Novel Machine-Learning Model for Estimating Construction Costs Considering Economic Variables and Indexes

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

Abstract: In addition to materials, labor, equipment, and method, construction cost depends on many other factors such as the project locality, type, construction duration, scheduling, and the extent of use of recycled materials. Further, the fluctuation of economic variables and indexes (EV&Is), such as liquidity, wholesale price index, and building services index, causes variation in costs. These changes may increase or reduce the construction cost, are hard to predict, and are normally ignored in the traditional cost estimation computation. This paper presents an innovative construction cost estimation model using advanced machine-learning concepts and taking into account the EV&Is. A data structure is proposed that incorporates a set of physical and financial (P&F) variables of the real estate units as well as a set of EV&Is variables affecting the construction costs. The model includes an unsupervised deep Boltzmann machine (DBM) learning approach along with a softmax layer (DBM-SoftMax), and a three-layer back-propagation neural network (BPNN) or another regression model, support vector machine (SVM). The role of DBM-SoftMax is to extract relevant features from the input data. The role of the BPNN or SVM is to turn the trained unsupervised DBM into a supervised regression network. This combination improves the effectiveness and accuracy of both conventional BPNN and SVM. A sensitivity analysis was performed within the algorithm in order to achieve the best results taking into account the impact of the EV&I factors in different times (time lags). The model was verified using the construction cost data for 372 low- and midrise buildings in the range of three to nine stories. Cost estimation errors of the proposed model were much less than those of both the BPNN-only and SVM-only models, thus demonstrating the effectiveness of the strategies employed in this research and the superiority of the proposed model. **DOI:** 10.1061/(ASCE)CO.1943-7862.0001570. © *2018 American Society of Civil Engineers.*

Author keywords: Construction cost estimation; Residential buildings; Deep learning; Deep Boltzmann machine; Softmax; Back-propagation neural networks; Support vector machine.

Introduction

Background Knowledge

In recent years, modern computer technologies have been applied to various construction problems such as modeling construction process using cloud computing (Sharif et al. 2017), construction simulation (Yu et al. 2017), construction laborer assignment (Yi and Wang 2016, 2017), construction crew allocation (Florez 2017), work-rest schedule design (Yi and Wang 2016), resource levelling (Ponz-Tienda et al. 2017), radio frequency identification (RFID)–enabled knowledge-based construction supply chain (Wang et al. 2017), and virtual design and construction implementation strategies (Mandujano et al. 2017).

Accurate cost estimation in early stages of construction projects leads to cost savings, thus contributing to a more sustainable project (Gao and Zhang 2013; Smith et al. 2014; Rafiei and Adeli 2016).

[1]Dept. of Civil, Environmental and Geodetic Engineering, Dept. of Physical Medicine Rehabilitation, Ohio State Univ., 470 Hitchcock Hall, 2070 Neil Ave., Columbus, OH 43210 (corresponding author). Email: rafiei.4@osu.edu

[2]Dept. of Civil, Environmental and Geodetic Engineering, Electrical and Computer Engineering, Biomedical Engineering, Neurology, and Neuroscience, Ohio State Univ., 470 Hitchcock Hall, 2070 Neil Ave., Columbus, OH 43210. Email: adeli.1@osu.edu

Note. This manuscript was submitted on February 20, 2018; approved on June 11, 2018; published online on September 25, 2018. Discussion period open until February 25, 2019; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Construction Engineering and Management***, © ASCE, ISSN 0733-9364.

The estimated cost is commonly computed based on the cost of project determinants such as construction materials, labor, equipment, and method (Myers 2016). The construction cost depends on many other factors such as the project locality, type, construction duration, scheduling, and extent of use of recycled materials (Fereshtehnejad and Shafieezadeh 2016). Further, the fluctuation of economic variables and indexes, such as liquidity, wholesale price index (WPI), and building services index (BSI), causes variation in costs, especially in a volatile or unstable economic environment (Hsiao et al. 2012; Adeli and Karim 1997; Karim and Adeli 1999; Jafarzadeh et al. 2013; Rafiei and Adeli 2015). These changes may increase or reduce the construction cost, are hard to predict, and are normally ignored in the traditional cost estimation computation (Smith 2002; Rafiei and Adeli 2015).

Previous Studies

In the past two decades, a number of studies have been published on construction cost estimation using neural network, regression, or stochastic techniques (Gribniak et al. 2016; El Hajj et al. 2017). Adeli and Wu (1998) presented a mathematical model and a regularization neural network for estimation of the cost of construction projects taking into account the noise in the data. They formulated the cost estimation problem in terms of an error function consisting of a standard error term and a regularization term whose purpose was to plank out the overfitting, thus improving the cost estimation accuracy for new data points. They applied the model to estimate the cost of reinforced concrete pavements.

Kim et al. (2004) used back-propagation neural networks (BPNNs) (Hung and Adeli 1993, 1994) to estimate the construction cost of residential buildings in Korea. The input included a building's variables such as floor area, number of stories, total units, construction time duration, and roof type. The output was the total construction cost. The model was evaluated using data of 530 residential buildings in Korea between 1997 and 2000. They used genetic algorithm (GA) (Mencia et al. 2016; Li et al. 2017) to find the optimum weights, biases, and other parameters of the BPNN model (Siddique and Adeli 2015). Wilmot and Mei (2005) used BPNN and economic variables such as annual bid volume and bid volume variance to estimate highway construction costs in Louisiana. Alex et al. (2009) employed BPNN to estimate the cost of installation of water and sewer services for residential facilities. Among the inputs of the model were project start date, duration, delay status, labor crew size, site operational requirements, and material estimations. They reported an accuracy of only 80%.

Kim and Kim (2010) used case-based reasoning (CBR) (Karim and Adeli 2003; Sirca and Adeli 2005) and GA (Pillon et al. 2016) to estimate bridge construction costs. CBR is a memory-based computational algorithm that assigns weights to predictors (inputs) for cost estimation (Waheed and Adeli 2005). CBR is used to determine the similarity of a cost estimation problem to similar projects in a previously collected database known as the case base (Li et al. 2011). Cheng et al. (2010) employed a fuzzy neural network (FNN) model (Karim and Adeli 2002) for construction cost estimation of buildings. Similar to Kim et al. (2004), they used GA to optimize the parameters of the FNN model. Hsiao et al. (2012) used GA for construction cost estimation of semiconductor hookup. Ma et al. (2016) presented an ontology for formalized representation of specifications for construction cost estimation. Kyriklidis and Dounias (2016) employed evolutionary computation (Rostami and Neri 2016; Wright and Jordanov 2017) for resource leveling optimization in project management. Bhargava et al. (2017) studied the probability of cost overrun during different phases of construction using the Monte Carlo simulation. The aforementioned studies employed mostly BPNNs, sometimes in combination with a GA. With the exception of Wilmot and Mei (2005), none of the published research incorporates any economic indicator in the cost estimation.

Current Research

Machine learning (Adeli and Hung 1995; Palomo and Lopez-Rubio 2016; Quintian and Corchado 2017; Fernandez et al. 2017) is a key technology in the 21st century that is already used in smart phones, drones, driverless cars, and numerous other applications (Wang et al. 2016; Rigos et al. 2016; Zeinalia and Story 2017; Zhang et al. 2017). It will be an increasingly pervasive technology in society. Deep-learning neural networks, in particular, have received a lot of attention in the past few years (Ortiz-Garcia et al. 2016; Morabito et al. 2017; Koziarski and Cyganek 2017; Ortega-Zamorano et al. 2017). The purpose of this research is to develop an innovative and more realistic construction cost estimation model using advanced machine-learning concepts and taking into account the economic variables and indexes (EV&Is). A data structure is proposed that incorporates a set of physical and financial (P&F) variables of the real estate units as well as a set of EV&I variables affecting the construction costs. The model includes an unsupervised deep Boltzmann machine (DBM) learning approach along with a softmax layer (DBM-SoftMax), and a three-layer BPNN or another regression model, support vector machine (SVM) (Cortes and Vapnik 1995). The role of DBM-SoftMax is to extract relevant features from the input data. The role of the BPNN or SVM is to turn the trained unsupervised DBM into a supervised regression network (DBM-BPNN or DBM-SVM). This combination improves the effectiveness and accuracy of both conventional BPNN and SVM. A sensitivity analysis was performed within the algorithm in order to achieve the best results taking into account the impact of the EV&I factors in different times (time lags).

General Data Structure

Following Rafiei and Adeli (2015), Fig. 1 presents the general data structure proposed for an example real estate unit (e.g., a single-family residential apartment). It consists of a vector of inputs, $\mathbf{X}$, and an output, C. The vector of inputs is made of two subvectors: (1) an I-dimensional row vector $\mathbf{V}$ that includes P&F variables of the real estate unit such as those summarized in Table 1; and (2) a K-dimensional row vector $\mathbf{U}$ that includes EV&I variables such as those summarized in Table 1, extracted from T different time lags before the beginning of construction ($\mathbf{X} = [\mathbf{V}, \mathbf{U}]$).

The EV&I variables can be computed in a specific time resolution such as a week, a month, or a quarter, depending on data availability. There is usually a time lag between changes of economic variables and their consequences on the construction costs. The time lag can be expressed in number of time resolutions before the start of the construction. For example, if the time resolution is a quarter, time lag can be one, two, or three quarters before the start of the construction.

Table 1 presents eight samples of P&F and 19 samples of EV&I variables for a residential multistory building (Rafiei and Adeli 2015). The proposed model can handle any type of real estate unit and any number of variables as long as data are available for those variables. EV&I variables are computed in nonoverlapping time lags and placed in $\mathbf{U}_t$, where the row vector $\mathbf{U}_t \in \mathbf{U} = \{\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_T\}$ contains the values of EV&I variables in the tth time lag (t time resolutions before the construction start time) and T is the total number of time lags (Fig. 1). For example, if time resolution is a quarter and $T = 5$, the EV&I variables in Table 1 in five quarters before the start of the construction will be used. The total number of P&F variables is I and the total number of EV&I variables computed in each time lag is J. Hence, the total

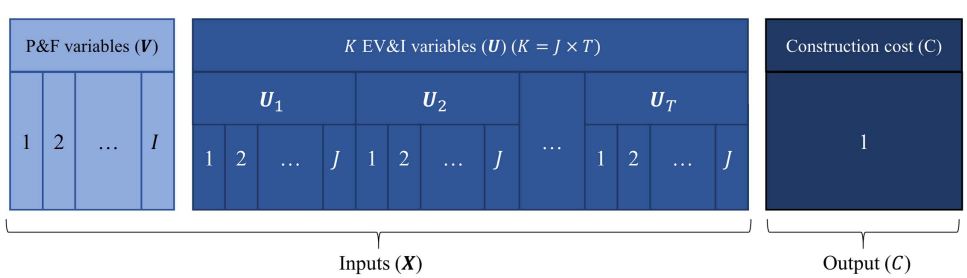

Fig. 1. General data structure of the model.

Table 1. Samples of P&F and EV&I variables

Variable	Description	Unit
P&F factors (first input subset, V)		
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	Total preliminary estimated construction cost based on the prices at the beginning of the project	Dollars
5	Preliminary estimated construction cost based on the prices at the beginning of the project	Dollars/m^2
6	Equivalent preliminary estimated construction cost based on the prices at the beginning of the project in a selected base year[a]	Dollars/m^2
7	Duration of construction	Quarter, month, or week
8	Price of the unit at the beginning of the project per square meter	Dollars/m^2
EV&I variables in each non-overlapping time lag, U_t, in the real estate unit's city		
1	Number of building permits issued	N/A
2	BSI[b] for a preselected base year[a]	N/A
3	WPI[c] of building materials for the base year	N/A
4	Total floor areas of building permits issued by the city or municipality	m^2
5	Cumulative liquidity[d]	Millions of dollars
6	Private sector investment in new buildings	Millions of dollars
7	Land price index for the base year[a]	Millions of dollars
8	Number of loans extended by banks in a time resolution[e]	N/A
9	Amount of loans extended by banks in a time resolution[e]	Millions of dollars
10	Interest rate for loan in a time resolution[e]	%
11	Average construction cost of buildings by private sector at the time of completion of construction	Millions of dollars/m^2
12	Average of construction cost of buildings by private sector at the beginning of the construction	Millions of dollars/m^2
13	Official exchange rate with respect to dollars	%
14	Nonofficial (street market) exchange rate with respect to dollars[f]	%
15	Consumer price index (CPI)[g] in the base year[a]	N/A
16	CPI of housing, water, fuel, and power in the base year[a]	N/A
17	Stock market index[h]	N/A
18	Population of the city	N/A
19	Gold price per ounce	Dollars

Source: Adapted from Rafiei and Adeli (2015), © ASCE.
[a]For example, 2004.
[b]Presents the total subcontractor's amount of contracts such as worker wages and pipe installation.
[c]Or producer price index presents cost of a basket of food/services to detect inflation/deflation (US Department of Labor 2015b).
[d]Represents how rapidly different types of assets can be changed to cash.
[e]For example, a quarter.
[f]Used only in countries with controlled currencies.
[g]Represents change in prices of a basket of goods and services purchased by urban households (US Department of Labor 2015a).
[h]Represents the payback condition of investment in stock market.

number of input variables is $I + K$, where $K = T \times J$. The output of the model is the construction cost, C.

Proposed Model

The proposed model requires a set of previously collected real estate training data and a smaller set of verification data per the data structure defined in the previous section (Fig. 1). The model consists of two consecutive phases: a training phase incorporating a sensitivity analysis similar to that used in Rafiei and Adeli (2017b) for an earthquake early warning model, using DBM-SoftMax and BPNN (or SVM) algorithms, and a verification phase. Fig. 2 presents the architecture of DBM-SoftMax and BPNN. One can substitute BPNN with another regression algorithm such as SVM.

DBM, DBM-SoftMax, DBM-BPNN, and DBM-SVM

DBM is an unsupervised learning algorithm, consisting of an encoder and a decoder (Hinton and Salakhutdinov 2006; Rafiei and Adeli 2015, 2018), each made of several layers of restricted Boltzmann machine (RBM) (Smolensky 1986), designed to simulate the learning functionality of neural layers in the human brain stochastically. The human brain extracts characteristics from perceptions received through sensory inputs. Similarly, the encoder extracts features from the input data and from the extracted features themselves multiple times consecutively. The decoder employs extracted features to reconstruct the inputs and simulates the ability of the brain in remembering the perceived perceptions. Details about DBM and its structure can be found in Hinton and Salakhutdinov (2006) and Rafiei and Adeli (2015, 2017b, a) and will not be repeated here for brevity.

The extracted features by the encoder indicate certain characteristics of the input data. Theoretically, a set of inputs or an object can have an infinite number of embedded features. DBM extracts a number of these features that might not necessarily convey the desired patterns in the problem at hand. Different randomly generated initial parameters, such as weights of the links of the neural network and biases, may lead to different sets of extracted features by DBM. To improve the situation, an additional two-layer softmax neural network with softmax activation function was employed in this research to control the DBM training such that the extracted features in the last layer of DBM address the desired patterns accurately (Fig. 3). In this network, the inputs of BPNN (or SVM) are

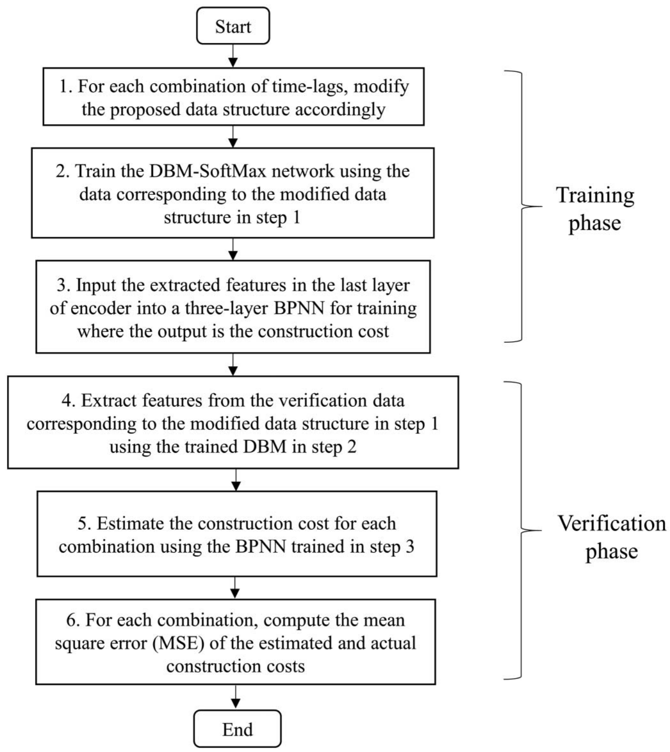

Fig. 2. Steps for each combination of time lag numbers in the proposed model.

the extracted features at the last layer of the encoder and the output is an M-dimensional vector, where M is the total number of classes in a classification problem, with all elements being 0 except the one corresponding to the class number of the input, which is 1.

The softmax activation function, also known as the entropy function, used for classification in this research is defined in the form of a conditional probability function as follows (Bishop 2006):

$$P[\mathrm{Class}(\mathbf{H}) = i|\mathbf{H}] = \frac{\exp(\mathbf{H}'\mathbf{W}_i)}{\sum_{j=1}^{M}\exp(\mathbf{H}'\mathbf{W}_j)} \tag{1}$$

where $\mathbf{H}$ = vector of extracted features from the last hidden layer of the encoder; the apostrophe represents the transpose of a function; and $\mathbf{W}_i = i$th column of the matrix of connection weights in the softmax network. Fig. 4 displays the softmax function with $M = 2$ classes using two randomly generated vectors $\mathbf{W}_1$ and $\mathbf{W}_2$, and 20,000 two-dimensional vectors of $\mathbf{H}$ in which each element (feature) of each vector was a real random number between -4 and $+4$. The vertical softmax axis presents the conditional probability of each random vector of $\mathbf{H}$ being in Class 1 or 2. The softmax function was found to be more robust and accurate compared with the sigmoid function commonly used in DBM when the outputs were in the form of special binary-valued class vectors. For example, if $M = 5$ and the class number of a data point is 4, the outputs of the softmax will be [0, 0, 0, 1, 0]. Another advantage of the softmax is that it is based on a stochastic concept and yields the probabilities of $\mathbf{H}$ being in different classes. The class assignment is based on the maximum probability.

The proposed data structure includes a real-valued output (not categories or class numbers), that is, construction cost. In order to employ such a softmax layer in DBM for a regression problem, one approach is to categorize the distribution of real-valued outputs of the training data into a predefined M classes by visual inspection of histogram of the natural logarithm of all outputs (natural logarithm usually yields a more normal distribution of data). An alternative approach, used in this research, is to use a clustering algorithm (Peng et al. 2016) such as k-means, wavelet-clustering-neural

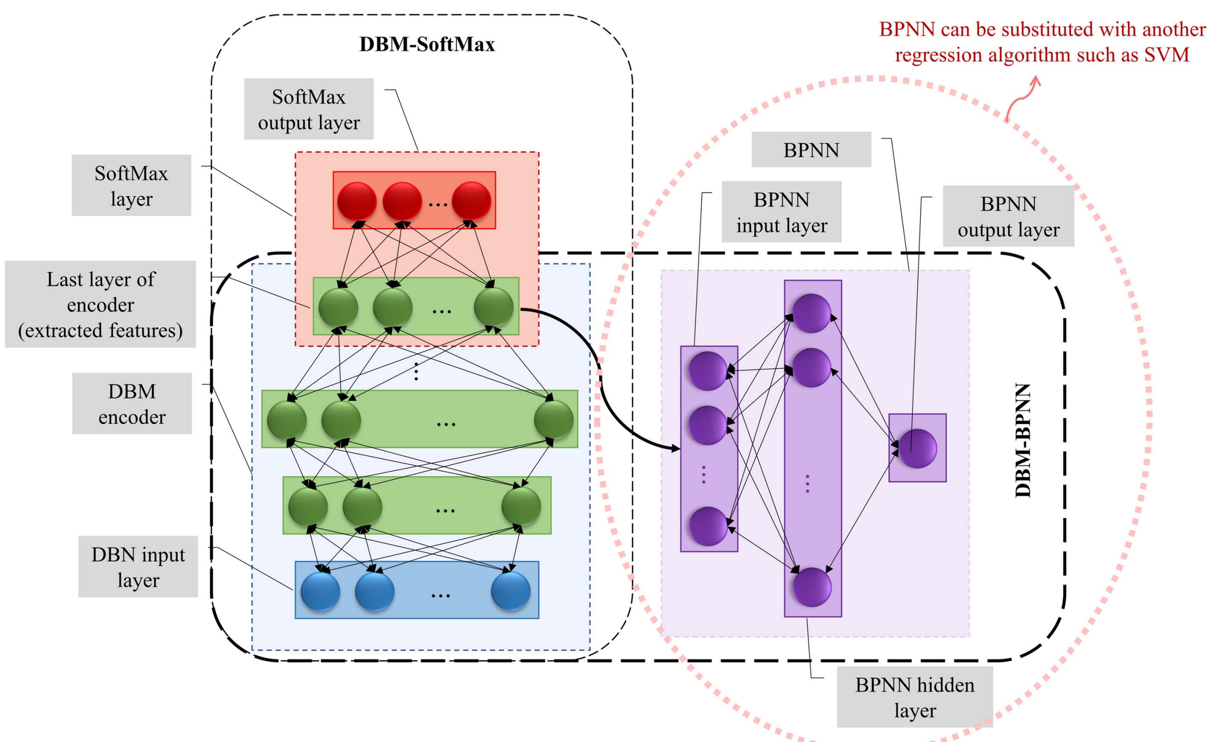

Fig. 3. Architecture of the integrated DBM-SoftMax and DBM-BPNN model.

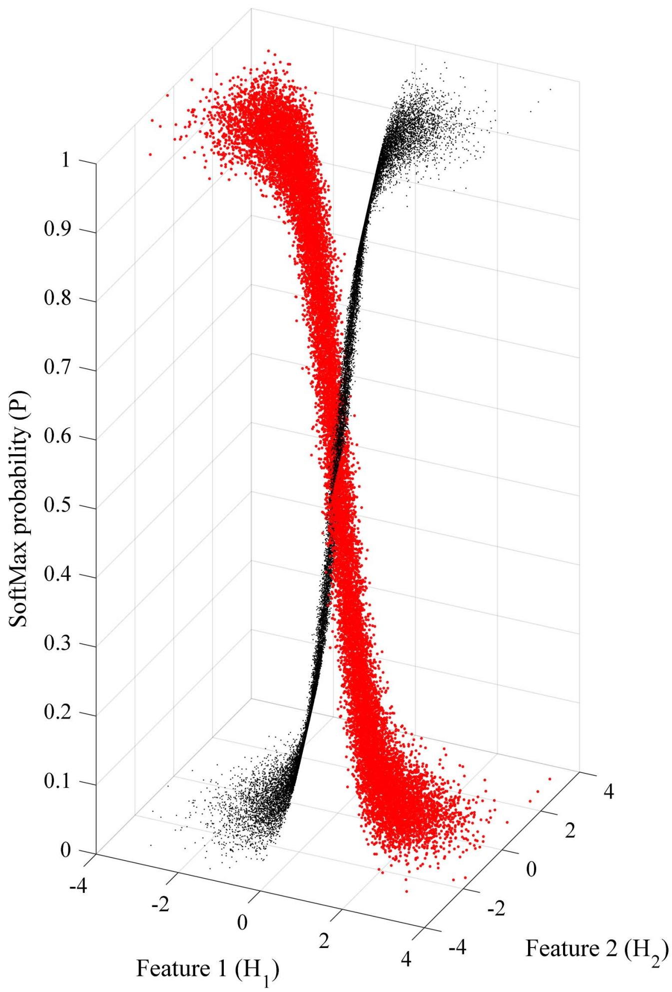

Fig. 4. Softmax function with $M = 2$ classes using two randomly generated vectors $\mathbf{W}_1$ and $\mathbf{W}_2$, and 20,000 two-dimensional vectors of $\mathbf{H}$ in which each element (feature) of each vector is a real random number between -4 and $+4$.

network (Ghosh-Dastidar and Adeli 2003), or expectation maximization (Ortiz-Rosario et al. 2017) to transform the real-valued output into M classes.

A gradient descent algorithm (Adeli 1994) was used to train the DBM-SoftMax network. However, the performance of DBM depends highly on the initial parameters (weights and biases). With initial parameters close enough to the trained parameters, training of DBM-SoftMax will be fast and accurate; otherwise, it requires a large number of iterations to converge. To solve this problem, Hinton and Salakhutdinov (2006) proposed a pretraining algorithm to identify appropriate initial parameters of DBM so that the DBM-SoftMax converges quickly. In the encoder, each layer of RBM is trained separately using a constructive divergence learning technique (Hinton and Salakhutdinov 2006; Rafiei and Adeli 2015) for a few iterations. The initial parameters of each layer of encoder are set to be the ones from the corresponding layer in the pretraining process (decoder is also modified since it is symmetric to encoder). This way, only a few iterations are needed to train the DBM-SoftMax network accurately.

Once the DBM-SoftMax is trained, the extracted features in the last hidden layer of encoder are used as the inputs of a three-layer BPNN (or input of SVM) (Fig. 3). More accurate estimation of construction cost is achieved by BPNN (or SVM) using the features extracted by DBM-SoftMax compared with using the inputs directly. Once the BPNN is trained, the trained DBM and the trained BPNN (SVM) collectively form a trained DBM-BPNN (DBM-SVM) for regression problems.

Training the Algorithm

Fig. 2 presents the steps in each combination of time lag numbers in the proposed model. In the first step of the proposed algorithm, identified in Step 1 of Fig. 2, to investigate the effect of EV&I variables in different time lags on the accuracy of the construction estimation, the proposed data structure is modified S times, where S is the total number of possible combinations of T time lags (Mathwords 2012)

$$S = \sum_{i=1}^{T} \frac{T!}{(T-i)!i!} \tag{2}$$

For every combination, the modified data structure always contains I number of P&F values (vector $\mathbf{V}$), and the output C, but includes the values of EV&I variables only in time lags that appear in that combination. Fig. 5 presents an example of a modified data structure when $T = 5$ [$S = 31$ per Eq. (2)] using a combination of time lags that includes EV&I values in only three time lag numbers: 2, 4, and 5.

In Step 2 of the proposed algorithm, the DBM-SoftMax network is trained using the data corresponding to the modified data structure presented in Step 1. In Step 3, the extracted features in the last layer of encoder in the trained DBM-SoftMax are input into a three-layer BPNN (or SVM) for training where the output is the construction cost (regression values).

Validating the Algorithm

In Step 4, for each combination of time lag numbers, the DBM trained in Step 2 is used to extract features from the verification

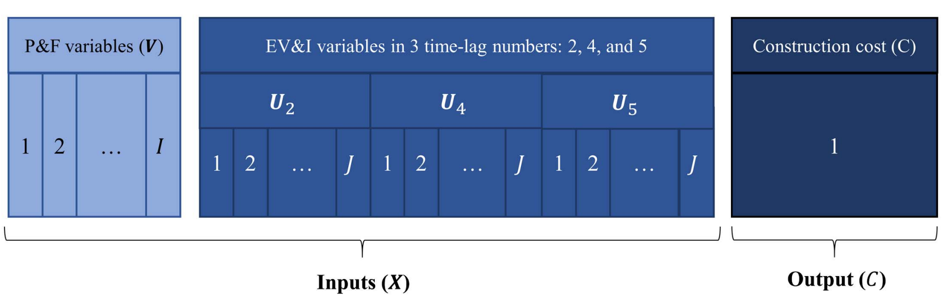

Fig. 5. Example of modified data structure with $T = 5$ and a combination that includes only time lag numbers 2, 4, and 5.

data corresponding to the modified data structure in Step 1. In Step 5, the BPNN trained in Step 3 is used to estimate the construction costs for each combination. In Step 6, the mean-square error (MSE) of the estimated and actual construction costs is computed for each combination. The trained DBM-BPNN corresponding to the combination with the maximum accuracy is the final trained model, which can be used to estimate the construction cost of any new real estate unit.

Case Study

The proposed model was applied to a set of data collected for 372 residential condominiums from as many 3- to 9-story buildings built between 1993 and 2008 in Tehran, Iran. Tehran is a city with a volatile economy, a large number of buildings under construction, and a metro population of around 8.2 million (Umetani et al. 2011). The data collected consisted of $I = 8$ P&F variables and $J = 19$ EV&I variables, presented in Table 1. The time resolution was a quarter and data for the EV&I variables for $T = 5$ time lags before the construction start of each condominium were available. The total number of P&F and EV&I variables with available data for each residential condominium was $8 + 5 \times 19 = 103$.

Implementation

The proposed model was implemented in MATLAB on a CPU core with four 2.4-GHz processors (Intel Core i7-3630QM, Santa Clara, California). The outputs in the last layer of softmax were the natural logarithm of construction costs categorized using the k-means clustering algorithm.

Results and Discussions

To prevent any statistical bias, the entire available data were divided into training and verification data using five different ratios of

Table 2. DBM-SoftMax architectures used to investigate the proposed model

			Number of neurons in the hidden layer			Total number of iterations	
Model	Architecture	M	First RBM	Second RBM	Third RBM	Pretraining	Training
DBM-SoftMax	1	10	100	50	20	1,000	1,000
	2	5	50	25	10	100	100
DBM-SVM	1	10	100	50	N/A	20	200

Table 3. Top five combinations with minimum DBM-BPNN average MSE for 100 sets of training and verification and their corresponding BPNN-only and DBM-SoftMax training accuracy percentages for different RVTs and two different networks

		Network 1						Network 2					
			Time lag						Time lag				
RVT (%)	Combination	MSE/accuracy percent	1	2	3	4	5	MSE/accuracy percent	1	2	3	4	5
10	1	0.049 (0.106) [95.1]	0	0	1	0	0	0.027 (0.146) [88.7]	0	1	0	1	0
	2	0.055 (0.246) [97.9]	0	0	0	1	1	0.027 (0.190) [88.2]	0	0	0	1	0
	3	0.055 (0.120) [99.5]	1	0	0	1	1	0.028 (0.147) [88.1]	0	0	1	1	0
	4	0.056 (0.175) [99.4]	1	0	1	0	1	0.029 (0.099) [88.7]	1	0	0	1	0
	5	0.056 (0.117) [96.1]	0	0	0	1	0	0.029 (0.099) [87.6]	0	0	1	0	0
	Average	0.054 (0.153) [97.6]	0.4	0	0.4	0.6	0.6	0.028 (0.136) [88.3]	0.2	0.2	0.4	0.8	0.0
20	1	0.061 (0.199) [99.6]	1	0	1	0	1	0.029 (0.123) [88.3]	0	0	0	1	0
	2	0.064 (0.208) [99.5]	1	1	0	0	1	0.031 (0.135) [88.5]	0	0	1	1	0
	3	0.064 (0.218) [99.6]	1	0	0	1	1	0.032 (0.187) [88.9]	0	1	0	1	0
	4	0.067 (0.213) [99.5]	0	1	1	1	0	0.032 (0.167) [88.5]	0	1	0	0	0
	5	0.068 (0.186) [98.6]	0	1	0	1	0	0.033 (0.154) [87.8]	0	0	1	0	0
	Average	0.065 (0.205) [99.3]	0.6	0.6	0.4	0.6	0.6	0.031 (0.153) [88.4]	0.0	0.4	0.4	0.6	0.0
30	1	0.071 (0.124) [98.8]	0	1	1	0	0	0.031 (0.120) [89.1]	0	0	0	1	0
	2	0.072 (0.154) [99.8]	1	0	0	1	1	0.034 (0.147) [89.8]	0	1	0	1	0
	3	0.073 (0.169) [99.8]	1	0	1	0	1	0.035 (0.174) [89.0]	0	1	0	0	0
	4	0.073 (0.134) [99.2]	0	1	0	1	0	0.035 (0.145) [88.8]	0	0	1	1	0
	5	0.074 (0.161) [97.1]	0	1	0	0	0	0.035 (0.131) [89.8]	1	0	0	1	0
	Average	0.072 (0.148) [99.0]	0.4	0.6	0.4	0.4	0.4	0.034 (0.143) [89.3]	0.2	0.4	0.2	0.8	0.0
40	1	0.087 (0.150) [99.9]	1	1	0	0	1	0.036 (0.106) [89.3]	0	0	0	1	0
	2	0.088 (0.169) [99.9]	1	0	1	0	1	0.037 (0.142) [89.3]	0	1	0	0	0
	3	0.094 (0.145) [98.9]	0	0	1	1	0	0.039 (0.172) [90.0]	0	1	0	1	0
	4	0.095 (0.179) [99.9]	1	0	0	1	1	0.039 (0.164) [88.8]	0	0	1	0	0
	5	0.095 (0.141) [99.9]	0	1	1	0	1	0.039 (0.139) [90.0]	1	0	0	1	0
	Average	0.092 (0.157) [99.7]	0.6	0.4	0.6	0.4	0.8	0.038 (0.145) [89.5]	0.2	0.4	0.2	0.6	0.0
50	1	0.109 (0.223) [99.2]	0	1	1	1	0	0.043 (0.131) [89.8]	0	0	0	1	0
	2	0.112 (0.192) [100.0]	1	0	1	0	1	0.047 (0.139) [89.7]	0	0	0	0	1
	3	0.115 (0.165) [98.7]	1	0	1	1	0	0.047 (0.165) [90.1]	0	1	0	0	0
	4	0.117 (0.164) [99.9]	0	0	1	1	1	0.048 (0.151) [89.5]	0	0	1	0	0
	5	0.125 (0.202) [100.0]	1	0	0	1	1	0.049 (0.176) [90.3]	0	0	1	1	0
	Average	0.116 (0.189) [99.6]	0.6	0.2	0.8	0.8	0.6	0.047 (0.152) [89.9]	0.0	0.2	0.4	0.4	0.2

Note: Values in parentheses are BPNN-only percentages; values in brackets are DBM-SoftMax training accuracy percentages. 1 = time lag was selected; and 0 = no time lag.

verification to training (RVTs) of 10%, 20%, 30%, 40%, and 50%. For each RVT, 100 randomly selected verification and training data sets were generated. There were $S = 31$ combinations of time lags for $T = 5$ used in this example. Two different DBM-SoftMax network architectures presented in Table 2 were investigated. Values of 0.004 and 5 were selected for the learning rate and number of minibatches, respectively. The first and second DBM-SoftMax networks extracted 20 and 10 features in the last layer of their encoders (hidden layer of the last or third RBM), respectively. The BPNN had one hidden layer with 20 neurons. The total number of BPNN training iterations was chosen to be the same as the number of DBM-SoftMax iterations. The results of the proposed model were compared with using BPNN alone. Alternatively, one can use SVM in place of BPNN. In this case, the proposed algorithm would be called DBM-SVM. For the sake of comparison, a DBM-SVM network architecture presented in Table 2 was also investigated in this research.

Table 3 presents the top five combinations with minimum DBM-BPNN average MSE for 100 training and verification data sets and their corresponding BPNN-only in parentheses and DBM-SoftMax training accuracy percentage in square brackets for different RVTs and two different networks (1 indicates time lag was selected and 0 indicates it was not). In all cases, the MSE of the proposed model was much less than that of the BPNN-only model. Network 1 has been trained using a relatively large number of pretraining and training iterations (1,000 each) that yielded average training accuracies in the range of 95.1% to 100.0%. The training accuracy of the second network varied within the smaller range of 87.6% to 90.3%, probably due to the smaller number of pretraining and training iterations (100 each) compared with the first network. However, the DBM-BPNN MSEs corresponding to the second network were much smaller than the first network. This might be due to overfitting of the first network (Adeli and Wu 1998). The average MSEs, training accuracies, and time lag selections were also computed in Table 3. According to the results of the second network, the most selected time lag number was 4, suggesting a relatively long 4-quarter delay between changes in EV&I variables and their effect on the construction cost at the end of the project.

Table 4 presents the top five combinations with minimum DBM-SVM average MSE for 10 training and verification data sets and their corresponding SVM-only in parentheses for five different RVTs. The average MSEs, training accuracies, and time lag selections are summarized in Table 4. According to the results of the second network, the most selected time lag number was 4, similar to that of the DBM-BPNN model. While DBM-SVM yielded more accurate results than DBM-BPNN, in all cases the MSE of the proposed DBM-SVM model was less than that of the SVM-only model.

Table 4. Top five combinations with minimum DBM-SVM average MSE for 10 sets of training and verification and their corresponding SVM-only for different RVTs

RVT (%)	Combination	MSE	Time lag 1	2	3	4	5
10	1	0.021 (0.032)	0	0	0	1	0
	2	0.022 (0.032)	0	0	1	1	0
	3	0.022 (0.034)	1	0	0	0	0
	4	0.022 (0.033)	0	1	0	1	0
	5	0.022 (0.032)	1	0	0	1	0
	Average	0.022 (0.032)	0.4	0.2	0.2	0.8	0
20	1	0.030 (0.046)	0	0	0	1	0
	2	0.031 (0.047)	0	0	0	1	1
	3	0.031 (0.048)	0	0	0	0	1
	4	0.032 (0.046)	0	0	1	1	0
	5	0.033 (0.048)	0	1	0	0	1
	Average	0.031 (0.047)	0	0.2	0.2	0.6	0.6
30	1	0.029 (0.041)	0	0	0	1	0
	2	0.030 (0.039)	0	0	0	1	1
	3	0.031 (0.042)	0	0	0	0	1
	4	0.031 (0.040)	0	0	1	1	0
	5	0.034 (0.039)	0	0	1	1	1
	Average	0.031 (0.040)	0	0	0.4	0.8	0.6
40	1	0.026 (0.042)	0	0	0	1	1
	2	0.026 (0.044)	0	0	0	0	1
	3	0.027 (0.042)	0	0	0	1	0
	4	0.028 (0.041)	0	0	1	1	0
	5	0.029 (0.043)	1	0	0	0	1
	Average	0.027 (0.042)	0.2	0	0.2	0.6	0.6
50	1	0.032 (0.041)	0	0	0	1	0
	2	0.033 (0.045)	0	0	1	0	0
	3	0.034 (0.042)	0	0	0	0	1
	4	0.035 (0.040)	0	0	0	1	1
	5	0.035 (0.044)	1	0	0	0	0
	Average	0.034 (0.043)	0.2	0	0.2	0.4	0.4

Note: Values in parentheses are SVM-only; 1 = time lag was selected; and 0 = no time lag.

Conclusion

In this paper, a new construction cost estimation model was presented using advanced machine-learning concepts and taking into account the economic variables and indexes. The model was verified using the construction cost data for 372 low- and midrise buildings in the range of three to nine stories. Cost estimation errors of the proposed model were much less than those of both the BPNN-only and SVM-only models, thus demonstrating the effectiveness of the strategies employed in this research and the superiority of the proposed model.

A purpose of the proposed research was to introduce an advanced machine-learning technology into the field of construction estimation. The proposed computational algorithm is general and can be used for different building construction projects for any city or region. The model has been illustrated using the collected data from one city. One can investigate the proposed algorithm for other construction projects and/or other regions. To the best of the authors' knowledge, the data used to verify the proposed algorithm are the first data set that includes both P&F and EV&I variables. A contribution of this paper is to bring the importance of economic variables in accurate construction cost estimation to the attention of the construction industry, researchers, and government agencies. This research should encourage researchers to collect such data in cities around the world, especially those with a volatile and fluctuating economic environment. The size of the data set used was large enough for the model to yield accurate results, but the accuracy can be improved further with additional data.

A concern about such data is the proportion of the number of data sets to the number of input variables. The larger the number of input variables, the larger the number of training samples required for accurate cost estimation, which can make the training of the proposed algorithm computationally intensive. This is known as the dimensionality curse in the field of machine learning (Adeli and Wu 1998; Hinton and Salakhutdinov 2006). To address the data dimensionality concern, the authors performed a sensitivity analysis using all possible combinations of time lags, and identified the combinations that yielded the best results (Table 1). This way, the number of inputs can be as few as only 27. In order to prevent any statistical bias, the authors used five different RVTs, and 100 randomly selected sets of verification and training data for each RVT.

This way, the probability of any statistical bias, including biases due to overfitting, is minimized.

One limitation of the proposed model would be the unavailability of some important economic data over a long period of time with an appropriate time resolution in a region or a city. In other words, for the proposed model to work, past data must be available in a region or a city. The other limitation is the unpredictability of government policies. Finally, the model may not produce accurate results in highly unstable economic environments.

Data Availability Statement

Data generated by the authors or analyzed during the study are available at: https://archive.ics.uci.edu/ml/datasets/Residential+Building+Data+Set. Information about the *Journal*'s data-sharing policy can be found here: http://ascelibrary.org/doi/10.1061/(ASCE)CO.1943-7862.0001263.

References

Adeli, H., ed. 1994. *Advances in design optimization*. London: Chapman & Hall.

Adeli, H., and S. Hung. 1995. *Machine learning, neural networks, genetic algorithms and fuzzy systems*. New York: Wiley.

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

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

Alex, D. P., M. Al Hussein, A. Bouferguene, and S. Fernando. 2009. "Artificial neural network model for cost estimation: City of Edmonton's water and sewer installation services." *J. Constr. Eng. Manage.* 136 (7): 745–756. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000184.

Bhargava, A., S. Labi, S. Chen, T. U. Saeed, and K. C. Sinha. 2017. "Predicting cost escalation pathways and deviation severities of infrastructure projects using risk-based econometric models and Monte Carlo simulation." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (8): 620–640. https://doi.org/10.1111/mice.12279.

Bishop, C. M. 2006. *Pattern recognition and machine learning*. New York: Springer.

Cheng, M., H. Tsai, and E. Sudjono. 2010. "Conceptual cost estimates using evolutionary fuzzy hybrid neural network for projects in construction industry." *Expert Syst. Appl.* 37 (6): 4224–4231. https://doi.org/10.1016/j.eswa.2009.11.080.

Cortes, C., and V. Vapnik. 1995. "Support-vector networks." *Mach. Learn.* 20 (3): 273–297. https://doi.org/10.1007/BF00994018.

El Hajj, B., F. Schoefs, B. Castanier, and T. Yeung. 2017. "A condition-based deterioration model for the stochastic dependency of corrosion rate and crack propagation in corroded concrete structure." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (1): 18–33. https://doi.org/10.1111/mice.12208.

Fereshtehnejad, E., and A. Shafieezadeh. 2016. "Multiple hazard incidents lifecycle cost assessment of structural systems considering state-dependent repair times and fragility curves." *Earthquake Eng. Struct. Dyn.* 45 (14): 2327–2347. https://doi.org/10.1002/eqe.2764.

Fernandez, A., C. J. Carmona, M. J. del Jesus, and F. Herrera. 2017. "A Pareto based ensemble with feature and instance selection for learning from multi-class imbalanced datasets." *Int. J. Neural Syst.* 27 (6): 1750028. https://doi.org/10.1142/S0129065717500289.

Florez, L. 2017. "Crew allocation system for the masonry industry." *Comput. Aided Civ. Infrastruct. Eng.* 32 (10): 874–889. https://doi.org/10.1111/mice.12301.

Gao, H., and X. Zhang. 2013. "A Markov-based road maintenance optimization model considering user costs." *Comput.-Aided Civ. Infrastruct. Eng.* 28 (6): 451–464. https://doi.org/10.1111/mice.12009.

Ghosh-Dastidar, S., and H. Adeli. 2003. "Wavelet-clustering-neural network model for freeway incident detection." *Comput. Aided Civ. Infrastruct. Eng.* 18 (5): 325–338. https://doi.org/10.1111/1467-8667.t01-1-00311.

Gribniak, V., H. A. Mang, R. Kupliauskas, G. Kaklauskas, and A. Juozapaitis. 2016. "Stochastic tension-stiffening approach for the solution of serviceability problems in reinforced concrete: Exploration of predictive capacity." *Comput.-Aided Civ. Infrastruct. Eng.* 31 (6): 416–431. https://doi.org/10.1111/mice.12183.

Hinton, G. E., and R. R. Salakhutdinov. 2006. "Reducing the dimensionality of data with neural networks." *Science* 313 (5786): 504–507. https://doi.org/10.1126/science.1127647.

Hsiao, F., S. Wang, W. Wang, C. Wen, and W. Yu. 2012. "Neuro-fuzzy cost estimation model enhanced by fast messy genetic algorithms for semiconductor hookup construction." *Comput.-Aided Civ. Infrastruct. Eng.* 27 (10): 764–781. https://doi.org/10.1111/j.1467-8667.2012.00786.x.

Hung, S. L., and H. Adeli. 1993. "Parallel backpropagation learning algorithms on Cray Y-MP8/864 supercomputer." *Neurocomputing* 5 (6): 287–302. https://doi.org/10.1016/0925-2312(93)90042-2.

Hung, S. L., and H. Adeli. 1994. "Object-oriented back propagation and its application to structural design." *Neurocomputing* 6 (1): 45–55. https://doi.org/10.1016/0925-2312(94)90033-7.

Jafarzadeh, R., J. Ingham, S. Wilkinson, V. González, and A. Aghakouchak. 2013. "Application of artificial neural network methodology for predicting seismic retrofit construction costs." *J. Constr. Eng. Manage.* 140 (2): 04013044. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000725.

Karim, A., and H. Adeli. 1999. "CONSCOM: An OO construction scheduling and change management system." *J. Constr. Eng. Manage.* 125 (5): 368–376. https://doi.org/10.1061/(ASCE)0733-9364(1999)125:5(368).

Karim, A., and H. Adeli. 2002. "Comparison of the fuzzy-wavelet RBFNN freeway incident detection model with the California algorithm." *J. Transp. Eng.* 128 (1): 21–30. https://doi.org/10.1061/(ASCE)0733-947X(2002)128:1(21).

Karim, A., and H. Adeli. 2003. "CBR model for freeway work zone traffic management." *J. Transp. Eng.* 129 (2): 134–145. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:2(134).

Kim, G., J. Yoon, S. An, H. Cho, and K. Kang. 2004. "Neural network model incorporating a genetic algorithm in estimating construction costs." *Build. Environ.* 39 (11): 1333–1340. https://doi.org/10.1016/j.buildenv.2004.03.009.

Kim, K. J., and K. Kim. 2010. "Preliminary cost estimation model using case-based reasoning and genetic algorithms." *J. Comput. Civ. Eng.* 24 (6): 499–505. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000054.

Koziarski, M., and B. Cyganek. 2017. "Image recognition with deep neural networks in presence of noise—Dealing with and taking advantage of distortions." *Integr. Comput.-Aided Eng.* 24 (4): 337–349. https://doi.org/10.3233/ICA-170551.

Kyriklidis, C., and G. Dounias. 2016. "Evolutionary computation for resource leveling optimization in project management." *Integr. Comput.-Aided Eng.* 23 (2): 173–184. https://doi.org/10.3233/ICA-150508.

Li, H., H. Adeli, J. Sun, and J. Han. 2011. "Hybridizing principles of TOPSIS with case-based reasoning for business failure prediction." *Comput. Oper. Res.* 38 (2): 409–419. https://doi.org/10.1016/j.cor.2010.06.008.

Li, W., H. Pu, P. Schonfeld, J. Yang, H. Zhang, L. Wang, and J. Xiong. 2017. "Mountain railway alignment optimization with bidirectional distance transform and genetic algorithm." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (8): 691–709. https://doi.org/10.1111/mice.12280.

Ma, Z., Z. Liu, and Z. Wei. 2016. "Formalized representation of specifications for construction cost estimation by using ontology." *Comput.-Aided Civ. Infrastruct. Eng.* 31 (1): 4–17. https://doi.org/10.1111/mice.12175.

Mandujano, M. G., C. Mourgues, L. F. Alarcón, and J. Kunz. 2017. "Modeling virtual design and construction implementation strategies considering lean management impacts." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (11): 930–951. https://doi.org/10.1111/mice.12253.

Mathwords. 2012. "Combination formula." Accessed June 22, 2015. http://www.mathwords.com/c/combination_formula.htm.

Mencıa, R., M. R. Sierra, C. Mencıa, and R. Varela. 2016. "Genetic algorithms for the scheduling problem with arbitrary precedence relations and skilled operators." *Integr. Comput.-Aided Eng.* 23 (3): 269–285. https://doi.org/10.3233/ICA-160519.

Morabito, F. C., et al. 2017. "Deep learning representation from electroencephalography of early-stage Creutzfeld-Jakob disease and features for differentiation from rapidly progressive dementia." *Int. J. Neural Syst.* 27 (2): 1650039. https://doi.org/10.1142/S0129065716500398.

Myers, D. 2016. *Construction economics: A new approach*. New York: Taylor & Francis.

Ortega-Zamorano, F., J. M. Jerez, I. Gómez, and L. Franco. 2017. "Layer multiplexing FPGA implementation for deep back-propagation learning." *Integr. Comput.-Aided Eng.* 24 (2): 171–185. https://doi.org/10.3233/ICA-170538.

Ortiz-Garcia, A., J. Munilla, J. M. Gorriz, and J. Ramirez. 2016. "Ensembles of deep learning architectures for the early diagnosis of the Alzheimer's disease." *Int. J. Neural Syst.* 26 (7): 1650025. https://doi.org/10.1142/S0129065716500258.

Ortiz-Rosario, A., H. Adeli, and J. A. Buford. 2017. "MUSIC—Expected maximization Gaussian mixture methodology for clustering and detection of task-related neuronal firing rates." *Behav. Brain Res.* 317 (Jan): 226–236. https://doi.org/10.1016/j.bbr.2016.09.022.

Palomo, E. J., and E. Lopez-Rubio. 2016. "Learning topologies with the growing neural forest." *Int. J. Neural Syst.* 26 (3): 1650019. https://doi.org/10.1142/S0129065716500192.

Peng, H., J. Wang, P. Shi, M. J. Perez-Jimenez, and A. Riscos-Nunez. 2016. "An extended membrane system with active membranes to solve automatic fuzzy clustering problems." *Int. J. Neural Syst.* 26 (3): 1650004. https://doi.org/10.1142/S0129065716500040.

Pillon, P. E., E. C. Pedrino, V. O. Roda, and M. C. Nicoletti. 2016. "A hardware oriented ad-hoc computer-based method for binary structuring element decomposition based on genetic algorithm." *Integr. Comput.-Aided Eng.* 23 (4): 369–383. https://doi.org/10.3233/ICA-160527.

Ponz-Tienda, J. L., A. Salcedo-Bernal, J. S. Rojas-Quintero, and E. Pellicer. 2017. "A parallel branch and bound algorithm for the resource levelling problem with minimal lags." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (6): 474–498. https://doi.org/10.1111/mice.12233.

Quintian, H., and E. Corchado. 2017. "Beta Hebbian learning as a new method for exploratory projection pursuit." *Int. J. Neural Syst.* 27 (6): 1750024. https://doi.org/10.1142/S0129065717500241.

Rafiei, M. H., and H. Adeli. 2015. "A novel machine learning model for estimation of sale prices of real estate units." *J. Constr. Eng. Manage.* 142 (2): 04015066. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001047.

Rafiei, M. H., and H. Adeli. 2016. "Sustainability in highrise building design and construction." *Struct. Des. Tall Spec. Build.* 25 (13): 643–658. https://doi.org/10.1002/tal.1276.

Rafiei, M. H., and H. Adeli. 2017a. "A novel machine learning-based algorithm to detect damage in high-rise building structures." *Struct. Des. Tall Spec. Build.* 26 (18): e1400. https://doi.org/10.1002/tal.1400.

Rafiei, M. H., and H. Adeli. 2017b. "NEEWS: A novel earthquake early warning model using neural dynamic classification and neural dynamic optimization." *Soil Dyn. Earthquake Eng.* 100 (Sep): 417–427. https://doi.org/10.1016/j.soildyn.2017.05.013.

Rafiei, M. H., and H. Adeli. 2018. "A novel unsupervised deep learning model for global and local health condition assessment of structures." *Eng. Struct.* 156 (Feb): 598–607. https://doi.org/10.1016/j.engstruct.2017.10.070.

Rigos, A., G. E. Tsekouras, M. I. Vousdoukas, A. Chatzipavlis, and A. F. Velegrakis. 2016. "A Chebyshev polynomial radial basis function neural network for automated shoreline extraction from coastal imagery." *Integr. Comput.-Aided Eng.* 23 (2): 141–160. https://doi.org/10.3233/ICA-15050710.3233/ICA-150507.

Rostami, S., and F. Neri. 2016. "Covariance matrix adaptation Pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm." *Integr. Comput.-Aided Eng.* 23 (4): 313–329. https://doi.org/10.3233/ICA-160529.

Sharif, M. M., M. Nahangi, C. T. Haas, and J. West. 2017. "Automated model-based finding of 3D objects in cluttered construction point cloud models." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (11): 893–908. https://doi.org/10.1111/mice.12306.

Siddique, N., and H. Adeli. 2015. "Nature inspired computing: An overview and some future directions." *Cognit. Comput.* 7 (6): 706–714. https://doi.org/10.1007/s12559-015-9370-8.

Sirca, G. F., Jr., and H. Adeli. 2005. "Case-based reasoning for converting working stress design-based bridge ratings to load factor design-based ratings." *J. Bridge Eng.* 10 (4): 450–459. https://doi.org/10.1061/(ASCE)1084-0702(2005)10:4(450).

Smith, G. S. 2002. *Managerial accounting for libraries & other not-for-profit organizations*. Chicago: American Library Association.

Smith, R., E. Ferrebee, Y. Ouyang, and J. Roesler. 2014. "Optimal staging area locations and material recycling strategies for sustainable highway reconstruction." *Comput.-Aided Civ. Infrastruct. Eng.* 29 (8): 559–571. https://doi.org/10.1111/mice.12089.

Smolensky, P. 1986. "Information processing in dynamical systems: Foundations of harmony theory." In *Parallel distributed processing*, 194–281. Cambridge, MA: MIT Press.

Umetani, T., K. Inoue, and T. Arai. 2011. "Pose estimation of construction materials using United Nations Statistics Division (2017)." Accessed December 26, 2017. http://data.un.org/Data.aspx?d=POP&f=tableCode:240.

US Department of Labor. 2015a. "Consumer price indexes: Bureau of Labor Statistics." Accessed June 22, 2015. http://www.bls.gov/cpi/.

US Department of Labor. 2015b. "Producer price indexes: Bureau of Labor Statistics." Accessed June 22, 2015. http://www.bls.gov/ppi/.

Waheed, A., and H. Adeli. 2005. "Case-based reasoning in steel bridge engineering." *Knowl.-Based Syst.* 18 (1): 37–46. https://doi.org/10.1016/j.knosys.2004.06.001.

Wang, R., Y. Zhang, and L. Zhang. 2016. "An adaptive neural network approach for operator functional state prediction using psychophysiological data." *Integr. Comput.-Aided Eng.* 23 (1): 81–97. https://doi.org/10.3233/ICA-150502.

Wang, Z., H. Hu, and W. Zhou. 2017. "RFID-enabled knowledge-based precast construction supply chain." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (6): 499–514. https://doi.org/10.1111/mice.12254.

Wilmot, C. G., and B. Mei. 2005. "Neural network modeling of highway construction costs." *J. Constr. Eng. Manage.* 131 (7): 765–771. https://doi.org/10.1061/(ASCE)0733-9364(2005)131:7(765).

Wright, J., and I. Jordanov. 2017. "Quantum inspired evolutionary algorithms with improved rotation gates for real-coded synthetic and real world optimization problems." *Integr. Comput.-Aided Eng.* 24 (3): 203–223. https://doi.org/10.3233/ICA-170545.

Yi, W., and S. Wang. 2016. "Multi-objective mathematical programming approach to construction laborer assignment with equity consideration." *Comput.-Aided Civ. Infrastruct. Eng.* 31 (12): 954–965. https://doi.org/10.1111/mice.12239.

Yi, W., and S. Wang. 2017. "Mixed-integer linear programming on work-rest schedule design for construction sites in hot weather." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (5): 429–439. https://doi.org/10.1111/mice.12267.

Yu, J., D. Zhong, B. Ren, D. Tong, and K. Hong. 2017. "Probabilistic risk analysis of diversion tunnel construction simulation." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (9): 748–771. https://doi.org/10.1111/mice.12276.

Zeinalia, Y., and B. Story. 2017. "Competitive probabilistic neural network." *Integr. Comput.-Aided Eng.* 24 (2): 105–118. https://doi.org/10.3233/ICA-170540.

Zhang, Y., Y. Wang, J. Jin, and X. Wang. 2017. "Sparse Bayesian learning for obtaining sparsity of EEG frequency bands based feature vectors in motor imagery classification." *Int. J. Neural Syst.* 244 (2): 1650032. https://doi.org/10.1016/j.quaint.2010.11.008.