IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 10, 2021, accepted January 14, 2021, date of publication January 18, 2021, date of current version January 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052680

Neural Networks and Imbalanced Learning for Data-Driven Scientific Computing With Uncertainties

FARHAD POURKAMALI-ANARAKI[1], (Member, IEEE), AND MOHAMMAD AMIN HARIRI-ARDEBILI[2,3]

[1]Department of Computer Science, University of Massachusetts, Lowell, MA 01854, USA
[2]Department of Civil Engineering, University of Colorado, Boulder, CO 80309, USA
[3]University of Maryland, College Park, MD 20742, USA

Corresponding author: Farhad Pourkamali-Anaraki (farhad_pourkamali@uml.edu)

ABSTRACT Uncertainty quantification in complex engineering problems is challenging because of necessitating large numbers of expensive model evaluations. This paper proposes a two-stage framework for developing accurate machine learning-based surrogate models in structural engineering. The studied numerical model considers aleatory and epistemic uncertainties, i.e., ground motion features and material properties. Our framework's first step trains classification algorithms on the collected data from our numerical model with a disproportionate ratio of observations from two categories, i.e., failed and safe simulations. We investigate the performance of imbalanced learning strategies along with artificial neural networks to achieve high classification accuracy. The second step of our framework aims to estimate three quantities of interest using the same network architecture, comparing our approach with regularized linear regression models. Moreover, we present a new approach to reducing the number of numerical simulations for developing machine learning-based surrogate models with limited training data. This approach employs Gaussian processes as a powerful probabilistic technique, providing an inherent uncertainty measure to determine the quality of estimated response values. Extensive numerical experiments demonstrate the superior performance of neural networks with three hidden layers compared to traditional machine learning algorithms for both classification and regression tasks. Also, empirical investigations corroborate that Gaussian processes enable us to predict the values of missing simulations for reducing the computational cost associated with numerical models. To conclude this work, we present several applications and future research directions.

INDEX TERMS Uncertainty, Gaussian processes, neural networks, imbalanced classification, regression.

I. INTRODUCTION

One of the vital tasks in the probabilistic risk assessment (PRA) of engineering structures is the proper prediction of structural responses (or as we define in the context of this paper: quantities of interest (QoIs)) [1], [2]. This is the first step towards the computation of failure probability. For many of the modern PRA, a detailed numerical simulation is required with all the nonlinearity and interaction sources. The nature of limit state function in this type of simulations is implicit (i.e., black box); thus, the efficiency of the probabilistic performance assessment relies on the type and number of simulations.

This paper proposes a comprehensive data-driven seismic risk analysis framework by considering uncertainties associated with ground motion features and material/modeling properties. These two uncertainty types belong to two different classes of random variables (RVs), i.e., aleatory and epistemic [3]. The former one refers to natural (inherent or stochastic) randomness in a process. It cannot be reduced by performing more experiments or exhaustive measurements. The latter one addresses the scientific uncertainty due to limited data and lack of knowledge. These two types of uncertainties are essentially separated and should be combined appropriately towards a hybrid model.

The associate editor coordinating the review of this manuscript and approving it for publication was Larbi Boubchir.

Several researchers provided distinctions between various uncertainty sources, their quantification, and combinations; among them: [4] for structures and pipelines, [5] for seismic analysis of reinforced concrete framed structures, [6] for buildings, [7] for flood frequency analysis, and [8] for coastal dike.

Nearly all the current applications in the field of seismic risk analysis of engineering structures with both aleatory and epistemic uncertainties are limited to the so-called ''direct'' methods. In this approach, each numerical model (as a representative of epistemic random variables) is directly connected with a ground motion record (as a representative of aleatory random variables), leading to a total of $N_{eps} \times N_{alt}$ numerical simulations, where N_{eps} is the number of numerical realizations, and N_{alt} is the number of ground motion records.

Furthermore, most of the risk- and reliability-based seismic assessments of structural systems need to be performed under multiple seismic intensity levels (SILs). This is because structural responses and potential damage should be quantified for various return periods of seismic hazard. This is the main objective in probabilistic seismic hazard analysis (PSHA) [9], where the seismic intensity is connected to the ground motion return periods. The outcome of such a study is typically presented as a fragility function [10], [11]. Therefore, one should redo all the $N_{eps} \times N_{alt}$ numerical simulations for N_{sil} intensity levels which leads to a total of $N_{eps} \times N_{alt} \times N_{sil}$ runs. Samples of such extensive simulations can be found in [12], [13] in the context of incremental dynamic analysis (IDA). For example, for a minimum of 40 ground motion records, each one scaled to at least 10 SILs, and applied to a set of 25 structural realizations sampled based on Latin Hypercube sampling (LHS), the ''direct'' method requires at least 10,000 nonlinear transient simulations. While it can be performed for a simple model with a limited number of elements [12], it is nearly impossible to afford such a huge computational cost for complex structures such as dams and nuclear power plants [14].

Therefore, it is essential to develop alternative ''indirect'' methods in which only a subset of simulations are enough for risk and reliability analysis purposes with the condition of preserving the accuracy of the original direct method to the extent possible. To this end, meta-models or surrogate models have been extensively used [15], [16]. One of the promising tools to achieve this goal is to employ machine learning (ML) methods.

In the remainder of this section, we review the current literature regarding the insertion of ML models into the field of seismic risk and reliability analysis of structures to reduce the cost of numerical simulations. In a recent review paper by [17], the authors provided a comprehensive survey of existing ML-based methods, including traditional learning algorithms and artificial neural networks (NNs). To mention the most relevant works, several papers [18]–[20] employed traditional learning algorithms, such as support vector machines, for predicting structural responses under uncertainties associated with ground motions. Another work [21] compared various traditional binary classification methods for the reliability analysis of concrete dams. In [22], the authors used nonlinear regression analysis techniques for accelerating seismic analysis. In [23], the authors developed a pool-based active learning algorithm to choose partial data from ground motion records to meet informativeness, representativeness, and diversity criteria. They applied this method to reduce the computation burden in finite element analyses. The authors of [24] applied multiple ML techniques to develop seismic fragility curves. Moreover, several works considered artificial NNs for developing regression models to quantify uncertainties related to ground motion features [25]–[31].

Therefore, based on the literature review, most existing data-driven approaches focus on developing regression models for quantifying uncertainties linked with variation in ground motion properties. Very few incorporate the material and modeling uncertainties, and nearly none of them included the ground motion meta-features in the process of uncertainty quantification. Thus, there is an intuitive research gap that we aim to appropriately fill out in this work.

The main contribution of this work is to develop a two-stage framework for predictive modeling: (1) classifying simulations into two categories (failed/safe); and (2) developing regression models to predict continuous-valued QoIs for safe simulations. As discussed, most existing data-driven methods for seismic hazard analysis restrict the use of ML methods for developing regression models. A reasonable explanation for this choice is the difficulty of training classification algorithms because obtained data sets are often imbalanced with a disproportionate ratio of observations in each category [32]. In seismic risk analysis, only a small proportion of input parameter combinations lead to structural failure. Therefore, identifying rare and extreme events is challenging, as most ML algorithms are biased towards the majority class. This issue arises from the fact each observation has an equal contribution to the total loss function and the corresponding optimization problem as we will discuss later. We will investigate the effectiveness of two primary imbalanced learning techniques [33], i.e., re-sampling methods and cost-sensitive learning, to alleviate this problem and enhance the performance of predictive models.

Another critical question that we answer is how to reduce the number of numerical simulations for varying ground motion scale factors to facilitate the predictive modeling process. While we considered eight different scale factors in the described model in Section IV, a valuable research direction is to perform regression analysis across the scale factor values so that we have to conduct a subset of numerical simulations. To this end, we propose to utilize a Bayesian nonlinear technique called Gaussian process regression [34] for extracting relationships between ground motion scale factors and the resulting structural response. The main advantage of using Gaussian process regression is the ability to generate distributions over functions, which provides a straightforward approach for capturing uncertainty levels in the estimation

process. We will present further details regarding the predictive modeling aspects in Section III.

This paper begins with a brief review of the foundations of ML and predictive modeling in Section II, followed by our proposed two-stage framework in Section III. The numerical model and the associated uncertainties are described in section IV. We present a detailed empirical evaluation of the proposed methodology in Section V, exemplifying the importance of imbalanced learning and the superior performance of neural networks. We also offer concluding remarks and a summary of our observations in Section VI.

In the following, we outline the four main contributions of this work.

- Developing classification methods for automatically identifying structural failures based on the information concerning ground motion features and material properties. Due to the inherent complex characteristics of imbalanced data collected from our numerical model, we investigate the effectiveness of two primary methods to solve the class imbalance problem. Moreover, we present a comprehensive comparison of traditional machine learning algorithms and artificial neural networks.
- Training regression models using both regularized linear models and neural networks with different network architectures. The resulting model based on neural networks allows accurate and reliable prediction of multiple QoIs, significantly outperforming linear machine learning models.
- Presenting a new approach to reducing the number of numerical simulations by utilizing Gaussian processes. We show that the Gaussian process regression algorithm is a powerful probabilistic technique to extract trends between QoIs and the seismic intensity level, producing both the most probable estimate and the confidence interval to quantify uncertainties associated with the predictive model.
- Application of hybrid uncertainty simulations for the first time on high-rise towers.

II. BACKGROUND ON PREDICTIVE MODELING

In this section, we discuss incorporating appropriate ML algorithms into complex engineering problems for constructing data-driven predictive models. In recent years, much research has focused on developing ML-based surrogate models in various engineering applications to reduce the computational complexity associated with numerical analysis techniques such as the finite element method. This section presents a brief problem formulation and an overview of widely used ML algorithms, including traditional approaches that work directly with input features (e.g., linear models) and more recent artificial NNs that provide high expressive power and flexibility.

Let us consider a numerical model with d input variables that are vectorized as $\mathbf{x} \in \mathbb{R}^d$, representing the input parameter space. For example, we often work with input vectors containing the characteristics of ground motions and material properties in structural engineering problems. We also denote the desired QoI or output variable, such as displacement or stress, by $y \in \mathbb{R}$. For the clarity of the presentation, we start our discussion by considering just a single output variable, and we will later describe the case of multiple outputs. Throughout this section, we use lower-case and upper-case boldface letters for vectors and matrices, respectively.

The goal of predictive modeling in scientific computing, also known as scientific ML, is to identify linear or nonlinear relationships between the input parameter space and the desired quantity of interest. That is, we aim to learn a function $f_{\boldsymbol{\theta}} : \mathbf{x} \mapsto y$, which maps the input parameters to the output space using available observations from a numerical model $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, referred to as the training data set. Therefore, ML methods often involve solving mathematical optimization problems in the following form:

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} l(f_{\boldsymbol{\theta}}(\mathbf{x}), y) \tag{1}$$

where $\boldsymbol{\theta}$ represents parameters (or weights) of the ML model and $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ denotes a suitable loss function that measures how well predictions match the observations.

When we obtain the optimal parameters $\boldsymbol{\theta}^*$ by solving Equation (1), the learned model enables making predictions regarding response values corresponding to new input parameters as an alternative for the numerical model, i.e., $\hat{y} = f_{\boldsymbol{\theta}^*}(\mathbf{x}_{\text{new}})$. A critical aspect of developing ML models is to utilize appropriate evaluation metrics to assess the generalization error, which is also known as the out-of-sample error. The evaluation process requires access to an additional set of observations $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{n'}$ that can be set aside and used as a test set. Therefore, we use $\mathcal{D}_{\text{train}}$ in the training phase to find the optimal parameter values, and we report the performance of ML algorithms based on the test data set $\mathcal{D}_{\text{test}}$. ML techniques can be divided into two categories based on the type of response values: classification and regression.

