A Novel Machine Learning Model for Estimation of Sale Prices of Real Estate Units

Mohammad Hossein Rafiei, S.M.ASCE[1]; and Hojjat Adeli, Dist.M.ASCE[2]

Abstract: Predicting the price of housing is of paramount importance for near-term economic forecasting of any nation. This paper presents a novel and comprehensive model for estimating the price of new housing in any given city at the design phase or beginning of the construction through ingenious integration of a deep belief restricted Boltzmann machine and a unique nonmating genetic algorithm. The model can be used by construction companies to gauge the sale market before they start a new construction and consider to *build or not to build*. An effective data structure is presented that takes into account a large number of economic variables/indices. The model incorporates time-dependent and seasonal variations of the variables. Clever stratagems have been developed to overcome the *dimensionality curse* and make the solution of the problem amenable on standard workstations. A case study is presented to demonstrate the effectiveness and accuracy of the model. **DOI: 10.1061/(ASCE)CO.1943-7862.0001047.** *© 2015 American Society of Civil Engineers.*

Author keywords: Quantitative methods.

Introduction

It is widely known that the precursor to the recent global economic crisis was the bust in the real estate market. Predicting the price of housing is of paramount importance for near-term economic forecasting of any nation. In an uncertain economic climate, construction companies are confronted with a daunting question: to *build or not to build*. Little research is reported in the construction literature on the price of housing. To find research on the price of housing, one needs to search mostly the business, economic, finance, and real estate journals. A number of researchers have attempted to describe variables affecting the real state price dynamics or movements. To study how information about local economic conditions affect the price of housing, Favara and Song (2014) describe a user-cost mathematical model based on the assumption that the housing supply is inelastic and demand variations are stochastic. Borowiecki (2009) notes that construction costs and population growth are among significant factors affecting housing prices in Switzerland, while Das et al. (2009) present a Bayesian (Spackova and Straub 2013; Huang et al. 2014) autoregressive model to predict the annualized real estate price growth rate for the small to large housing market in South Africa considering the following factors: income, interest rates, construction costs, labor market variables, stock prices, industrial production, consumer confidence index, and factors representing the global economy. Égert and Mihaljek (2007) report that gross domestic product (GDP) per capita and interest rates have the highest influence on housing prices in eastern and central Europe and Organization for Economic Cooperation and Development (OECD) countries.

Housing prices, dynamics, and performance in the United States have been studied using several different statistical and pattern recognition approaches such as artificial neural networks (ANNs) (Dai and Wang 2014; Story and Fry 2014; Butcher et al. 2014). As an example, Khalafallah (2008) predicts the short-term (3 months in advance) housing market performance in Orlando, Florida, defined as the average sale price over average asking price. The author applies a simple backpropagation (BP) neural network (Adeli and Hung 1995; Siddique and Adeli 2013) using eight parameters as inputs: time in years and months, the average interest rate, the percentage change in sales volume compared to the previous year, percentage change in median house price compared to the previous year, average days a house has been on the market, the volume of inventory, and the inventory month's supply. In a similar study, using housing price data sets in Turkey, Selim (2009) compares the performance of the BP neural networks and regression methods to estimate future housing prices and reports better performance for the former.

The aforementioned neural network models are based on the simple BP neural network and include a limited number of factors. It is well known that BP requires thousands or millions of iterations for convergence depending on the complexity of the problem and cannot model complicated pattern recognition problems effectively in a reasonable amount of computing time (Hung and Adeli 1993, 1994; Adeli and Park 1998; Adeli 2001).

This paper presents a novel and comprehensive model for estimating the price of new housing in any given city at the design phase or at beginning of the construction incorporating a large number of factors through ingenious integration of a more powerful and recent machine learning model known as deep restricted Boltzmann machine (DRBM) and a nonmating genetic algorithm (Adeli and Sarma 2006; Pedrino et al. 2013; Jia et al. 2014; Chow 2014). The model can be used by construction companies to gauge the sale market *before* they start a new construction and consider to *build or not to build*. In the next section, first the data structure of the real estate sale price prediction model is presented. It is shown that the total number of input variables can be large, in the order of

[1]Ph.D. Student, Dept. of Civil, Environmental and Geodetic Engineering, Ohio State Univ., 470 Hitchcock Hall, 2070 Neil Ave., Columbus, OH 43220. E-mail: rafiei.4@buckeyemail.osu.edu

[2]Professor, Dept. of Civil, Environmental and Geodetic Engineering, Ohio State Univ., 470 Hitchcock Hall, 2070 Neil Ave., Columbus, OH 43220 (corresponding author). E-mail: adeli.1@osu.edu

Note. This manuscript was submitted on November 24, 2014; approved on June 18, 2015; published online on August 10, 2015. Discussion period open until January 10, 2016; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Construction Engineering and Management***, © ASCE, ISSN 0733-9364/04015066(10)/$25.00.

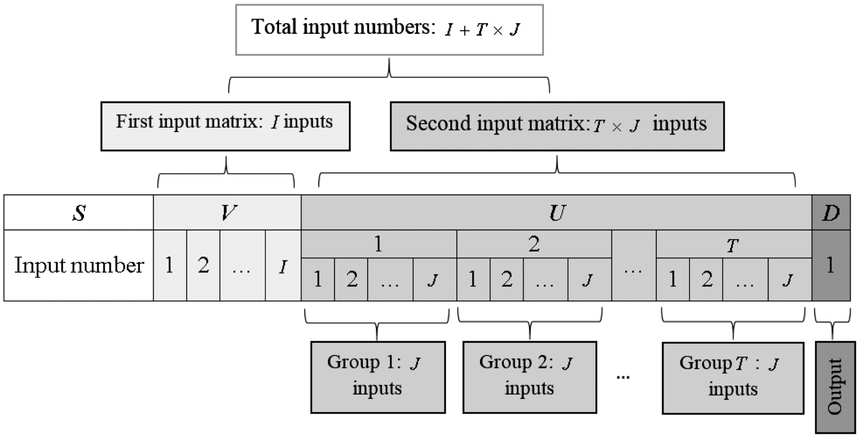
Total input numbers: $I + T \times J$
First input matrix: I inputs
Second input matrix: $T \times J$ inputs
S
V
U
D
Input number
1
2
…
I
1
2
…
T
1
2
…
J
1
Group 1: J inputs
Group 2: J inputs
…
Group T : J inputs
Output

Fig. 1. General data structure of the model for a sample real estate data (a single-family house, a residential or an office condominium)

100 or more, in the complex prediction problem. Next, DRBM is described briefly. Then, clever optimization strategies are presented to reduce the number of inputs followed by a case study. The paper ends with conclusions.

Data Structures for the Real Estate Sale Price Prediction Model

The general data structure of the model for sample real estate data (a single-family house, or a residential or office condominium) is shown in Fig. 1. There are two sets of inputs and one output. The first set of inputs is denoted by the row matrix $\boldsymbol{V}$ with I elements representing the physical and financial properties of the real estate unit. The second set of inputs denoted by $\boldsymbol{U}$ consists of T subgroups representing T periods prior to the start of the construction. The period can be a quarter, a month, or a week depending on the resolution of the available data. Each subgroup has J inputs representing J key economic variables affecting the price of the unit permanently. The output, the unit price at the time of completion of the project, is represented by D. Types and number of inputs in each set vary for different types of units and also from region to region for any given type of real estate unit such as a residential or office multistory building or condo.

