Accepted by Journal of Geotechnical and Geoenvironmental Engineering, ASCE, March 2020

SS-XGBoost: a machine learning framework for predicting Newmark sliding displacements of slopes

Mao-Xin Wang[1], Duruo Huang[2], Gang Wang[3], M.ASCE and Dian-Qing Li[4], M.ASCE

Abstract: Estimation of Newmark sliding displacement plays an important role for evaluating seismic stability of slopes. Current empirical models generally utilize predefined functional forms and relatively large model uncertainty is involved. On the other hand, machine learning method has superior capacity in processing comprehensive datasets in a non-parametric way. In this study, a machine learning framework is proposed to predict Newmark sliding displacements using the extreme gradient boosting model (XGBoost) and the Next Generation Attenuation (NGA)-West2 database, where the subset simulation (SS) is coupled with *K*-fold cross validation (CV) technique for the first time to tune hyper-parameters of the XGBoost model. The framework can achieve excellent generalization capability in predicting displacements and prevent data overfitting by using

[1] Ph.D. Student, State Key Laboratory of Water Resources and Hydropower Engineering Science, Institute of Engineering Risk and Disaster Prevention, Wuhan University, 299 Bayi Road, Wuhan 430072, China

[2] Associate Professor, Department of Hydraulic Engineering, Tsinghua University, Beijing 100084, China. Corresponding Author (huangduruo@tsinghua.edu.cn)

[3] Associate Professor, Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

[4] Professor, State Key Laboratory of Water Resources and Hydropower Engineering Science, Institute of Engineering Risk and Disaster Prevention, Wuhan University, 299 Bayi Road, Wuhan 430072, China

optimized hyper-parameters. The developed data-driven Newmark displacement models can better satisfy both sufficiency and efficiency criteria, and produce considerably smaller standard deviations compared with traditional empirical models. Application of the models in probabilistic seismic slope displacement hazard analysis is also demonstrated. The proposed SS-XGBoost framework has great potential in developing data-driven prediction models for a wide range of engineering applications.

Keywords: Machine learning; Newmark displacement model; Seismic slope stability; Gradient boosted regression tree; Subset simulation

Introduction

Earthquake-induced landslides are one of the most catastrophic effects of earthquakes, as evidenced by many historic events over the past decades. For example, during the 2008 Wenchuan earthquake in China, 15,000 incidences of earthquake-induced landslides, rockfalls and debris flows increased the death toll by 20,000 (Yin et al. 2009). At Tangjiashan, earthquake-induced mass movement over 2,037 million m^3 blocked the main river channel, posing a significant threat to lives downstream. Of the many landslide assessment methods, the Newmark sliding mass model and its variants have been extensively used to estimate earthquake-induced displacements in natural slopes, earth dams and landfills since the 1960s (Newmark 1965; Rathje and Bray 2000, Bray and Travasarou 2007; Du et al. 2018a, 2018b). Newmark sliding mass model and its variants have been extensively used to estimate earthquake-induced displacements in natural slopes, earth dams and landfills since the 1960s (Jibson et al. 2000, Rathje and Bray 2000, Jibson 2007, Bray and Travasarou 2007, Du et al. 2018a, 2018b). Under the assumptions of rigid sliding block procedure, the sliding initializes once the input acceleration exceeds the yield acceleration (k_y) that is determined by the properties of slope (e.g., soil properties and geometric conditions), and the permanent displacement continues to increase until the relative velocity between the block and sliding surface becomes zero. The Newmark displacement is computed through the integral of the velocity-time history of the sliding block. This rigid-block procedure is applicable to thin landslides in relatively stiff materials and is justified because almost 90% of earthquake-induced landslides are shallow slides and falls (Jibson 2011). Although the original Newmark model has been extended through some efforts, it is still the most common

analytical procedure for predicting the seismic displacements of natural slopes (Saygili and Rathje 2008; Rathje and Saygili 2009) and has become an important tool for constructing seismic landslide hazard maps (e.g., Du and Wang 2014).

In recent years, various empirical models (e.g., Bray and Travasarou 2007; Jibson 2007; Saygili and Rathje 2008; Du and Wang 2016; Song et al. 2016; Du et al. 2018b) have been proposed to correlate ground motion intensity measures (IMs) or seismological parameters to the Newmark displacement. Among them, the vector-IM models can better satisfy both the "sufficiency" and "efficiency" criteria than the scalar-IM models (Saygili and Rathje 2008, Wang 2012). However, the model uncertainty contributed from the Newmark displacement models alone is still much larger than that in ground-motion prediction equations (GMPEs) (Du et al. 2018a). Because traditional Newmark prediction models predefine the functional forms then adjust their terms according to residual analyses, the development of models strongly depends on experience. It is still challenging to capture the highly nonlinear behavior of the sliding block using only limited IMs and simple functional forms, which might be the reason that limits further reduction of the standard deviation of empirical prediction models throughout years.

As the fast development of artificial intelligence (AI) techniques, machine learning (ML) methods have been gradually introduced to solve engineering problems for their robust predictive ability and excellent generalization capability. A comprehensive overview of the application of ML methods to seismology can be found in Kong et al. (2018). Other examples, such as assessment of soil liquefaction resistance (e.g., Juang et al. 2003), simulation of ground motion recordings (e.g., Alimoradi and Beck 2014) and ground-motion prediction

models (e.g., Alavi and Gandomi 2011; Derras et al. 2012; Khosravikia et al. 2018) can be found in the literature. Recently compiled Next Generation Attenuation NGA-West2 database (Ancheta et al. 2014) provides an opportunity to develop data-driven Newmark displacement prediction models, which are expected to have better generalization capability (e.g., less bias and smaller standard deviation on the unseen dataset) than traditional Newmark displacement models. Also, updating Newmark models using the NGA-West2 database is essential because it contains more than 17500 additional recordings and more than doubled number of records for magnitude larger than 5.5 compared with the NGA-West1 database (Ancheta et al. 2014).

There are two main challenges to develop data-driven Newmark models. First, the Newmark displacement data is highly nonlinear and imbalanced. Because sliding occurs only if the peak ground acceleration (PGA) exceeds k_y, non-zero displacement data becomes much less for regression when k_y is relatively larger. This is why some empirical models (e.g., Lee and Green 2015; Du and Wang 2016) developed separate prediction equations for different k_y. To address this challenge, an advanced implementation of the gradient boosted regression tree (GBRT) (Friedman 2001) known as the extreme gradient boosting (XGBoost) (Chen and Guestrin 2016), is adopted in this study to predict the Newmark displacement. The GBRT combines a sequence of regression trees (RTs) into a powerful prediction model and has been proven to be an advantageous tool compared with other ML algorithms for some data mining problems (e.g., Youssef et al. 2016). Compared to the traditional GBRT, the regularization idea was introduced into XGBoost to penalize the tree complexity for a better model performance (Chen and Guestrin 2016).

The second challenge in developing a data-driven Newmark model is the model

generalization capability. In this study, we propose a "hyper-parameter tuning" method using subset simulation to optimize XGBoost model configuration in order to achieve a bias-variance trade-off and better model generalization (e.g., Goodfellow et al. 2016). On the contrary, traditional strategies such as the rules-of-thumb (e.g., Hinton 2012) or the grid search (e.g., Pedregos et al. 2011) tune model configuration manually, which is time-consuming and impractical for a high-dimensional case. The subset simulation (SS) was originally developed for reliability analysis (Au and Beck 2001) and optimization (Li and Au 2010; Li 2011). For the first time, it is coupled with the *K*-fold cross validation (CV) for hyper-parameter tuning of the XGBoost model in this study.