A. CLASSIFICATION

In this discussion, we first focus on classification methods that assume the desired quantity of interest is discrete, i.e., y takes a finite number of values. We mainly explain the problem of binary classification in which y indicates one of two states, such as failure/safety in a structural system. Hence, the goal is to find a function that automatically categorizes input parameter vectors into one of the two classes, i.e., $y \in \{0, 1\}$. Among existing ML methods, logistic regression [35] is a simple and practical technique that applies the sigmoid function to a linear combination of the input variables as:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^T\mathbf{x}) = \frac{1}{1+\exp(-\boldsymbol{\theta}^T\mathbf{x})} \tag{2}$$

where $\sigma(z) = 1/(1+\exp(-z))$ represents the sigmoid function [36] and has two horizontal asymptotes at 0 and 1. Also, $\boldsymbol{\theta}^T\mathbf{x} \in \mathbb{R}$ is the inner product between the two

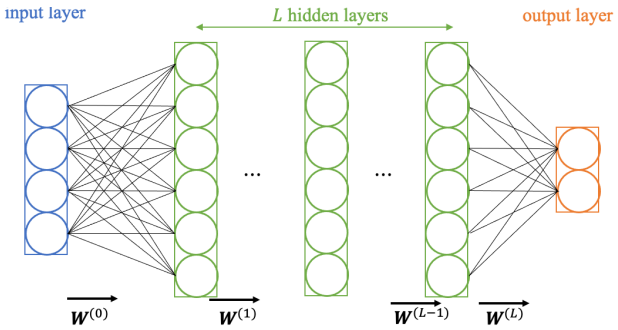


FIGURE 1. Schematic representation of artificial NNs with L hidden layers. The input and output layers are assumed to be layer 0 and $L+1$, respectively. Therefore, $\mathbf{W}^{(i)}$ is a weight matrix for moving from layer i to $i+1$.

vectors, i.e., linear combinations of the input features. Thus, one can design a classification rule by introducing a threshold parameter, such as $y = 1$ if $f_{\boldsymbol{\theta}}(\mathbf{x}) \geq 0.5$ and zero otherwise. Training the logistic regression model boils down to solving an optimization problem in the form of Equation (1) using the following loss function (known as the binary cross-entropy loss [37]):

$$l(f_{\boldsymbol{\theta}}(\mathbf{x}), y) = -y \log(f_{\boldsymbol{\theta}}(\mathbf{x})) - (1-y)\log(1-f_{\boldsymbol{\theta}}(\mathbf{x})), \quad (3)$$

where $y \in \{0, 1\}$. A downside of logistic regression is that the argument inside the sigmoid function is restricted to be a linear combination of the input features. Therefore, the performance of this technique relies heavily on the selection of appropriate and informative features.

Artificial NNs [38] can be viewed as a generalization of the logistic regression model, consisting of three layers: input, hidden, and output layers (illustrated in Figure 1). Like logistic regression, we determine the number of neurons in the input and output layers according to the number of input variables d and the size of the output space, i.e., number of QoIs. Compared to traditional methods such as logistic regression, the expressive power of NNs can be seen as a function of the depth, i.e., number of hidden layers, and the width, i.e., number of neurons per hidden layer. Therefore, we can express the mapping between input parameters and output variables as a nested set of functions with the parameter set $\boldsymbol{\theta} = \{\mathbf{W}^{(i)}\}_{i=0}^{L}$:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma_{L+1}\big(\mathbf{W}^{(L)}\sigma_L(\mathbf{W}^{(L-1)} \cdots \sigma_1(\mathbf{W}^{(0)}\mathbf{x}))\big), \quad (4)$$

where L represents the number of hidden layers and $\sigma_i(\cdot)$, for each $i = 1, \ldots, L+1$, is a coordinate-wise scalar activation function for the i-th layer to introduce nonlinearities into the model. In this formulation, each $\mathbf{W}^{(i)}$ represents the weight of the connection from layer i to $i+1$; thus, logistic regression is a NN without hidden layers, i.e., $L = 0$.

For binary classification problems, it is common to use the sigmoid activation function for the output layer since we can still employ the cross-entropy loss given in Equation (3). However, a widely used choice for hidden layers is the rectified linear unit (ReLU) activation function, defined as $\sigma(z) = \max\{0, z\}$ [39]. The reason for using nonlinear activation functions is to improve the predictive model's performance on data sets that are not linearly separable. Assigning linear activation functions to all layers will result in a surrogate model with a linear decision boundary and impractical for complex nonlinear data. When considering the nested function in Equation (4), we use the stochastic gradient descent (SGD) algorithm or other iterative techniques such as the ADAM optimization algorithm [40], [41] for solving the problem in Equation (1).

The above formulation of NNs is known as Multi-Layer Perceptron (MLP) networks because every layer is fully connected. That is, every neuron in each layer connects to all neurons in other layers. The main advantage of utilizing such network architectures is to provide a powerful framework for extracting relationships between inputs and desired QoIs without making assumptions concerning their structures. In some areas such as computer vision, the primary goal is to analyze high-dimensional image data with spatial structures [42]. Therefore, using convolutional layers has gained a lot of attention in image processing tasks, where each neuron connects to only a restricted sub-area of the previous layer [43]. This work considers MLP networks because the input parameter space consists of a set of predefined features, eliminating the need to perform feature extraction using Convolutional Neural Networks (CNNs). Section V provides an example to demonstrate the effect of using convolutional layers in our case study.

In addition to logistic regression and artificial NNs, other widely used classification algorithms that work directly on the input parameter space include support vector machines (SVMs) and the random forest (RF) algorithm [44], [45]. The former aims to find an optimal separating hyperplane or decision boundary that maximizes the geometric margin for producing a classification rule. Therefore, the main objective of SVMs is to separate the two classes with a large gap. The RF classifiers are ensemble learning methods that fit multiple decision trees during the training stage. In a nutshell, decision trees aim to search for a variable which gives the maximum information gain or divides the data in the most homogeneous way. Hence, a decision tree constructs a flowchart-like tree structure that has a root node and terminal nodes. An advantage of employing these two classification methods is that they require tuning fewer hyper-parameters than training NNs (e.g., depth and width of the network and selecting appropriate activation functions). One of the main objectives of this work is to provide a thorough comparison of artificial NNs with traditional learning algorithms.

B. REGRESSION

Another critical form of predictive modeling is to predict continuous response values or QoIs, i.e., $y \in \mathbb{R}$, known as regression analysis. One can extract a linear relationship between the input parameter space and the QoI by considering $f_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T\mathbf{x}$, where we replaced the sigmoid function in Equation (2) with the identity function, i.e., $\sigma(z) = z$. Thus,

we can represent a general optimization problem for linear regression with a regularization term that controls the trade-off between model complexity and the fit to the training data as follows [46]:

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda g(\boldsymbol{\theta}), \tag{5}$$

where $\lambda > 0$ is a regularization parameter, and $\|\boldsymbol{\theta}\|_2$ represents the Euclidean norm or ℓ_2-norm of the parameter vector.

In this formulation, $\mathbf{y} \in \mathbb{R}^n$ contains all outputs values given in the training data set $\mathcal{D}_{\text{train}}$ as a column vector and the rows of the data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ are corresponding input parameters. Two popular choices of regularization functions are $g(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$ and $g(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$, where $\|\boldsymbol{\theta}\|_1$ is the ℓ_1 norm of the weight vector, i.e., sum of absolute values of entries. The former is known as the ridge regression problem [47] and the latter is referred to as the least absolute shrinkage and selection operator (LASSO) [48]. Compared to classification tasks discussed before in Section II-A, we use the square error loss function (instead of the cross-entropy loss) for regression problems. When the objective is to predict multiple QoIs, the easiest way is to train an independent linear regression model for each output variable.

We can utilize artificial NNs with several hidden layers for performing nonlinear regression analysis [49]. To this end, we replace the sigmoid activation function for the output layer with the identity function or other suitable activation functions for enabling the prediction of continuous values that are not restricted to the interval (0, 1). In these situations, we extract nonlinear relationships using activation functions such as ReLU for the hidden layers. A significant advantage of using neural networks for the regression problem is the ease of incorporating multiple QoIs in the analysis by adding more neurons to the output layer. Therefore, we can share the same network architecture (i.e., depth and width) and the weights $\mathbf{W}^{(i)}, i = 0, 1, \ldots, L-1$, for predicting various QoIs. In this case, the size of the last weight matrix $\mathbf{W}^{(L)}$ depends on the number of output variables.

III. PROPOSED TWO-STAGE FRAMEWORK FOR PREDICTIVE MODELING AND GAUSSIAN PROCESSES

This section provides a detailed discussion concerning the proposed data-driven framework. We also discuss a new application of Gaussian process regression for understanding the influence of seismic intensity levels. As mentioned before, the proposed data-driven framework consists of two steps. The first stage involves investigating the accuracy of various binary classification algorithms for automatically identifying combinations of input parameters, i.e., different realizations of aleatory and epistemic uncertainties, that lead to a collapse or failure. This stage can thus be viewed as an initial surrogate model to divide the input parameter space into two categories for assessing the severity of structural damage. We thoroughly examine the performance of traditional learning algorithms (namely, logistic regression, support vector machines, and random forest) and artificial NNs with one to three hidden layers, i.e., $L = 1, 2, 3$.

Regardless of the specific type of machine learning method being used, a significant challenge is the unequal distribution of classes within a data set, referred to as imbalanced learning [32], [50]–[53]. In our problem of interest with skewed class proportions, many observations belong to the safe category (related to as the majority class), and much fewer samples fit in the structural failure group (referred to as the minority class). Without taking additional steps, standard machine learning algorithms produce classifiers with poor predictive accuracy for the minority class and tend to classify most new samples in the majority class. It is worth pointing out that the appropriate assessment of the performance in this scenario is vital. Using standard evaluation metrics, such as the fraction of correct predictions, can be misleading and not representative of the trained model's actual performance [54].

To solve this problem, we will incorporate two major imbalanced learning strategies, re-sampling methods and cost-sensitive learning, into our predictive modeling pipeline for enhancing the performance of binary classifiers; see Figure 2 for an illustration. Re-sampling methods are pre-processing techniques that either add repetitive observations to the minority class (i.e., over-sampling) or remove samples from the majority class (i.e., under-sampling) for re-balancing the original training data set $\mathcal{D}_{\text{train}}$. Among these methods, a practical and influential algorithm is titled synthetic minority over-sampling technique [55] or SMOTE for short. The main contribution of SMOTE is to carry out an interpolation among neighboring minority class instances, creating new synthetic examples to increase the size of the minority class. In the following, we briefly summarize the three main steps of SMOTE for creating such synthetic samples. Let us denote the minority class by $\mathcal{M}$ and recall that SMOTE does not process the majority class. For each $\mathbf{x} \in \mathcal{M}$, we compute the K nearest neighbors of $\mathbf{x}$, where K is a parameter that should be set. The second step randomly selects one of the examples from this set that we call $\mathbf{x}_{\text{neighbor}}$. In the third step, we pick a random number β from the range (0, 1) for generating a new synthetic example $\mathbf{x}_{\text{syn}}$ as follows:

$$\mathbf{x}_{\text{syn}} = \mathbf{x} + \beta\big(\mathbf{x}_{\text{neighbor}} - \mathbf{x}\big), \tag{6}$$

where this interpolation creates a sample on the line between $\mathbf{x}_{\text{neighbor}}$ and $\mathbf{x}$. We repeat this process for other data samples in the minority class until we generate the desired synthetic examples for re-balancing the training data set. After this pre-processing step on the given training data set, we utilize standard machine learning algorithms without modifying their loss functions.

