PREDICTION OF SEISMIC DAMAGE SPECTRA USING COMPUTATIONAL INTELLIGENCE METHODS

Sadjad Gharehbaghi[a], Mostafa Gandomi[b], Vagelis Plevris[c,d], and Amir H. Gandomi[e,]*

[a] *Department of Civil Engineering, Sharif University of Technology, Tehran, Iran*
[b] *School of Civil Engineering, University of Tehran, Tehran, Iran*
[c] *Department of Civil and Architectural Engineering, Qatar University, Doha, Qatar*
[d] *Department of Civil Engineering and Energy Technology, Oslo Metropolitan University, Oslo, Norway*
[d] *Faculty of Engineering & IT, University of Technology Sydney, Australia*

ABSTRACT. Seismic damage prediction can be effectively used in performance-based seismic design of structures. Predicting seismic damage spectra, capturing both structural and earthquake features, is useful in quantifying the potential seismic damage of structures. The objective of this paper is to accurately predict the seismic damage spectra using computational intelligence methods. For this purpose, an inelastic single-degree-of-freedom system subjected to a set of earthquake ground motion records is used to compute the (exact) spectral damage. The Park-Ang damage index is used to quantify the seismic damage. Both structural and earthquake features are involved in the prediction models where multi-gene genetic programming (MGGP) and artificial neural networks (ANNs) are applied. Common performance metrics were used to assess the models developed for seismic damage spectra, and indicated that their accuracy was higher than a corresponding model in the literature. Although the performance metrics revealed that the ANN model is more accurate than the MGGP model, the explicit MGGP-based mathematical model renders it more practical in quantifying the potential seismic damage of structures.

Keywords: Computational Intelligence; Genetic programming; Artificial neural networks; Regression analysis; Seismic damage spectra; Inelastic SDOF systems; Park-Ang damage index; Resiliency.

1. INTRODUCTION

Today, predicting earthquake-induced damage is of crucial importance both when designing new resilient structures and to assess the resiliency of existing structures. The potential damage of a structure due to an earthquake, which is a qualitative problem, can be quantified using an index that captures both the main structural and the main earthquake characteristics. The peak demands, such as peak ductility and peak ground acceleration, typically used by seismic design codes, do not appropriately reflect the potential seismic damage of a structure. The deformation demands imposed on a structural component by an earthquake ground motion are cyclic in nature, and the associated results of the cumulative damage have a significant effect on the seismic resistance of a structure. Indeed, these demands include a conceptual limitation due to the fact that they do not explicitly capture the cumulative damage resulting from the number of response cycles, yield excursions, and energy dissipation [1-4]. Moreover, they do not take into account the duration and frequency content of a possible earthquake. Although these demands are essential, the sole reliance on such demands is inadequate, particularly for structures where the plastic deformations and energy dissipation are expected to be under the design-basis earthquakes. This inference is based on evidence and observations from past earthquake sequences, and it has also been confirmed by researchers (e.g. [5]). The design spectra currently included in most seismic design codes do not take these critical effects into account. As a result, the design base shear coefficient could be insufficient to control the structural damage of ductile structures subjected to ground motions exhibiting large energy content [6, 7]. The seismic design spectra obtained by means of spectral acceleration (known as elastic design spectra) and the response modification factor (known as ductility-based inelastic spectra) ignore the role of the chief factors

* Corresponding author, Professor
Email: gandomi@uts.edu.au
Tel.: +61 2 95145081

DOI: https://doi.org/10.1016/j.compstruc.2021.106584

described above. Therefore, there is a need to estimate the potential damage in a way that considers the chief parameters of both structure and earthquake. A solution is to use an inelastic spectrum that captures the cumulative damage. This approach has been investigated by researchers, and some relevant works are presented in the next paragraph.

Bozorgnia and Bertero [8] proposed damage spectra by combining the normalized hysteretic energy and deformation ductility. The damage spectra were computed by considering hundreds of horizontal ground motion records recorded during the Landers and Northridge earthquakes. Kunnath and Chai [9] proposed a simple procedure based on establishing a relation between seismic input energy and dissipated hysteretic energy to determine a damage-based inelastic cyclic demand spectrum able to be used in seismic analysis and the design of structures. Cosenza et al. [10] proposed a simple method incorporating the cumulative damage effect, which is applicable for seismic design and assessment of structures. An equivalent damage factor was introduced to determine the modified strength or displacement inelastic spectra capable of capturing the potential damage caused by an earthquake. Zhai et al. [11] proposed a mathematical model to estimate spectral damage for mainshock-aftershock sequence-type earthquake ground motions. They captured the influence of the fundamental vibration period of the structure, strength reduction factor, damping ratio, post-yield stiffness and site condition in the proposed spectral damage model. Greco et al. [12] proposed a stochastic approach to determine the seismic damage-based inelastic spectra, which incorporates the cumulative damage effects by using the Park-Ang damage model [13] and adopting the peak theory of random processes. Deng et al. [14] proposed a new and applicable damage indicator based on the response spectrum, which includes an intensity component and a spectral shape component. Based on the fact that the fundamental vibration period of the structure changes during an earthquake, they presented a circular rule to determine the critical period range of the response spectrum needed to calculate the spectral shape used. Wen et al. [15] proposed damage spectra for mainshock-aftershock sequence-type for soft soil sites. It was found that the influence of soft soil on the damage spectra could account for more than 40%. Wen et al. [16] also investigated the damage spectra of global crustal mainshock-aftershock sequences and proposed a mathematical model to estimate the damage spectra. They also looked at the effects of degradation and pinching behavior of the single-degree-of-freedom system, as well as the scale factor of aftershock sequences.

As described, predicting the potential damage for a broad range of structures, which can be provided by damage spectra directly, can be used in the design of new resilient structures and/or to assess the seismic resiliency of existing structures. Most of the works carried out so far have aimed to determine the potential damage indirectly by modifying the inelastic deformation-based spectra, while only a few works have computed the damage spectra directly. Moreover, although the former are well-studied and useful, the researchers have dealt with a limited number of variables in determining the damage spectra, while the main contribution of their work concerns mainshock-aftershock effects. In addition, all the previously mentioned works have used conventional regression methods to estimate the inelastic (and/or damage) spectra of interest.

The recent advances in computational intelligence methods render them suitable for solving challenging problems in science and engineering. It has been proven that they could reliably solve complex and highly nonlinear problems where conventional methods, such as regression analysis, fail or perform poorly [17, 18]. One of the most renowned computational intelligence-based predictive methods is Artificial Neural Networks (ANNs). ANNs belong inherently to the class of artificial intelligence methods, being inspired by the biological neural networks of the human brain. Although ANNs typically build black-box models and do not represent a governing equation of a predicted relation directly, they are capable of accurately predicting the outputs of complex problems. ANNs require the user to predefine the network architecture and some network parameters. Another well-known predictive method is Genetic Programming (GP), which is a learning algorithm that originated from genetic algorithms. GP, which lies in the class of evolutionary computation algorithms, is a method commonly used for developing nonlinear mathematical models to solve complicated problems. The standard GP was extended, and its variants such as gene expression programming [19] and multi-stage genetic programming [20] have been used in the literature. Multi-gene genetic programming (MGGP), one of the most robust variants of GP, has an additional capability of conventional regression in parameter estimation with respect to the standard GP. The effectiveness of MGGP over other GP approaches and ANNs has been proven in the works conducted by Gandomi et al. (e.g. [21-25]). Unlike ANNs, MGGP is capable of constructing the predicted relations between inputs and outputs as an analytic equation. The literature shows various successful applications of computational intelligence methods (e.g. ANNs and GP) in solving different problems in the field of structural and earthquake engineering, exhibiting its suitability for handling such problems. Some of these works are described in the paragraph below.

Lagaros and Papadrakakis [26] proposed a framework to predict the inelastic time-history responses of three-dimensional structures by using ANNs. Alavi and Gandomi [27] predicted some principal ground-motion parameters by using a hybrid method coupling ANNs and simulated annealing. Gholizadeh and Fattahi [28] used ANNs to predict the inelastic response of steel structures needed during the performance-based design optimization process. Gharehbaghi et al. [29] used ANNs to predict the inelastic seismic response of a high-rise reinforced concrete framed structure. Lagaros et al. [30] used ANNs to solve reliability-robust design optimization problems. Plevris and Asteris [31] proposed the modelling of masonry failure surface under biaxial compressive stress by using ANNs. They have also proposed the use of ANNs to approximate the failure surface of masonry materials in a dimensionless form [32]. An ANN model was proposed by Rizzo and Caracoglia [33] to the flutter velocity of suspension bridges. Gholizadeh et al. [34-36] used ANNs to predict the seismic response required in design optimization process of both steel and reinforced concrete structures. Alavi et al. [37] used multi expression programming to propose a set of attenuation models to derive some important parameters in earthquake engineering. Lim et al. [38] assessed the ultimate conditions of FRP-confined concrete columns using GP. Gharehbaghi et al. [25] proposed some efficient formulations to predict the seismic spectral energy by using MGGP. Kiani et al. [39] presented a work describing the application of machine learning techniques such as ANNs to derive seismic fragility curves.

In this paper, two computational intelligence methods, namely ANN and MGGP, are used to predict the seismic damage spectra when considering both main structural and main earthquake features. To achieve this, several single-degree-of-freedom (SDOF) systems with the structural properties of different hardening ratios of a bilinear hysteretic behavior model, damping ratio, response modification factor, ultimate ductility, and a constant related to the damage model, were used to determine the seismic damage spectra. Moreover, four different sets of earthquake ground motion records based on different soil types (soft, firm, stiff and rock) with magnitudes greater than 5.5 and source-to-site distances of more than 17.5 km were considered. Finally, the spectral damage was predicted using both ANN and MGGP predictive models. In the case of MGGP, a mathematical equation is also proposed for predicting the spectral damage. The effectiveness of the models is then assessed using six well-known performance metrics in comparison with another model available in the literature.

In the present study, the seismic damage analysis is described in Section 2. This section also introduces the Park-Ang damage index, which is appropriate for quantifying the earthquake-induced damage of structures. Section 3 presents ANN and MGGP, the two computational intelligence methods employed, while Section 4 describes a framework for the prediction of seismic damage spectra. In Section 5, the predictive models of seismic damage spectra are proposed, discussed and benchmarked in relation to a model recently proposed in the literature.

2. SEISMIC DAMAGE SPECTRUM

Computing potential seismic damage is of great importance in the damage-based seismic assessment and design of structures in earthquake-prone areas. The degree of structural damage could be used for assessments of structure functionality and robustness, which are key parameters in structural resilience [40]. In effect, computing the level of damage could be significant in estimating the resiliency of both existing and new structures. The deformation demand of a structure subjected to an earthquake is inherently cyclic. Hence, the effects of cumulative damage resulting from a number of response cycles, yield excursions, and energy dissipation should be considered when computing the seismic structural damage. Moreover, the use of peak displacement and ductility does not consider the effects of the duration and frequency content of a possible earthquake.

The seismic damage of structures is mostly a qualitative problem in nature. However, the seismic damage index can be used to quantify the extent of damage of a structure during an earthquake. Several damage models have been proposed in the literature, and among them, the Park-Ang damage index [13] has been widely used to quantify the seismic damage of structures [4, 8, 10, 15]. The index is the linear combination of the maximum displacement and the hysteretic energy dissipation of a structure, and is expressed mathematically as follows:

$$DI = \frac{x_m}{x_y \mu_u} + \beta \frac{E_H}{x_y \mu_u F_y} \tag{1}$$

where x_m is the maximum displacement during an earthquake; x_y is the yield displacement; μ_u is the ultimate ductility capacity based on monotonic loading; F_y is the yield strength; E_H is the cumulative hysteretic energy at the end of an earthquake; and β is a positive empirical constant that depends on structural features that contribute to the rate of damage through hysteretic energy. The constant parameter β is commonly considered with values between 0.05 and 0.3 [41]. As presented in Table 1, Park et al. [42] calibrated the overall damage index with the observed damage states in different statuses.