This study aims at proposing an efficient SS-XGBoost framework to develop data-driven Newmark displacement prediction models. This paper starts with an introduction to the dataset used for model development. Next, methods and implementation procedure for the proposed SS-XGBoost framework are presented, followed by evaluation of the model performance and comparison with traditional empirical models based on various metrics. Finally, the application of the data-driven models to the probabilistic seismic slope displacement hazard analysis (PSSDHA) is illustrated using three hypothetical slope cases.

Ground Motion Database and Data Preparation

The sequential procedure for generating the final dataset primarily includes four steps: First, a subset of the NGA-West2 database from Pacific Earthquake Engineering Research (PEER) Center (http://ngawest2.berkeley.edu/) was generated by excluding low-quality recordings based on selection criteria by Campbell and Bozorgnia (2014). This resulted in 15521 pairs of

motions with two horizontal components from 322 worldwide earthquake events. Note that two horizontal recordings at a same station are treated as independent motions in Newmark analysis; Second, IMs (i.e., PGA, PGV, Ia) of individual recordings are calculated to generate the predictor variable samples; Third, Newmark displacements of individual recordings are computed to generate the target variable samples; Finally, samples whose displacements smaller than 1×10^{-4} cm are excluded from the database. The final dataset includes 43832 data points, whose corresponding distribution of moment magnitude (M_w) and rupture distance (R_{rup}) is shown in Fig. 1.

Following Saygili and Rathje (2008), the PGA, peak ground velocity (PGV) and Arias intensity (I_a) are considered to develop Newmark displacement models. Individual samples have a form of [$\boldsymbol{x}_i$, y_i], where $\boldsymbol{x}_i$ = i-th sample of predictor variables (i.e., [PGA, PGV, k_y], or [PGA, I_a, k_y] or [PGA, PGV, I_a, k_y]); and y_i = i-th sample of target variable (i.e., natural logarithm of observed Newmark displacement). Note that neither the commonly used logarithmic transformation (e.g., Jibson 2007; Saygili and Rathje 2008) or normalization (e.g., Derras et al. 2012; Jones et al. 2018) of predictor variables is needed for our tree-based model, which is invariant to scaling of predictor variables. This can somewhat simplify the data preprocessing compared to other ML methods such as the neural network. The whole dataset was randomly divided into training and testing sets following the ratios of 80% and 20%, respectively (e.g., Alavi and Gandomi 2011; Khosravikia et al. 2018). The former set was involved in the training process, and the latter set was used for testing model performance on the unseen data (generalization capability) (e.g., Ren et al. 2018; Jones et al. 2018). In ML studies, the testing error is generally larger than the training error, so the key challenge is to

low the training error and narrow the gap between the training and testing errors to avoid both underfitting and overfitting problems (e.g., Goodfellow et al. 2016).

Gradient Boosted Regression Tree for Predicting Newmark Displacement

Regression Tree

The regression tree (RT) is introduced first because it is the base model of the gradient boosted regression tree (GBRT) (Friedman 2001). For illustration, a simple RT model using X_1 and X_2 as predictor variables is shown in Fig. 2. The predictor variable space is divided into several regions as shown in Fig. 2(a) and each region is represented by a path from the root split node to the corresponding leaf node as seen in Fig. 2(b). Each leaf node has a specific leaf score (w), which represents the predicted value for that region and will be fitted as the average of target variable samples in that region. The training of a RT is to search the optimal split nodes and continue the partition process (i.e., growing tree) until the stopping criterion is reached. In this study, we adopted the pruning technique that allows for removing unnecessary split nodes in a RT to prevent overfitting. Details can be referred to Hastie et al. (2009).

XGBoost: An Advanced Implementation of Gradient Boosted Regression Tree

The idea of ensemble modeling is to construct a powerful model by taking advantage of a collection of "weak" base models (e.g., Hastie et al. 2009; Shao and Deng 2018). The parallel ensemble technique has been widely used to reduce the model uncertainty in probabilistic seismic hazard analysis (PSHA) and PSSDHA through a logic tree framework (e.g., Du et al.

2018a). As a sequential ensemble technique, the gradient boosted regression tree (GBRT) develops a series of base regression trees (RTs) over time to enlarge the model capacity. In a forward stepwise manner, the additive training process of the boosted model can be expressed as:

$$
\begin{aligned}
&\hat{y}^{(0)} = 0 \\
&\hat{y}^{(1)} = v\, f_1\left(\boldsymbol{x};\ \boldsymbol{\Theta}_1\right) = \hat{y}^{(0)} + v\, f_1\left(\boldsymbol{x};\ \boldsymbol{\Theta}_1\right) \\
&\hat{y}^{(2)} = v\sum_{j=1}^{2} f_j\left(\boldsymbol{x};\ \boldsymbol{\Theta}_j\right) = \hat{y}^{(1)} + v\, f_2\left(\boldsymbol{x};\ \boldsymbol{\Theta}_2\right) \\
&\quad \cdots \\
&\hat{y}^{(T)} = v\sum_{j=1}^{T} f_j\left(\boldsymbol{x};\ \boldsymbol{\Theta}_j\right) = \hat{y}^{(T-1)} + v\, f_T\left(\boldsymbol{x};\ \boldsymbol{\Theta}_T\right)
\end{aligned}
\tag{1}
$$

where T = number of RTs for boosting; $\boldsymbol{\Theta}_j$ = structure of the j-th RT (including all sections of a tree such as split and leaf nodes in Fig. 2); v = shrinkage factor (also known as learning rate that satisfies $0 < v < 1$ for shrinking the contribution of individual RTs) ; $\hat{y}^{(j)}$ = prediction of target variable using first j RTs; and $f_j()$ = output of the j-th RT without shrinkage, which uses predictor variables $\boldsymbol{x}$ to approximate $y - \hat{y}^{(j-1)}$ (i.e., residuals) with tree structure $\boldsymbol{\Theta}_j$. Therefore, the residuals will generally decrease as the number of RTs increases. A schematic diagram of GBRT is shown in Fig. 3 for demonstration.

To penalize the complexity of individual RTs, Chen and Guestrin (2016) proposed a scalable tree boosting system known as XGBoost, which follows the general idea of regularized learning and combines a regularization term with the traditional loss function in GBRT. The core task in boosted tree modeling is to find optimal $\boldsymbol{\Theta}_j$ and build $f_j(\boldsymbol{X};\ \boldsymbol{\Theta}_j)$ at the j-th step, which is fulfilled by minimizing the objective function:

$$\hat{\Theta}_j = \arg\min_{\Theta_j}\left\{\sum_{i=1}^{N} L\left[y_i, \hat{y}_i^{(j-1)} + v f_j\left(x_i;\ \mathbf{\Theta}_j\right)\right] + \Omega\left(\mathbf{\Theta}_j\right)\right\}$$
$$= \arg\min_{\Theta}\left\{\sum_{i=1}^{N}\left[\hat{y}_i^{(j-1)} - y_i + v f_j\left(x_i;\ \mathbf{\Theta}_j\right)\right]^2 + \Omega\left(\mathbf{\Theta}_j\right)\right\} \quad (2)$$

where N = number of considered samples; $L\left(y,\ \hat{y}\right) = \left(\hat{y} - y\right)^2$ = commonly used square loss function; and $\Omega(\mathbf{\Theta}_j)$ = regularization term on the j-th RT, which is written as:

$$\Omega\left(\mathbf{\Theta}_j\right) = \gamma M_j + \frac{1}{2}\lambda\left\|w_k\right\| = \gamma M_j + \frac{1}{2}\lambda\sum_{k=1}^{M_j}\left(w_k^{(j)}\right)^2 \quad (3)$$

where $w_k^{(j)}$ = leaf score of the k-th leaf node in the j-th RT (see Fig. 2); M_j = number of leaf nodes in the j-th RT; γ = minimum loss reduction needed for a further node partition in RT; and λ = L2 regularization term on leaf scores in RT. Through the second-order Taylor expansion, Eq. (2) can be approximated as:

$$\hat{\mathbf{\Theta}}_j \simeq \arg\min_{\mathbf{\Theta}_j}\left\{\sum_{i=1}^{N}\left[L\left(y_i,\ \hat{y}_i^{(j-1)}\right) + v g_i^{(j)} f_j\left(x_i;\ \mathbf{\Theta}_j\right) + \frac{1}{2}v^2 h_i^{(j)} f_j^2\left(x_i;\ \mathbf{\Theta}_j\right)\right] + \gamma M_j + \frac{1}{2}\lambda\sum_{k=1}^{M_j}\left(w_k^{(j)}\right)^2\right\} \quad (4)$$

where $g_i^{(j)} = \partial L\left(y_i,\ \hat{y}_i^{(j-1)}\right)\Big/\partial\hat{y}_i^{(j-1)} = 2\left(\hat{y}_i^{(j-1)} - y_i\right)$ and $h_i^{(j)} = \partial^2 L\left(y_i, \hat{y}_i^{(j-1)}\right)\Big/\left(\partial\hat{y}_i^{(j-1)}\right)^2 = 2$. It is evident that more leaves (larger M_j) will be penalized by a larger γ and a larger λ will produce more regular distribution of leaf scores (smaller $w_k^{(j)}$). As y_i is the given sample and $\hat{y}^{(j-1)}$ has been determined at the (j-1)-th step, $L\left(y_i, \hat{y}_i^{(j-1)}\right)$ can be considered as a constant term and will be removed from Eq. (4). The XGBoost model finishes training after determining all of $\mathbf{\Theta}_j$ (j = 1, 2, …, T), such that Eq. (1) can be used to perform a prediction. Interested readers are referred to Chen and Guestrin (2016) for more details about XGBoost.

The above T, v, γ and λ are known as XGBoost hyper-parameters ($\boldsymbol{\theta_h}$). Unlike the model parameters $\mathbf{\Theta}$ determined in the training process, the $\boldsymbol{\theta_h}$ should be specified before training and are somewhat analogous to the number of neural hidden layers in the neural network.

Besides them, another two hyper-parameters were also considered: (1) $d_{\max}$ = maximum depth of RT (i.e., number of edges with the farthest distance between the root split node and the leaf node, e.g., $d_{\max} = 3$ in Fig. 2) and (2) w_{mc} = minimum sum of instance weight needed for a further partition in RT. The problem that how to choose $\boldsymbol{\theta_h}$ from initial ranges will be solved in the next section.

Subset Simulation and *K*-fold Cross Validation for Hyper-parameter Tuning

K-fold Cross Validation

The search of the hyper-parameters should not be performed on the training data (that is seen by the training algorithm) because the model generalization performance on the unseen data is truly concerned. In this study, the *K*-fold cross validation (CV) is employed for such a purpose, in which the training set is divided into *K* folds randomly such that the training of model is performed on the (*K* - 1) folds using the specified hyper-parameters and then the left one fold is used for validation based on the trained model. In other words, each round of validation can be somewhat considered as the evaluation of the model generalization capability on unseen data. After repeating this process *K* times, the average value of performance measures from the *K* rounds is taken as the overall measure to evaluate the model performance with the corresponding hyper-parameters. Based on the literature (e.g., Hastie et al. 2009) and the consideration of our data size, *K* is chosen as 5 in this study.

A common error index in ML namely the root mean square error (RMSE) (e.g., Alavi and Gandomi 2011; Jones et al. 2018) is taken as the performance measure during the 5-fold CV and is expressed as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i^{(T)}\right)^2} \tag{5}$$

It is evident that RMSE should be minimized to obtain the appropriate hyper-parameters of XGBoost. The process of 5-fold CV considering the RMSE is illustrated in Fig. 4. Based on the RMSE value, the early stopping strategy (e.g., Goodfellow et al. 2016) is adopted to determine the optimal T; that is, T will be added until the RMSE for the j-th step is larger than that for the (j-5)-th step (j = 1, 2, …, T). The upper limit of T is set as 2000, which is an appropriate value as the contribution of adding RT becomes very small when $j > 2000$. Since there is a trade-off between choices of T and other hyper-parameters (e.g., a smaller v or $d_{\max}$ resulting in a larger T), this adaptive strategy is useful in tuning hyper-parameters of the gradient boosted model (Hastie et al. 2009).

Subset Simulation for Hyper-parameter Tuning

Traditionally, the hyper-parameter search in ML is achieved manually through the rules-of-thumb (e.g., Hinton 2012) or the grid search (e.g., Pedregos et al. 2011), which are relatively time-consuming and impractical especially when the number of hyper-parameters is large. An automatic hyper-parameter search method was proposed in this paper based on the subset simulation (SS) (Au and Beck 2001), which has been widely used in geotechnical reliability analysis (e.g., Li et al. 2016; Wang et al. 2020) and recently modified to solve stochastic optimization problems (Li and Au 2010; Li 2011). In the context of SS, the search process is achieved through finding the minimum RMSE and the associated hyper-parameter set of XGBoost. A brief introduction of the implementation procedure of SS is given below in an order of the simulation level (l = 1, 2, …m), where m = number of total levels in SS.

In the first simulation level (l = 1), N_l random samples sets of $\boldsymbol{\theta_h}$ are generated through the artificial distributions, which are specified as truncated normal distributions (Li and Au 2010). Obviously, a large N_l will increase the computational cost while a small N_l may not produce an acceptable result as the search space is not covered well. Based our test, N_l = 200 can achieve such a trade-off and is adopted in this study. After configuring each set of $\boldsymbol{\theta_h}$ for the XGBoost model, 200 RMSE values are computed through Eq. (5) using the 5-fold CV. Then, these RMSE values are ranked in an ascending order and the ($p_l N_l$+1)-th (i.e., 101-th) RMSE is regarded as $rmse_l$ (i.e., $rmse_1$ in this level), where p_l = conditional probability. A descending strategy that set p_1 = 0.5 in the first level and reduce it to 0.2 if the largest estimate of the standard deviation of $\boldsymbol{\theta_h}$ sample is less than 0.1, and further reduce p_l to 0.1 when the largest estimate of the standard deviation of $\boldsymbol{\theta_h}$ sample is less than 0.01, is adopted to achieve a better convergence (Li 2011).

The 100 sets of "good" $\boldsymbol{\theta_h}$ corresponding to RMSE < $rmse_1$ are taken as "seeds" to generate the left 100 sets of $\boldsymbol{\theta_h}$ using the Markov Chain Monte Carlo Simulation (MCMCS) (e.g., Xiao et al. 2016; Wang et al. 2020). During MCMCS, a candidate sample (in independent standard normal space) for the next state in the Markov Chain is first generated from a proposal probability density function defined through the current Markov Chain state and this candidate will be accepted or rejected based on the acceptance ratio (Li et al. 2016). Therefore, 200 sets of $\boldsymbol{\theta_h}$ (100 seeds and 100 new samples) are obtained again and thereby entering the next level of SS. Similarly, configuring the 100 sets of new $\boldsymbol{\theta_h}$ for XGBoost can produce 100 new RMSE values and a total of 200 RMSE values (100 seeds and 100 new RMSE values) are computed, followed by ranking them and identifying $rmse_l$ again. The

above procedure will be repeated until $\left|rmse_l - rmse_{l-1}\right| \le 1\times10^{-5}$. Note that the number of seed samples will be reduced at the subsequent levels because of the descending strategy for p_l. As the level of SS increases, the 200 sets of $\boldsymbol{\theta_h}$ will gradually gather from a broad space to a narrower region.

SS-XGBoost Framework for Developing Newmark Displacement Prediction Model

Implementation Procedure

As is shown in Fig. 5, the proposed SS-XGBoost framework can be systematically divided into four sub-parts: (1) dataset preparation, (2) subset simulation and *K*-fold cross validation for tuning hyper-parameters ($\boldsymbol{\theta_h}$), (3) final training and testing of XGBoost model, and (4) model prediction for estimating median value and standard deviation of Newmark displacement. The proposed framework is also applicable to other ML algorithms such as neural network and support vector machine. Such steps can readily be programmed as user-friendly functions using Python or R languages, whose corresponding XGBoost packages are freely accessible. These organized functions can be directly used and geotechnical practitioners only need to import the region-specific dataset if they want to develop a new model. Then, the trained Newmark displacement model serves as output to be saved for practical use. Like many other engineering problems, there is a trade-off between result accuracy (model performance) and operation convenience (model complexity). Practitioners can make a choice between traditional models and machine learning models according to the specific conditions. More easily, they can also use the developed models in this paper.

Hyper-parameter Tuning and Model Development

Five relatively important hyper-parameters of XGBoost (v, $d_{\max}$, w_{mc}, λ and γ) were tuned using SS, while T was determined adaptively through the mentioned early stopping strategy. Other hyper-parameters (e,g, α) were taken as the default values, which can be found in the XGBoost documentation (https://xgboost.readthedocs.io/en/release_0.80/parameter.html). To improve the tuning effectiveness and efficiency, the search ranges of hyper-parameters were specified based on our preliminary trials, such that some manifestly unreasonable values can be bypassed. Note that v can directly reduce the contribution of each RT and a smaller v will produce a more conservative training and require a larger T (more RTs). A relatively small v ($0.05 < v < 0.1$) is adopted in this paper because the generalization capability on new data may be better when $v < 0.1$ (Friedman 2001). All of the search ranges are summarized in Table 1.

The mean and minimum RMSE values in each SS level corresponding to the three vector-IM models versus the level are illustrated in Fig. 6. It can be seen that different RMSE values in the 5-fold cross validation (CV) decrease as the level of SS increases and the SS stops automatically when its stopping criteria is reached. The reason for the gradually slower decline of curves is that the sample sets at the subsequent simulation levels become repeated, illustrating the convergence of SS for the hyper-parameter tuning. Note that the relatively small decrease of RMSE values can be attributed to the strong prediction capability of XGBoost and the careful setting of hyper-parameter tuning ranges. Among them, the (PGA, PGV, I_a) model produces the smallest RMSE value while the (PGA, I_a) model has the largest RMSE value, which is consistent with the result of Saygili and Rathje (2008) using the same

IM combinations. The smallest minimum RMSE values at the final level for the (PGA, PGV), (PGA, I_a) and (PGA, PGV, I_a) combinations are 0.576, 0.610 and 0.503, respectively. The corresponding hyper-parameter tuning results for the three models are also listed in Table 1. It is seen that the hyper-parameters for the combinations of (PGA, PGV) and (PGA, I_a) are relatively similar. Although a smaller shrink factor v is determined for the (PGA, PGV, I_a) model, a smaller T is still needed because one more IM is considered. In addition, a larger λ is needed in 3-IMs model to penalize the leaf scores of RTs.

The RMSE values corresponding to all rounds in 5-fold CV (refer to Fig. 3) and their average value against the number of RTs using the optimal hyper-parameters are presented in Fig. 7, where a break point is set for the region from 1 to 3 on the Y axis. The RMSE values decrease steeply at first 100 boosting steps, and then the improvement of validation performance gradually becomes insignificant as the residuals of prediction in following boosting steps become smaller. Note that the validation errors decrease continuously and the general rule for overfitting behavior that the validation performance starts to become worse (e.g., Hastie et al. 2009; Goodfellow et al. 2016) is not observed. Additionally, the RMSE values in individual rounds of CV are close to their average value, indicating that the determined optimal hyper-parameters can produce low variance among the 5 rounds and control the model generalization error effectively. Finally, these hyper-parameters were configured for the corresponding XGBoost models, followed by the ultimate model training.

Results and Comparisons

This section will sequentially illustrate the sufficiency, overall performance, model

uncertainty, median prediction and interpretation of the developed XGBoost models, where the traditional models with functional forms will also be involved for comparison.

Traditional Newmark Displacement Prediction Model

Because of its popularity in predicting Newmark displacements and the same considered predictor variables, The Saygili and Rathje (2008) models (referred to as SR08 models) are selected as representatives of traditional models. The respective functional forms of the SR08 (PGA, PGV), (PGA, Ia) and (PGA, PGV, Ia) models are expressed as follows:

$$\ln D = a_1 + a_2\left(\frac{k_y}{\mathrm{PGA}}\right) + a_3\left(\frac{k_y}{\mathrm{PGA}}\right)^2 + a_4\left(\frac{k_y}{\mathrm{PGA}}\right)^3 + a_5\left(\frac{k_y}{\mathrm{PGA}}\right)^4 + a_6\ln(\mathrm{PGA}) + a_7\ln(\mathrm{PGV}) + \varepsilon\sigma_{\ln D} \tag{6a}$$

$$\ln D = a_1 + a_2\left(\frac{k_y}{\mathrm{PGA}}\right) + a_3\left(\frac{k_y}{\mathrm{PGA}}\right)^2 + a_4\left(\frac{k_y}{\mathrm{PGA}}\right)^3 + a_5\left(\frac{k_y}{\mathrm{PGA}}\right)^4 + a_6\ln(\mathrm{PGA}) + a_7\ln(I_a) + \varepsilon\sigma_{\ln D} \tag{6b}$$

$$\ln D = a_1 + a_2\left(\frac{k_y}{\mathrm{PGA}}\right) + a_3\left(\frac{k_y}{\mathrm{PGA}}\right)^2 + a_4\left(\frac{k_y}{\mathrm{PGA}}\right)^3 + a_5\left(\frac{k_y}{\mathrm{PGA}}\right)^4 + a_6\ln(\mathrm{PGA}) + a_7\ln(\mathrm{PGV}) + a_8\ln(I_a) + \varepsilon\sigma_{\ln D} \tag{6c}$$

where D = Newmark displacement (cm); PGA, PGV, I_a and k_y are in units of g, cm/s, m/s and g, respectively; $\boldsymbol{a}$ = [a_1, a_2, …, a_8] = regression coefficients of functional form, which are summarized in Table 2; ε = standard normal variable; $\sigma_{\ln D}$ = standard deviation in natural logarithm units, which is usually used to quantify the model uncertainty and a smaller $\sigma_{\ln D}$ corresponds to a more efficient model (Saygili and Rathje 2008).

For a fair comparison, we also used our dataset and R software to obtain new regression coefficients of the three models, which are referred to as the update SR08 models throughout

this paper. Their model parameters are also listed in Table 2. The following sub-sections will illustrate that the results for SR08 models and updated SR08 models are generally consistent. Both of the SR08 and updated SR08 models will be considered to compare with the XGBoost models. However, the updated SR08 models are involved in most of subsequent comparisons considering a desirable purpose of comparing the model performances given the same dataset.

Comparison of Model Sufficiency

It is important to demonstrate that the developed XGBoost models can satisfy sufficiency criterion (e.g., Rathje and Saygili 2009), namely the model can sufficiently predict Newmark displacements without the need for specifying earthquake magnitude (e.g., M_w) and source-to-site distance (e.g., R_{rup}). The residuals ($\ln D_{\mathrm{obs}}$ - $\ln D_{\mathrm{pred}}$) against M_w and R_{rup} for the updated SR08 models and the XGBoost models are shown in Fig. 8 and 9, respectively, where D_{obs} and D_{pred} = observed and predicted Newmark displacements (cm), respectively. Note that the means and error bars of residuals for the XGBoost models are calculated based on all of data. In Fig. 9(c), the positive residuals at R_{rup} < 5 km for the updated SR08 (PGA, Ia) model shows that the polynomial model is biased and may underestimate the Newmark displacement near the source. Such a biased trend was also observed in the original SR08 (PGA, I_a) model although not shown here for brevity. By contrast, XGBoost models generally have an unbiased mean of residuals with much reduced scattering against M_w and R_{rup}, indicating that the developed models can better satisfy the sufficiency criterion. Note that similar patterns appear on training and testing sets, indicating good generalization capabilities of developed models.

Comparison of Overall Model Performance

Some regression metrics will be introduced in this sub-section to illustrate the overall performance of developed models. Fig. 10 plots D_{pred} versus D_{obs} for updated SR08 models and XGBoost models against the 1:1 line. The coefficient of determination (R^2) is used to reflect the model efficiency (e.g., Jibson 2007; Wang 2012). Obviously, data distribution in the XGBoost models is closer around the ideal fitting line than that in the SR08 forms, which is illustrated quantitatively by the overall larger R^2 for the XGBoost models. Note that XGBoost models have higher R^2 than any one of the updated SR08 models. In each model, R^2 increases following the same order of (PGA, I_a), (PGA, PGV) and (PGA, PGV, I_a) combination. In addition, R^2 is similar for both training and testing data. These results clearly demonstrate that the developed data-driven models have both good fitting and generalization performances.

To further evaluate the model generalization capability, external validation for the XGBoost models is performed using the optimal hyper-parameters. Three regression metrics are presented in Table 3, including Pearson correlation coefficient (R), RMSE and mean absolute error (MAE) for both the training and testing datasets in XGBoost models. Generally, the predicted values are thought to have a strong correlation with the observed ones if $R \geq 0.8$ and the error metrics (e.g., RMSE and MAE) are minimized (e.g., Alavi and Gandomi 2011; Khosravikia et al. 2018). As a result, all of the developed XGBoost models have a high R and relatively low RMSE, MAE values for both the training and testing sets. In addition, external validation metrics suggested by Golbraikh and Tropsha (2002) are adopted for overall performance verification. As is seen in Table 3, the developed models satisfy all the required

criteria. It is interesting to note that the RMSE values associated with the testing set for the three models are almost equal to the respective RMSE values in the hyper-parameter tuning (refer to Figs. 6 and 7), which shows the model generalization capability again. From the overall perspective, these validation results reflect the efficiency of the developed models in predicting the Newmark displacement. The reason for the general better training performance than the testing performance is that the residual data has significant scattering on the logarithmic scale if displacement is small (e.g., displacements smaller than 0.01 cm). Although not shown here for brevity, the training and testing performances will become very similar if these small displacements of little engineering importance (e.g., Du and Wang 2016; Du et al. 2018a) are excluded.

Table 4 presents the R^2, MAE and RMSE values for the SR08 models, updated SR08 models and XGBoost models on testing set after excluding small displacement data ($D_{\text{pred}} <$ 0.01 cm). It is seen that XGBoost models can produce stronger correlations and smaller errors than other models on this unseen dataset. Particularly, the 2-IM XGBoost (PGA, I_a) model can achieve even better performance than both 3-IM SR08 and updated SR08 (PGA, PGV, Ia) models.

Comparison of Model Uncertainty

Smaller model uncertainty is usually reflected by lower standard deviation $\sigma_{\ln D}$ and indicates the Newmark model can better satisfy the efficiency criterion. Fig. 11 plots $\sigma_{\ln D}$ versus D_{pred} on the training and testing sets. Three observations can be made. First, $\sigma_{\ln D}$ decreases with an increase of D_{pred} for all of IM combinations. Second, considering an additional IM (i.e., 3-IMs model) can produce a smaller $\sigma_{\ln D}$. This is not unexpected because more information

about the ground motion can be complemented by more IMs. Third, the $\sigma_{\ln D}$ between training and testing sets are generally similar at displacement level of engineering interest (e.g., $D_{\text{pred}} > 1$ cm) (Bray and Travasarou 2007). Therefore, the generalization capabilities of the developed models on the future data (testing set) are verified again. The $\sigma_{\ln D}$ is usually derived from all of data when developing polynomial-based Newmark displacement models, while this way is not preferred by machine learning methods. In this study, the $\sigma_{\ln D}$ on the testing set is recommended to quantify the model uncertainty. The reason will be illustrated in the following with the aid of *K*-fold cross validation (CV).

Within the 5-fold CV scheme, $\sigma_{\ln D}$ in individual validation rounds with the optimal hyper-parameters are plotted in Fig. 12. Also, $\sigma_{\ln D}$ on the testing set shown in Fig. 11, is reproduced in Fig. 12 for comparison. Note that the validation data in individual validation rounds is unseen by the model trained in that round and different validation rounds can be regarded as "bind tests" for quantifying model uncertainty to some extent (refer to Fig. 3), while the testing set is truly unseen throughout the model development. It is observed that the $\sigma_{\ln D}$ on the testing set has similar trend to $\sigma_{\ln D}$ in different CV rounds, indicating that the $\sigma_{\ln D}$ on the testing set can be used to quantify the model uncertainty well.

To derive the $\sigma_{\ln D}$ in probabilistic calculations conveniently, the $\sigma_{\ln D}$ versus D_{pred} relationships for three models shown in Fig. 12 are fitted by trilinear functions:

$$\sigma_{\ln D}^{\text{PGA, PGV}} = \begin{cases} 0.816 & \textit{if} \quad D_{\text{pred}} \le 0.0015\ cm \\ -0.103\log_{10}\left(D_{\text{pred}}\right) + 0.525 & \textit{if} \quad 0.0015\ cm < D_{\text{pred}} \le 300\ cm \\ 0.270 & \textit{if} \quad D_{\text{pred}} > 300\ cm \end{cases} \tag{7}$$

$$\sigma_{\ln D}^{\mathrm{PGA},\, I_a} = \begin{cases} 0.881 & if \quad D_{\mathrm{pred}} \le 0.007\ cm \\ -0.151\log_{10}\left(D_{\mathrm{pred}}\right) + 0.556 & if \quad 0.007\ cm < D_{\mathrm{pred}} \le 30\ cm \\ 0.333 & if \quad D_{\mathrm{pred}} > 30\ cm \end{cases} \tag{8}$$

$$\sigma_{\ln D}^{\mathrm{PGA,\, PGV},\, I_a} = \begin{cases} 0.798 & if \quad D_{\mathrm{pred}} \le 0.003\ cm \\ -0.147\log_{10}\left(D_{\mathrm{pred}}\right) + 0.427 & if \quad 0.003\ cm < D_{\mathrm{pred}} \le 20\ cm \\ 0.235 & if \quad D_{\mathrm{pred}} > 20\ cm \end{cases} \tag{9}$$

To make a comparison of $\sigma_{\ln D}$ between the developed models and traditional models, Fig. 13 shows the $\sigma_{\ln D}$ versus D_{pred} on the testing set for the SR08 models, updated SR08 models and XGBoost models. Several observations can be made. First, similar trends are observed for different models and the performance rankings of different IM combinations are the same. Second, the $\sigma_{\ln D}$ for SR08and updated SR08 models are very similar although the former ones use a much larger dataset in regression, implying the model capacity of traditional methods; that is using more data is not helpful to better satisfy the efficiency criteria. Third, all XGBoost models produce considerably smaller $\sigma_{\ln D}$ than 2-IMs SR08 model (and updated SR08 models) and 2-IMs XGBoost models may even achieve better performance than the 3-IMs SR08 (PGA, PGV, I_a) model. Based on the results of updated SR08 and XGBoost models, the relative percentage of $\sigma_{\ln D}$ reduction at several D_{pred} levels are listed in Table 5. Generally, the percentage reduction of $\sigma_{\ln D}$ increases with D_{pred}. Among three IM combinations, the model uncertainty of (PGA, I_a) reduces most significantly using the XGBoost model. The results indicate that the advantage of the XGBoost models will be more significant with an increase of displacement. Note again that updated SR08 model are regressed through the whole dataset including this comparison set (i.e., testing set) while this set is always unseen by XGBoost models.

Comparison of Median Prediction

For a deterministic earthquake scenario: that M_w = 7, R_{rup} = 5 km and 30-m shear wave velocity V_{s30} = 760 m/s, the corresponding values of PGA, PGV and I_a to predict the displacement are 0.33 g, 30 cm/s and 1.07 m/s, respectively (Saygili and Rathje 2008). Fig. 14 shows the median predicted displacements for XGBoost models with respect to various k_y. Besides SR08 and updated SR08 models, the original BT07 (M_w, PGA) model (Bray and Travasarou 2007) and J07 (PGA, I_a) model (Jibson 2007) are also presented for comparison. As shown, the curves associated with different prediction models are generally comparable. Specifically, the median predictions for the SR08 and updated SR08 models are also similar. The large difference in J07 model is caused by its low order of polynomial and the limited data for the regression. Additionally, BT07 model produces larger displacements than SR08 models and XGBoost models, which is in accordance with the finding by Saygili and Rathje (2008) using the same IM combinations.

Moreover, model prediction for varying earthquake scenarios is also investigated. For the three IMs involved in this study, the ground-motion prediction equations (GMPEs) proposed by Campbell and Bozorgnia (2012, 2014) are adopted to estimate PGA, PGV, and I_a, respectively. The median predicted displacements for a strike-slip fault with magnitudes M_w = 5.5 and 7.5, V_{s30} = 400 m/s (soil site) and V_{s30} = 760 m/s (rock site), k_y = 0.1 g are plotted in Fig. 15 against R_{rup}. Similarly, the developed XGBoost models can generally predict comparable median displacements with other models while the curves corresponding to the combination of (PGA, I_a) are relatively unsmooth. This may be attributed to large correlation between PGA and I_a, so information conveyed by these two IMs are not complementary.

However, these curves are still situated within the clusters of others. Similar unsmoothness of the median prediction curves for data-driven methods can also be found in the literature (e.g., Alavi and Gandomi 2011).

Interpretation of Models

Median predicted displacements versus IMs are shown in Fig. 16 using contour maps, where ky is fixed as 0.1 g and only 2-IMs models are considered herein. It can be seen that the displacement contour maps for the updated corresponding SR08 models and XGBoost models are generally comparable. Since PGA and IA are two intensity measures that are highly correlated, data distribution in the predictor space is narrowly focused, as shown in Fig. 16(d). The contour lines of XGBoost (PGA, Ia) is locally unsmooth and can be influenced by individual data. That explains why displacement prediction by XGBoost (PGA, Ia) may even increase with increasing Rrup, as shown in Fig. 15. On the contrary, the data points for the XGBoost (PGA, PGV) model are well-distributed in the whole predictor space, so its contour seems relatively smooth. Therefore, Newmark displacements predicted by data-driven method are greatly influenced by the range and distribution of ground motion data.

Furthermore, the relative importance of each predictor variable in predicting the Newmark displacement is identified based on the developed model. As discussed early, the optimal split nodes of variables should be found to partition the predictor space at each boosting step. Therefore, the number of times to split the data for a predictor variable in all regression trees is taken as the feature importance score. Table 6 lists the feature importance scores for different models. It is seen that PGA has the highest score in both the 2-IMs and 3-IMs models, which is not unexpected because the importance of PGA is reflected in both

triggering and accumulating sliding displacement. This can explain why PGA is the primary IM to predict the Newmark displacement in the literature (e.g., Bray and Travasarou 2007; Jibson 2007; Rathje and Saygili 2009). In addition, PGV is more important than I_a in the 3-IMs model, which is consistent with the result that the larger correlation between PGA and I_a leads to a larger standard deviation for (PGA, I_a) model than that for (PGA, PGV) model, because PGV can supplement more intermediate frequency content information for PGA. The lowest score for k_y is because it is a nominal variable and needs less split in regression trees. The importance score may be helpful for selecting predictor variables to develop sufficient and efficient predictive models in engineering problems, especially when the number of candidate variables is large.

Application to Probabilistic Seismic Slope Displacement Hazard Analysis

The application of the developed XGBoost prediction models to the probabilistic seismic slope displacement hazard analysis (PSSDHA) (Rathje and Saygili 2008; Wang and Rathje 2018) is illustrated based on three deterministic values of slope yield acceleration, which are assumed as 0.05, 0.1 and 0.15 g, respectively. The inherent variability of soil properties (e.g., Qi and Li 2018; Xiao et al. 2018) is not considered because we focus on comparing the model uncertainty herein. The stiff soil site (V_{s30} = 400 m/s) with a R_{rup} of 5 km from a point source is considered and the following Gutenberg–Richter (G-R) recurrence law is used to describe the seismicity of the source:

$$\log_{10} \lambda_m = 4.2 - 1.0 M_w \tag{10}$$

where λ_m = mean annual rate of exceedance of M_w. A truncated G-R distribution with a minimum M_w of 4.2 and a maximum M_w of 8.0 is adopted, in which the magnitude bin is

equal to 0.2. Considering the limited length of paper, only the recommended combination (PGA, PGV) by both this study and Saygili and Rathje (2008) is considered herein for a demonstration purpose. The $\sigma_{\ln D}$ of the updated SR08 (PGA, PGV) model is derived by Fig. 13, although the original $\sigma_{\ln D}$ from Saygili and Rathje (2008) can produce similar hazard curves based on our test. Again, the previous GMPE (Campbell and Bozorgnia 2014) is used to determine the median and the standard deviation of PGA and PGV as well as the correlation between them.

Fig. 17 plots the displacement hazard curves for the XGBoost model and the updated SR08 model for the three slope cases. It is observed that the hazard curves for the updated SR08 model and the XGBoost model are consistent at the small-to-median displacement level while the former model will result in a larger hazard at the large displacement level, although the comparable median predictions are produced as discussed previously. Furthermore, three specific hazard levels (10%, 5% and 1% probability of exceedance in 50 years) are chosen to compare the corresponding Newmark displacements for the two models explicitly, as listed in Table 7. Because of the reduction of model uncertainty, the XGBoost model can reduce displacement hazard by 23-36% in the three cases, indicating that a large uncertainty accompanied with the Newmark displacement prediction model will result in an overestimation of the landslide hazard and lead to a conservative engineering design.

Summary and Conclusions

This study proposes a SS-XGBoost framework to develop data-driven models for predicting the Newmark displacement. The framework proposes a subset simulation (SS) and *K*-fold

cross validation (CV) procedure, which is efficient for tuning hyper-parameters of XGBoost model. Three data-driven Newmark displacement models are developed using different vector IMs, namely, the XGBoost (PGA, PGV), XGBoost (PGA, I_a) and XGBoost (PGA, PGV, I_a) models. The developed models have excellent generalization capability, and do not require predefined functional forms.

Residual analyses clearly reveal that the developed XGBoost models can better satisfy sufficiency and efficiency criteria, when compared with the SR08 models using same IMs and dataset. Generally, the XGBoost models have reduced standard deviations ($\sigma_{\ln D}$) by 20%-50% compared with SR08 models, and the reduction becomes most significant at large displacement levels. Based on the 5-fold CV, the standard deviation on the testing dataset is recommended to describe the model uncertainty. In the end, three trilinear equations are proposed to quantify the model uncertainty for practical use.

Generally, the developed data-driven models can produce the median predicted displacements comparable with existing empirical models. Yet, the developed model is more flexible in capturing high nonlinearity embedded in the dataset. Attention should be paid to the number and range of training data, which has significant influence on the generalization of the data-driven models. In addition, it is indicated that PGA is the most important IM and the XGBoost (PGA, PGV) and XGBoost (PGA, PGV, I_a) models are recommended for use for their better generalization capability and robustness. Furthermore, probabilistic seismic slope displacement hazard analysis (PSSDHA) is conducted using the developed models. Compared with empirical models, it is found that the data-drive models result in smaller displacement hazards because of their reduced model uncertainty.

The developed Newmark displacement models utilize the most updated NGA-West2 database, which can be regarded as alternatives to existing empirical models (e.g., Jibson 2007; Saygili and Rathje 2008; Du and Wang 2016). For practical use, the developed models are provided in executable files at http://gwang.people.ust.hk/XGB-Newmark.html. Geotechnical practitioners only need to import predictor variables (i.e., IMs and k_y) in a spreadsheet and then the predicted Newmark displacements and associated standard deviations can be obtained. The developed model can also be combined with spatial cross-correlation models of PGA, PGV and I_a (Wang and Du 2013) for risk analysis of spatially distributed slopes (e.g., Du and Wang 2014). In addition, the proposed framework can also be applied to solve other data-driven problems such as ground-motion prediction and liquefaction assessment.

Data Availability Statement

The developed executable file for XGBoost Newmark displacement models is available at http://gwang.people.ust.hk/XGB-Newmark.html

Acknowledgements

The authors acknowledge support from Hong Kong Research Grants Council (Grant No. 16214118), the National Natural Science Foundation of China (Grant No. 51579190), and Joint Research Fund for Overseas Chinese Scholars and Scholars in Hong Kong and Macao (Grant No. 51828902) from National Natural Science Foundation of China. The first author wishes to thank the Department of Civil and Environmental Engineering, Hong Kong

University of Science and Technology, for hosting his visit as an exchange Ph.D. student.

Notation

The following symbols are used in this paper:

$\boldsymbol{a}$ = regression coefficients of functional form;

D_{obs} = observed Newmark displacement;

D_{pred} = predicted Newmark displacement;

$d_{\max}$ = maximum depth of regression tree;

$f_j()$ = output of the j-th regression tree without shrinkage;

I_a = Arias intensity;

IM = ground motion intensity measure;

k_y = slope yield acceleration;

$L()$ = square loss function;

M_j = number of leaf nodes in the j-th regression tree;

M_w = moment magnitude;

MAE = mean absolute error;

m = number of total levels in subset simulation;

N = number of considered samples;

N_l = number of samples generated in each level of subset simulation;

PGA = peak ground acceleration;

PGV = peak ground velocity;

p_l = conditional probability in subset simulation;

R = Pearson correlation coefficient;

R^2 = coefficient of determination;

R_{rup} = rupture distance;

RMSE = root mean square error;

$rmse_l$ = specific threshold of root mean square error in the l-th level of subset simulation;

T = number of regression trees for boosting;

V_{s30} = average shear wave velocity in the upper 30 m;

$w_k^{(j)}$ = leaf score of the k-th leaf node in the j-th regression tree;

w_{mc} = minimum sum of instance weight needed for a further node partition in regression tree;

$\boldsymbol{x}_i$ = i-th sample of predictor variables;

y_i = i-th sample of target variable;

$\hat{y}^{(j)}$ = prediction of target variable using first j regression trees;

γ = minimum loss reduction needed for a further node partition in regression tree;

ε = standard normal variable;

λ = L2 regularization term on leaf scores;

λ_m = mean annual rate of exceedance of moment magnitude;

$\boldsymbol{\Theta}_j$ = structure of the j-th regression tree;

$\boldsymbol{\theta_h}$= model hyper-parameters;

v = shrinkage factor;

$\sigma_{\ln D}$ = standard deviation in natural logarithm units; and

$\Omega()$ = regularization term on regression tree.

References

Alimoradi, A., and J. L. Beck. 2014. "Machine-learning methods for earthquake ground motion analysis and simulation." *J. Eng. Mech.*, 141(4), 04014147. https://doi.org/10.1061/(ASCE)EM.1943-7889.0000869.

Au, S. K., and J. L. Beck. 2001. "Estimation of small failure probabilities in high dimensions by subset simulation." *Probab. Eng. Mech.*, 16(4), 263-277. https://doi.org/10.1016/S0266-8920(01)00019-4

Alavi, A. H., and A. H. Gandomi. 2011. "Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing." *Comput. Struct.*, 89(23-24), 2176-2194. https://doi.org/10.1016/j.compstruc.2011.08.019

Ancheta, T. D., et al. 2014. "NGA-West2 database." *Earthquake Spectra*, 30(3), 989-1005. https://doi.org/10.1193/070913EQS197M.

Bray, J. D., and T. Travasarou. 2007. "Simplified procedure for estimating earthquake-induced deviatoric slope displacements." *J. Geotech. Geoenviron. Eng.*, 133(4), 381-392, https://doi.org/10.1061/(ASCE)1090-0241(2007)133:4(381)

Campbell, K. W., and Y. Bozorgnia. 2012. "A comparison of ground motion prediction equations for Arias intensity and cumulative absolute velocity developed using a consistent database and functional form." *Earthquake Spectra*, 28(3), 931-941. https://doi.org/10.1193/1.4000067

Campbell, K. W., and Y. Bozorgnia. 2014. "NGA-West2 ground motion model for the average horizontal components of PGA, PGV, and 5% damped linear acceleration response spectra." *Earthquake Spectra*, 30(3), 1087-1115. https://doi.org/10.1193/

062913EQS175M.

Chen, T., and C. Guestrin. 2016. "XGBoost: A scalable tree boosting system." *Proc*., *22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 785–794. New York: ACM.

Derras, B., P. Y. Bard, F. Cotton, and A. Bekkouche. 2012. "Adapting the neural network approach to PGA prediction: An example based on the KiK-net data." *Bull. Seismol. Soc. Am*., 102(4), 1446-1461. https://doi.org/10.1785/0120110088

Du, W., and G. Wang. 2014. "Fully probabilistic seismic displacement analysis of spatially distributed slopes using spatially correlated vector intensity measures." *Earthquake Eng. Struct. Dyn*., 43(5), 661-679. https://doi.org/10.1002/eqe.2365

Du, W., and G. Wang. 2016. "A one-step Newmark displacement model for probabilistic seismic slope displacement hazard analysis." *Eng. Geol*., 205, 12-23. https://doi.org/10.1016/j.enggeo.2016.02.011

Du, W., D. Huang, and G. Wang. 2018a. "Quantification of model uncertainty and variability in Newmark displacement analysis." *Soil. Dyn. Earthquake Eng*., 109, 286-298. https://doi.org/10.1016/j.soildyn.2018.02.037

Du, W., G. Wang, and D. Huang. 2018b. "Evaluation of seismic slope displacements based on fully coupled sliding mass analysis and NGA-West2 database." *J. Geotech. Geoenviron. Eng*., 144(8), 06018006. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001923

Golbraikh, A., and A. Tropsha. 2002. "Beware of q2!." *J. Mol. Graph. Model*. 20(4), 269–276.

Friedman, J. H. 2001. "Greedy function approximation: A gradient boosting machine." *Ann.*

Stat., 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Goodfellow, I., Y. Bengio, and A. Courville. 2016. Machine learning basics., *Deep learning*, pp. 98-164, MIT Press, Cambridge, MA.

Hinton, G. E. 2012. Neural networks: Tricks of the trade., *A practical guide to training restricted Boltzmann machines*, pp. 599-619, Springer, Berlin, Heidelberg.

Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The elements* of *statistical learning*, 2nd Ed., Springer, New York.

Jibson, R. W. 2007. "Regression models for estimating coseismic landslide displacement." *Eng*. *Geol*., 91(2-4), 209-218. https://doi.org/10.1016/j.enggeo.2007.01.013

Jibson, R. W. 2011. "Methods for assessing the stability of slopes during earthquakes—A retrospective." *Eng*. *Geol*., 122(1-2), 43-50. https://doi.org/10.1016/j.enggeo.2010.09.017

Jones, R. E., J. A. Templeton, C. M. Sanders, and J. T. Ostien. 2018. "Machine learning models of plastic flow based on representation theory." *Comput. Model. Eng. Sci*., 117(3), 309-342.

Juang, C. H., H. Yuan, D. H. Lee, and P. S. Lin. 2003. "Simplified cone penetration test-based method for evaluating liquefaction resistance of soils." *J. Geotech*. *Geoenviron*. *Eng*., 129(1), 66-80. https://doi.org/10.1061/(ASCE)1090-0241(2003)129:1(66)

Khosravikia, F., Y. Zeinali, Z. Nagy, P. Clayton, and E. Rathje. 2018. "Neural network-based equations for predicting PGA and PGV in Texas, Oklahoma, and Kansas." *Proc*., *5th Geotech*. *Earthq*. *Eng*. *Soil Dynam*. *Conference*, Austin, Texas.

Kong, Q., D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft. 2018.

"Machine learning in seismology: turning data into insights." *Seismol. Res. Lett.*, 90(1), 3-14. https://doi.org/10.1785/0220180259

Lee, J., and R. A. Green. 2015. "Empirical predictive relationship for seismic lateral displacement of slopes." *Géotechnique*, 65(5), 374-390. https://doi.org/10.1680/geot.SIP.15.P.011

Li, H. S., and S. K. Au. 2010. "Design optimization using subset simulation algorithm." *Struct. Saf.*, 32(6), 384-392. https://doi.org/10.1016/j.strusafe.2010.03.001

Li, H. S. 2011. "Subset simulation for unconstrained global optimization." *Appl. Math. Model.*, 35(10), 5108-5120. https://doi.org/10.1016/j.apm.2011.04.023

Li, D. Q., T. Xiao, Z. J. Cao, C. B. Zhou, and L. M. Zhang. 2016. "Enhancement of random finite element method in reliability analysis and risk assessment of soil slopes using Subset Simulation." *Landslides*, 13(2), 293-303. https://doi.org/10.1007/s10346-015-0569-2

Newmark, N. M. 1965. "Effects of earthquakes on dams and embankments." *Géotechnique*, 15(2), 139–160. http://dx.doi.org/10.1680/geot.1965.15.2.139

Pedregosa, F., et al. 2011. "Scikit-learn: Machine learning in Python." *J. Mach. Learn. Res.*, 12, 2825–2830.

Qi, X. H., and D. Q. Li. 2018. "Effect of spatial variability of shear strength parameters on critical slip surfaces of slopes." *Eng. Geol.*, *239*, 41-49. https://doi.org/10.1016/j.enggeo.2018.03.007

Rathje, E. M., and J. D. Bray 2000. "Nonlinear coupled seismic sliding analysis of earth structures." *J. Geotech. Geoenviron. Eng.*, 126(11), 1002-1014.

Rathje, E. M., and G. Saygili. 2008. "Probabilistic seismic hazard analysis for the sliding displacement of slopes: scalar and vector approaches." *J. Geotech. Geoenviron. Eng.*, 134(6), 804-814. https://doi.org/10.1061/(ASCE)1090-0241(2008)134:6(804)

Rathje, E. M., and G. Saygili. 2009. "Probabilistic assessment of earthquake-induced sliding displacements of natural slopes." *Bull. N. Z. Soc. Earthquake Eng.*, 42(1), 18–27.

Ren, S., G. Chen, T. Li, Q. Chen, and S. Li. 2018. "A deep learning-based computational algorithm for identifying damage load condition: an artificial intelligence inverse problem solution for failure analysis." *Comput. Model. Eng. Sci.*, 117(3), 287-307.

Saygili, G., and E. M. Rathje. 2008. "Empirical predictive models for earthquake-induced sliding displacements of slopes." *J. Geotech. