szeri, A. J. (2007). Synchronization measures of the scalp EEG can discriminate healthy from alzheimers subjects. *International Journal of Neural Systems*, *17*(2), 61–69.

Lan, L., & Zhu, K. Y. (2007). Biomedical stability analysis of the lambda*symbol*-model controlling one joint. *International Journal of Neural Systems*, *17*(3), 193–206.

Lee, H., Cichocki, A., & Choi, S. (2007). Nonnegative matrix factorization for motor imagery EEG classification. *International Journal of Neural Systems*, *17*(4), 305–317.

Lian, H. C., & Lu, B. L. (2007). Multi-view gender classification using multi-resolution local binary patterns and support vector machines. *International Journal of Neural Systems*, *17*(6), 479–487.

Liu, H., Wang, X., & Qiang, W. (2007). A fast method for implicit surface reconstruction based on radial basis functions network from 3D scattered points. *International Journal of Neural Systems*, *17*(6), 459–465.

Liu, M., & Zhang, S. (2008). An LMI approach to design H_∞ controllers for discrete-time nonlinear systems based on unified models. *International Journal of Neural Systems*, *18*(5), 443–452.

Mayorga, R. V., & Arriaga, M. (2007). Non-linear global optimization via parameterization and inverse function approximation: An artificial neural networks approach. *International Journal of Neural Systems*, *17*(5), 353–368.

Mayorga, R. V., & Carrera, J. (2007). A radial basis function network approach for the computational of inverse continuous time variant functions. *International Journal of Neural Systems*, *17*(3), 149–160.

McGuire, J., Boettcher, M., & Jordan, T. (2005). Foreshock sequences and short-term predictability in the east pacific rise transform faults. *Nature*, *434*(1), 457–461.

Mersch, B., Glasmachers, T., Meinicke, P., & Igel, C. (2007). Evolutionary optimization of sequence kernels for detection of bacterial gene starts. *International Journal of Neural Systems*, *17*(5), 369–381.

Mohan, V., & Morasso, P. (2007). Towards reasoning and coordinating actions in the mental space. *International Journal of Neural Systems*, *17*(4), 329–341.

Montina, A., Mendoza, C., & Arecchi, F. T. (2007). Role of refractory period in homoclinic models of neural synchronization. *International Journal of Neural Systems*, *17*(2), 79–86.

Mosavi, M. R. (2007). GPS receivers timing data processing using neural networks: Optimal estimation and errors modeling. *International Journal of Neural Systems*, *17*(5), 383–393.

Nemissi, M., Seridi, H., & Akdag, H. (2008). The labeled systems of multiple neural networks. *International Journal of Neural Systems*, *18*(4), 321–330.

Neumann, D., Eckmiller, R., & Baruth, O. (2007). Combination of biometric data and learning algorithms for both generation and application of a secure communication link. *Integrated Computer-Aided Engineering*, *14*(4), 345–352.

Ni, H., & Yin, H. (2008). Self-organizing mixture autoregressive model for non-stationary time series prediction. *International Journal of Neural Systems*, *18*(6), 469–480.

Osterhage, H., Mormann, F., Wagner, T., & Lehnertz, K. (2007). Measuring the directionality of coupling: Phase versus state space dynamics and application to EEG time series. *International Journal of Neural Systems*, *17*(3), 139–148.

Panakkat, A., & Adeli, H. (2007). Neural network models for earthquake magnitude prediction using multiple seismicity indicators. *International Journal of Neural Systems*, *17*(1), 13–33.

Panakkat, A., & Adeli, H. (2008). Recent efforts in earthquake prediction (1990–2007). *National Hazard Review*, *9*(2), 70–80.

Pande, A., & Abdel-Aty, M. (2008). A Computing approach using probabilistic neural networks for instantaneous appraisal of rear-end crash risk. *Computer-Aided Civil and Infrastructure Engineering*, *23*(7), 549–559.

Park, H. S., & Adeli, H. (1995). A neural dynamics model for structural optimization — Application to plastic design of structures. *Computers and Structures*, *57*(3), 391–399.

Park, H. S., & Adeli, H. (1997). Distributed neural dynamics algorithms for optimization of large steel structures. *Journal of Structural Engineering, ASCE*, *123*(7), 880–888.

Postnov, D. E., Ryazanova, L. S., Zhirin, R. A., Mosekilde, E., & Sosnovtseva, O. V. (2007). Noise controlled synchronization in potassium coupled neural networks. *International Journal of Neural Systems*, *17*(2), 105–113.

Raftopoulos, K., Papadakis, N., Ntalianis, K., & Kollias, S. (2007). Shape-based invariant classification of gray scale images. *Integrated Computer-Aided Engineering*, *14*(4), 365–378.

Rigatos, G. G. (2008). Adaptive fuzzy control with output feedback for H-infinity tracking of SISI nonlinear systems. *International Journal of Neural Systems*, *18*(4), 305–320.

Rigatos, G. G., & Tzafestas, S. G. (2007). Neurodynamics and attractors in quantum associative memories. *Integrated Computer-Aided Engineering*, *14*(3), 224–242.

Rodríguez-Sánchez, A. J., Simine, E., & Tsotsos, J. K. (2007). Attention and visual search. *International Journal of Neural Systems*, *17*(4), 275–288.

Ruiz-Pinales, J., Jaime-Rivas, R., Lecolinet, E., & Castro-Bleda, M. J. (2008). Cursive word recognition based on interactive activation and early visual processing models. *International Journal of Neural Systems*, *18*(5), 419–431.

Sabourin, C., Madani, K., & Bruneau, O. (2007). Autonomous biped gait pattern based on fuzzy-CMAC neural networks. *Integrated Computer-Aided Engineering*, *14*(2), 173–186.

Samant, A., & Adeli, H. (2001). Enhancing neural network incident detection algorithms using wavelets. *Computer-Aided Civil and Infrastructure Engineering*, *16*(4), 239–245.

Schaefer, A. M., & Zimmermann, H. G. (2007). Recurrent neural networks are universal approximators. *International Journal of Neural Systems*, *17*(4), 253–263.

Schneider, N. C., & Graupe, D. (2008). A modified lamstar neural network and its applications. *International Journal of Neural Systems*, *18*(4), 331–337.

Senouci, A. B., & Adeli, H. (2001). Resource scheduling using neural dynamics model of Adeli and Park. *Journal of Construction Engineering and Management, ASCE*, *127*(1), 28–34.

Sirca, G., & Adeli, H. (2001). Neural network model for uplift load capacity of metal roof panels. *Journal of Structural Engineering*, *127*(11), 1276–1285.

Specht, D. (1990). Probabilistic neural networks. *Neural networks*, *3*(1), 110–118.

Stathopoulos, A., Dimitriou, L., & Tsekeris, T. (2008). Fuzzy modeling approach for combined forecasting of urban traffic flow. *Computer-Aided Civil and Infrastructure Engineering*, *23*(7), 521–535.

Tashakori, A. R., & Adeli, H. (2002). Optimum design of cold-formed steel space structures using neural dynamic model. *Journal of Constructional Steel Research*, *58*(12), 1545–1566.

Taylor, N. R., Panchev, C., Hartley, M., Kasderidis, S., & Taylor, J. G. (2007). Occlusion, attention and object representations. *Integrated Computer-Aided Engineering*, *14*(4), 283–306.

Terai, A., & Nakagawa, M. (2007). A neural network model of metaphor understanding with dynamic interaction based on a statistical language analysis. *International Journal of Neural Systems*, *17*(4), 265–274.

Tsapatsoulis, N., Rapantzikos, K., & Pattichis, C. (2007). An embedded saliency map estimator scheme: Application to video encoding. *International Journal of Neural Systems*, *17*(4), 289–304.

Wang, L., Jiang, M., Lu, Y., Sun, M., & Noe, F. (2007). A comparative study of clustering methods for molecular data. *International Journal of Neural Systems*, *17*(6), 447–458.

Vatsa, M., Singh, R., & Noore, A. (2007). Integrating image quality in 2ν-SVM biometric match score fusion. *International Journal of Neural Systems*, *17*(5), 343–351.

Villaverde, I., Grana, M., & d'Anjou, A. (2007). Morphological neural networks and vision based simultaneous localization and mapping. *Integrated Computer-Aided Engineering*, *14*(4), 355–363.

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2007). Spatio-temporal short-term urban traffic flow forecasting using genetically-optimized modular networks. *Computer-Aided Civil and Infrastructure Engineering*, *22*(5), 317–325.

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2008). Temporal evolution of short-term urban traffic flow: A non-linear dynamics approach. *Computer-Aided Civil and Infrastructure Engineering*, *23*(7), 536–548.

Wasserman, P. (1993). *Advanced methods in neural networks*. New York, NY: Van Nostrand Reinhold.

Wersing, H., Kirstein, S., Götting, M., Brandl, H., Dunn, M., Mikhailova, I., et al. (2007). Online learning of objects in a biologically motivated visual architecture. *International Journal of Neural Systems*, *17*(4), 219–230.

Working Group on California Earthquake Probabilities, (2004). *Earthquake probabilities in the San Fransisco bay region, United States geological survey open-file report*. 03-214.

Yau, W. C., Kumar, D. K., & Arjunan, S. P. (2007). Visual recognition of speech consonants using facial movement features. *Integrated Computer-Aided Engineering*, *14*(1), 49–61.