The seismic damage could be estimated for a broad range of structures by computing the seismic damage spectrum. The spectral damage index is a function of both structural and earthquake characteristics. The most important features are arguably the fundamental period of vibration *T*, elastic spectral acceleration S_a, yield strength F_y, ultimate ductility capacity ratio μ_u, damping ratio ξ, post-yield stiffness ratio η, the positive constant of the Park-Ang damage index β and soil class *S*. Mathematically, the spectral damage can be written as:

$$S_{DI} = f(T, R_u, \mu_u, \xi, \alpha, \beta, S) \tag{2}$$

in which R_u is the response modification factor (known as *R*-factor) which is determined using the following relationship:

$$R_u = \frac{mS_a}{F_y} \tag{3}$$

where *m* is the seismic mass of the structure of interest. Despite the ductility and elastic response and design spectra, because of computing the potential seismic damage using the Park-Ang damage index directly, the effect of the cumulative effects of damage is made available in the damage spectrum.

Table 1. Interpretation of structural damage index [42]

Degree	Physical Appearance	Damage Index	Damage State
Collapse	Partial or total collapse of building	> 1.0	Loss of building
Severe	Extensive crashing of concrete; disclosure of buckled reinforcement	0.4 – 1.0	Beyond repair
Moderate	Extensive large cracks; spalling of concrete in weaker elements	0.25 – 0.4	Repairable
Minor	Minor cracks; partial crushing of concrete in columns	0.1 – 0.25	Minor damage
Slight	Sporadic occurrence of cracking	< 0.1	No damage

3. COMPUTATIONAL INTELLIGENCE METHODS

Typically, the wording *computational intelligence* denotes the ability of a computer program to learn a specific task from real data or experimental observations [43]. Today, researchers are confronted by complex and highly nonlinear real-world problems where mathematical and traditional modelling methods fail or perform poorly. It has been proven that computational intelligence methods, such as the nature-inspired computational methodologies, can solve these types of problems. The recent advances in computational intelligence methods have made them suitable for solving challenging problems in science and engineering. In this study, two of the most well-known methods, ANN and GP, are applied to a case study in the field of structural and earthquake engineering and their results are presented, assessed and compared.

3.1. Artificial Neural Networks (ANNs)

In the early 1940s, a computational model was first developed for neural networks by McCulloch and Pitts [44], who simulated a biological nervous system that resulted in emerging ANNs. These artificial networks consist of artificial neurons that work in a manner similar to the elementary functions of a biological neuron. The structure of ANNs is such that they are able to learn based on experiences they gain and to generalize based on making a relationship between the inputs and outputs. Without the need for a predefined mathematical equation between model inputs and outputs, ANNs only use the data to determine the structure of the model and the unknown model parameters. They are able to learn, adapt and improve as more data become available without having to repeat the learning process from the beginning, and hence overcome the considerable limitations of conventional methods. Rapid mapping of a given input into the desired output with the smallest error can be provided by a trained ANN, which can thus be used to improve the computational efficiency of a numerical analysis process.

An ANN architecture comprises a set of processing elements usually arranged in a set of sequential layers including an input layer, one or more hidden layers and an output layer. The neurons in the input layer distribute the input signal x_i to the neurons of the next layer. The neurons in the input layer weight the received signal x_i and all the weighted signals are subsequently summed up. A bias b, which plays an important role, is added to the summation. The resulting summation process with the added bias can be mathematically expressed as [43]:

$$net = \left(\sum_{i=1}^{n} w_i \, x_i \right) + b \tag{4}$$

In order to allow for varying input conditions and their effect on the output, it is usually necessary to include a transfer function (or activation function) f (.) in the neuron arrangement. The activation function is a monotonically increasing, continuous, differentiable function, applied to the weighted input of a neuron to produce the final output. This is so that the adequate levels of amplification may be used where necessary for small input signals, which avoids the risk of driving the output to unacceptable limits. The output of neuron y, is then computed as follows [43]:

$$y = f(net) \tag{5}$$

One of the common classes of ANNs is the multilayered perceptron (MLP) networks in which feedforward architectures are used, and the backpropagation algorithm is commonly used for training. This type of ANNs is capable of accurately predicting any continuous function [43]. An input layer, one or more hidden layers and an output layer are the components of an MLP network. Each unit of these networks is associated with weighted connections to the units of the subsequent layer. The output is obtained by passing the sum of the products of the inputs and weights through a nonlinear activation function. The use of hidden layers and nonlinear activation functions enhance the ability of ANNs to learn the complicated relationship between a set of input and a set of output data. The connection between neurons of each layer is called a link. This link is stored as a weighted value that provides a measure of the connection between two nodes [45]. The supervised learning step changes these weights (and biases) in order to minimize a selected error function, generally using the mean squared error (*MSE*) approach.

Each layer includes a weight matrix ***W***, a bias vector ***B*** and an output vector ***Y***. This relationship could be mathematically expressed as follows [43]:

$$\boldsymbol{Y} = f\left(\boldsymbol{W}\ \boldsymbol{X} + \boldsymbol{B}\right) \tag{6}$$

The selection of the network architecture (or topology), i.e. the number of neurons, layers, connections and choice of transfer functions, is usually fixed from the beginning and it is largely determined by the characteristics of the problem being solved.

3.2. Genetic Programming

The use of conventional techniques such as regression analysis cannot guarantee that an accurate and reliable model for predicting the behavior of complex problems will be found [24]. Because of the available uncertainties in structural and earthquake engineering problems, determining the response of structures under an earthquake is a complex problem [25].

Genetic programming (GP), an efficient behavioral modeling methodology, was firstly proposed by Koza in 1990 [46]. It is an extension of genetic algorithm in which the relation of data, between inputs and outputs, could be formulated using a tree structure technique. Unlike ANNs and classic regression models, GP is capable of formulating the relationship regardless of a predefined structure. GP and its variants have been successfully used to solve numerous engineering problems (e.g. [23, 25, 47]. Multi-gene genetic programming (MGGP), one of the variants of GP, was proposed by Searson et al. [48] to improve the performance of the standard GP, where the capability of conventional regression is added. Studies have shown that MGGP outperforms the standard GP in model prediction [25].

3.2.1 Multi-Gene Symbolic Regression

MGGP is one of the efficient variants of GP [21, 22, 48]. It is used to predict mathematical models that are inherently multi-gene and consists of a linear combination of low order nonlinear transformations of the input variables. MGGP uses a single GP particle swarm model selection program consisting of a number of genes in which each gene has a tree expression with some relatively simple, fixed-depth sub-models [49].

In order to develop a population of genes, the symbolic regression method is implemented using standard GP in which a symbolic mathematical expression is directly encoded by each of the genes. Figure 1 shows a typical multi-gene model with three input variables (x_1, x_2, and x_3). The overall model consists of some coefficients (g_0, g_1, and g_2) utilized for a weighted linear combination of genes, although the nonlinear terms such as “sin” and “log” are used. The model is generally known as multi-gene symbolic regression, and it can be mathematically written as [24]:

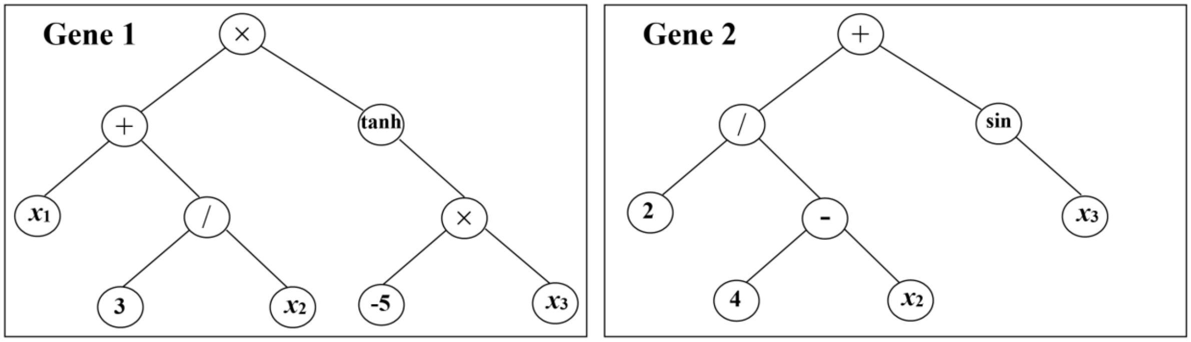
$$\hat{y}(\boldsymbol{x},\boldsymbol{g},\boldsymbol{\theta}) = g_o + \sum_{i=1}^{n} \left(g_i G_i(\boldsymbol{\theta},\boldsymbol{x}) \right) \tag{7}$$

where g_o is a bias term; g_i is the gene weight; $G_i(\boldsymbol{\theta},\boldsymbol{x})$ is the output vector from the ith gene encompassing a multi-gene individual; $\boldsymbol{\theta}$ is the vector of the unknown parameters for each gene; and n is the number of genes. The algorithmic structure of GP and MGGP is the same in general, except for crossover and mutation of multi-gene individuals. MGGP is more accurate than the standard GP in solving complex nonlinear problems [21, 22] since it does not include a simplifying assumption in the model development process.

To construct an initial population in MGGP, random individuals are generated by means of different nonlinear functions, input variables, and a range of random constants, where each individual can contain between 1 to G_{max} genes. In the algorithm, an attempt is made to maximize diversity, provided that the individuals do not include the same genes. The genes are randomly selected, and the vector of unknown coefficients $\boldsymbol{g}$ is estimated by the least-squares normal equation as follows [48]:

$$\boldsymbol{g} = \left(\boldsymbol{G}^T \boldsymbol{G} \right)^{-1} \boldsymbol{G}^T \boldsymbol{y} \tag{8}$$

in which $\boldsymbol{G} = [1\ G_1 \ldots G_\mathrm{n}]$ is the gene response matrix. Since the columns of matrix $\boldsymbol{G}$ can be collinear, the Moore–Penrose pseudo-inverse $(\boldsymbol{G}^T\boldsymbol{G})^{\#}$ can be calculated through the singular value decomposition instead of the standard matrix inverse $(\boldsymbol{G}^T\boldsymbol{G})^{-1}$. During the MGGP run, genes can be employed or eliminated by a tree crossover operator (a high-level crossover) whilst this action is performed in addition to the sub-tree crossover in the standard GP (a low-level crossover). The low-level crossover chooses a gene randomly from each parent individually. The standard sub-tree crossover is then used, and the created trees will replace the parent trees in the otherwise unaltered individual in the next generation. The high-level crossover allows the interchange of one or more genes with another selected individual provided that the maximum number of genes of an individual is G_{max}. Thus, if the number of genes passes G_{max}, more genes are randomly selected and removed [50].



$$\hat{y} = g_0 + g_1\times[x_1+3/x_2]\times\tanh(-5x_3) + g_2\times[2/(4-x_2)+\sin(x_3)]$$

Figure 1. An example of a multi-gene mathematical model [24].

3.2.2. Multi-Objective Genetic Programming (MOGP)

The overall process of both standard GP and MGGP is structured based on a single objective optimization problem for each individual in which a fitness function is considered. For a symbolic regression, the goodness-of-fit to the training data is considered as a single objective to be maximized. Due to this maximization, the optimization procedure could result in overly complex, impractical and non-robust models [24]. In addition, for MGGP, additional genes may be employed by a multi-gene model, which has a minor effect on the final model performance. The first and simple solution could be provided by limiting G in a model to G_{max}, which is a hard-to-determine unique value for any given problem parall].