The second class of imbalanced learning methods, known as cost-sensitive learning [56], adjust the optimization problem given in Equation (1) during the training process by introducing a weighted loss function. To explain this line of work, let us further denote the majority class by $\mathcal{S}$. The weighted loss function and related optimization problem for extracting the mapping between the input parameters and the

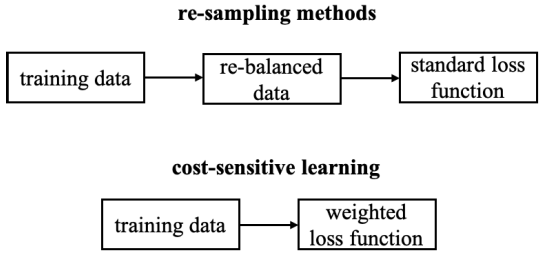


FIGURE 2. Two main techniques for tackling the class imbalance problem.

desired output can be expressed in the following form:

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} \Big(\beta_{\mathcal{M}} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} \mathbb{I}_{\mathcal{M}}(\mathbf{x}) l(f_{\boldsymbol{\theta}}(\mathbf{x}), y) + \beta_{\mathcal{S}} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} \mathbb{I}_{\mathcal{S}}(\mathbf{x}) l(f_{\boldsymbol{\theta}}(\mathbf{x}), y) \Big), \quad (7)$$

where $\mathbb{I}_{\mathcal{M}}(\mathbf{x})$ and $\mathbb{I}_{\mathcal{S}}(\mathbf{x})$ are indicator functions that specify each training data point $\mathbf{x}$ belongs to one of the two classes. Also, $\beta_{\mathcal{M}}$ and $\beta_{\mathcal{S}}$ are two positive scalars representing class weights to asymmetrically penalize the prediction error. When $\beta_{\mathcal{M}} = \beta_{\mathcal{S}}$, the optimization problem in Equation (7) reduces to Equation (1), ignoring the disproportionate ratio of observations in the training data set $\mathcal{D}_{\text{train}}$. However, assigning larger weights to the minority class, i.e., $\beta_{\mathcal{M}} > \beta_{\mathcal{S}}$, allows us to improve the predictive accuracy of classification algorithms when facing imbalanced data. Compared to re-sampling techniques, cost-sensitive learning is task-specific because the modification process differs for various machine learning algorithms.

So far, we explained the first stage of our data-driven framework for classifying the input parameter space into two classes, i.e., safe and failed simulations. The second task is to train accurate regression models for predicting the desired structural responses for safe simulations. While the previous research has primarily focused on developing regression models based on just ground motion parameters, we extend this line of work by considering uncertainties associated with ground motion signals and material properties, i.e., both aleatory and epistemic uncertainties. Notably, this paper presents a detailed analysis of artificial NNs with varying numbers of hidden layers L for predicting various QoIs. This problem is known as multi-target regression [57] in the machine learning literature, where the goal is to simultaneously predict multiple outputs given an input vector. This work investigates the performance of NNs with the number of neurons in the output layer set to the number of QoIs. Hence, this technique allows us to share the network architecture for predicting different structural responses, eliminating the need to train individual NNs. To provide a fair empirical evaluation, we compare the performance of nonlinear regression analysis using NNs with a family of methods for solving the least-squares problem with regularization.

The main objective of the discussed regression model is to capture the relationship between the input parameter space, which is a combination of the ground motion features and material properties, and the desired QoIs. In this paper, we present a novel application of regression analysis for predicting structural responses as a function of the seismic intensity level. The primary purpose of this task is to reduce the number of numerical simulations when taking into consideration multiple values of the scale factor. This step aims to extract and exploit the relationship between structural responses and the scale factor to conduct a subset of numerical simulations to save computing resources. To capture the trend between the ground motion scale factor and the desired structural responses, we can view the input variable as the index of the scale factor, i.e., an integer that shows the ordering of ground motion scale factors. Thus, we fix all other input variables, such as the material properties, to reveal the connection between the scale factor value and the desired structural response.

From the engineering perspective, it is reasonable to consider scale factors over a small range of values; thus, the number of training examples is limited, and training NNs with a large number of parameters is impractical. To tackle this problem and develop a model that allows quantifying uncertainties associated with predictions, we propose to employ a Bayesian technique known as Gaussian process regression [58]. A Gaussian process is a random process, where any input variable $\mathbf{x}$ is assigned to a random variable $f(\mathbf{x})$ and the joint distribution of $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^T \in \mathbb{R}^n$ is Gaussian, i.e., $\mathcal{N}(\mathbf{0}, \mathbf{K})$. In this work, we follow the common practice of assuming the mean vector is set to $\mathbf{0}$ for simplifying the presentation [34]. The main ingredient of Gaussian processes is the covariance matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, which controls the distribution over functions such as smoothness. That is, the kernel matrix encodes nonlinear similarities between all pairs of data points $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e., $[\mathbf{K}]_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, where a widely used choice of the kernel function is the radial basis function with the length-scale parameter $r > 0$ [59]:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\Big(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2r^2} \Big). \quad (8)$$

We can use this information regarding the kernel matrix in order to compute the predictive distribution $p(\hat{y}|\mathbf{x}_{\text{new}}, \mathcal{D}_{\text{train}}) = \mathcal{N}(\mu_{\text{new}}, \sigma^2_{\text{new}})$ with the following mean and variance [60]:

$$\mu_{\text{new}} = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{y}, \quad \sigma^2_{\text{new}} = k_{\text{new}} - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}, \quad (9)$$

where $\mathbf{k} = [\kappa(\mathbf{x}_1, \mathbf{x}_{\text{new}}), \ldots, \kappa(\mathbf{x}_n, \mathbf{x}_{\text{new}})]^T \in \mathbb{R}^n$, $k_{\text{new}} = \kappa(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{new}}) \in \mathbb{R}$, and $\mathbf{y} = [y_1, \ldots, y_n]^T \in \mathbb{R}^n$.

Hence, the Gaussian process regression algorithm provides a probabilistic model of the target function for introducing confidence intervals. Unlike standard regression analysis techniques, Gaussian processes offer a straightforward framework for quantifying uncertainties using the posterior variance σ^2_{new}, which is desirable in settings with limited training data. We will present experimental results in the next section to demonstrate the application of Gaussian processes when the training data set is a subset of numerical simulations corresponding to a fraction of ground motion scale factors.

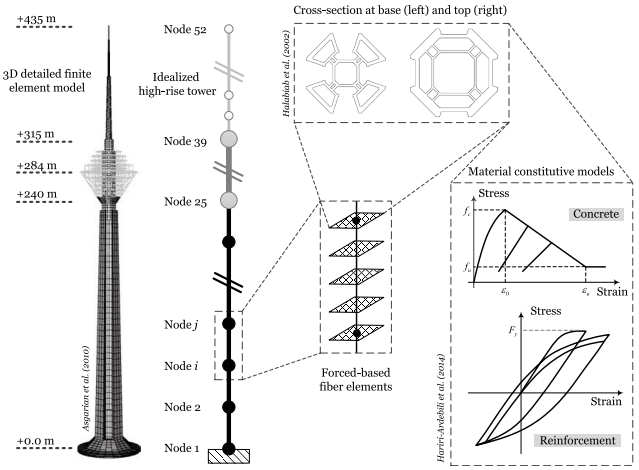


FIGURE 3. Summary of the finite element modeling.

IV. NUMERICAL MODEL WITH HYBRID UNCERTAINTIES

A high-rise telecommunication tower which is already developed and analyzed by the authors [61] is used for case study; see Figure 3. This example has been selected as it is a computationally expensive finite element model compared to the typical framed structures. It includes four main parts: foundation, shaft, head structure, and antenna mast. The shaft is a 315 m high reinforced concrete (RC) structure (varying from 8 m wide at the bottom to 17 m at the top), and is the main load-carrying element of the tower that transfers the lateral and gravitational loads to the foundation.

As mentioned, we perform a large number of transient analyses in this paper. On the other hand, the modeling aspects such as material nonlinearities, i.e., cracking, crushing, and damage, and geometric nonlinearities, i.e., P-Delta effects, and large displacements, as well as interaction between the different structural components, should be considered. As a consequence, performing 3D finite element analyses to address the transient effects together with the other nonlinearity issues becomes very time-consuming [62]. Therefore, it is desirable to look for a model that not only requires fewer elements and less computational time, but also provides the desired outputs with an acceptable loss of accuracy. One of the best alternatives is to model the structure using uniaxial fiber beam-column elements [63].

A 2D nonlinear model of the tower, including the head structure, shaft and transition, is developed using OpenSees [64]; see Figure 3. The service core is modeled using 2D force-based nonlinear fiber beam-column elements [65] with five integration points. The core cross-section is discretized into concrete and steel fibers. The superstructure is idealized using equivalent mass of the floors. The base elevation of the building is constrained in the lateral and rotational degrees of freedom, excluding the effects of soil-structure interaction [61]; however, some research shows the response of tall towers affected by modeling the beneath foundation and the soil flexibility [66], [67].

The concrete is modeled based on the uniaxial Kent-Scott-Park constitutive model [68] with degraded linear unloading/reloading stiffness as shown in Figure 3. In this model f_c is the compressive strength of concrete and ε_0 is the strain at the peak strength. In addition, f_u and ε_u are the ultimate compressive stress and its corresponding stain. Steel is modeled based on the Giuffrè-Menegotto-Pinto model with isotropic strain hardening, in which a transition curve is defined to avoid the unsmoothed response of bi-linear kinematic hardening behavior at the yield point, and consequently, the path-dependent nature of the material can be traced effectively [69], [70]. Furthermore, the Bauschinger effect [71] is intrinsically defined in the material stress-strain curve so that the deterioration of strength in the element behavior is automatically modeled.

In this study, Rayleigh damping is adopted where the mass proportional part is constant during the analysis; however, the stiffness proportional part alters according to the updated stiffness matrix of the structure. In other words, the damping property is updated for each load step of the transient analysis. One should note that there are some preliminary studies on the vibration characteristics of the telecommunications towers proposing a new form of damping model [72] which is beyond the objective of this paper.

The reader should note that the emphasis of this paper is on proposing a generic framework for combining epistemic and aleatory uncertainties in the context of PRA with limited and imbalanced data. Therefore, the precise calibration and verification of the numerical models is not within the scope of the current study. However, we provided as many as possible peer-reviewed publications to support the engineering aspects. We made some more assumptions: In the numerical model, the bounded Gaussian distributional model is used. A total of 10 random models are generated using LHS, to consider the variability in 18 material parameters (since there is no correlation among the random variables, we choose a small batch of samples). A flat coefficient of variation (COV) of 10% is assumed in all cases (except area which is 5%).

- Component level - Concrete: Compressive strength, strain at maximum strength, crushing strength, strain at crushing strength, weight per volume;
- Component level - Steel: yield strength, initial elastic tangent, strain-hardening ratio, area of bars;
- System level: Damping ratio.

A total of 100 ground motions are selected randomly from the PEER database [73] to cover a wide range of potential ground motions. This is in line with Cloud analysis, where a large data set of signals are applied to the structural system [74]. For each ground motion, 31 intensity measure (IM) parameters are extracted to develop a side information matrix [45]. These 31 IM parameters are selected from a comprehensive list of over 70 IM parameters found in [14]. Finally, in order to simulate the structural response under high intensity seismic actions, the acceleration time history of the initial un-scaled records are multiplied by 8 different

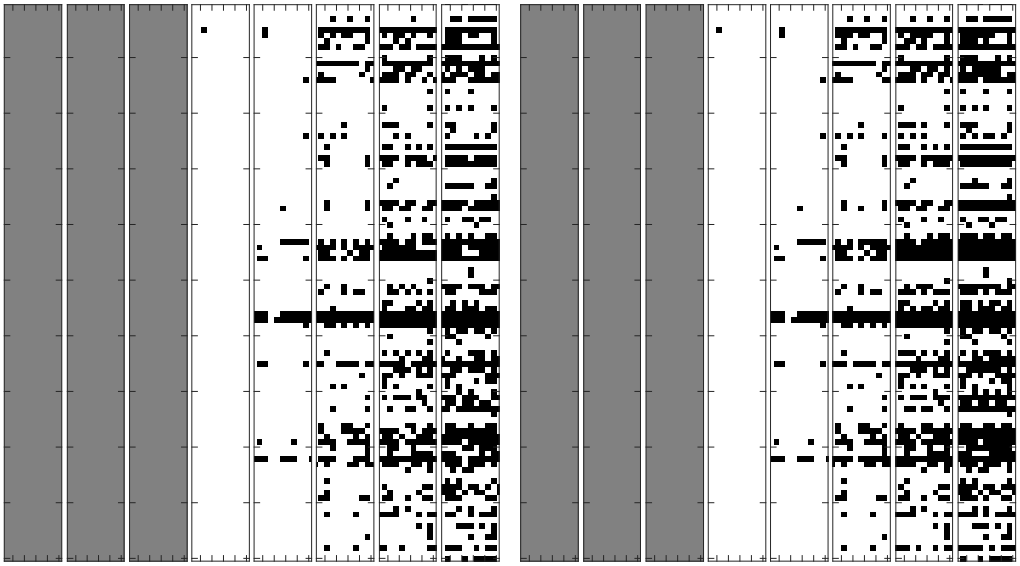


FIGURE 4. Matrix of binary responses (safe and failed) used for classification and regression. Left to right the scale factors are increased.

scale factors (SF), i.e., 1, 5, 10, 20, 40, 60, 80, and 100. This method is called scaled cloud analysis. While some of these SF values seem to be very high, one should note that the objective of this paper is to push a handful of records to collapse regions to generate a database consisting of both safe and failed simulations.

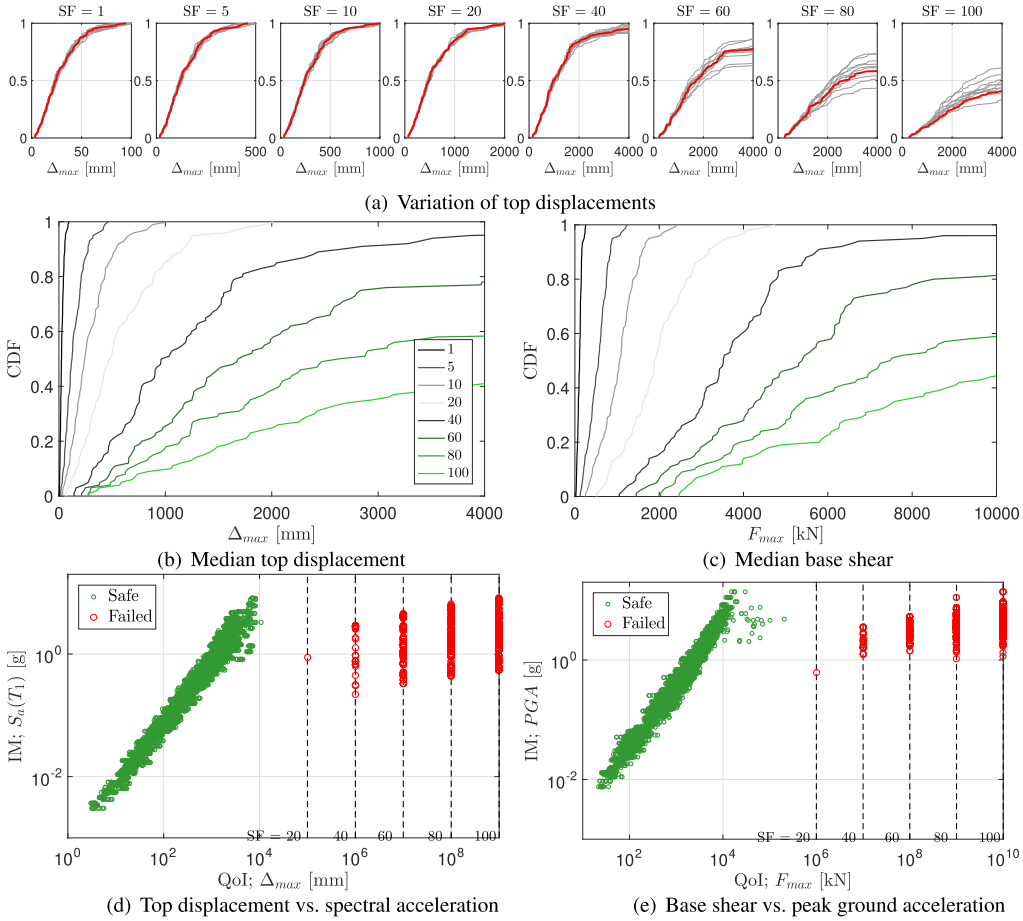
Figure 4(a) shows the binary (black: failed, white: safe) response combination of numerical simulations. The eight matrices present the 8 SFs and within each matrix the rows are 100 (scaled) records and the columns are 10 structural realizations. The first three matrices are gray since there is no failure reported in those SFs (SF = 1, 5 and 10). The fourth SF = 20 contains only one failure over 1,000 simulations. Technically, if there is a failure in a particular cell in SF i, all higher SFs should also report a failure in that particular cell. However, there are few cases which show failure in SF i and not $i + 1$. This can be attributed to the so-called resurrection phenomenon explained in [75]. In the context of this paper (and for simplicity) we revise/update the matrix of the structural responses to have failure in $i + 1$, if there is already a reported failure in i. This processed matrix is shown in Figure 4(b).

In this paper, we consider three quantities of interest (QoIs): maximum top displacement (Δ_{max}), maximum base shear (F_{max}), and maximum total relative displacement (δ_{max}). Figure 5(a) presents the variation of Δ_{max} as a function of SF and structural realizations. For lower SILs where there is no (i.e., SF = 1, 5, and 10) or only few (i.e., SF = 20) failed cases, the Δ_{max} is limited to 100, 500, 1000, and 2000 mm, respectively. In these cases the variation of 10 different structural realizations is small compared to seismic responses. This is also partially true for SF = 40. However, for higher SFs (i.e., SF = 60, 80, and 100) there is a large variation among different LHS-based structural models before failure. These figures also show the percentage of failed cases for each SF. For example, for SF = 100, 40-65% of models are failed (with the median of 55%). This is also consistent with the physics of the problem in which some random variables associated with high intensity ground motions are activated.

Figures 5(b) and 5(c) compare the median curves from 10 LHS-based random structural realizations and 8 different SFs for Δ_{max} and F_{max}, respectively. While the general trend is similar (as expected), the base shear has larger lower bounds especially for higher SFs. Finally, Figures 5(d) and 5(e) present the relationship between different input and output quantities (in green) including the failed simulations (in red) for higher scale factors (i.e., SF = 20 to 100). As seen, the (safe) input-output pairs have a linear trend in logarithmic scale implying a nonlinear relationship in Cartesian coordinate system. The failed cases (red) follow a log-normal distribution as well (not shown here).

V. EMPIRICAL EVALUATION

In this section, we evaluate our proposed data-driven framework using observations from the numerical model discussed in Section IV. The first part of this section relates to training and assessing binary classification algorithms for automatically categorizing combinations of input parameters into two classes, i.e., safe and failed simulations. Our goal is to demonstrate the performance of artificial NNs compared to traditional ML algorithms. We also exhibit the significance of utilizing imbalanced learning strategies in our problem of interest. We then develop regression models for extracting relationships between the input parameter space and three QoIs, i.e., maximum top displacement, Δ_{max}, maximum base shear, F_{max}, and maximum total relative displacement, δ_{max}.



(a) Variation of top displacements

(b) Median top displacement

(c) Median base shear

(d) Top displacement vs. spectral acceleration

(e) Base shear vs. peak ground acceleration

FIGURE 5. Dependency of responses to scale factors, structural realizations, and ground motion intensity measures.

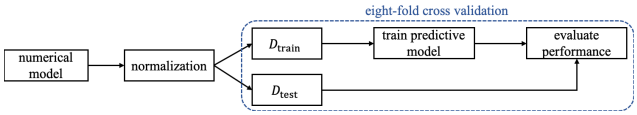


FIGURE 6. Evaluating classification and regression models.

Particularly, we evaluate the performance of NNs for the case of multi-target regression. Figure 6 summarizes the main steps involved in developing classification (Section V-A) and regression (Section V-B) models in our framework. Note that in these two sections, the input parameter space includes various combinations of ground motion features and material properties. However, regression analysis in Section V-C considers just a single input variable, i.e., ground motion scale factor. Thus, unlike Section V-B, the main objective is to extract the trend between a structural response and the ground motion scale factor (instead of considering all input variables together).

In our numerical model, we have a total of 31 ground motion features and 18 material properties; thus, the dimension of the input parameter space is $d = 49$. Moreover, we consider 8 scale factors, 100 realizations of ground motions, and 10 different material combinations, resulting in a database of 8,000 numerical simulations to be used for training and evaluation purposes. As the input variables fall in significantly differing ranges, the first step of our data normalization is to apply the z-score transformation [76]. Thus, we scale each input variable to have the mean 0 and a variance of 1. For predicting QoIs simultaneously, we employ a normalization technique, known as min-max normalization [77], so that the minimum value of each QoI gets transformed into 0, the maximum value gets transformed into 1. After normalizing the input parameters and target values, we use 8-fold cross-validation to generate training and test data sets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. We split the available observations from our numerical model into 8 smaller sets or "folds," and we use 7 of these groups as the training data set and the remaining part of the data is used for reporting accuracy. Thus, repeating this process 8 times allows us to include all of the data exactly once for assessing predictive power. For example, for the binary classification task, we divide the available 8,000 simulations into 8 groups, which means that we train 8 classifiers using 7,000 training data samples and the remaining 1,000 simulations are used for reporting performance. We present evaluation metrics over the 8 trials in the form of a boxplot.

The proposed data-driven framework is implemented in Python using a machine with Intel Core *i*9 3.6 GHz CPU and

32 GB RAM. We use the `scikit-learn` implementation of traditional machine learning algorithms, such as logistic regression and regularized linear regression. We implement artificial NNs using `Keras`, which is an high-level API for `TensorFlow`. In all experiments, we set the number of epochs to 50, batch size is equal to 128, and the Stochastic Gradient Descent (SGD) optimization method is used with a fixed learning rate of 0.001. To implement cost-sensitive learning algorithms, we use built-in functions in `scikit-learn` and `Keras`. We use `imblearn` implementation [78] of SMOTE in our experimental results. All parameters associated with ML algorithms are set to their default values unless otherwise stated.

A. BINARY CLASSIFICATION

This section compares traditional ML algorithms with artificial NNs consisting of one to three hidden layers (i.e., $L = 1, 2, 3$). The three traditional classification algorithms are support vector machines (SVM), random forests (RF), and logistic regression (LogReg). We select the radial basis function kernel for SVM and the number of trees in RF is 100. These three methods are widely used in the literature and cover various learning techniques, such as margin-based classification and ensemble learning. Since we are focusing on a binary classification problem, we use target values 0 and 1 for failure and safe simulations, respectively. In this problem, the input training data is highly imbalanced because only 1,063 simulations lead to structural failure (13% of total simulations). Thus, we will investigate the effectiveness of two imbalanced learning strategies discussed in the previous section (illustrated in Figure 2). For each ML algorithm used in this experiment, we refer to re-sampling methods as ''sampled'' and use ''weighted'' for cost-sensitive learning methods. For over-sampling using SMOTE, the default value of the number of nearest neighbors is set to 5, and the number of samples in the two classes will be equalized. Hence, the number of training examples is significantly higher than 7,000 when using SMOTE because of generating synthetic samples. The ''weighted'' mode automatically adjusts the class weights, i.e., $\beta_{\mathcal{M}}$ and $\beta_{\mathcal{S}}$ in Equation (7), inversely proportional to class frequencies in the input training data; thus, the number of training examples in $\mathcal{D}_{\text{train}}$ will be unchanged.

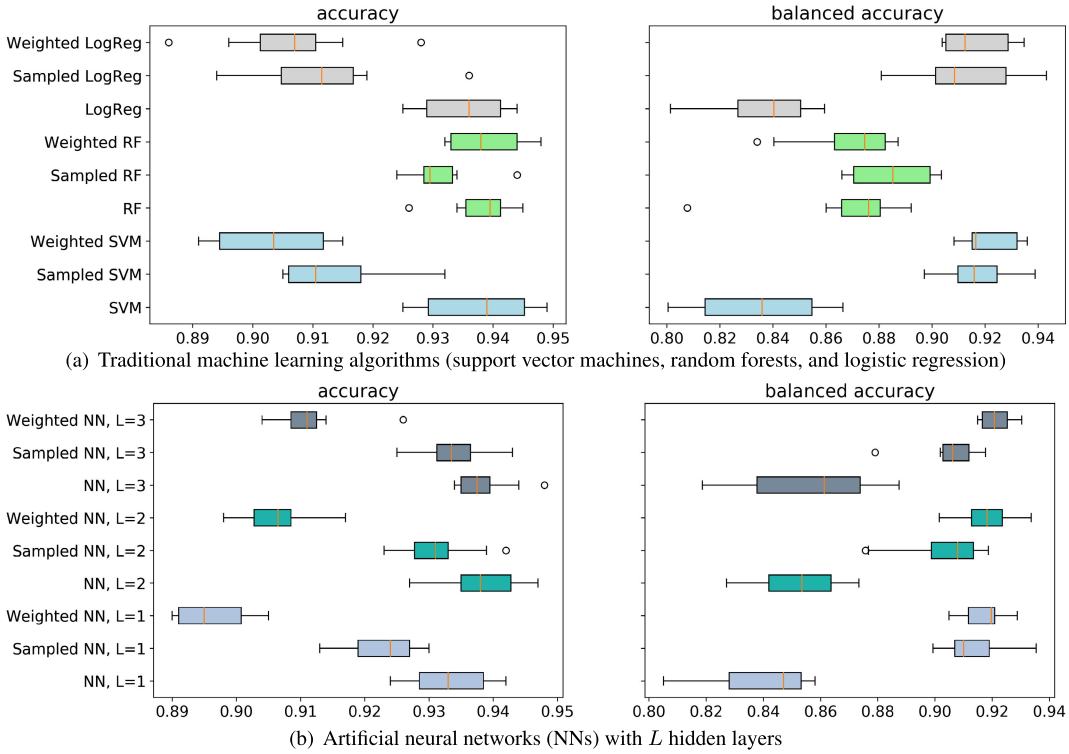
Figure 7 presents classification accuracy results in the form of a boxplot using two evaluation metrics. In the binary classification problem with $y = 0$ (negative class) and $y = 1$ (positive class), true negatives and true positives are samples that are correctly classified. On the other hand, false negatives and false positives are samples that are misclassified. A widely used metric to assess classification is basic accuracy, which is the ratio of correct predictions to the total number of samples in the test data set $\mathcal{D}_{\text{test}}$ (i.e., 1,000 test samples in our setting). However, in the case of imbalanced data, this metric is misleading since the minority class holds a small effect on this measure [79]. Therefore, this section utilizes ''balanced accuracy,'' which avoids inflated performance estimates on imbalanced data sets. Balanced accuracy is defined as the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate) in the following form:

$$\text{balanced accuracy} = \frac{1}{2}\Big(\frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}}\Big) \quad (10)$$

where both accuracy and balanced accuracy take values between 0 and 1, and higher values indicate better predictive models.

Based on Figure 7(a), we see that utilizing imbalanced learning methods generally improves the performance of the three traditional ML algorithms. When facing imbalanced data, using standard learning methods without modifying them leads to higher accuracy levels, but the balanced accuracy score is lower because of overlooking the minority class. Therefore, we mainly use the balanced accuracy score to compare the performance of classification methods in this experiment. Except for random forests, cost-sensitive learning (referred to as ''weighted'' in the plot) turns out to be more effective than SMOTE. Another advantage of cost-sensitive learning is its lower time complexity during the training phase because the number of examples from the minority class is not increased to re-balance the training data. Among traditional learning algorithms *with* imbalanced learning, support vector machines and logistic regression have similar performance (the median balanced accuracy gets very close to 0.92) and outperform random forests. Interestingly, we observe that the random forest algorithm's standard loss function is less sensitive to the class imbalance problem. Thus, if we cannot utilize imbalanced learning methods, random forests outperform the other two classification algorithms.

Figure 7(b) reports classification results for NNs with varying numbers of hidden layers. We use a brute-force approach to determine the number of neurons in each hidden layer. We use fully connected hidden layers with the ReLU activation function. For all three cases, we set the number of neurons per hidden layer to 16. The main reason for this choice is that we did not observe significant improvements by changing the number of neurons per hidden layer when $L = 2$ and $L = 3$. Moreover, using the same number of neurons allows us to keep the width of the designed neural network unchanged. Like traditional learning algorithms, we see that employing imbalanced learning techniques enhances the accuracy of NNs, and cost-sensitive learning leads to improved results compared to SMOTE. When comparing weighted NNs as we increase L, we see a consistent trend that increasing the number of hidden layers improves the trained model's predictive power. However, increasing the number of hidden layers requires more training observations, and there is a trade-off between the model complexity and the training size. Based on Figure 7(b), we observe that NNs with $L = 3$ hidden layers leads to highly accurate results with low standard deviation. For example, using NNs with $L = 1, 2$ and weighted logistic regression, the minimum value of the balanced accuracy score over 8 trials is close to 0.90. However, the minimum value is slightly lower than 0.92 when using NNs with $L = 3$. Therefore, we can argue



(a) Traditional machine learning algorithms (support vector machines, random forests, and logistic regression)

(b) Artificial neural networks (NNs) with L hidden layers

FIGURE 7. Comparing performance of classification algorithms using 8-fold cross-validation and two evaluation metrics.

that using 3 hidden layers produces an accurate classification model to be used as a surrogate model in this experiment.

To further demonstrate the significance of employing imbalanced learning, we plot the confusion matrix for one of the 8 trials when the classifier is a NN with $L = 3$ hidden layers. A confusion matrix is another evaluation metric with a matrix format, where each row represents the samples in an actual class, i.e., $y \in \{0, 1\}$. Each column represents the instances in a predicted class, i.e., $\hat{y} \in \{0, 1\}$. Therefore, the entries off the main diagonal indicate incorrect predictions, and we can easily identify false positive and negative rates. The confusion matrix in Figure 8(a) corresponds to the NN classifier without using any imbalanced learning strategy. As we can observe from this confusion matrix, the trained model correctly classifies 855 out of 886 samples from the majority class (approximately 97% of instances). However, only 74% of test data points from the minority class are correctly classified, which is significantly lower than the majority class rate. Such a model is not valuable because the minority group corresponds to simulations that lead to structural failure, and identifying them is extremely important. Fortunately, based on Figure 8(b), we observe that using cost-sensitive learning substantially improves the ratio of correctly classified points from the minority class, where 98% of test samples are correctly classified. At the same time, we notice a modest decrease in the classification accuracy for the majority class.

The next experiment examines the impact of utilizing convolutional layers on the classification performance. As mentioned earlier, a Convolutional Neural Network (CNN) consists of both convolutional and fully-connected layers to extract features from data with spatial or temporal patterns. Neurons in a convolutional layer will only be connected to a small region of the layer before it, instead of all neurons in a fully-connected manner. The sets of weights for convolutional layers is referred to as a filter or kernel. Figure 9(a) plots the confusion matrix for MLP with $L = 3$ fully-connected hidden layers on a trial with 1000 test data points (we set the number of neurons per hidden layer to 16). In Figure 9(b), we implement a CNN by using the `Conv1D` layer of `Keras` for extracting features before the three hidden layers. The convolutional layer has 64 filters of size 2. As the main purpose of this experiment is understanding the effect of using CNNs, we did not use any imbalanced learning technique. These results show that using CNNs does not noticeably affect the classification accuracy level because the input

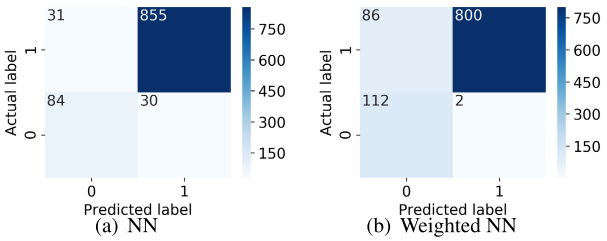


FIGURE 8. Confusion matrix for one of the trials with 1000 test data points.

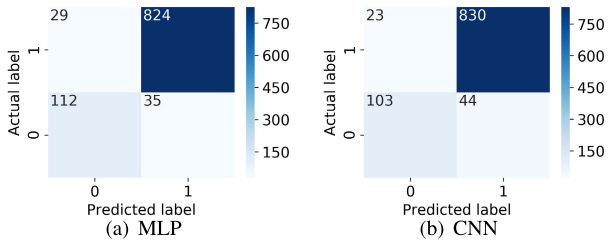


FIGURE 9. Examining the classification performance on a trial with 1000 test data points using MLP ($L = 3$ hidden layers) and CNN (`Conv1D` layer plus $L = 3$ hidden layers). Each fully-connected layer has 16 neurons.