Tables 1 and 2 present sample representative lists of input set one and inputs in each subgroup of second input set, for a single-family house or a residential condominium, respectively. Some of these parameters are for a preselected base year, which is necessary because prices/indices vary from year to year due to inflation, deflation, or other economic conditions. The effects of these parameters on the housing prices have been noted by local organizations and in several papers (Borowiecki 2009; Égert and Mihaljek 2007; Khalafallah 2008; Das et al. 2009; Rapach and Strauss 2006). For instance, in general there is an inverse relationship between the interest rates and the price of housing.

The total number of input variables is $I + T \times J$, which can be large, in the order of 100 or more. Such a large number of input variables requires a very large number of training samples (in the thousands) that may make the task of data collection impractical if not impossible and the training of the prediction model computationally very intensive. This is known as the *dimensionality curse* in the neural network literature (Adeli and Wu 1998; Hinton and Salakhutdinov 2006).

Deep Belief Restricted Boltzmann Machine (DRBM)

Machine learning is a core technology in the development of intelligent systems and has been a focus of substantial research in the past two decades (Adeli and Hung 1995; Siddique and Adeli 2013). Among the recent developments are fuzzy neural network algorithms (Boutalis et al. 2013; Hsu 2013; Rigatos 2013; Zhang and Ge 2013; Vlahogianni and Karlaftis 2013; Kwon et al. 2014), wavelet neural network (Kodogiannis et al. 2013), hybrid evolutionary models, swarm intelligence (Shafahi and Bagherian 2013; Forcael et al. 2014; Zeng et al. 2014; Wu et al. 2014), spiking neural networks (Friedrich et al. 2014; Rosselló et al. 2014; Zhang et al. 2014a; Shapero et al. 2014), and DRBM based on integration of the earlier restricted Boltzmann machine (RBM) (Smolensky 1986) and more recent deep belief concept (Hinton et al. 2006; Hinton 2007).

Hinton and Salakhutdinov (2006) proposed DRBM made of several layers of RBMs, as shown schematically in Fig. 2(b), where N is the number of layers of RBM. The hidden layer of each RBM with M_i terms ($i \in [1,2, \ldots, M]$) is the visible layer of the next RBM. The goal in RBM is to identify the best combination of weights and biases to decrease the energy of the visible vector and increase the energy of a corresponding so-called *fantasy vector*, the vector that is created by considering hidden units as visible layer

Table 1. Sample Physical and Financial Properties for Residential Condos (First Input Set, $\boldsymbol{V}$)

Input number	Description	Unit
1	Project locality defined in terms of zip codes	N/A
2	Total floor area of the building	m^2
3	Lot area	m^2
4	Preliminary estimated construction cost per m^2 based on the prices at the beginning of the project	$\$/m^2$
5	Equivalent preliminary estimated construction cost per m^2 based on the prices at the beginning of the project in a selected base year (e.g., 2004)	$\$/m^2$
6	Duration of construction	Quarter, month, or week
7	Price of the unit at the beginning of the project per m^2	$\$/m^2$

Table 2. Sample Economic Variables/Indices for Residential Condos (Second Input Set, ***U***)

Second input number	Descriptions
1	Number of building permits issued by the city/municipality
2	Building services index[a] for a preselected base year (e.g., 2004)
3	Wholesale price index (WPI)[b] of building materials for the base year
4	Total floor areas of building permits issued by the city/municipality
5	Cumulative liquidity[c] (millions of dollars)
6	Private sector investment in new buildings in the city (millions of dollars)
7	Land price index in the city for the base year
8	Number of loans extended by banks in a quarter
9	Amount of loans extended by banks in a quarter (millions of dollars)
10	Interest rate for loan in the quarter
11	Average construction cost of buildings by private sector at the time of completion of construction per m^2 in the city (millions of dollars)
12	Average of construction cost of buildings by private sector at the beginning of the construction per m^2 in the city (millions of dollars)
13	Official exchange rate with respect to dollars
14	Nonofficial (street market) exchange rate with respect to dollars (used only in countries with controlled currencies)
15	Consumer price index (CPI)[d] in the base year
16	CPI of housing, water, fuel, and power in the base year
17	Stock market index[e]
18	Population of the city
19	Gold price per ounce (dollars)

[a] Building services index presents the total subcontractors amount of contracts (such as worker wages, painting, plaster work, pipe installation, etc., during the selected duration of building construction such as day or month).

[b] WPI presents the cost of a determined basket of food or services in some countries to detect inflation or deflation. In the United States, the producer price index (PPI) may be considered instead of WPI (United States Department of Labor 2015).

[c] Liquidity represents how rapidly different types of assets can be changed to cash.

[d] CPI is an indicator representing the change in prices of a determined basket of goods and services purchased for consumption by urban households over a specific time period (United States Department of Labor 2015).

[e] An indicator representing the payback condition of investment in stock market.

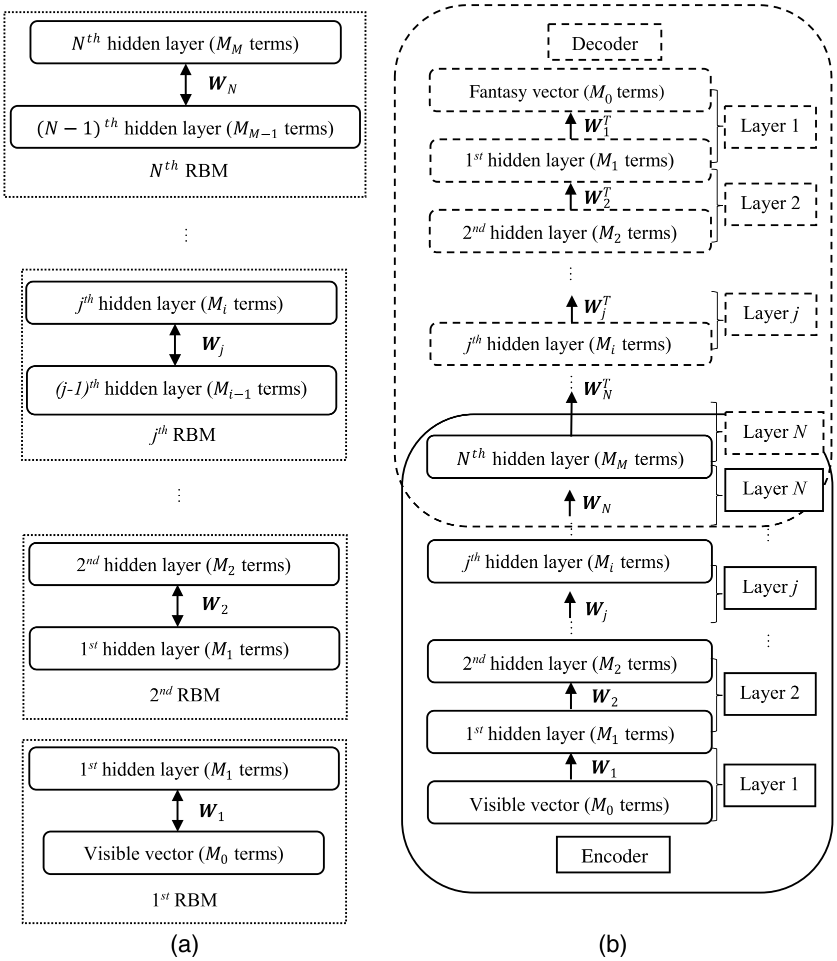
N^{th} hidden layer (M_M terms)
$\boldsymbol{W}_N$
$(N-1)^{th}$ hidden layer (M_{M-1} terms)
N^{th} RBM
j^{th} hidden layer (M_i terms)
$\boldsymbol{W}_j$
$(j\text{-}1)^{th}$ hidden layer (M_{i-1} terms)
j^{th} RBM
2^{nd} hidden layer (M_2 terms)
$\boldsymbol{W}_2$
1^{st} hidden layer (M_1 terms)
2^{nd} RBM
1^{st} hidden layer (M_1 terms)
$\boldsymbol{W}_1$
Visible vector (M_0 terms)
1^{st} RBM
(a)
Decoder
Fantasy vector (M_0 terms)
$\boldsymbol{W}_1^T$
Layer 1
1^{st} hidden layer (M_1 terms)
$\boldsymbol{W}_2^T$
Layer 2
2^{nd} hidden layer (M_2 terms)
$\boldsymbol{W}_j^T$
Layer j
j^{th} hidden layer (M_i terms)
$\boldsymbol{W}_N^T$
Layer N
N^{th} hidden layer (M_M terms)
$\boldsymbol{W}_N$
Layer N
j^{th} hidden layer (M_i terms)
$\boldsymbol{W}_j$
Layer j
2^{nd} hidden layer (M_2 terms)
$\boldsymbol{W}_2$
Layer 2
1^{st} hidden layer (M_1 terms)
$\boldsymbol{W}_1$
Layer 1
Visible vector (M_0 terms)
Encoder
(b)