As the main goal is to establish a model with high accuracy and low complexity, the use of the multi-objective concept in the multi-gene symbolic regression was proposed by Searson [50]. As such, both the goodness-of-fit and the complexity of candidate models can be optimized simultaneously by searching what is known as the Pareto front (non-dominated solutions) set. The methodology is commonly referred to as multi-objective genetic programming (MOGP). In this paper, the GPTIPS 2 toolbox [50], associated with the related subroutines coded in MATLAB [51], is used to solve the MOGP task by using a non-dominated sorting technique [52]. The non-dominated solutions are sorted based on their complexity accuracy and exactness by using the non-dominating sorting technique at the end of each generation of the MOGP algorithm. The non-dominating sorting technique is described in detail in [50, 53].

3.2.3. Accelerating GP Process

The term “Big data” can be used to define data sets that are very large, or when the data mining is complex enough to mean that traditional methods either cannot handle it or they are not practical to employ [24]. In engineering studies and especially in experimental results, the data sets are generally not very large, but they are typically complex [54]. Although successful applications of GP techniques in modelling engineering systems have been reported in the literature, there are some difficulties because of the big data problem. The evolutionary approaches are often time-consuming compared with statistical data mining approaches (e.g. regression analysis). GP is one of the slowest among them because it is used to find the structure of a model. The extra process of non-dominated sorting of a MOGP accentuates the problem. In this paper, in order to overcome this shortcoming, the following two solution strategies are used in the prediction procedure:

- 60% of the data were randomly selected and used for the training process, and the rest are used as a testing set for each run.
- The final Pareto front was determined by merging the Pareto fronts for all runs.

Two classes of machine learning algorithms are trajectory-based algorithms and population-based algorithms in general. The trajectory algorithms (for example, ANNs and regression analysis) deal with a single solution during the learning process. Unlike them, GP, a population-based algorithm, deals with a set of solutions (programs) in each generation. This feature makes GP suitable for parallel processing. Therefore, the performance of the MOGP procedure could be accelerated by parallelizing the computations using a distributed computing machine that makes it able to deal with Big Data in GP. The schematic of parallel processing in the GP process is shown in Figure 2. Herein, *twelve* cores were only used to evolve and evaluate new models while this number could be increased up to the population size using this framework. As shown in Figure 2, the population is split evenly among the available cores (nodes). For larger problems, the GP system can be parallelized by Cloud computing in which Cloud nodes replace computer cores. Further information about the parallel processing of multi-gene MOGP used in this study and GP in general in a distributed system can be found in [53] and [54], respectively.

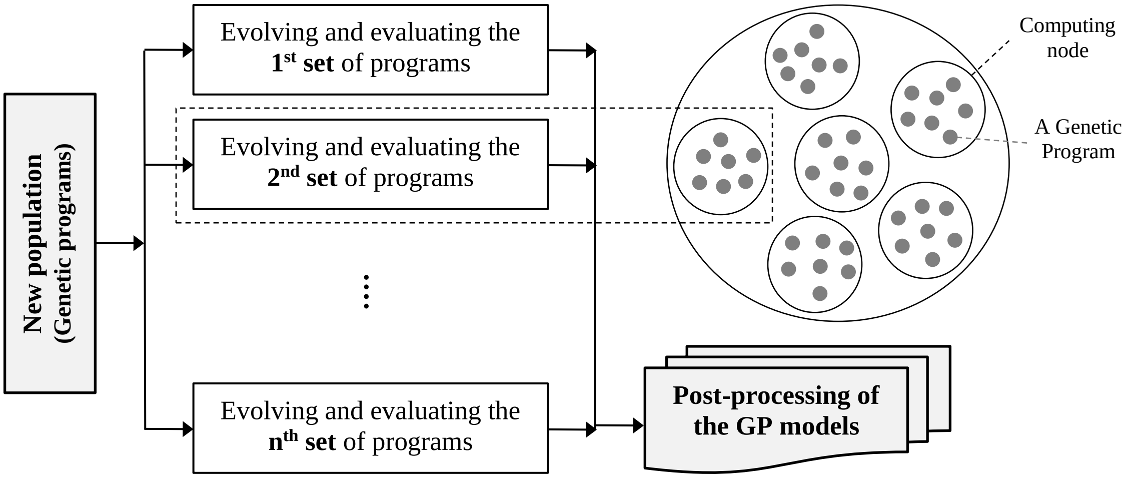


Figure 2. A parallel processing schematic in the GP.

4. PREDICTING SEISMIC DAMAGE SPECTRA

4.1. Exact Database

The structural and earthquake ground motion properties could greatly (or poorly – as investigation is needed) affect the inelastic responses of an SDOF system. In this paper, these properties, which are frequently used for seismic damage evaluations of structures, are used as the input variables of the prediction models, and they are described in detail in the next subsections.

4.1.1. Inelastic SDOF Systems

The period (T), hardening ratio of hysteretic behavior (α), damping ratio (ξ), response modification factor (R_u), the ultimate capacity ductility ratio (μ_u), and the constant of the Park-Ang damage index (β) are the structural properties used to determine the spectral damage. The period's range studied for the prediction of the seismic damage spectra is between 0.1 to 6.0 s with a step of 0.1 s. The values assumed for α are 0, corresponding to the elastic-perfectly-plastic (EPP) model, and 0.05 and 0.1 indicating bilinear models. The values used for the other parameters can be found in Table 2. The assumed values for the considered variables (T, α, ξ, R_u, μ_u and β) resulted in a total number of 60×3×3×3×3×3=14,580 individual inelastic SDOF systems used for the prediction of the S_{DI}. It should be noted that only T, α, ξ and R_u are used in the structural analyses (i.e. 60×3×3×3=1620 analyses for each earthquake ground motion record) while μ_u and β are only used for post-processing of the spectral damage computation.

Table 2. Structural properties of the inelastic SDOF systems and other parameters used

Parameter	Values used		
Period (T)	From 0.1 s to 6.0 s, with step 0.1 s (60 values)		
Hardening ratio (α)	0	0.05	0.1

Damping ratio (ξ)	0.02	0.05	0.1
Response modification factor (R_u)	2	4	6
Ult. capacity ductility ratio (μ_u)	6	8	10
Constant of Park-Ang DI (β)	0.05	0.15	0.3

4.1.2. Earthquake Ground Motions

Three factors of the site class, source-to-site distance, and S_a are the three variables considered for the earthquake ground motion records used. Based on the shear wave velocities corresponding to the 30 m in depth ($V_{s,30}$) of more than 750, 360 to 750, 180 to 360 and less than 180 m/s, four soil types (S1, S2, S3, and S4) were assumed for the records used. The soil types S1, S2, S3, and S4 correspond to the soft, firm, stiff and rock soil types, respectively. To consider the source-to-site distance, the records having the Joyne-Boor distance (R_{JB}) in the range of more than 17.5 km and less than 150 km were used (known as far-fault records). All records are non-pulse-like and correspond to events with a magnitude (M) greater than 5.5. The NGA-West-II project of the PEER ground motion database [55] was used for the selection of the records. The diversity of M, R_{JB} and $V_{s,30}$, and the number of records are shown in Figure 3. As shown in this figure, 281 records were selected in such way that they cover a large variety of the mentioned properties. In addition, they have been differentiated on the basis of their allocated soil types.

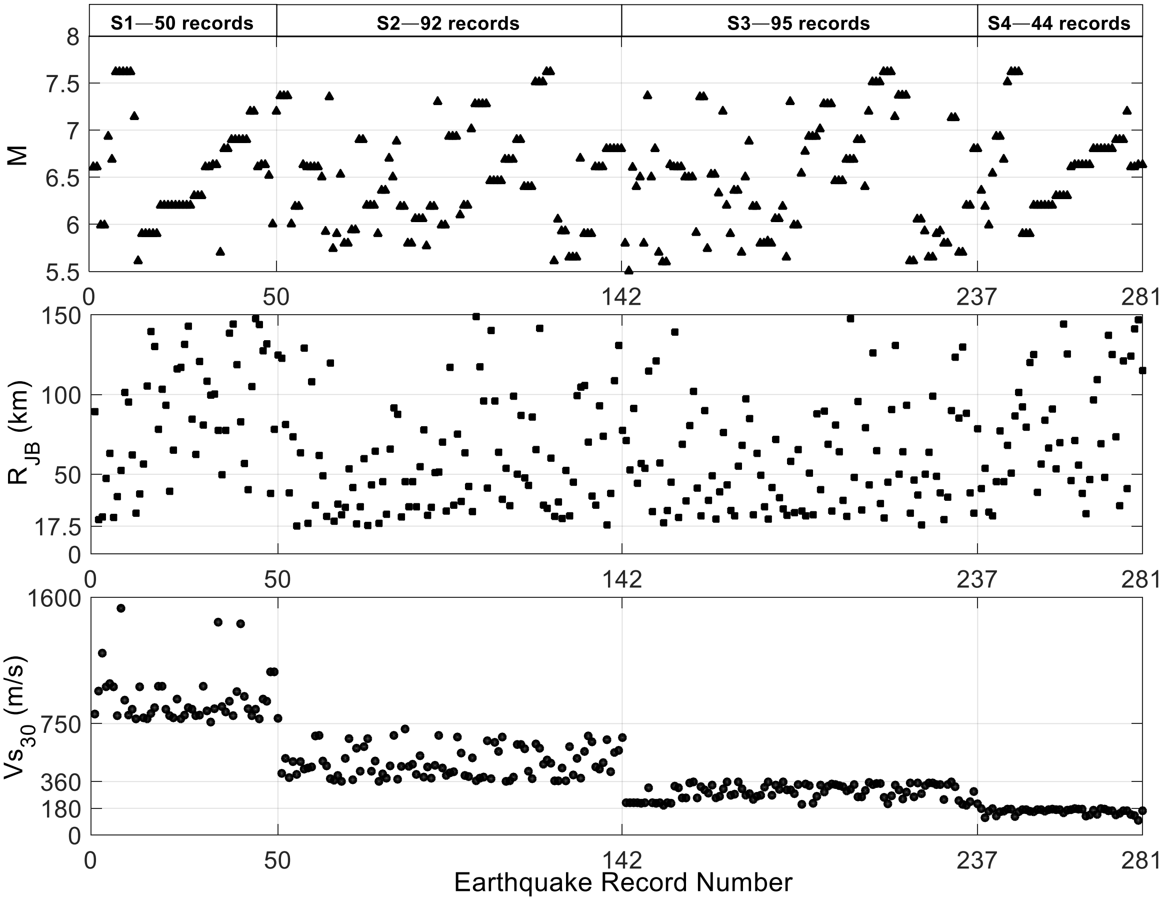


Figure 3. Parameters M, R_{JB} and $V_{s,30}$, and the number of records used

Finally, based on the values assumed for the structural variables and the four sets of earthquake records, a large number of inelastic time history dynamic analyses (more than 455,000) were carried out, and the exact values of S_{DI} were determined. The entire process was simulated in MATLAB [51].

In order to compute the damage spectra, the individual spectral damage responses of the SDOF systems under each earthquake record (of a set of interest) were firstly computed. Then, based on the normal distribution, the mean spectral damage of each set of the earthquake records was determined, to be predicted later on.

4.2. Damage Assessment

In this section, the exact results of the Park-Ang damage index are presented, and the effects of the variables R_u, ξ, μ_u, η, β and soil type on the S_{DI} are investigated. It should be noted that for all the spectral damages presented herein, such as, for example, the investigation of the effects of R_u, the spectral damage presented is the mean across all the other variables. Figure 4 shows the damage spectra in terms of all the mentioned variables. To show the changes clearly, the vertical axis is arranged in a logarithmic scale.

The variations of S_{DI} for the three values of R_u equal to 2, 4 and 6 are depicted in Figure 4(a), where it can be seen that the spectral damage increases in step with increasing R_u for all the structures with short-to-long periods. It could be said that, because of decreasing F_y, higher degrees of nonlinearity are experienced at the same level of earthquake intensity, and the induced damage is then increased with increasing R_u. The variations of S_{DI} from the structures with an R_u of 2 to 4 is much greater than those from 4 to 6.

Figure 4(b) shows the variation of S_{DI} for three values of μ_u equal to 6, 8 and 10. As can be seen, the spectral damage is decreased when the ductility ratio increases. This is because structures with higher ductility have more deformation capability and energy dissipation capacity, resulting in a lower degree of damage.

The effect of the damping ratio of the SDOF systems on the S_{DI} is discussed herein. The three values considered for the damping ratio are 0.02, 0.05 and 0.1, which are the common values used in the analysis of engineering structures [56]. This effect is shown in Figure 4(c). A clear relation between them is that S_{DI} increases as the damping ratio ξ increases. This effect has been confirmed in detail by the results found by Zhai et al. [11]. As they have also noted, increasing the damping ratio leads to a decrease in spectral acceleration (S_a), and at a given R_u, in accordance with Eq. (3), the F_y of the SDOF system will decrease, resulting in increased damage. As shown in Figure 4(c), the differences due to the effect of the damping ratio on S_{DI} is decreased as the period of structures is increased. Moreover, by increasing the damping ratio, a greater range of short-period structures experience collapse status.

Herein, the effects of the hardening ratio (α) of bilinear hysteretic behavior of SDOF systems on the S_{DI} are studied. The ratio is equal to the post-yield stiffness of the bilinear model to the initial one. Three values of 0.0, 0.05 and 0.1 are considered for α, where 0.0 indicates the EPP behavior model. The damage spectra for different α values are shown in Figure 4(d), where the overall trend of damage spectra is seen to descend, and a greater hardening ratio leads structures to exhibit a lower degree of damage. By considering 0.05 as the hardening ratio for the behavior model, the spectral damage is significantly decreased, and by increasing the hardening ratio from 0.05 to 0.1, the spectral damage is slightly decreased. The former difference is almost zero for the short-period structures, particularly those with a period of less than 0.6 s.

A positive constant, β, is used to combine two terms of the Park-Ang damage index by which the portion of hysteretic energy is contributed. Several values have been proposed for β, which are common in the range of 0.05 to 0.3. Figure 4(e) shows the variation of S_{DI} with respect to β. As expected, S_{DI} increases as β increases.

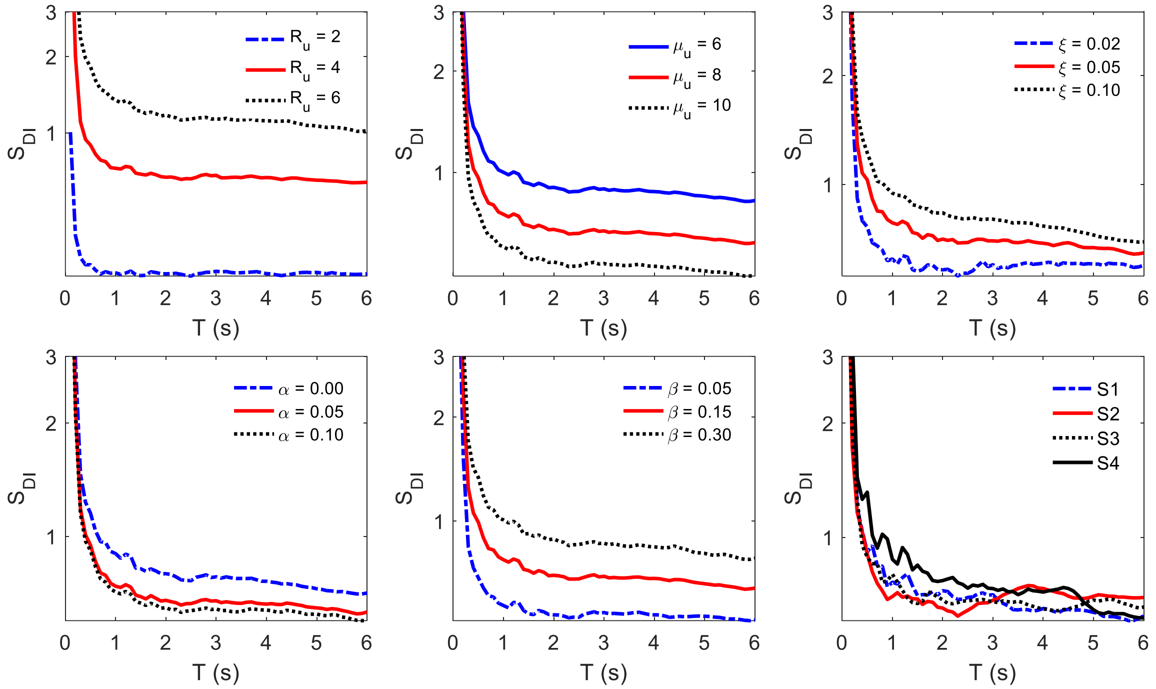


Figure 4. Damage spectra for the variables: (a) R_u, (b) μ_u, (c) ξ, (d) α, (e) β, (f) S1 to S4 (Note: the vertical axis is in logarithmic scale)

The effects of soil types S1 to S4 were also captured and are shown in Figure 4(f). As can be seen, the effect of soil type on S_{DI} is not as significant as the other abovementioned parameters. In general, except for a limited region of long-period structures, S_{DI} for S4 is more than that for S1, S2 and S3. In addition, for most of the periods in the range of short and medium, S_{DI} corresponding to S1 is higher than that for S2 and S3 and vice versa for long-period structures. Moreover, as a general trend, the value of S_{DI} decreases as the period increases, in all cases.

4.3. Model Development Using MOGP

An efficient model needs appropriate parameters to be utilized as a part of the MOGP predictive algorithm. To develop an optimum MOGP model, some mathematical functions (e.g. *tanh*, *ln*) and basic arithmetic operators (+, -, ×, /) have to be used. To form the model, the elements chosen from the functional set and the terminal set are combined randomly. The population size is used to determine the number of programs in the population, and the number of levels (which the algorithm would apply before it terminates) is determined by the number of generations. These important values are set based on three factors: the nature of the data set; the number of variables; and the problem complexity. To restrict the complexity of the models, the parameters G_{max} and D_{max} need to be defined, which respectively are the upper bounds of an individual and the maximum tree depth. The values are obtained from a trade-off between the running time and the complexity of the evolved solutions.

Herein, a "rate-based high-level crossover" is employed with the aim of generating new genes for individuals, decreasing the overall number of genes for one model and increasing the total number of genes for the other. A crossover rate parameter (CR) is used to capture the process. A uniform random number (r_0) in the range [0, 1] with a mean value of 0.5 is produced separately for each gene in the parents. If $r_0 < CR$, the corresponding gene is transferred to the other individual. If an exchange of gene results in offspring that violate G_{max}, the gene is deleted to satisfy the constraint [50]. For the analyses and to develop the models, data are split into a training subset and a validating subset. To fit the genetic model evolution, the training subset is used only for the learning, and the performance of the evolved programs on unseen data is measured by the performance of the algorithm on the validation subset. Note that the validation subset is only used for validation and plays no role in building the models.

A consistent data division is obtained by considering different combinations of training and testing sets. To designate the best-encoded expression, the evolved expressions are evaluated by the fitness function. For the MOGP, the minimum of the root mean square error (*RMSE*) is used as the fitness function expressed as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(e_i - p_i\right)^2}{n}}, \tag{9}$$

in which, p_i and e_i are the predicted, and exact output values for the *i*-th output, respectively; and *n* is the number of samples. In the MOGP approach, the two objectives of maximizing the correlation coefficient (*R*) and minimizing the model complexity are considered for selecting the best final predictive model.

4.4. Model Development Using ANNs

In order to develop a predictive model using ANN, some main factors need to be addressed. In general, this comprises choosing the data set, data preprocessing, selecting adequate input variables, determining an appropriate network architecture, assigning convergence criteria, choosing the optimization method and model validation. One of the most important among these factors for improving the model performance is the selection of a network architecture that is compatible with the input and output data and the nature of the problem.

The ANN model development procedure starts by inputting a training data set into the input layer. A learning rule is then used to find a set of weights by producing an input-output mapping with the smallest error. After successfully implementing the training phase, another data set (i.e. the testing data set) is applied to assess the performance of the developed model.

In this paper, three training methods, namely *trainlm*, *trainscg* and *trainbr* were used:

- *trainlm* is a network training function that updates weight and bias values according to the Levenberg-Marquardt optimization [57].
- *trainscg* is a network training function that updates weight and bias values according to the scaled conjugate gradient method.
- *trainbr* is another network training function that updates the weight and bias values according to the Levenberg-Marquardt optimization. However, it minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. This process is called Bayesian regularization.

Determining the optimum value of the performance ratio parameter is a challenging problem during the regularization process. The large values of this ratio can result in overfitting, which leads to a fitting of the noise and a loss of generalization of the network. Hence, there is a need to automatically find the optimum values. The regularization parameters are automatically set by the Bayesian regularization function, which is related to the unknown variances associated with these distributions. This reduces the potential for overfitting by which a nonlinear system is converted into the "well-posed" statistical problems [37, 38]. One basic feature of Bayesian regularization is providing a measure for recognizing the effective network parameters, including weights and biases. The weights are considered as the random variables in the Bayesian network, and their density function is thereby written according to the Bayes' rules. A detailed description of the process can be found in [58].

4.5. Model Accuracy

It should be noted that all the models can be valid in the range of the actual data used for the model generation. In order to investigate the effectiveness and accuracy of the models in the range of our database, common performance metrics, including the mean absolute percentage error (*MAPE*), relative root mean square error (*RRMSE*), linear correlation coefficient (*R*), performance index (*PI*), coefficient of efficiency (*E*), and the index of agreement (*d*) corresponding to the predicted formulation of each EDP are obtained. The equations for calculating these quantities are expressed as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{e_i - p_i}{e_i}\right| \tag{10}$$

$$RRMSE = \frac{1}{|\overline{e}|}\sqrt{\frac{\sum_{i=1}^{n}(e_i - p_i)^2}{n}} \tag{11}$$

$$R = \frac{\sum_{i=1}^{n}(e_i - \overline{e}_i)(p_i - \overline{p}_i)}{\sqrt{\sum_{i=1}^{n}(e_i - \overline{e}_i)^2 \cdot \sum_{i=1}^{n}(p_i - \overline{p}_i)^2}} \tag{12}$$

$$PI = \frac{RRMSE}{R+1} \tag{13}$$

$$E = 1 - \frac{\sum_{i=1}^{n}(e_i - p_i)^2}{\sum_{i=1}^{n}(e_i - \overline{e}_i)^2} \tag{14}$$

$$d = 1 - \frac{\sum_{i=1}^{n}(e_i - p_i)^2}{\sum_{i=1}^{n}\left(|e_i - \overline{e}_i| + |p_i - \overline{e}_i|\right)^2} \tag{15}$$

where $\overline{e}_i$ and $\overline{p}_i$ are the average values of the exact and predicted outputs, respectively; and n, $e_{i,}$ and p_i have been predefined. Lower *MAPE* and *RRMSE* values as well as higher *R* (or R^2), *E,* and *d* indicate the accuracy and effectiveness of the prediction model used. Based on Eq. (13), higher *R* values and lower *RRMSE* values result in lower *PI* and, subsequently, indicate a more precise model. It should be noted that the *PI* ranges from 0 to ∞, and values close to 0 indicate that the model fits very well to the exact (actual) values.

It is worth noting that two sources of complexity affect the accuracy of the models: Firstly, the behavior of the structures under consideration as they experience inelastic deformations with high nonlinearity, and secondly, the nature of earthquake excitations includes some effective characteristics, such as frequency content, which make a structure experience different cyclic excursions associated with complex behavior [25].

5. RESULTS AND DISCUSSION

5.1. MOGP predictive model

The parameter settings used for the MOGP are listed in Table 3. The values were obtained based on the authors' experience, the suggestions available in the literature [48-50] and the employment of a case-dependent trial-and-error process. Finally, using the MOGP, the S_{DI} was predicted, and its optimal mathematical model was determined.

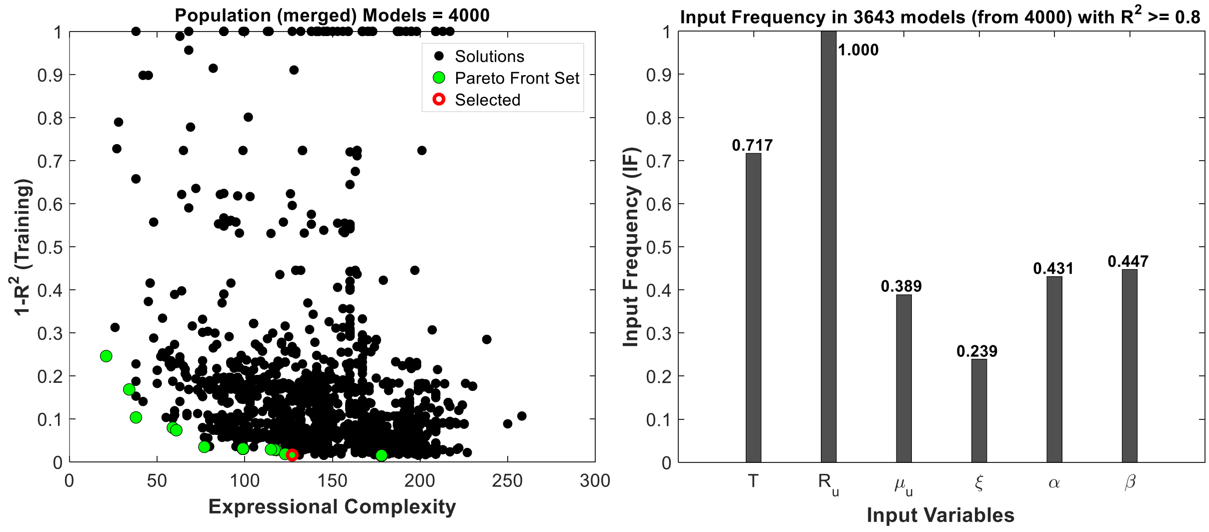
Four cases based on the different soil types (S1 to S4) were considered in the model development process. Therefore, in this paper, the complete database (which includes four soil types, S1 to S4) is employed in order to develop a unique prediction model for S_{DI}. The linear regression coefficients are then updated for each soil type. The final mathematical model is selected based on a compromise between the prediction accuracy (as measured by the correlation coefficient *R*) and the model complexity (as measured by the number of input variables). After that, the complete database is divided into four groups based on soil types S1 to S4. Using each group of data, the predicted coefficients of the final model are re-evaluated by conducting a regression analysis to reflect the influence of the soil types. Finally, a unique mathematical model is presented containing the structural variables (and PGA of earthquake records) with an application to all soil types (S1 to S4), with four different groups of coefficients. The participation of each input variable in the mathematical model and the Pareto front obtained by using a non-dominated sorting method is also presented.

Twenty independent GP runs with random initial populations were conducted, and the final populations are merged in a pool with 4,000 models (20×200=4,000). The results of all models developed by MOGP for S_{DI} are shown in Figure 5(a). The Pareto front sets are shown with green circles, and the rest of the models are shown in solid black circles. As mentioned earlier, the Pareto front set is obtained by using a non-dominated sorting of populations at the end of all MOGP runs. This process simultaneously optimizes the accuracy and complexity of all developed models. The final model in the Pareto front set is selected and highlighted in a red circle.

Table 3. Parameter settings for the MOGP algorithm

Parameter settings	Setting
Function set	+, -,×,/,√ , ^2, ^3, exp., ln, tanh, mul3[a]
Population size	100 – 500 (**200**[b])
Number of generations	10,000
Maximum number of genes allowed in an individual (G_{max})	3 – **4**
Maximum tree depth (D_{max})	5
Tournament size	10% of the population
Pareto Tournament	30% of tournaments
Crossover events	0.85
High-level crossover	0.2
Low-level crossover	0.8
Mutation events	12
Sub-tree mutation	9
Replacing input terminal with another random terminal	0.05
Gaussian perturbation of randomly selected constant	0.05
Direct reproduction	0.05
Ephemeral random constants	[-10,10]

[a] mul3 means the product of three factors; [b] Bold set is the final set



(a) (b)

Figure 5. (a) Pareto front, and (b) Frequency of the model developed by MOGP

The frequencies of each input variable measure the contribution of each input variable in the mathematical prediction models. The maximum contribution of each input variable corresponds to the frequency value of 1.0, indicating that it has the maximum contribution within the best-generated models [59]. It was assumed that the models with R^2> 0.8 are the best-generated models. The frequency histograms of the input variables for the predicted spectral damage are shown in Figure 5(b). As shown, for the selected database, R-factor and ξ, respectively, show the most and the least statistically significant contributions in the best-generated MOGP models.

The mathematical model obtained for S_{DI} is expressed as follows:

$$S_{DI} = c_0 + c_1G_1 + c_2G_2 + c_3G_3 + c_4G_4 \, , \tag{15}$$

where the parameters G_1, G_2, G_3 and G_4 are expressed as

$$G_1 = \frac{\beta + \sqrt{\xi}}{\mu_u + 2\alpha}\left(\ln(R_u)\right)^2 , \tag{16-a}$$

$$G_2 = \frac{R_u^2}{\left(\mu_u + \xi + 6.398\right)T^4} , \tag{16-b}$$

$$G_3 = -\frac{R_u^{\,2}\tanh\left(\alpha\right)}{T^{\,3}} , \tag{16-c}$$

$$G_4 = \frac{\beta R_u^3}{\mu_u T^{\,2}} \tag{16-d}$$

and c_0 is the bias term, while c_1 to c_4 are the gene weight for the prediction model. These coefficients are listed in Table 4.

Table 4. The coefficients c_0, c_1, c_2, c_3, c_4 used in equation (17) for each soil type

Soil Type	Coefficients – Eq. (16)				
	c_0	c_1	c_2	c_3	c_4
S1	0.130275	5.832123	0.000492	0.002978	0.035622
S2	0.111011	6.216587	0.000366	0.002046	0.026355
S3	0.095521	6.366112	0.000329	0.001634	0.024222
S4	0.090168	6.875437	0.000876	0.006611	0.056876

The correlation coefficient (R) is an essential output of the regression analysis measuring the correlation between the exact (target) and predicted data. For a more rigorous investigation, the square of R, i.e. R^2 is used herein. Figure 6 shows the regression plots for the training, test and validation data. As can be seen from all the plots, the predicted data are well concentrated around the 45° line. Moreover, the results of R^2 marked on the figure are equal to 0.984, 0.987 and 0.986, respectively, for the training, test and validation data set, demonstrating their excellent agreement with the target data.

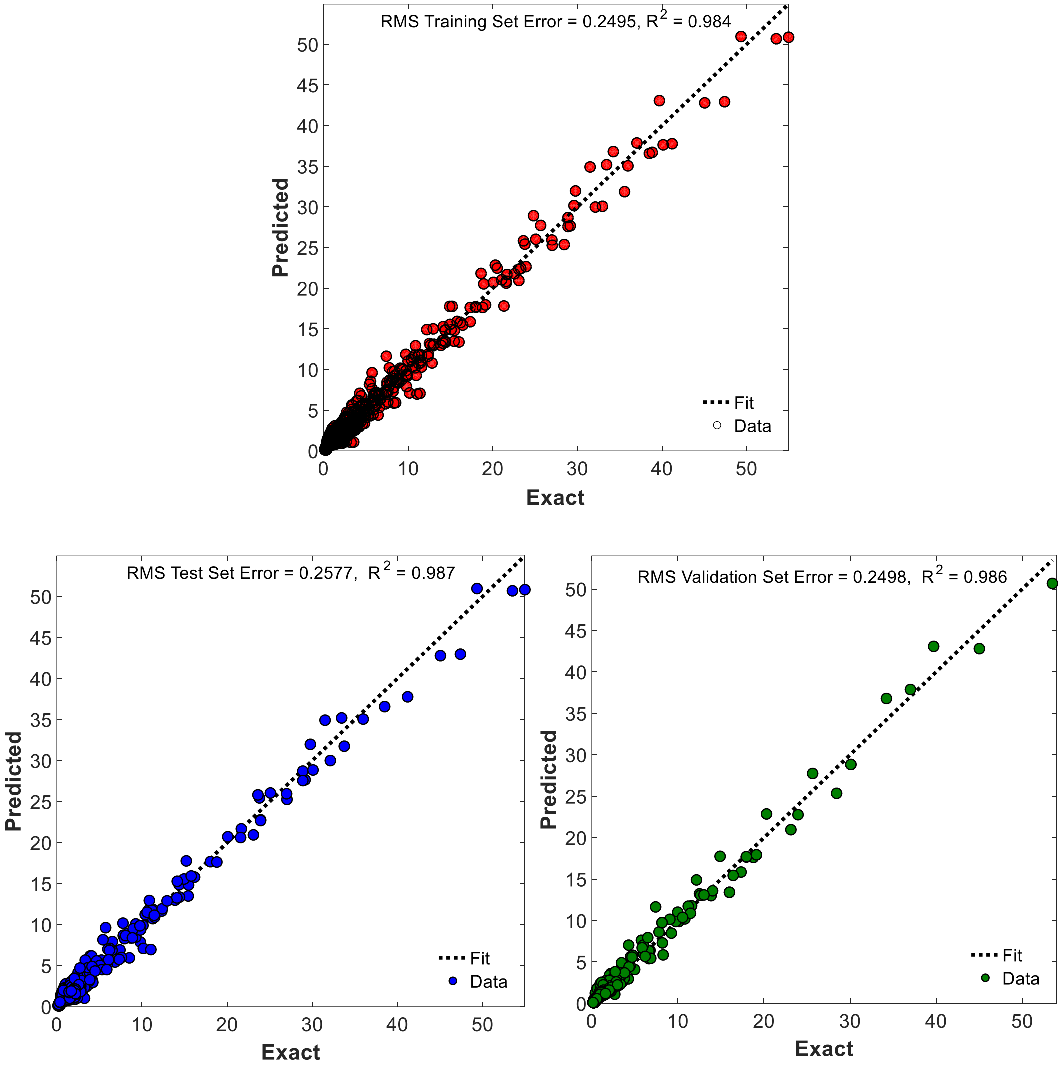


Figure 6. Regression plots for exact versus predicted results for the MOGP model

5.2. ANN predictive model

The hyperbolic tangent sigmoid transfer function and a linear transfer function were considered for the hidden layer and the output layer, respectively. To update weight and bias values, the Levenberg-Marquardt algorithm with 1000 epochs was used as the network training function. Backpropagation was used to adjust the weights and biases in the steepest descent direction to reduce prediction errors. The error corresponding to the adjusted weights and biases was then calculated and fed back to the network. The mean square error (*MSE*) with a tolerance of 10^{-5} was used as the performance function, and 85% and 15% of the data were used for training and testing the ANN model, respectively.

The number of input-output neurons is primarily determined on the basis of the requirements, but there is no certain rule for determining the number of neurons in the hidden layers. Due to the nature of the problem, all the trained networks have 6 neurons in the input layer and 4 neurons in the output layer. It was decided that the number of neurons in the hidden layer would be 30. The best number of neurons was obtained using a trial-and-error process. The criteria used to determine this number were that less time was required to achieve less error in the testing and training of the ANNs. The parameter settings for the ANN are listed in Table 5.

Table 5. Parameter settings for the ANN used

Parameter	Setting
Number of neurons in the input layer	6
Number of neurons in the output layer	4

Number of neurons in the hidden layer	1-30 (**28**[a])
Epochs	1000
Goal	0
Momentum constant	0.005
Training, validation, and testing data (%)	70, 15, 15
Training optimization algorithm	Levenberg–Marquardt
Transfer function (hidden layer)	Hyperbolic tangent sigmoid
Transfer function (output layer)	Linear
Regularization function	Bayesian

[a] Bold set is the final set

Three network training functions are implemented here to update the network weights, including Levenberg–Marquardt optimization, Bayesian Regularization, and Scaled Conjugate Gradient. Training using Scaled Conjugate Gradient was not successful, and therefore, only results of Levenberg–Marquardt optimization and Bayesian Regularization are presented in Figure 7. The results show that both algorithms have almost the same performance, and their results are converged with more than 30 neurons. Here, the best model is selected based on the training error, which has the lowest testing error. The final model is the networked trained with Bayesian Regularization and has 28 nodes.

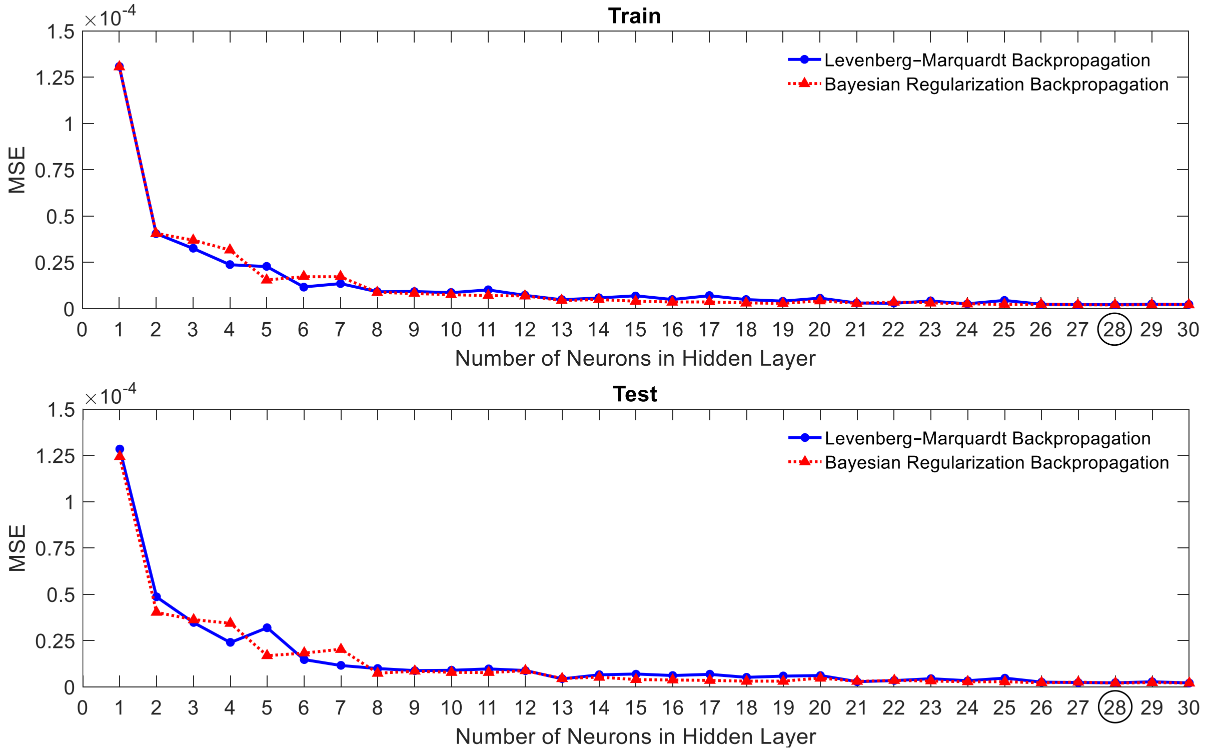


Figure 7. MSE values vs the number of hidden neurons for different training functions

Herein, the performance of the best ANN applied is evaluated through the *MSE*. Figure shows the performance curve of the developed model for predicting S_{DI}. As can be seen, *MSE* becomes small as the number of epochs is increased, and no overfitting is observed, while the training process has been performed well. The best performance in terms of *MSE* was about 0.005 and occurred at epoch 1000.

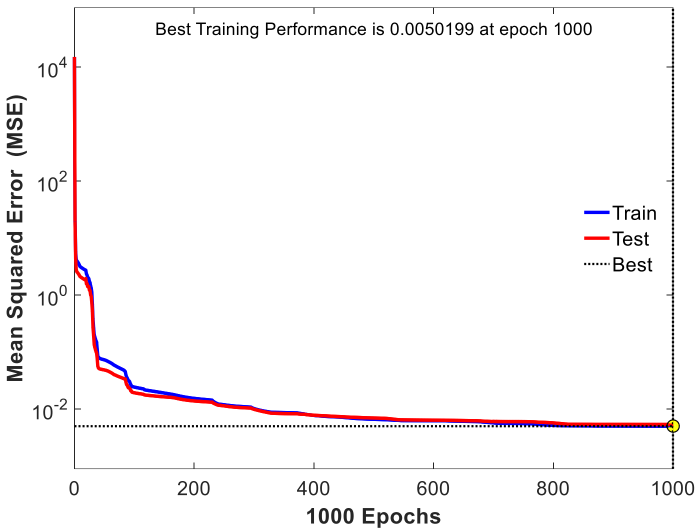


Figure 8. The ANN performance during the training step

Additional verification of the ANN performance could be provided using the error histogram, which visualizes the difference between values predicted by the ANN model and the actual values after the training. Normally, the near-zero errors indicate the better-trained model. Figure shows the error histogram plot with 20 bins for the ANN model used. As can be seen, the data-fitting errors are distributed within an appropriate range near zero, confirming the superior performance of the training step.

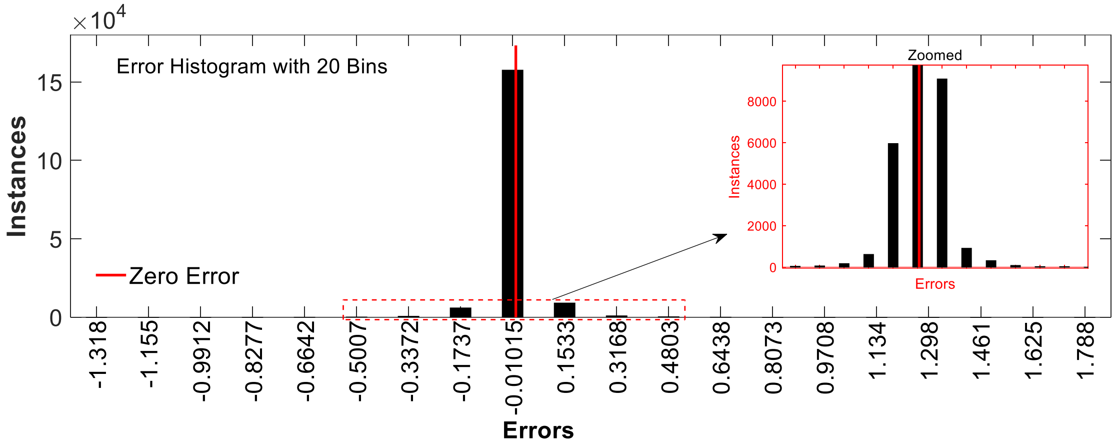


Figure 9. Error histogram of ANN (with 20 Bins) during the training step

The training state is also shown in Figure , which shows the plots of gradient, momentum and validation checks against epochs. The descending trend of gradient during epochs are obvious, indicating that the model is converging, and its performance is improving. The variation of *Mu* values, which is used to control the weights of the neuron updating process during the training, is also depicted in the figure. The final values for gradient and *Mu* are marked on the figure. In addition, validation fails during the training process are shown in Figure , indicating that there are no fails during the training. Note also that a high number of fails during the training step shows overtraining, resulting in the process being stopped.

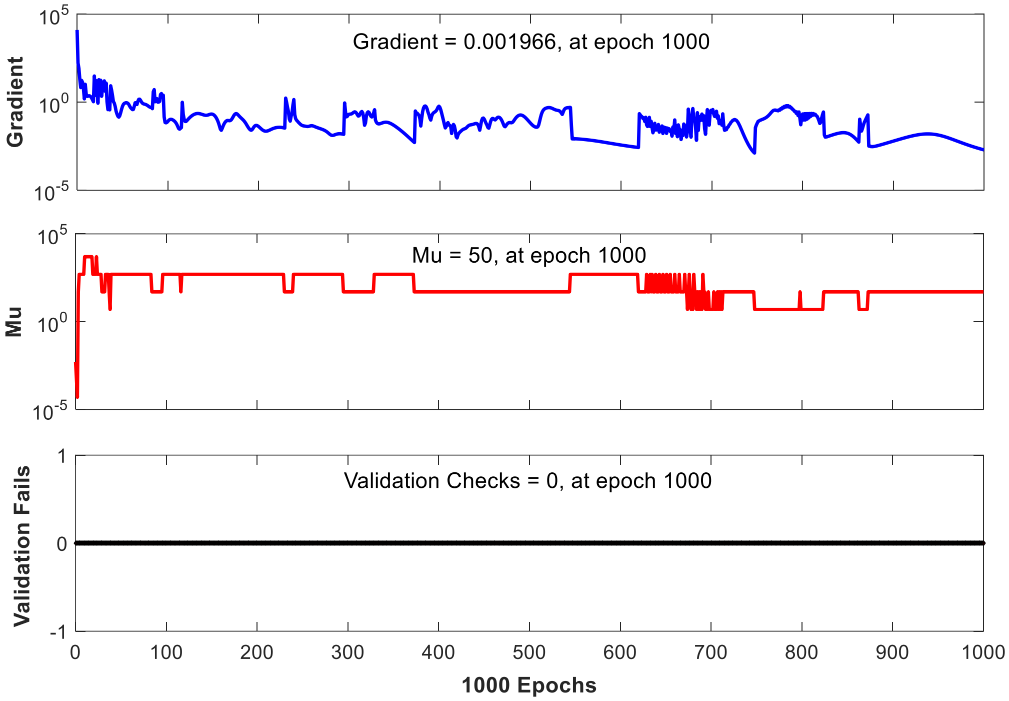


Figure 10. The training state of ANN in terms of gradient, Mu and validation checks

Figure shows the regression plots for the developed ANN model for all soil types (S1 to S4). As shown in this figure, the predicted data are well concentrated around the 45° line. In addition, the results of R^2 marked on the figure are more than 0.998 for all soil types, indicating that the predicted values are in excellent agreement with the *exact* data. In addition, it confirms the high accuracy of the developed ANN model.

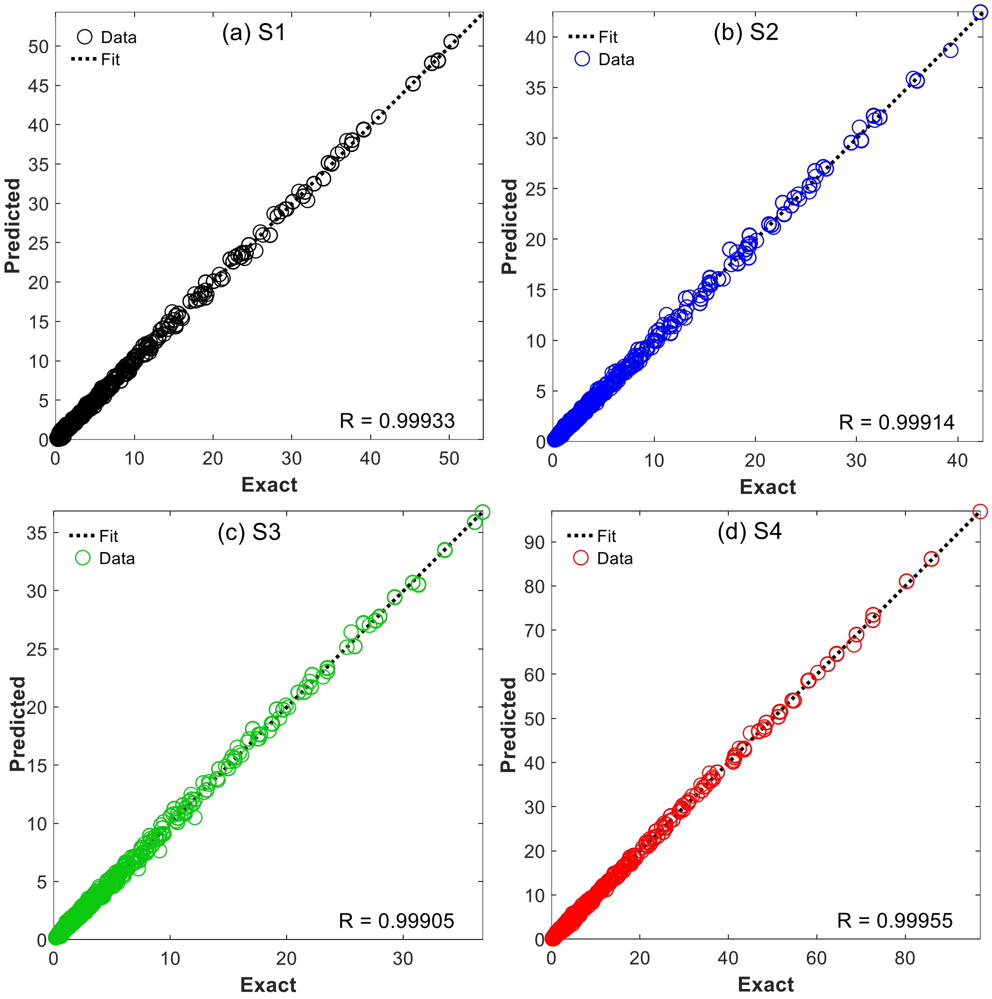


Figure 7. Regression plots for exact versus predicted results for the ANN model

5.3. Comparison

In the previous subsections, the accuracy of the predictive models developed by both MOGP and ANN was confirmed. Herein, in order to compare the models developed by the MOGP and the ANN with a model presented in the literature, a relevant model proposed by Zhai et al. [11] is applied. The model is mathematically expressed as follows [11]:

$$S_{DI,\ Zhai\,et\,al.} = \frac{1}{\mu_u} + \left(b_1 + \frac{b_2}{T^{1.5}}\right)\left(R_u - 1\right)^{b_3}\frac{10}{\mu_u} \tag{17}$$

where b_1 to b_3 are the coefficients depending on the soil types, and the other parameters have been predefined. The values proposed for the coefficients corresponding to each soil type are listed in Table 6.

To make a general comparison, the mean values of S_{DI} predicted by all of the models developed across all variables are used. The above-defined performance metrics, including *MAPE*, *RRMSE*, R^2, *PI*, *E*, *d*, and are used for the comparison. To make an informative comparison, all the performance metrics are listed in Table 7. The results are separately presented for each soil type (S1 to S4). The differences between the performance metrics of the two models are also computed and presented. As shown in this table, the results of MOGP and ANN are excellent in terms of all performance metrics, and the ANN model has higher accuracy compared with the MOGP model. Nevertheless, the mathematical relationship of equation (16) is an advantage for the MOGP model when compared to the ANN model, which provides no such elegant solution. In addition, the comparison of the performance metrics shows that both MOGP and ANN models have higher accuracy (i.e. lower *MAPE*, *RRMSE*, *PI* and higher R^2, *d*, *E*) than the Zhai et al. model. This is true for all four soil types.

Table 6. The values proposed for b_1 to b_3 used in the model developed by Zhai et al [11].

Soil Type	Coefficients		
	b_1	b_2	b_3
S1	0.08	0.02	1.25
S2	0.09	0.04	1.2
S3	0.09	0.05	1.1
S4	0.08	0.1	1.18

To make a more informative and general assessment of the proposed MOGP and ANN models, as well as the Zhai et al. model, the average values of the performance metrics of all soil types, for each of the models, are also presented in Table 7. As shown, the *MAPE*, *RRMSE* and *PI* of the MOGP model have almost 53.1, 76.1 and 77.6% lesser values, respectively, and R^2, *E* and *d* have almost 29, 98.6, and 15.7% higher values, respectively, compared with those of the model of Zhai et al. [11]. In addition, based on the table, the *MAPE*, *RRMSE* and *PI* of the ANN model have almost 83.5, 93.9 and 94.3% lesser values, respectively, while the R^2, *E* and *d* have almost 31.6, 102.6 and 16.2% higher values, respectively, compared with those of the Zhai et al. model. Finally, the higher accuracy of the MOGP and ANN models compared with the Zhai et al. model is confirmed by the performance metrics listed in Table 7. It can also be seen that the ANN model outperforms the MOGP model.

Next, a graphical comparison is made to show the varying trend of the predicted models using MOGP, and comparing the best model from the literature with the exact results. For this purpose, the input variables were selected as $R_u = 4$, $\mu_u = 10$, $\xi = 0.05$ $\eta = 0.0$ and $\beta = 0.15$. The S_{DI} corresponding to the soil types S1 to S4 are shown in Figure . The figure shows the plots of *exact* and predicted S_{DI} using the MOGP, ANN and Zhai et al. models, and all the S_{DI} curves show a descending trend. The S_{DI} corresponding to the ANN model is very close to the exact values as compared to the MOGP and Zhai et al. models. This is also true for the MOGP model when compared to the Zhai et al. model.

Table 7. Performance metrics for the ANN, MOGP and Zhai et al. models

Soil Type	Performance Metrics	Prediction models for S_{DI}			Difference (Improvement) %	
		Zhai et al. [11]	MOGP	ANN	MOGP to Zhai et al.	ANN to Zhai et al.
S1	*MAPE*	27.17	12.18	4.48	-55.171	-83.511
	RRMSE	217.77	31.91	8.36	-85.347	-96.161
	R^2	0.714	0.980	0.999	37.255	39.916
	PI	1.180	0.160	0.042	-86.441	-96.441
	E	-3.837	0.980	0.999	-74.459	-73.964
	d	0.674	0.995	0.9996	47.626	48.309
S2	*MAPE*	23.32	12.33	4.26	-47.127	-81.732
	RRMSE	91.50	27.15	7.69	-70.328	-91.596
	R^2	0.786	0.979	0.998	24.555	26.972
	PI	0.485	0.136	0.038	-71.959	-92.165
	E	0.559	0.979	0.998	75.134	78.533
	d	0.917	0.995	0.9996	8.506	9.008
S3	*MAPE*	27.52	12.12	3.72	-55.959	-86.483
	RRMSE	85.33	23.63	7.68	-72.308	-91.000
	R^2	0.787	0.982	0.998	24.778	26.811
	PI	0.452	0.119	0.038	-73.673	-91.593
	E	0.678	0.982	0.998	44.838	47.198
	d	0.931	0.995	0.9995	6.874	7.358
S4	*MAPE*	29.63	13.81	5.29	-53.392	-82.146

	RRMSE	141.47	45.46	8.88	-67.866	-93.723
	R^2	0.747	0.977	0.999	30.790	33.735
	PI	0.759	0.229	0.044	-69.829	-94.203
	E	0.626	0.977	0.999	56.070	59.585
	d	0.918	0.994	0.9997	8.279	8.900
Average values for all soil types	*MAPE*	26.91	12.61	4.44	-53.140	-83.501
	RRMSE	134.02	32.04	8.15	-76.093	-93.919
	R^2	0.759	0.979	0.999	28.986	31.621
	PI	0.719	0.161	0.041	-77.608	-94.298
	E	-0.493	0.979	0.999	98.580	102.637
	d	0.860	0.995	0.9996	15.698	16.233

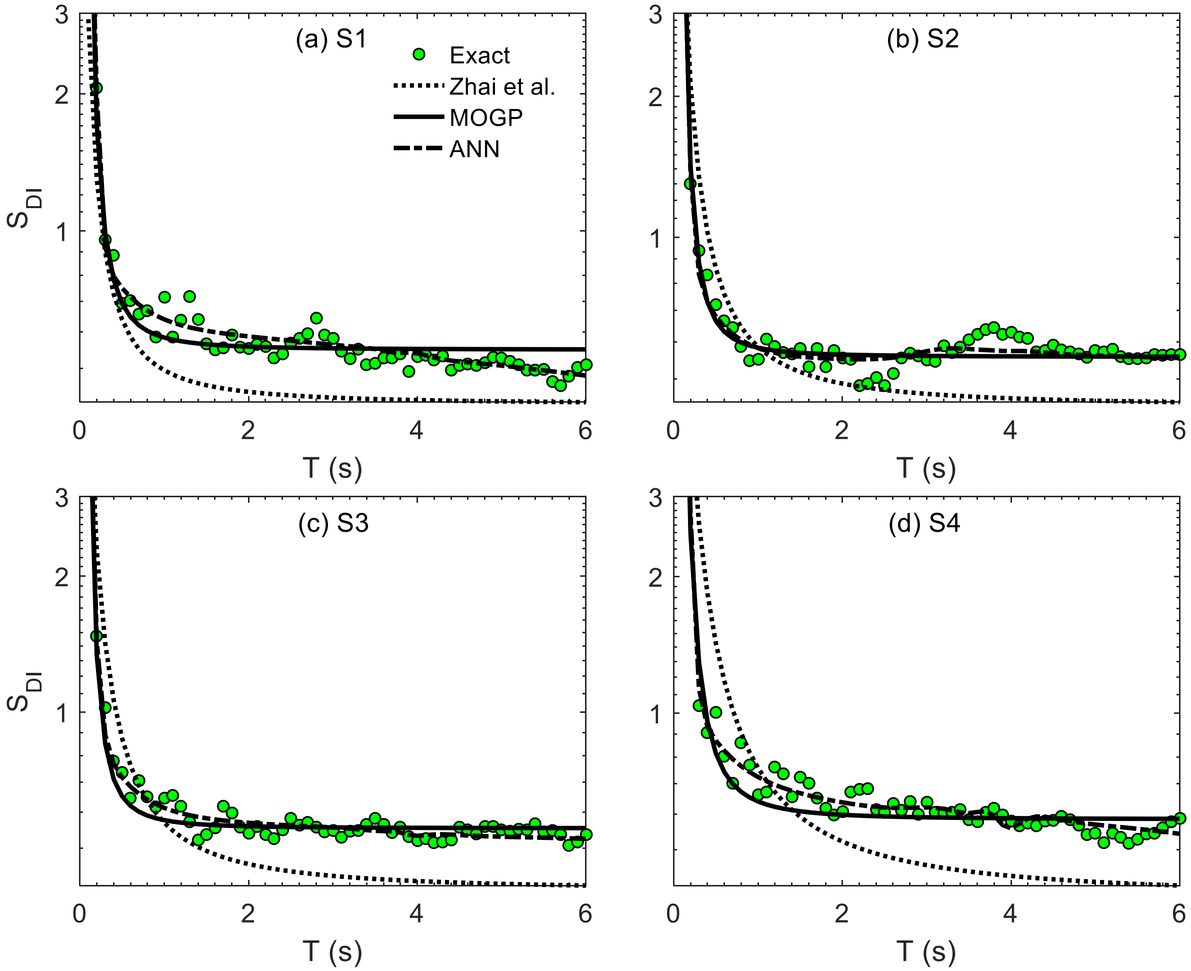


Figure 8. S_{DI} plots corresponding to the exact and the MOGP, ANN and Zhai et al. models: (a) S1 to (d) S4

6. CONCLUSIONS

Determining the potential seismic damage of a structure is of great importance both when designing new structures and to assess the resiliency of existing ones. In this work, to determine the earthquake-induced damage of a wide range of structures, the seismic damage spectra were predicted by using computational intelligence methods, specifically ANN and MGGP. In order to generate a database of the spectral damage, the most important structural and earthquake features were captured, and the Park-Ang damage index was used for the computation of structural damage. The structural features used were the period (T), hardening ratio of hysteretic behavior (α), damping ratio (ξ), response modification factor (R_u), ultimate capacity ductility ratio (μ_u), and the constant of the Park-Ang damage index (β). Also, some earthquake features including moment magnitude (M), the Joyne-Boor distance (R_{JB}) and soil shear wave velocity corresponding to 30 m in depth ($V_{s,30}$) were considered when determining the spectral damage.

Two predictive models based on ANN and MGGP were developed, and their predictions were assessed by performing a comparison with exact results using performance metrics. A spectral damage model available in the literature was also used in a comparison, and finally, all three models were compared with one another. Common performance metrics, including the mean absolute percentage error (*MAPE*), relative root mean square error (*RRMSE*), linear correlation coefficient (*R*), performance index (*PI*), coefficient of efficiency (*E*), and the index of agreement (*d*) were employed to measure the developed predictive models.

Although the performance metrics show that the ANN model is more accurate than the MGGP model, the explicit MGGP-based mathematical model makes it more practical in quantifying the potential seismic damage of structures. In addition, a comparison of the results of the developed models with those of other models available in the literature shows the superiority of the models proposed in this paper.

References:

1. Basu, B. and V.K. Gupta, *A note on damage-based inelastic spectra.* Earthquake engineering & structural dynamics, 1996. **25**(5): p. 421-433.

2. Fajfar, P., *Equivalent ductility factors, taking into account low-cycle fatigue.* Earthquake Engineering & Structural Dynamics, 1992. **21**(10): p. 837-848.

3. Mahin, S.A. and V.V. Bertero, *An evaluation of inelastic seismic design spectra.* Journal of the Structural Division, 1981. **107**(9): p. 1777-1795.

4. Gharehbaghi, S., *Damage controlled optimum seismic design of reinforced concrete framed structures.* Structural Engineering and Mechanics, 2018. **65**(1): p. 53-68.

5. Elenas, A. and K. Meskouris, *Correlation study between seismic acceleration parameters and damage indices of structures.* Engineering Structures, 2001. **23**(6): p. 698-704.

6. Samanta, A., K. Megawati, and T. Pan. *Duration-dependent inelastic response spectra and effect of ground motion duration*. in *Proceedings of the 15th world conference on earthquake engineering, Lisbon*. 2012.

7. Teran-Gilmore, A., A. Sanchez-Badillo, and M. Espinosa-Johnson, *Performance-based seismic design of reinforced concrete ductile buildings subjected to large energy demands.* Earthquakes and Structures, 2010. **1**(1): p. 69-91.

8. Bozorgnia, Y. and V.V. Bertero, *Damage spectra: characteristics and applications to seismic risk reduction.* Journal of Structural Engineering, 2003. **129**(10): p. 1330-1340.

9. Kunnath, S. and Y. Chai, *Cumulative damage-based inelastic cyclic demand spectrum.* Earthquake engineering & structural dynamics, 2004. **33**(4): p. 499-520.

10. Cosenza, E., G. Manfredi, and M. Polese, *Simplified method to include cumulative damage in the seismic response of single-degree-of-freedom systems.* Journal of engineering mechanics, 2009. **135**(10): p. 1081-1088.

11. Zhai, C.-H., et al., *Damage spectra for the mainshock–aftershock sequence-type ground motions.* Soil Dynamics and Earthquake Engineering, 2013. **45**: p. 1-12.

12. Greco, R., G.C. Marano, and A. Fiore, *Damage-based inelastic seismic spectra.* International Journal of Structural Stability and Dynamics, 2017. **17**(10): p. 1750115.

13. Park, Y.-J. and A.H.-S. Ang, *Mechanistic seismic damage model for reinforced concrete.* Journal of structural engineering, 1985. **111**(4): p. 722-739.

14. Deng, P., et al., *A response spectrum-based indicator for structural damage prediction.* Engineering Structures, 2018. **166**: p. 546-555.

15. Wen, W., C. Zhai, and D. Ji, *Damage spectra of global crustal seismic sequences considering scaling issues of aftershock ground motions.* Earthquake Engineering & Structural Dynamics, 2018. **47**(10): p. 2076-2093.

16. Wen, W., et al., *Damage spectra of the mainshock-aftershock ground motions at soft soil sites.* Soil Dynamics and Earthquake Engineering, 2018. **115**: p. 815-825.

17. Khan, S.A., D. Shahani, and A. Agarwala, *Sensor calibration and compensation using artificial neural network.* ISA transactions, 2003. **42**(3): p. 337-352.

18. Gandomi, A.H. and D.A. Roke, *Assessment of artificial neural network and genetic programming as predictive tools.* Advances in Engineering Software, 2015. **88**: p. 63-72.

19. Ferreira, C., *Gene expression programming: mathematical modeling by an artificial intelligence*. Vol. 21. 2006: Springer.

20. Gandomi, A.H. and A.H. Alavi, *Multi-stage genetic programming: a new strategy to nonlinear system modeling.* Information Sciences, 2011. **181**(23): p. 5227-5239.

21. Gandomi, A.H. and A.H. Alavi, *A new multi-gene genetic programming approach to nonlinear system modeling. Part I: materials and structural engineering problems.* Neural Computing and Applications, 2012. **21**(1): p. 171-187.

22. Gandomi, A.H. and A.H. Alavi, *A new multi-gene genetic programming approach to non-linear system modeling. Part II: geotechnical and earthquake engineering problems.* Neural Computing and Applications, 2012. **21**(1): p. 189-201.

23. Tahmassebi, A. and A.H. Gandomi, *Building energy consumption forecast using multi-objective genetic programming.* Measurement, 2018. **118**: p. 164-171.

24. Gandomi, A.H., et al., *Genetic programming for experimental big data mining: A case study on concrete creep formulation.* Automation in Construction, 2016. **70**: p. 89-97.

25. Gharehbaghi, S., et al., *A hybrid computational approach for seismic energy demand prediction.* Expert Systems with Applications, 2018.

26. Lagaros, N.D. and M. Papadrakakis, *Neural network based prediction schemes of the non-linear seismic response of 3D buildings.* Advances in Engineering Software, 2012. **44**(1): p. 92-115.

27. Alavi, A.H. and A.H. Gandomi, *Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing.* Computers & Structures, 2011. **89**(23-24): p. 2176-2194.

28. Gholizadeh, S. and F. Fattahi, *Damage-controlled performance-based design optimization of steel moment frames.* The Structural Design of Tall and Special Buildings, 2018: p. e1498.

29. Gharehbaghi, S., H. Yazdani, and M. Khatibinia, *Estimating inelastic seismic response of reinforced concrete frame structures using a wavelet support vector machine and an artificial neural network.* Neural Computing and Applications: p. 1-14.

30. Lagaros, N.D., V. Plevris, and M. Papadrakakis, *Neurocomputing strategies for solving reliability-robust design optimization problems.* Engineering Computations, 2010. **27**(7): p. 819-840.

31. Plevris, V. and P.G. Asteris, *Modeling of Masonry Failure Surface under Biaxial Compressive Stress Using Neural Networks.* Construction and Building Materials, 2014. **55**: p. 447-461.

32. Asteris, P.G. and V. Plevris, *Anisotropic masonry failure criterion using artificial neural networks.* Neural Computing and Applications, 2017. **28**(8): p. 2207-2229.

33. Rizzo, F. and L. Caracoglia, *Artificial Neural Network model to predict the flutter velocity of suspension bridges.* Computers & Structures, 2020. **233**: p. 106236.

34. Gholizadeh, S., *Performance-based optimum seismic design of steel structures by a modified firefly algorithm and a new neural network.* Advances in Engineering Software, 2015. **81**: p. 50-65.

35. Gholizadeh, S. and V. Aligholizadeh, *Reliability-based optimum seismic design of RC frames by a metamodel and metaheuristics.* The Structural Design of Tall and Special Buildings, 2019. **28**(1): p. e1552.

36. Gholizadeh, S. and E. Salajegheh, *Optimal design of structures for earthquake loading by self organizing radial basis function neural networks.* Advances in Structural Engineering, 2010. **13**(2): p. 339-356.

37. Alavi, A.H., et al., *New ground-motion prediction equations using multi expression programing.* Journal of Earthquake Engineering, 2011. **15**(4): p. 511-536.

38. Lim, J.C., M. Karakus, and T. Ozbakkaloglu, *Evaluation of ultimate conditions of FRP-confined concrete columns using genetic programming.* Computers & Structures, 2016. **162**: p. 28-37.

39. Kiani, J., C. Camp, and S. Pezeshk, *On the application of machine learning techniques to derive seismic fragility curves.* Computers & Structures, 2019. **218**: p. 108-122.

40. Bruneau, M., et al., *A framework to quantitatively assess and enhance the seismic resilience of communities.* Earthquake spectra, 2003. **19**(4): p. 733-752.

41. Cosenza, E. and G. Manfredi, *Damage indices and damage measures.* Progress in Structural Engineering and Materials, 2000. **2**(1): p. 50-59.

42. Park, Y., A.H. Ang, and Y. Wen, *Damage-limiting aseismic design of buildings.* Earthquake spectra, 1987. **3**(1): p. 1-26.

43. Siddique, N. and H. Adeli, *Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing*. 2013: John Wiley & Sons.

44. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity.* Bulletin of Mathematical Biophysics, 1943. **5**: p. 115-133.

45. Garson, G.D., *Interpreting neural-network connection weights.* AI expert, 1991. **6**(4): p. 46-51.

46. Koza, J.R., *Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems*. Vol. 34. 1990: Stanford University, Department of Computer Science Stanford, CA.

47. Babanajad, S.K., et al., *Numerical modeling of concrete strength under multiaxial confinement pressures using linear genetic programming.* Automation in Construction, 2013. **36**: p. 136-144.

48. Searson, D., M. Willis, and G. Montague, *Co-evolution of non-linear PLS model components.* Journal of Chemometrics: A Journal of the Chemometrics Society, 2007. **21**(12): p. 592-603.

49. Searson, D.P., D.E. Leahy, and M.J. Willis. *GPTIPS: an open source genetic programming toolbox for multigene symbolic regression*. in *Proceedings of the International multiconference of engineers and computer scientists*. 2010. IMECS Hong Kong.

50. Searson, D.P., *GPTIPS 2: an open-source software platform for symbolic data mining*, in *Handbook of genetic programming applications*. 2015, Springer. p. 551-573.

51. MATLAB, *The Language of Technical Computing*. 2018, Math Works Inc.

52. Deb, K., et al., *A fast and elitist multiobjective genetic algorithm: NSGA-II.* IEEE Transactions on Evolutionary Computation, 2002. **6**(2): p. 182-197.

53. Gandomi, A.H. and E. Atefi, *Software review: the GPTIPS platform.* Genetic Programming and Evolvable Machines, 2019: p. 1-8.

54. Veeramachaneni, K., Arnaldo, I., Derby, O., O'Reilly, UM., *FlexGP: Cloud-Based Ensemble Learning with Genetic Programming for Large Regression Problems*. Journal of Grid Computing, 2015. **13**(3):391-407.

55. *PEER Strong Motion Database*. 2017; Available from: http://ngawest2.berkeley.edu/.

56. Chopra, A.K., *Dynamics of structures. theory and applications to Earthquake Engineering*. 2017.

57. Hagan, M.T. and H.B. Demuth. *Neural networks for control*. in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*. 1999. IEEE.

58. Burden, F. and D. Winkler, *Bayesian regularization of neural networks*, in *Artificial neural networks*. 2008, Springer. p. 23-42.

59. Gandomi, A.H., et al., *Nonlinear genetic-based models for prediction of flow number of asphalt mixtures.* Journal of Materials in Civil Engineering, 2010. **23**(3): p. 248-263.