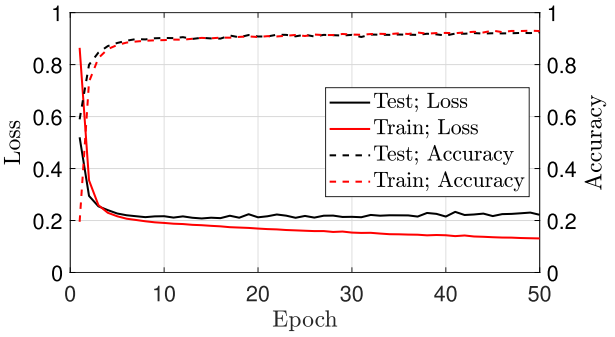


FIGURE 10. Learning curves of model performance on the train and test data.

parameter space is relatively small (i.e., $d = 49$ features) without spatial or temporal patterns. Moreover, we observed that increasing the kernel size leads to a similar trend and MLPs perform on par with CNNs in our case study.

In the final experiment of this section, we plot the values of the cross-entropy loss and the classification accuracy as a function of the epoch number for MLP with $L = 3$ hidden layers and 16 neurons per hidden layer. These results are shown in Figure 10, where we use 6400 samples for training and the remaining 1600 points form the test set. Comparing the test metrics to the train metrics reveal that the neural network with $L = 3$ hidden layers manages to avoid overfitting. This is consistent with our observation in Figure 7 because the variation in cross-validation is relatively small.

B. REGRESSION ANALYSIS

In this section, our goal is to evaluate the performance of regression models using traditional machine learning algorithms and NNs that enable capturing nonlinear relationships. We remove the minority class, i.e., structural failure, from the entire data set used in the previous section. Also, 8-fold cross-validation is used to generate training and test data sets. The input parameter space consists of $d = 49$ features representing ground motion and material properties. As mentioned before, we consider three quantities of interest: maximum top displacement, Δ_{max}, maximum base shear, F_{max}, and maximum total relative displacement, δ_{max}. In the data normalization step, we use the min-max normalization so that all three output variables are in the interval (0, 1).

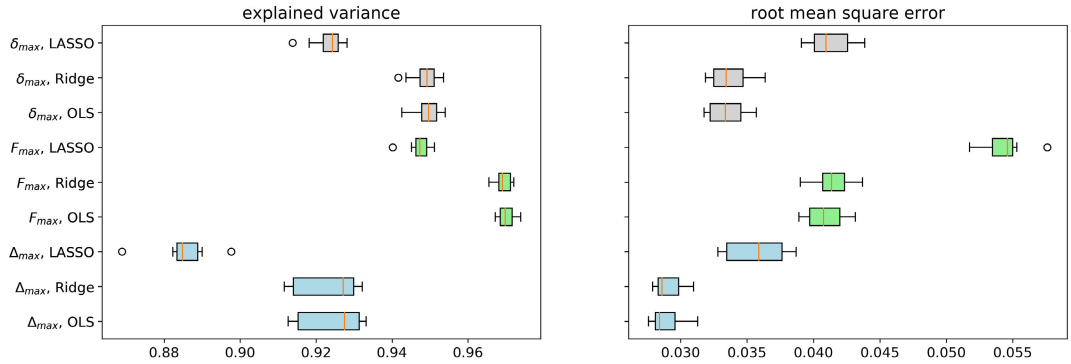
Figure 11(a) compares three standard regression algorithms: ordinary least squares or OLS for short, ridge regression (Ridge), and LASSO. Recall that based on Equation (5), OLS sets the regularization parameter $\lambda = 0$. Ridge and LASSO use regularization terms based on the ℓ_2-norm and ℓ_1-norm of the weight vector. Using hyper-parameter tuning, we set $\lambda = 0.1$ and $\lambda = 0.01$ for Ridge and LASSO, respectively. Moreover, we use two evaluation metrics in this experiment: explained variance and root mean square error. To define these two scores, let $\mathbf{y}$ denote the actual target outputs, and we use $\hat{\mathbf{y}}$ to denote the estimated outputs generated by a regression model. The explained variance is defined as follows:

$$\text{explained variance} = 1 - \frac{\text{Var}(\mathbf{y} - \hat{\mathbf{y}})}{\text{Var}(\mathbf{y})}, \tag{11}$$

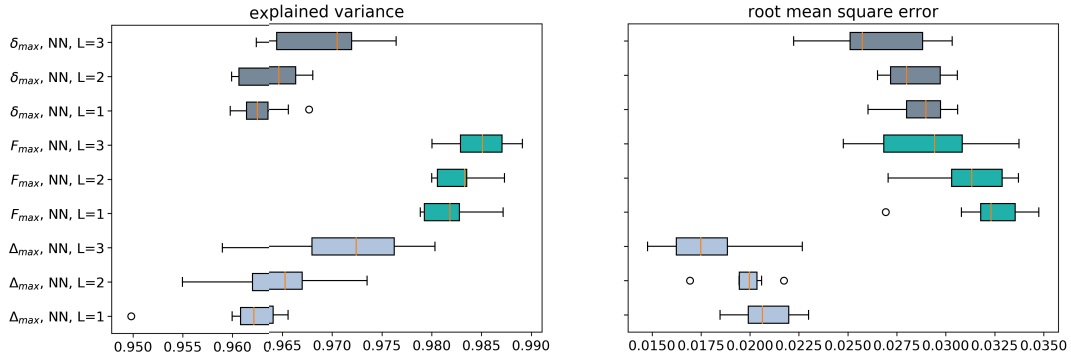
where $\text{Var}(\cdot)$ computes variance of its argument. In this case, the best possible score is 1, and lower values indicate an inferior model. Root mean square error represents the square root of the second sample moment of the differences between predicted values and observed values for $\mathcal{D}_{\text{test}}$. That is, we get $\sqrt{(1/n')\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2}$, where n' is the size of the test data set. To interpret this regression score, note that a value of 0 means a perfect fit to the data, and lower values indicate better performance (hence, opposite of explained variance).

We observe that OLS and ridge regression outperform LASSO for all three quantities of interest according to the reported results. Moreover, the median explained variance score is greater than 0.92 when using OLS and ridge regression across three output variables. Also, the median root mean square error is lower than 0.045, except for LASSO. Therefore, we conclude that linear models provide reasonable surrogate models for extracting relationships between the input variables and the three QoIs. Another interesting observation is that a model with a higher explained variance score does not necessarily lead to better performance concerning other evaluation metrics. For example, based on explained variance, estimating base shear is more accurate than the other output variables. However, we notice that the trained model for estimating base shear is the least accurate model according to the root mean square error.

We present the two metrics for regression tasks in Figure 11(b) for artificial NNs with $L = 1$ to $L = 3$ hidden layers. When $L = 1$, the number of neurons for the hidden layer is set to 32. For the other two cases, i.e., $L = 2, 3$, we select (32, 16) and (32, 16, 16) neurons per hidden layer and all activation functions for hidden layers are rectified linear units (ReLUs). Moreover, we set the number of neurons in the output layer to 3 since we aim to estimate three target values. As we see, all trained NNs result in improved regression results for the three quantities of interest. Notably, the minimum root mean square score using linear regression models is roughly 0.03, whereas the maximum value of root mean square error for NNs is approximately 0.035. Therefore, utilizing NNs significantly increases the explained variance score and reduces root mean square error



(a) Traditional machine learning algorithms (ordinary least squares, ridge regression, and LASSO)



(b) Artificial neural networks (NNs) with L hidden layers

FIGURE 11. **Comparing performance of regression models for three quantities of interest using 8-fold cross-validation.**

for all three quantities of interest. This experiment shows the importance of nonlinear regression techniques for extracting relationships between the input parameter space and the structural responses. While we used the identity function for the output layer, the rectified linear unit (ReLU) activation function in hidden layers enabled capturing nonlinear trends in the training data set. We did not observe any significant improvement in this experiment by increasing the number of hidden layers L. For example, Figure 12 compares the performance of MLP with $L = 3$ amd $L = 6$ hidden layers for predicting Δ_{max} (we use 16 neurons in each additional hidden layer). In this experiment, we also explore the use of dropout for regularizing neural networks. The dropout layer randomly sets input units to 0 with a rate specified by the probability parameter p. Based on the reported results in Figure 12, we notice that increasing the number of hidden layers does not substantially improve the performance, and using dropout with $p = 0.1$ degrades the performance in our case study. Therefore, we conclude that using $L = 3$ hidden layers provides a reasonable trade-off between the model performance and complexity due to the limited number of observations in our case study.

To further demonstrate the predictive power of the neural network with $L = 3$ hidden layers (which achieved the best scores in the previous experiment), we present three scatter plots of actual versus predicted values in Figure 13. We focus on the neural network's predictions in one of the eight folds used in cross-validation. We observe that almost all points are

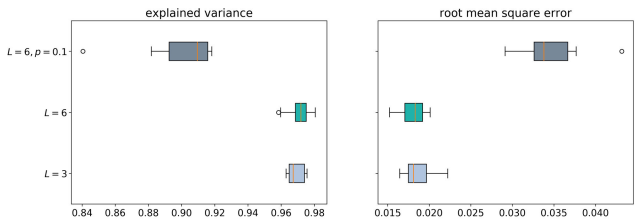


FIGURE 12. **Investigating the impact of increasing the number of hidden layers ($L = 3$ vs. $L = 6$) and using dropout for regularization.**

close to a regressed diagonal line (i.e., equity line), and evenly distributed on both sides. Therefore, there is a strong correlation between the model's predictions and its actual results. Moreover, as expected, we see that the obtained predictive models are more accurate in regions with a greater density of training instances. Thus, this experiment suggests that we can improve regression models' performance by generating additional training data points in low-density areas, i.e., closer to 1.

We conclude this section by discussing the computational complexity of machine learning-based surrogate modeling in scientific computing. Two main factors contribute to the overall cost: running numerical simulations of complex models to generate training data and the time complexity of training machine learning algorithms. The former is typically the dominant cost due to the high computational cost of running numerical models for complex problems, requir-

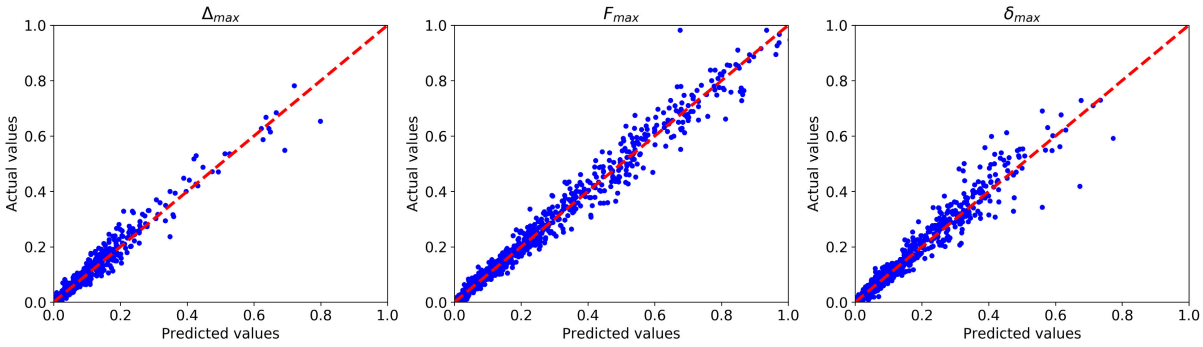


FIGURE 13. Scatter plots of actual versus predicted values for three quantities of interest when using the neural network with $L = 3$ hidden layers.

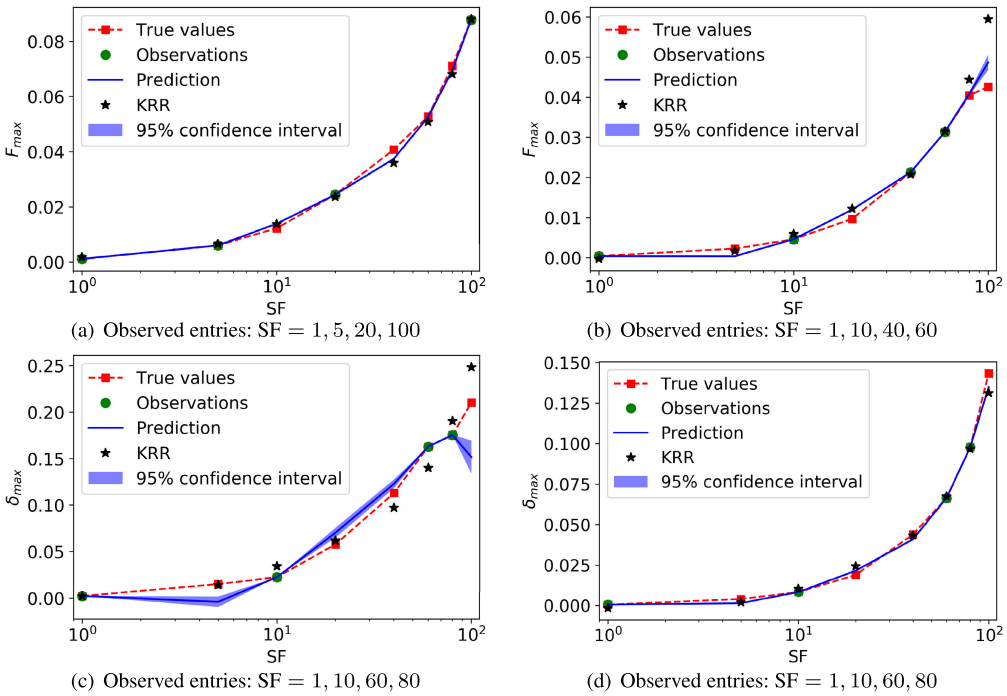
ing days or weeks of computing time. On the other hand, training machine learning algorithms on a data set with a few thousand samples requires much less computation time. In our regression analysis, linear regression models and neural networks take about 0.1 and 15 seconds, respectively. Therefore, the increase in time complexity by utilizing neural networks is negligible compared to the cost of running complex numerical simulations. Additionally, neural networks allow for developing more accurate predictive models as we saw.

C. GAUSSIAN PROCESSES

A comprehensive data-driven seismic risk analysis framework requires selecting various ground motion scale factors and conducting numerical simulations for all of those values. Therefore, a challenging task is to generate sufficient amounts of training data from numerical simulations to be used for training ML algorithms. An interesting question that has not been addressed in the previous research is whether we can extract nonlinear relationships between scale factors and the structural responses. In this problem, the input variable is the the ground motion scale factor (SF), which takes eight values in our numerical model, i.e., 1, 5, 10, 20, 40, 60, 80, and 100. Unlike the regression analysis in Section V-B, we have access to a limited number of observations for training and evaluating a regression model. For example, if we conduct numerical simulations for half of scale factor values, we only have access to 4 observed values, and the goal is to estimate the remaining 4 target outputs. Therefore, training neural networks and using cross-validation is impractical. To solve this issue, we proposed to use the Gaussian process regression algorithm (Section III).

Figure 14 exemplifies the performance of Gaussian processes for estimating the desired structural responses as a function of the scale factor. In this experiment, we use the radial basis kernel function, and the nice feature of Gaussian processes is that the kernel's hyper-parameters are optimized automatically during the training process. In each case, the input to the Gaussian process regression algorithm consists of 4 observations (denoted by green circles) and the output variables are either base shear or relative displacement. Gaussian processes allow us to obtain a point prediction using its mean and uncertainty quantification using its posterior variance. To this end, we plot the 95% confidence interval in each case for comparison with true values obtained by performing numerical simulations. As we see in Figure 14(a) and 14(d), the trained regression models are confident in their predictions (variance is almost 0), and they provide accurate estimates of missing entries or simulations. On the other hand, Figure 14(b) represents a situation where the trained model accurately predicts missing target values for SF $= 5, 20$. However, we can easily recognize that the estimated value for SF $= 100$ is not reliable because of the high variance for that point. Therefore, in this case, we can only perform another numerical simulation for SF $= 100$, which still leads to computational savings. To have a fair comparison, we present another example in Figure 14(c) that the trained model leads to high variances for most missing entries and; thus, the inherent uncertainty quantification tells us that we need to perform a complete set of numerical simulations for this case. Overall, this experiment exhibits that the Gaussian process regression algorithm allows us to (1) extract nonlinear relationships between the scale factor and structural responses, and (2) understand our uncertainty regarding the predictions.

To clarify the previous observation concerning the estimation of confidence intervals, we compare Gaussian process regression with kernel ridge regression (KRR) [80]. Similar to support vector machines, KRR uses a kernel function to extract nonlinear relationships. In this experiment, we use the polynomial kernel function with degree 2. In Figure 14(a) and 14(d), we observe that KRR provides accurate estimates of the missing simulations, similar to Gaussian process regression. However, KRR is not capable of returning confidence intervals, which is problematic when predicted values are not accurate. For example, we see in Figure 14(c) that the estimated value for SF $= 100$ using KRR is far from the actual value. On the other hand, the confidence interval obtained from Gaussian process regression allows us to understand when predicted values are reliable, which is essential for scientific computing applications.



(a) Observed entries: SF $= 1, 5, 20, 100$
(b) Observed entries: SF $= 1, 10, 40, 60$
(c) Observed entries: SF $= 1, 10, 60, 80$
(d) Observed entries: SF $= 1, 10, 60, 80$

FIGURE 14. Gaussian process regression (in blue color) and kernel ridge regression (KRR) as a function of scale factor for estimating missing entries or simulations.

VI. CONCLUSION

This paper presented a two-stage framework for developing surrogate models, consisting of classification and regression steps. We demonstrated the superior performance of neural networks and the effectiveness of imbalanced learning for training accurate classifiers that automatically identify failed simulations. Although the training data set for the classification task is challenging because of being imbalanced, our experiments show that the integration of neural networks with three hidden layers and cost-sensitive learning results in 90% accuracy and balanced accuracy. Therefore, we confirmed the practicality of the proposed framework for distinguishing severe damages to civil structures. Moreover, our regression analysis using two evaluation matrices, i.e., explained variance and root mean square error, reveals that neural networks are more expressive than linear regression models, providing valuable predictive modeling tools in scientific computing. For example, we observed that using neural networks with three hidden layers lead to accurate predictions with a minimum explained variance of 0.97. The last experiment explored the use of Gaussian processes for reducing the number of simulations when considering various scale factors. Unlike the previous research, our numerical model incorporated various scale factors to provide a comprehensive seismic hazard analysis study. We showed that Gaussian processes enable us to predict the values of missing simulations with corresponding uncertainties for reducing the computational cost associated with numerical models.

An important future research direction is to explore the trade-off between the number of numerical simulations and the accuracy of the proposed framework. In this work, we considered all combinations of aleatory and epistemic uncertainties for the eight scale factors. On the other hand, the Gaussian process regression model provides reliable estimates when performing simulations for a fraction of scale factors. Therefore, our future work will utilize partial observations from Gaussian processes to train the proposed two-stage framework. We will study the impact of using estimated values on the performance of both classification and regression steps. Last but not least, an important step towards developing an optimal surrogate model is to identify a group of efficient and sufficient input parameters. The future research should focus on developing a group of scalar meta-features for a stochastic time-series in order to reduce the computational burden of meta-modeling.

The proposed framework applies to a wide range of scientific computing problems, including computational materials science. The central goal is to predict material properties given a set of input parameters. While the previous research focused on just developing regression models, our framework allows for identifying combinations of input parameters that lead to materials with undesirable properties. Thus, our framework's second step trains a regression model on only a fraction of the available data set comprising informative input-output pairs. Other applications of the proposed framework include constructing surrogate models in computational biology, fluid dynamics, and risk analysis of natural hazards.

REFERENCES

[1] R. L. Winkler, "Uncertainty in probabilistic risk assessment," *Rel. Eng. Syst. Saf.*, vol. 54, nos. 2–3, pp. 127–132, Nov. 1996.

[2] M. Stewart and R. E. Melchers, *Probabilistic Risk Assessment of Engineering Systems*. Cham, Switzerland: Springer, 1997.

[3] A. D. Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?" *Struct. Saf.*, vol. 31, no. 2, pp. 105–112, Mar. 2009.

[4] S. D. Koduru, "Strategies for separation of aleatory and epistemic uncertainties," in *Proc. 12th Int. Conf. Appl. Statist. Probab. Civil Eng.*, Vancouver, BC, Canada, Jul. 2015, pp. 1–5.

[5] O. C. Celik and B. R. Ellingwood, "Seismic fragilities for non-ductile reinforced concrete frames–role of aleatoric and epistemic uncertainties," *Struct. Saf.*, vol. 32, no. 1, pp. 1–12, Jan. 2010.

[6] M. Dolšek, "Simplified method for seismic risk assessment of buildings with consideration of aleatory and epistemic uncertainty," *Struct. Infrastruct. Eng.*, vol. 8, pp. 939–953, Dec. 2012.

[7] B. Merz and A. H. Thieken, "Separating natural and epistemic uncertainty in flood frequency analysis," *J. Hydrol.*, vol. 309, nos. 1–4, pp. 114–132, Jul. 2005.

[8] J. C. Helton, J. D. Johnson, W. L. Oberkampf, and C. J. Sallaberry, "Representation of analysis results involving aleatory and epistemic uncertainty," *Int. J. Gen. Syst.*, vol. 39, no. 6, pp. 605–646, Aug. 2010.

[9] R. McGuire, "Probabilistic seismic hazard analysis and design Earthquakes: Closing the loop," *Bull. Seismol. Soc. Amer.*, vol. 85, pp. 1275–1284, 1995.

[10] K. Porter, R. Kennedy, and R. Bachman, "Creating fragility functions for performance-based earthquake engineering," *Earthq. Spectra*, vol. 23, no. 2, pp. 471–489, May 2007.

[11] J. W. Baker, "Efficient analytical fragility function fitting using dynamic structural analysis," *Earthq. Spectra*, vol. 31, no. 1, pp. 579–599, Feb. 2015.

[12] D. Vamvatsikos and M. Fragiadakis, "Incremental dynamic analysis for estimating seismic performance sensitivity and uncertainty," *Earthq. Eng. Struct. Dyn.*, vol. 39, pp. 141–163, 2010.

[13] M. Dolsek, "Incremental dynamic analysis with consideration of modeling uncertainties," *Earthq. Eng. Struct. Dyn.*, vol. 38, no. 6, pp. 805–825, May 2009.

[14] M. A. Hariri-Ardebili and V. E. Saouma, "Probabilistic seismic demand model and optimal intensity measure for concrete dams," *Struct. Saf.*, vol. 59, pp. 67–85, Mar. 2016.

[15] M. R. Rajashekhar and B. R. Ellingwood, "A new look at the response surface approach for reliability analysis," *Struct. Saf.*, vol. 12, no. 3, pp. 205–220, Oct. 1993.

[16] J. Ghosh, J. E. Padgett, and L. Dueñas-Osorio, "Surrogate modeling and failure surface visualization for efficient seismic vulnerability assessment of highway bridges," *Probabilistic Eng. Mech.*, vol. 34, pp. 189–199, Oct. 2013.

[17] Y. Xie, M. Ebad Sichani, J. Padgett, and R. DesRoches, "The promise of implementing machine learning in Earthquake engineering: A state-of-the-art review," *Earthq. Spectra*, vol. 3, Jul. 2020, Art. no. 8755293020919419.

[18] D. Yinfeng, L. Yingmin, L. Ming, and X. Mingkui, "Nonlinear structural response prediction based on support vector machines," *J. Sound Vib.*, vol. 311, nos. 3–5, pp. 886–897, Apr. 2008.

[19] J. Tezcan and Q. Cheng, "Support vector regression for estimating earthquake response spectra," *Bull. Earthq. Eng.*, vol. 10, no. 4, pp. 1205–1219, Aug. 2012.

[20] M. A. Hariri-Ardebili and F. Pourkamali-Anaraki, "Support vector machine based reliability analysis of concrete dams," *Soil Dyn. Earthq. Eng.*, vol. 104, pp. 276–295, Jan. 2018.

[21] M. A. Hariri-Ardebili and F. Pourkamali-Anaraki, "Simplified reliability analysis of multi hazard risk in gravity dams via machine learning techniques," *Arch. Civil Mech. Eng.*, vol. 18, no. 2, pp. 592–610, Feb. 2018.

[22] A. Alimoradi and J. L. Beck, "Machine-learning methods for Earthquake ground motion analysis and simulation," *J. Eng. Mech.*, vol. 141, no. 4, Apr. 2015, Art. no. 04014147.

[23] J. Kiani, C. Camp, S. Pezeshk, and N. Khoshnevis, "Application of pool-based active learning in reducing the number of required response history analyses," *Comput. Struct.*, vol. 241, Dec. 2020, Art. no. 106355.

[24] J. Kiani, C. Camp, and S. Pezeshk, "On the application of machine learning techniques to derive seismic fragility curves," *Comput. Struct.*, vol. 218, pp. 108–122, Jul. 2019.

[25] N. D. Lagaros and M. Papadrakakis, "Neural network based prediction schemes of the non-linear seismic response of 3D buildings," *Adv. Eng. Softw.*, vol. 44, no. 1, pp. 92–115, Feb. 2012.

[26] K. Morfidis and K. Kostinakis, "Approaches to the rapid seismic damage prediction of r/c buildings using artificial neural networks," *Eng. Struct.*, vol. 165, pp. 120–141, Jun. 2018.

[27] T. Kim, O.-S. Kwon, and J. Song, "Response prediction of nonlinear hysteretic systems by deep neural networks," *Neural Netw.*, vol. 111, pp. 1–10, Mar. 2019.

[28] A. Derakhshani and A. H. Foruzan, "Predicting the principal strong ground motion parameters: A deep learning approach," *Appl. Soft Comput.*, vol. 80, pp. 192–201, Jul. 2019.

[29] T. Kim, J. Song, and O.-S. Kwon, "Probabilistic evaluation of seismic responses using deep learning method," *Struct. Saf.*, vol. 84, May 2020, Art. no. 101913.

[30] B. K. Oh, B. Glisic, S. W. Park, and H. S. Park, "Neural network-based seismic response prediction model for building structures using artificial earthquakes," *J. Sound Vib.*, vol. 468, Mar. 2020, Art. no. 115109.

[31] R. Zhang, Y. Liu, and H. Sun, "Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling," *Eng. Struct.*, vol. 215, Jul. 2020, Art. no. 110704.

[32] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[33] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9268–9277.

[34] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," *J. Math. Psychol.*, vol. 85, pp. 1–16, Aug. 2018.

[35] J. Huggins, T. Campbell, and T. Broderick, "Coresets for scalable Bayesian logistic regression," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 4080–4088.

[36] M. A. Hariri-Ardebili, "Analytical failure probability model for generic gravity dam classes," *Proc. Inst. Mech. Eng., O, J. Risk Rel.*, vol. 231, no. 5, pp. 546–557, Oct. 2017.

[37] Y. Ho and S. Wookey, "The Real-World-Weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[39] H. Zhang, T. Weng, P. Chen, C. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 4939–4948.

[40] A. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 4148–4158.

[41] P. Mehta, M. Bukov, C.-H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, "A high-bias, low-variance introduction to machine learning for physicists," *Phys. Rep.*, vol. 810, pp. 1–124, May 2019.

[42] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights into Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018.

[43] D. Kang, D. Dhar, and A. Chan, "Incorporating side information by adaptive convolution," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 3867–3877.

[44] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[45] M. A. Hariri-Ardebili and S. Barak, "A series of forecasting models for seismic evaluation of dams based on ground motion meta-features," *Eng. Struct.*, vol. 203, Jan. 2020, Art. no. 109657.

[46] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large-scale ℓ_1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 8, pp. 1519–1555, Jul. 2007.

[47] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proc. Int. Conf. Mach. Learn.*, 1998, pp. 515–521.

[48] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., Ser. B Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[49] P. A. Gutierrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez, "Ordinal regression methods: Survey and experimental study," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 127–146, Jan. 2016.

[50] N. B. Abdel-Hamid, S. ElGhamrawy, A. E. Desouky, and H. Arafat, "A dynamic spark-based classification framework for imbalanced big data," *J. Grid Comput.*, vol. 16, no. 4, pp. 607–626, Dec. 2018.
[51] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[52] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1567–1578.
[53] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, Feb. 2020, Art. no. 107377.
[54] S. Daskalaki, I. Kopanas, and N. Avouris, "Evaluation of classifiers for an uneven class distribution problem," *Appl. Artif. Intell.*, vol. 20, no. 5, pp. 381–417, Jun. 2006.
[55] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, Apr. 2018.
[56] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.
[57] D. Xu, Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen, "Survey on multi-output learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2409–2429, Jul. 2020.
[58] C. Williams and C. Rasmussen, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[59] F. Pourkamali-Anaraki, S. Becker, and M. Wakin, "Randomized clustered Nyström for large-scale kernel machines," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3960–3967.
[60] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *J. Mach. Learn. Res.*, vol. 11, pp. 1865–1881, Mar. 2010.
[61] M. A. Hariri-Ardebili, H. Rahmani-Samani, and M. Mirtaheri, "Seismic stability assessment of a high-rise concrete tower utilizing endurance time analysis," *Int. J. Structural Stability Dyn.*, vol. 14, no. 06, Aug. 2014, Art. no. 1450016.
[62] M. Yahyai, B. Rezayibana, and A. S. Daryan, "Nonlinear seismic response of milad tower using finite element model," *Struct. Des. Tall Special Building*, vol. 18, no. 8, pp. 877–890, Dec. 2009.
[63] B. Asgarian, M. Yahyai, M. Mirtaheri, H. R. Samani, and P. Alanjari, "Incremental dynamic analysis of high-rise towers," *Struct. Des. Tall Special Buildings*, vol. 19, no. 8, pp. 922–934, Dec. 2010.
[64] F. McKenna, "Opensees: A framework for earthquake engineering simulation," *Comput. Sci. Eng.*, vol. 13, no. 4, pp. 58–66, 2011.
[65] M. H. Scott and G. L. Fenves, "Plastic hinge integration methods for force-based Beam–Column elements," *J. Struct. Eng.*, vol. 132, no. 2, pp. 244–252, Feb. 2006.
[66] A. M. Halabian and M. H. E. Naggar, "Effect of foundation flexibility on seismic response of reinforced concrete TV-towers," *Can. J. Civil Eng.*, vol. 28, no. 3, pp. 465–481, Jun. 2001.
[67] A. M. Halabian, M. H. El Naggar, and B. J. Vickery, "Nonlinear seismic response of reinforced-concrete free-standing towers with application to TV towers on flexible foundations," *Struct. Des. Tall Buildings*, vol. 11, no. 1, pp. 51–72, 2002.
[68] B. D. Scott, R. Park, and M. J. Priestley, "Stress-strain behavior of concrete confined by overlapping hoops at low and high strain rates," *J. Proc.*, vol. 79, no. 1, pp. 13–27, 1982.
[69] A. Giuffrà, "Il comportamento del cemento armato per sollecitazioni cicliche di forte intensità," *Giornale del Genio Civile*, 1970.
[70] M. Menegotto, "Method of analysis for cyclically loaded rc plane frames including changes in geometry and non-elastic behavior of elements under combined normal force and bending," in *Proc. Symp. Resistance Ultimate Deformability Struct. Acted Defined Repeated Loads*, 1973, pp. 15–22.
[71] J. Bauschinger, "On the change of the elastic limit and the strength of iron and steel, by drawing out, by heating and cooling, and by repetition of loading (summary)," *Minutes Proc. Inst. Civil Eng. Sel. Abstracted Papers*, vol. 87, p. 463, May 1886.
[72] M. M. Amiri and M. Yahyai, "Estimation of damping ratio of TV towers based on ambient vibration monitoring," *Struct. Des. Tall Special Buildings*, vol. 22, no. 11, pp. 862–875, Aug. 2013.
[73] PEER. (Jul. 2017). *Ground Motion Database*. [Online]. Available: http://ngawest2.berkeley.edu/PacificEarthquakeEngineering.ResearchCenter
[74] F. Jalayer, R. De Risi, and G. Manfredi, "Bayesian cloud analysis: Efficient structural fragility assessment using linear regression," *Bull. Earthq. Eng.*, vol. 13, no. 4, pp. 1183–1203, Apr. 2015.
[75] D. Vamvatsikos and C. A. Cornell, "Incremental dynamic analysis," *Earthq. Eng. Structural Dyn.*, vol. 31, no. 3, pp. 491–514, 2002.
[76] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognit.*, vol. 38, no. 12, pp. 2270–2285, 2005.
[77] L. Munkhdalai, T. Munkhdalai, K. Park, H. Lee, M. Li, and K. Ryu, "Mixture of activation functions with extended min-max normalization for forex market prediction," *IEEE Access*, vol. 7, pp. 183680–183691, 2019.
[78] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 559–563, 2017.
[79] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0177678.
[80] F. Pourkamali-Anaraki, M. Hariri-Ardebili, and L. Morawiec, "Kernel ridge regression using importance sampling with application to seismic response prediction," in *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, Sep. 2020, pp. 1–14.

FARHAD POURKAMALI-ANARAKI (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Colorado Boulder, in 2017. He is currently an Assistant Professor with the Department of Computer Science, University of Massachusetts Lowell. He received the Best Paper Award at the 19th IEEE International Conference on Machine Learning and Applications (ICMLA). His research interests include the intersection of statistics, optimization, and machine learning on the one hand and the intersection of theory, applications, and implementations on the other side. In particular, he works on extending the use of machine learning algorithms to other domains, including reliability analysis of critical infrastructures in civil engineering and advanced manufacturing in mechanical engineering.

MOHAMMAD AMIN HARIRI-ARDEBILI is currently a Researcher with the National Institute of Standards and Technology (NIST), and affiliated with the University of Maryland College Park and University of Colorado Boulder. His main research interests are performance-based earthquake assessment of infra-structures, risk and resilience, alkali-silica reaction, uncertainty quantification, and applied machine learning. He has published more than 80 peer-reviewed journal articles. His recent book entitled *Aging, Shaking and Cracking of Infrastructures: From Mechanics to Dams and Nuclear Structures* will be published by Springer-Nature, in early 2021.