Fig. 2. Pretraining and DRBM processes

and using the transpose of the weight vector. Using the general concept of thermal equilibrium in physics, the energy of a configuration of visible and hidden units is defined as follows (Smolensky 1986), which is similar to that of a Hopfield network (Hopfield 1982):

$$E(\boldsymbol{X},\boldsymbol{H}) = -\sum_{i}^{M}\Psi_i X_i - \sum_{j}^{N}\Phi_j H_j - \sum_{i}^{M}\sum_{j}^{N} X_i H_j W_{ij}$$
$$= -\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{X} - \boldsymbol{\Psi}^T\boldsymbol{X} - \boldsymbol{\Phi}\boldsymbol{H} \qquad (1)$$

where $\boldsymbol{X}$ and $\boldsymbol{H}$ are the binary row vectors of visible and hidden nodes with M and N terms, respectively, Ψ_i and Φ_j are the bias terms corresponding to X_i and H_i, respectively, and W_{ij} is the weight between X_i and H_i.

In Fig. 2(a), the DRBM consists of two parts, an encoder and decoder. The last hidden layer of the encoder is the visible layer of the decoder. The encoder is the overarching guider of the system, the *big picture guy*, intended to perform the high functions of the brain, and mathematically the dimensionality reducer. The decoder is the *details guy*, remembering the higher dimensional aspects. Each hidden layer in the encoder is a feature detector. The decoder is applied to retrieve the so-called fantasy vector using the transpose of the corresponding weight matrices in a way similar to the encoder. It is claimed the larger the number of layers in the DRBM, the better the approximation of the training data.

To improve the network training speed, Hinton and Salakhutdinov (2006) proposed a layer-by-layer learning technique known as *pre-training*, depicted schematically in Fig. 2(a). Once the weights for one layer are adjusted using gradient decent methods (Adeli 1994; Adeli and Soegiarso 1999), the hidden units are applied as the visible units of the next layer, and this process is repeated layer by layer. The adjusted weights are now ready to be applied to DRBM [Fig. 2(b)].

Optimization Strategies to Reduce the Number of Inputs and Overcome the Dimensionality Curse

Reducing Dimensionality and NGA Operation

In this research, to overcome the dimensionality curse, clever strategies are developed to reduce the number of inputs in $\boldsymbol{U}$ with $T \times J$ terms to a much smaller preselected number K. The reduced row vector with K terms denoted by $\boldsymbol{B}$ is selected such that it includes the most influential set of inputs thus yielding the most accurate results. Fig. 3 shows the reduced input data structure. The number of possible combinations of F objects out of E objects is equal to (Mathwords 2012)

$$W(E,F) = \frac{E!}{(E-F)! \times F!} \qquad (2)$$

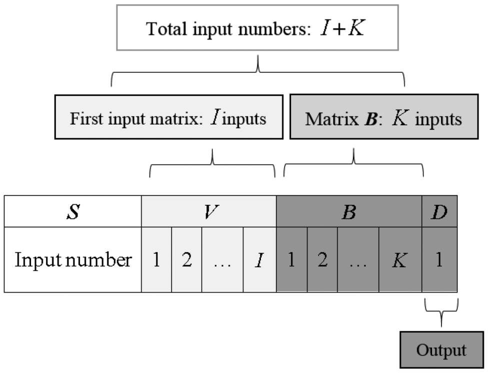

Fig. 3. Reduced input data structure

Hence, the number of possible combinations of K inputs out of $T \times J$ inputs is equal to

$$W(T \times J, K) = \frac{(T \times J)!}{(T \times J - K)! \times K!} \qquad (3)$$

As an example, for the data presented in Table 2, $J = 19$ and using $T = 5$ (say, for 5 quarters), $T \times J = 95$. When the number of inputs is decreased to a smaller and more manageable size, say, $K = 20$, the number of possible combinations W is about 1.71×10^{20}, an astronomical number. In this research, a unique nonmating genetic algorithm (NGA) (Hejazi et al. 2013; Fuggini et al. 2013) is developed to determine the most influential set of inputs yielding the most accurate results.

The NGA operation and optimization strategies to reduce the number of inputs is shown schematically in Fig. 4. First, all available data are divided into training and testing data sets randomly using a preselected ratio of test-to-training data (RTT) (for example, 10%) (step 1 in Fig. 4). Next, the first generation of chromosomes in NGA is created randomly (step 2 in Fig. 4), say 20 chromosomes in the population. Each chromosome in the population is a binary number with $T \times J$ bits. It includes K ones and $T \times J - K$ zeroes (Fig. 4) selected randomly. Chromosomes with such a structure are referred to as *admissible chromosomes* in this paper. In an admissible chromosome, when a bit positioned as the ith bit among $T \times J$ bits has a value of 1, it indicates the selection of the ith input in $\boldsymbol{U}$ (Fig. 1), whether it is testing or training data, to be one of the K terms in $\boldsymbol{B}$ (Fig. 3). This process is called the *translation process* (step 3 in Fig. 4) and is performed for all training and testing samples. The translation process, based on the position of bits with a value of 1 in each chromosome of population, reduces the input data structure, for both testing and training data, to a lower dimensional vector as depicted in Fig. 3. Hence, each chromosome in the population represents training and testing data corresponding to the combination of 0 and 1 in the structure of that chromosome. Next, the DRBM is trained a number of times equal to the number of chromosomes in the selected population, each time using the training data set corresponding to that chromosome in the population (step 4 in Fig. 4). Hence, a DRBM is trained for each chromosome. Then, the testing data set of each chromosome is applied to the corresponding trained DRBM (step 5 in Fig. 4) and the minimum squared error (MSE) of the testing samples for each chromosome is computed (step 6 in Fig. 4). Steps 3–6 in Fig. 4 represent the NGA fitness function. The minimum of the MSEs (MMSE) is the fitness value of the NGA fitness function (step 7 in Fig. 4). Next, a new generation of the chromosomes is generated using a strategy to be discussed later in this section (steps 10–13 in Fig. 4). The aforementioned fitness function, a one-input one-output function, can be presented (Fig. 4) as follows:

$$Y = g(x) | x \in C \qquad (4)$$

where x = admissible chromosome; $g(x)$ = fitness function; Y = MSE of the admissible chromosome; and C = set of all admissible chromosomes. As mentioned earlier, admissible chromosomes contain K ones and $T \times J - K$ zeroes in their binary scales. Hence, they all have just K ones out of $T \times J$ bits. Therefore, the cardinality of the set C is $W(T \times J, K)$ (Eq. 2).

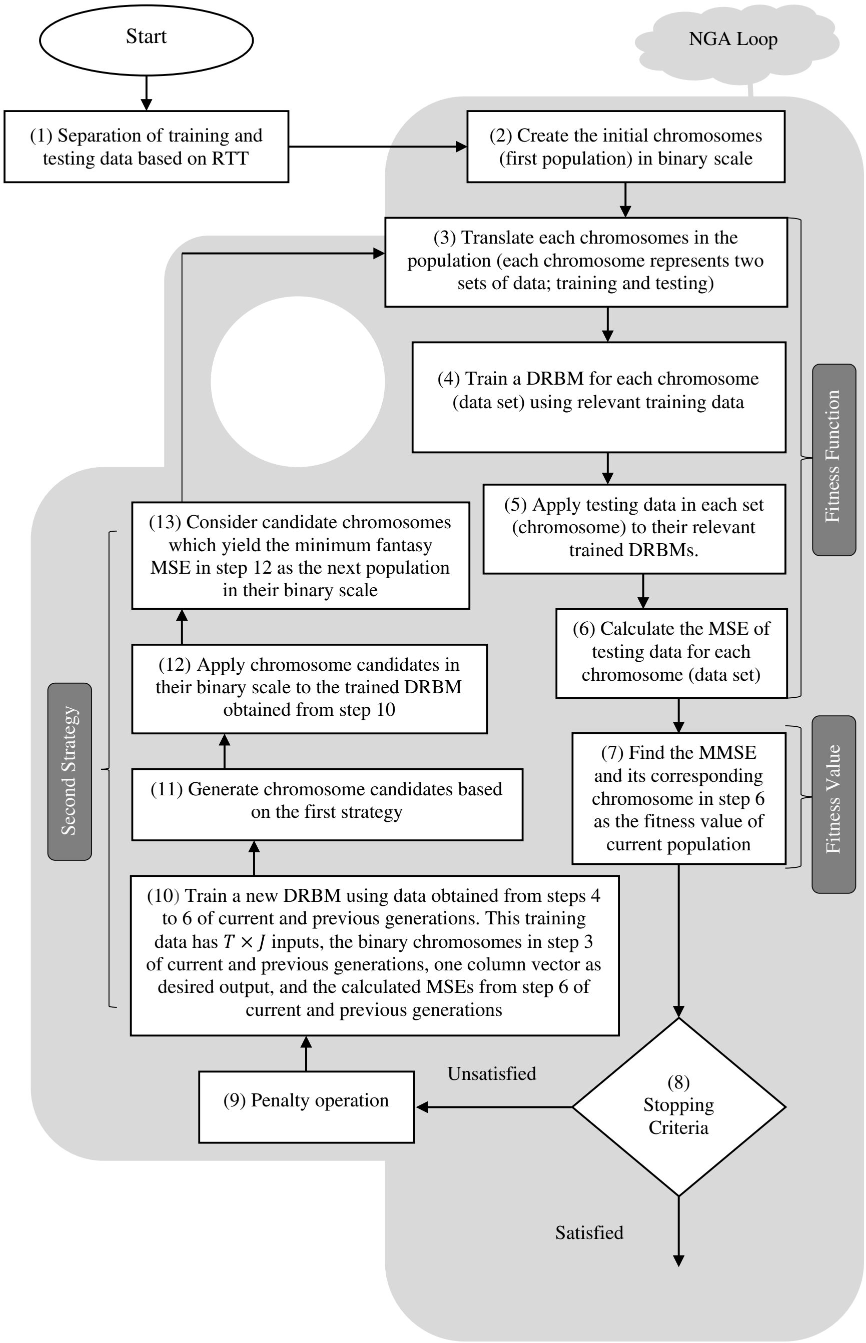
Start
NGA Loop
(1) Separation of training and testing data based on RTT
(2) Create the initial chromosomes (first population) in binary scale
(3) Translate each chromosomes in the population (each chromosome represents two sets of data; training and testing)
(4) Train a DRBM for each chromosome (data set) using relevant training data
(5) Apply testing data in each set (chromosome) to their relevant trained DRBMs.
(6) Calculate the MSE of testing data for each chromosome (data set)
(7) Find the MMSE and its corresponding chromosome in step 6 as the fitness value of current population
Fitness Function
Fitness Value
(8) Stopping Criteria
Unsatisfied
Satisfied
(9) Penalty operation
(10) Train a new DRBM using data obtained from steps 4 to 6 of current and previous generations. This training data has T × J inputs, the binary chromosomes in step 3 of current and previous generations, one column vector as desired output, and the calculated MSEs from step 6 of current and previous generations
(11) Generate chromosome candidates based on the first strategy
(12) Apply chromosome candidates in their binary scale to the trained DRBM obtained from step 10
(13) Consider candidate chromosomes which yield the minimum fantasy MSE in step 12 as the next population in their binary scale
Second Strategy

Fig. 4. NGA operation and optimization strategies to reduce the number of inputs

Complexity of the Research Problem

Suppose the size of binary chromosomes is 95 ($T \times J = 95$) and $K = 20$ (Figs. 1 and 4). The smallest binary chromosome is $00 \ldots 0$ (all bits are 0) and the largest binary chromosome is $11 \ldots 1$ (all bits are 1), but because the total number of ones in each chromosome should remain $K = 20$, the smallest and largest binary chromosomes are changed to $00 \ldots 011 \ldots 1$ (the last 20 bits are all ones and the rest are zeroes) with a decimal equivalent (DE) of 1048575 and $11 \ldots 100 \ldots 0$ (the first 20 bits are all ones and the rest are all zeroes) with DE of about 3.9614×10^{28}, respectively. Chromosomes with DE values between these smallest and largest values are admissible for the translation process if the number of ones in their binary scale is $K = 20$. Table 3 shows a number of these admissible chromosomes in binary scale and their DE values. According to Eq. (2), the number of combinations of 20 ones in 95 bits is around 1.71×10^{20}. Consequently, the 1.71×10^{20} number of admissible chromosomes are distributed between minimum chromosome (1,048,575) and maximum chromosome (3.9614×10^{28}) with variable interval because the number of ones should remain $K = 20$. Table 3 shows these chromosomes in their binary and decimal scales. For this example, since there are 1.71×10^{20} admissible chromosomes (the cardinality of set C), according to Eq. (2), there are 1.71×10^{20} possible MSEs (Y). This means a very large population size as well as a very large number of generations is needed to increase the probability of finding the chromosome with the smallest MSE. In addition, the iteration history of the fitness function value may become noisy.

Table 3. Admissible Chromosome Binary Scales versus Their DE Values ($T \times J = 95$ and $K = 20$)

Chromosome number	Admissible chromosome binary scale (the number of ones in all chromosomes binary scale remains 20)	DE value
1	00000000000000000000000000000000000000000000000000000000000000000000000000011111111111111111111	1,048,575
2	00000000000000000000000000000000000000000000000000000000000000000000000000101111111111111111111	1,572,863
3	00000000000000000000000000000000000000000000000000000000000000000000000000110111111111111111111	1,835,007
4	00000000000000000000000000000000000000000000000000000000000000000000000000111011111111111111111	1,966,079
5	00000000000000000000000000000000000000000000000000000000000000000000000000111101111111111111111	2,031,615
$\sim 1.71 \times 10^{20}$	11111111111111111110100000000000000000000000000000000000000000000000000000000000000000000000000	$\sim 3.9614 \times 10^{28}$
$\sim 1.71 \times 10^{20}$	11111111111111111111000000000000000000000000000000000000000000000000000000000000000000000000000	$\sim 3.9614 \times 10^{28}$

Strategies to Determine the Best Chromosome

In a typical genetic algorithm, the crossover and mutation operations are used to generate a new population in every iteration using the roulette wheel method (Szeto et al. 2014; Lin and Ku 2014; Zhu et al. 2014). Two strategies are developed in this research and used simultaneously to increase the chance of finding the best chromosome (the chromosome that yields the lowest MSE) and also making the methodology more efficient and tractable so that the optimization problem can be solved in a much fewer number of populations (in the order of 20) and much fewer iterations (in the order of 200).

The first strategy is to increase the selection chance of admissible chromosomes with higher decimal values, that is, with more ones in the first bits of their binary scale. The input variables in times closer to project's construction beginning dates (the inputs in the first subgroups of $\boldsymbol{U}$) are assumed to have more influence on the real estate market compared with those at later times. This strategy helps increase the probability of finding the chromosome with the lowest MSE. This assumption is based on the experience of the authors with the available data. The computational model presented in the paper allows reversal of this assumption if the data for another country or region indicates that. In other words, the computational model presented in the paper is general does not depend on this assumption.

In the optimization model presented in this research (Fig. 4), DRBM is used twice in each NGA loop, first to estimate the lowest MSE of the chromosomes in the population and second as a *population generator* to be used in the next generation. The latter is the second strategy used in the proposed model to increase the probability of finding the chromosome with the lowest MSE instead of traditional operations such as crossover and mutation.

In each generation, the population generator is trained using all binary chromosomes and their MSEs obtained from previous generations. Suppose the population size and number of generations are 20 and 250, respectively. In the first generation, 20 MSEs for 20 chromosomes (chromosomes in the first population) are calculated (step 6 in Fig. 4). The 20 binary chromosomes represent the training data inputs of the population generator. The 20 chromosomes' MSEs (step 6 in Fig. 4) represent the desired output of the population generator. The population generator is trained using this training data set (step 10 in Fig. 4). Once the population generator is trained, a number of new admissible chromosomes, for example, 4,000, are generated randomly based on the first strategy. These admissible chromosomes (step 11 in Fig. 4) are called *candidate chromosomes* as each of them has the capability of being selected for the next generation. Their outputs (fantasy MSEs) are estimated by the trained population generator (step 12 in Fig. 4). Chromosomes with smaller estimated fantasy MSEs will have a better chance of yielding lower real MSEs, that is, the MSEs revealed from the fitness function procedure in the NGA operation (Fig. 4), in the next generation. Therefore, the top 20 chromosomes with the lowest estimated fantasy MSEs are considered as the population for the next generation (step 13 in Fig. 4). If a sufficient amount of random access memory (RAM) is available, say around 64 gigabytes, the number of candidate chromosomes can be in the order millions. A higher number of candidate chromosomes increases the probability of finding the best chromosome because more chromosomes among 1.71×10^{20} [set C in Eq. (3)] are evaluated by the trained population generator.

In the next generation, steps 3 to 7 in Fig. 4 are repeated in the same way as the previous generation, but with a newly generated population. The training data set for the population generator (step 10 in Fig. 4) now has $20 + 20 = 40$ binary values as inputs and 40 MSEs as output (20 from the current generation and 20 from the previous generation). This number becomes $20 \times 250 = 5{,}000$ in the final 250th generation. These 5,000 chromosomes are called *survived chromosomes* as they are selected in different generations among many candidate chromosomes (step 13 in Fig. 4). In addition, during the NGA process, $4{,}000 \times 250 = 1{,}000{,}000$ possible chromosomes are evaluated using the population generator (steps 10 to 12 in Fig. 4). If the number of new possible chromosomes is chosen to be, say, four million in each generation, the total number of evaluated chromosomes by population generator will be $4 \times 10^6 \times 250 = 1 \times 10^9$ in the final 250th generation. This means many more chromosomes among 1.71×10^{20} (set C) are evaluated. Hence, the probability of finding the best chromosome is increased.

Using the second strategy, the NGA operation (Fig. 4) gradually learns about the chromosomes' environment and will identify the chromosomes with a higher chance to yield more accurate results, and eventually help increase the chance to find the best $\boldsymbol{B}$. Two issues to be decided in the NGA operation presented in Fig. 4 are the stopping criteria and the penalty operation. For the former, one can specify a maximum number of iterations and/or a minimum value for the MSE. The purpose of the latter is to eliminate or reduce the oscillation usually encountered in the NGA convergence for which a number of strategies, such as the elitist strategy, can be used.

Strategies to Prevent Statistical Biases

A two-step strategy is used to minimize statistical biases (1) in the NGA operation and (2) in DRBM. First, the NGA operation depicted in Fig. 4 finds the best chromosome among all generated chromosomes that yields the minimum MSE based on a specific RTT (step 1 in Fig. 4). To minimize any statistical bias in the training of the proposed model, the NGA operation is repeated 100 times, for a number of different RTTs, say 5 (10, 20, 30, 40, and 50%), each time using a different randomly selected sets of training and testing samples. Hence, 5 sets of the 100 best chromosomes

(BCs), a total of 500 BCs, are revealed, each having the potential to be considered as the best $\boldsymbol{B}$.

Considering the chromosome that yields the smallest MSE among all BCs (SMBC) as the best chromosome and the corresponding best $\boldsymbol{B}$ can introduce another statistical bias because the MSE generated by SMBC is based on just one RTT coming from the NGA operation. When the input number of a new set of training and testing data of a different RTT is reduced through the translation process of SMBC and the MSE is estimated using DRBM, it may be bigger than the smallest MSE obtained through the NGA operation. To minimize this statistical bias in the DRBN, in the second step each BC is translated 100 times, for all 5 RTTs (10, 20, 30, 40, and 50%), each time using a different randomly selected set of training and testing samples. Hence, 5 sets of 100 training and testing data set are generated, for a total of 500 training and testing data sets. For each BC, the estimation tool DRBM is trained, each time using one of the 500 training data set, and relevant testing data are applied to estimate the MSE of the testing data. Therefore, 5 sets of 100 MSEs are estimated for each BC, for a total of 500 MSEs for each BC. The average of these MSEs is calculated for each BC. The best chromosome among BCs (termed the BBC) is the one that yields the minimum average MSE and consequently the best $\boldsymbol{B}$. To summarize, the BBC is found in the second step out of the 500 BCs obtained in the first step.

Estimation of Sale Prices of Real Estate Units

DRBM is necessary for the efficient operation of the NGA model presented in the previous section. The outcome of the model is the best $\boldsymbol{B}$ with the total number of input variables $I + K$, substantially smaller than the original input number of $I + T \times J$ (Fig. 4) in $\boldsymbol{U}$. The reduced learning problem can be solved by a number of different supervised artificial neural network models to estimate the sale prices of real estate units before construction. In this research, DRBM is used. In the next section, a case study is presented using the data obtained for a large metropolitan city.

Case Study

The model proposed herein is applied to a set of data collected for 360 residential condominiums (3–9 stories) that were built between 1993 and 2008 in Tehran, Iran, a city with a metro population of around 8.2 million (United Nations Statistics Division 2014) and much building construction activity. The first set of inputs and each subgroup of the second set of inputs are the same as those presented in Tables 1 and 2 ($I = 7$ and $J = 19$), respectively. The number of subgroups in second input set is assumed as 5 ($T = 5$), which represents 5 quarters prior to the start of the construction. Hence, the second set of inputs has $T \times J = 95$ inputs and the total input numbers is $I + T \times J = 102$. The total number of data in the second set of inputs is aimed to be reduced to $K = 20$. Hence, the total input number is reduced to $I + K = 27$, which seems to be appropriate for 360 samples.

Identification of BBC

The numbers of NGA generations and population number are selected as 200 and 20, respectively. In this case study, the number of candidate chromosomes is assumed to be 40,000 based on the RAM availability of the workstation used in this research (32 GB of RAM and two 2.7-GHz processors). Therefore, $40{,}000 \times 200 = 8{,}000{,}000$ possible chromosomes are evaluated in one NGA operation. The first population (20 chromosomes) is generated randomly. The computer model presented in this paper was implemented in *MATLAB version 8.2.0.701* programming tool and run over five CPU cores simultaneously for about a week.

The BBC is identified as the chromosome vector number 87 ($\boldsymbol{B}$) for RTT = 10% with an MSE of 0.0029. Table 4 presents the inputs and their selected subgroups using this chromosome. It is seen that input numbers 4, 10, 12, and 18 are eliminated as inputs with insignificant influence. The remaining 15 inputs are determined to be influential with inputs 2, 3, 11, and 16 deemed to be more influential than the rest because they contain two subgroups or periods. The first subgroup has the highest number of appearance

Table 4. Sample Inputs for Residential Condos Based on the Best Chromosome Structure

Second input number	Subgroup number	Descriptions
1	1	Number of building permits issued by the city/municipality
2	1,2	Building services index for a preselected base year (e.g., 2004)
3	1,2	Wholesale price index (WPI) of building materials for the base year
4	—	Total floor areas of building permits issued by the city/municipality
5	1	Cumulative liquidity (millions of dollars)
6	1	Private sector investment in new buildings in the city (millions of dollars)
7	1	Land price index in the city for the base year
8	2	Number of loans extended by banks in a quarter
9	1	Amount of loans extended by banks in a quarter (millions of dollars)
10	—	Interest rate for loan in the quarter
11	1,2	Average construction cost of buildings by private sector at the time of completion of construction per m^2 in the city (millions of dollars)
12	—	Average of construction cost of buildings by private sector at the beginning of the construction per m^2 in the city (millions of dollars)
13	1	Official exchange rate with respect to dollars
14	1	Nonofficial (street market) exchange rate with respect to dollars (used only in countries with controlled currencies)
15	2	Consumer price index (CPI) in the base year
16	1,2,4	CPI of housing, water, fuel, and power in the base year
17	1	Stock market index
18	—	Population of the city
19	3	Gold price per ounce (dollars)

Table 5. Comparative Performance of DRBM and BP

Number of epochs	Processing time for training (s)		Testing error (%)	
	BP	DRBM	BP	DRBM
2	4.9	16.1	56.8	5.7
4	9.5	24.4	56.6	5.5
6	13.7	33.5	56.4	5.3
8	18.3	40.3	56.3	5.2
10	23.9	49.2	56.1	5.2
50	112.6	223.7	49.1	4.8
100	234.1	443.5	22.2	4.7
1,000	2,248.5	4,632.6	6.3	3.7
1,820	4,073.9	8,612.8	5.7	3.6

(12 times) in BBC, while the last subgroup, number 5, did not appear at all. This confirms the assumption in the first strategy that the input variables in times closer to the project's construction beginning date (the inputs in the first subgroups of $\boldsymbol{U}$), in general, have more influence on the real estate market compared with those at later times.

Estimation of the Sale Price

Out of the 360 available data points for the period 1993–2009 described in the previous section, 10 were chosen randomly for testing and the remaining 350 were used for training. The original data set has $I + T \times J = 102$ features. The BBC presents the most influential features with a reduced set of input equal to $I + K = 27$. This process was repeated 100 times. The model yields an average error of 5.7% with the strategies used to reduce the dimensionality of the input data using only two epochs of training. This compares with an error of 17.4% without using the strategies.

For the sake of comparison, the sale estimation part of the methodology was also solved using the feedforward backpropagation (BP) neural network with one hidden layer and 10 nodes in the hidden layer. The results are summarized in Table 5, which includes the number of time periods used for training, processing time for training on an Intel Core 2.40-GHz laptop, and the testing error. After two time periods, DRBM and BP yield results with an error of 5.7% and 56.8%, respectively. As such, DRBM yields reasonably accurate results using only two periods of training and training time of only 16.1 s. On the other hand, BP requires 1,820 periods of training and training time of 4,073.9 s to yield the same testing error of 5.7%. BP as an estimator is about 253 times slower compared with DRBM to achieve the same testing error. Consequently, DRBM is a substantially faster and more accurate machine learning algorithm than BP.

In order to solve the entire sale price prediction problem for the case study, 4,000 applications of the estimator (steps 4–6 in Fig. 4), DRBM or BP, are needed. For just one specific RTT, say 10%, it takes around 16.1 s for DRBM and 4,073.9 s for BP to calculate the MSE of 100 randomly selected training and testing samples using one chromosome in each population. If the numbers of populations and generations are 20 and 200, respectively, the total time needed by DRBM, neglecting the time needed for generating the new population using the population generator (steps 10–13 in NGA operation, Fig. 4), is $20 \times 200 \times 16.1/3{,}600 \cong 18$ h. In contrast, using BP, about $20 \times 200 \times 4{,}073.9/3{,}600 \cong 4{,}530$ h would be needed to obtain results comparable to DRBM.

The error can be due to a variety of reasons *partly* due to the data peculiarity of the locality, Tehran, a city under severe economic sanctions, unpredictable government policies that can change virtually overnight, and unstable and soft currency of the country, which create huge volatility. The validity of the model was demonstrated using a chaotic real estate market as proof of the concept. The model was able to predict in a highly volatile and arguably unpredictable market with a reasonable accuracy. It is expected to yield more accurate results in advanced and developed countries.

Table 6. Average Error Percentages of 100 Sets of Training and Testing

Classifier	Methodology			
	Genetic search	Best first	Linear forward selection	Correlation-based feature subset
Naïve Bayes	20.3	19.2	19.4	10.3
Bagging	19.2	21.4	19.2	18.1
SVM	47.2	56.3	53.5	64.8

The accuracy of the nonmating GA (NGA) proposed in the paper is compared with four widely used feature selection methodologies (FSM), standard genetic search, best first, linear forward selection, and correlation-based feature subset, using three different classifiers, naïve Bayes, bagging, and support vector machine (SVM). Average error percentages of 100 sets of training and testing using these methods are summarized in Table 6. The error ranges from 10.3 to 64.8%. This compares with an error of 3.7% obtained from the NGA incorporating the DRBM. Thus, the superiority of the new method is substantiated.

Conclusion

In this paper, an expansive and novel model was presented for estimating the sale price of a new building unit before the start of the construction. The original contribution of the current submission is twofold. First, the paper formulates a comprehensive model for estimating the price of new housing in any given city at the design phase or beginning of the construction. No previous study presents such a comprehensive formulation. An effective data structure is presented that takes into account a large number of economic variables/indices. The model incorporates time-dependent and seasonal variations of the variables. The second contribution of the paper is a novel solution strategy using a powerful neural network model. Since the model considers a large number of variables, training the model would require a large training data set and correspondingly significant computational resources. Consequently, a focus of the paper is reducing the data dimensionality.

Clever stratagems have been developed to overcome the dimensionality curse and make the solution of the problem amenable on standard workstations. The model was tested with new data successfully. It estimates the sale price with reasonable accuracy. The accuracy can be partly improved by increasing the number of hidden layers, but that will increase the computation time significantly, requiring the use of a parallel machine or a supercomputer. Part of the difficulty of estimating the sale price accurately is due to unforeseen economic conditions such as (1) error in data collection, (2) speculation, (3) the political decisions by the government authorities, (4) impact of global economy, and (5) sudden world events that can reverberate from one nation to others.

Since the model is comprehensive and incudes a large number of variables, data dimensionality becomes necessary for its efficient execution. NGA was developed not only for reducing data dimensionality but also to increase the probability of finding the best combination of features to yield accurate estimation results exploiting the global optimization capability of the GA. The capability of the new model was tested using data from a locality under severe economic sanctions, unpredictable government policies that can change virtually overnight, and unstable and soft currency of the country that create huge volatility. Even though the results obtained using data for this particular locality are very promising (Tables 5 and 6), the model needs to be evaluated further using data from other localities, which is currently unavailable. This research can provide the impetus to collect the pertinent data for other regions.

This research shows that DRBM is effective in learning high-dimensional and complicated pattern recognition problem addressed in this research in a reasonable amount of time. The authors assert that this paper breaks new grounds for developing a tool for estimation for the price of new construction. Such a tool will be of great value for the entire construction industry. In the model presented, the same set of input parameters is used for each period. The model, however, can be extended to have different sets of input parameters for different periods.

The new model can assist construction companies to address the question of whether to *build or not to build*. In a down market, depending on the future sale price provided by model, the builder may decide to postpone the beginning of construction. In an up market, the builder may prefer to sell the units at the time of completion instead of the presale to maximize the profit. Therefore, the proposed model helps the builder decide to *build or not to build* at a particular time in a logical manner. As such, the proposed model can be a valuable tool for construction companies. Further, a construction company with multiple projects in different regions can use the tool for resource scheduling (Adeli and Karim 1997), such as allocating large cranes (Senouci and Adeli 2001) to localities with a better price prospect.

Another application would be in the emerging area of *big data*, which is mostly unexplored in the construction field. Increasingly, the society at large is driven by large amounts of data. The same phenomenon will happen in the field of construction. The authors envisage the creation of large depositories of data for the building construction of an entire large metro area. Currently, building information model (BIM) data (Khosrowshahi et al. 2014; Zhang et al. 2014b; Cho et al. 2014) is stored and used for each individual building in isolation. In the future, large databases will include such BIM data for the entire city, following similar trends in other fields such as banking and medicine.

References

Adeli, H., ed. (1994). *Advances in design optimization*, Chapman and Hall, London.

Adeli, H. (2001). "Neural networks in civil engineering: 1989–2000." *Comput.-Aided Civ. Infrastruct. Eng.*, 16(2), 126–142.

Adeli, H., and Hung, S. L. (1995). *Machine learning—Neural networks, genetic algorithms, and fuzzy systems*, Wiley, New York.

Adeli, H., and Karim, A. (1997). "Scheduling/cost optimization and neural dynamics model for construction." *J. Constr. Manage. Eng.*, 10.1061/(ASCE)0733-9364(1997)123:4(450), 450–458.

Adeli, H., and Park, H. S. (1998). *Neurocomputing for design automation*, CRC Press, Boca Raton, FL.

Adeli, H., and Sarma, K. (2006). *Cost optimization of structures—Fuzzy logic, genetic algorithms, and parallel computing*, Wiley, Chichester, U.K.

Adeli, H., and Soegiarso, R. (1999). *High-performance computing in structural engineering*, CRC Press, Boca Raton, FL.

Adeli, H., and Wu, M. (1998). "Regularization neural network for construction cost estimation." *J. Constr. Eng. Manage.*, 10.1061/(ASCE)0733-9364(1998)124:1(18), 18–24.

Borowiecki, K. J. (2009). "The determinants of house prices and construction: An empirical investigation of the Swiss housing economy." *Int. Real Estate Rev.*, 12(3), 193–220.

Boutalis, Y., Christodoulou, M., and Theodoridis, D. (2013). "Indirect adaptive control of nonlinear systems based on bilinear neuro-fuzzy approximation." *Int. J. Neural Syst.*, 23(5), 1350022.

Butcher, J. B., Day, C. R., Austin, J. C., Haycock, P. W., Verstraeten, D., and Schrauwen, B. (2014). "Defect detection in reinforced concrete using random neural architectures." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(3), 191–207.

Cho, Y. S., Lee, S. I., and Bae, J. S. (2014). "Reinforcement placement in a concrete slab object using structural building information modeling." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(1), 47–59.

Chow, J. Y. J. (2014). "Activity-based travel scenario analysis with routing problem reoptimization." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(2), 91–106.

Dai, H., and Wang, W. (2014). "An adaptive wavelet frame neural network method for efficient reliability analysis." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(10), 801–814.

Das, S., Gupta, R., and Kabundi, A. (2009). "Could we have predicted the recent downturn in the South African housing market?" *J. Housing Econ.*, 18(4), 325–335.

Égert, B., and Mihaljek, D. (2007). "Determinants of house prices in central and eastern Europe." *Comp. Econ. Stud.*, 49(3), 367–388.

Favara, G., and Song, Z. (2014). "House price dynamics with dispersed information." *J. Econ. Theor.*, 149, 350–382.

Forcael, E., González, V., Orozco, F., Vargas, S., Moscoso, P., and Pantoja, A. (2014). "Ant colony optimization model for tsunamis evacuation routes." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(10), 723–737.

Friedrich, J., Urbancziky, R., and Senn, W. (2014). "Code-specific learning rules improve action selection by populations of spiking neurons." *Int. J. Neural Syst.*, 24(5), 1450002.

Fuggini, C., Chatzi, E., Zangani, D., and Messervey, T. B. (2013). "Combining genetic algorithm with a meso-scale approach for system identification of a smart polymeric textile." *Comput.-Aided Civ. Infrastruct. Eng.*, 28(3), 227–245.

Hejazi, F., Toloue, I., Noorzaei, J., and Jaafar, M. S. (2013). "Optimization of earthquake energy dissipation system by genetic algorithm." *Comput.-Aided Civ. Infrastruct. Eng.*, 28(10), 796–810.

Hinton, G. E. (2007). "Learning multiple layers of representation." *Trends Cognit. Sci.*, 11(10), 428–434.

Hinton, G. E., Osindero, S., and Teh, Y. W. (2006). "A fast learning algorithm for deep belief nets." *Neural Comput.*, 18(7), 1527–1554.

Hinton, G. E., and Salakhutdinov, R. R. (2006). "Reducing the dimensionality of data with neural networks." *Science*, 313(5786), 504–507.

Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities." *Proc. Natl. Acad. Sci. U.S.A.*, 79(8), 2554–2558.

Hsu, W. Y. (2013). "Single-trial motor imagery classification using asymmetry ratio, phase relation and wavelet-based fractal features, and their selected combination." *Int. J. Neural Syst.*, 23(2), 1350007.

Huang, Y., Beck, J. L., Wu, S., and Li, H. (2014). "Robust Bayesian compressive sensing for signals in structural health monitoring." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(3), 160–179.

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

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

Jia, L., Wang, Y., and Fan, L. (2014). "Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm." *Integr. Comput. Aided Eng.*, 21(1), 77–90.

Khalafallah, A. (2008). "Neural network based model for predicting housing market performance." *Tsinghua Sci. Technol. J.*, 13(S1), 325–328.

Khosrowshahi, F., Ghdous, P., and Sarshar, M. (2014). "Visualization of the modeled degradation of building flooring systems in building maintenance." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(1), 18–30.

Kodogiannis, V. S., Amina, M., and Petrounias, I. (2013). "A clustering-based fuzzy-wavelet neural network model for short-term load forecasting." *Int. J. Neural Syst.*, 23(5), 1350024.

Kwon, M., Kavuri, S., and Lee, M. (2014). "Action-perception cycle learning for incremental emotion recognition in a movie clip using 3D fuzzy

GIST based on visual and EEG signals." *Integr. Comput.-Aided Eng.*, 21(3), 295–310.

Lin, D. Y., and Ku, Y. H. (2014). "Using genetic algorithms to optimize stopping patterns for passenger rail transportation." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(4), 264–278.

Mathwords. (2012). "Combination formula." ⟨http://www.mathwords.com/c/combination_formula.htm⟩ (Jun. 22, 2015).

MATLAB version 8.2.0.701 [Computer software]. Natick, MA, MathWorks.

Pedrino, E. C., et al. (2013). "A genetic programming based system for the automatic construction of image filters." *Integr. Comput.-Aided Eng.*, 20(3), 275–287.

Rapach, D. E., and Strauss, J. K. (2006). "The long-run relationship between consumption and housing wealth in the Eighth District states." *Econ. Dev.*, 2(2), 140–147.

Rigatos, G. G. (2013). "Adaptive fuzzy control for differentially flat MIMO nonlinear dynamical systems." *Integr. Comput.-Aided Eng.*, 20(2), 111–126.

Rosselló, J. L., Canals, V., Oliver, A., and Morro, A. (2014). "Studying the role of synchronized and chaotic spiking neural ensembles in neural information processing." *Int. J. Neural Syst.*, 24(5), 11.

Selim, H. (2009). "Determinants of house prices in Turkey: Hedonic regression versus artificial neural network." *Expert Syst. Appl.*, 36(2), 2843–2852.

Senouci, A. B., and Adeli, H. (2001). "Resource scheduling using neural dynamics model of Adeli and Park." *J. Constr. Eng. Manage.*, 10.1061/(ASCE)0733-9364(2001)127:1(28), 28–34.

Shafahi, Y., and Bagherian, M. (2013). "A customized particle swarm method to solve highway alignment optimization problem." *Comput.-Aided Civ. Infrastruct. Eng.*, 28(1), 52–67.

Shapero, S., Zhu, M., Hasler, P., and Rozell, C. (2014). "Optimal sparse approximation with integrate and fire neurons." *Int. J. Neural Syst.*, 24(5), 1440001.

Siddique, N., and Adeli, H. (2013). *Computational intelligence—Synergies of fuzzy logic, neural networks and evolutionary computing*, Wiley, Chichester, U.K.

Smolensky, P. (1986). "Information processing in dynamical systems: Foundations of harmony theory." Chapter 6, *Parallel distributed processing*, Vol. 1, MIT Press, Cambridge, MA, 194–281.

Spackova, O., and Straub, D. (2013). "Dynamic Bayesian networks for probabilistic modeling of tunnel excavation processes." *Comput.-Aided Civ. Infrastruct. Eng.*, 28(1), 1–21.

Story, B. A., and Fry, G. T. (2014). "A structural impairment detection system using competitive arrays of artificial neural networks." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(3), 180–190.

Szeto, W. Y., Wang, Y., and Wong, S. C. (2014). "The chemical reaction optimization approach to solving the environmentally sustainable network design problem." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(2), 140–158.

United Nations Statistics Division. (2014). "City population by sex, city and city type." ⟨http://data.un.org/Data.aspx?d=POP&f=tableCode:240⟩ (Mar. 11, 2014).

U.S. Department of Labor. (2015). "Producer price indexes." Bureau of Labor Statistics, ⟨http://www.bls.gov/ppi/⟩ (Jun. 22, 2015).

Vlahogianni, E. I., and Karlaftis, M. G. (2013). "Fuzzy-entropy neural network freeway incident duration modeling with single and competing uncertainties." *Comput.-Aided Civ. Infrastruct. Eng.*, 28(6), 420–433.

Wu, J. W., Tseng, J. C. R., and Tsai, W. N. (2014). "A hybrid linear text segmentation algorithm using hierarchical agglomerative clustering and discrete particle swarm optimization." *Integr. Comput.-Aided Eng.*, 21(1), 35–46.

Zeng, Z., Xu, J., Wu, S., and Shen, M. (2014). "Antithetic method-based particle swarm optimization for a queuing network problem with fuzzy data in concrete transportation systems." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(10), 771–800.

Zhang, G., Rong, H., Neri, F., and Perez-Jimenez, M. J. (2014a). "An optimization spiking neural P system for approximately solving combinatorial optimization problems." *Int. J. Neural Syst.*, 24(5), 1440006.

Zhang, J. P., Yu, F. Q., Li, D., and Hu, Z. Z. (2014b). "Development and implementation of an industry foundation classes-based graphic information model for virtual construction." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(1), 60–74.

Zhang, Y., and Ge, H. (2013). "Freeway travel time prediction using Takagi-Sugeno-Kang fuzzy neural network." *Comput.-Aided Civ. Infrastruct. Eng.*, 28(8), 594–603.

Zhu, W., Hu, H., and Huang, Z. (2014). "Calibrating rail transit assignment models with genetic algorithm and automated fare collection data." *Comput.-Aided Civ. Infrastruct. Eng.*, 29(7), 518–530.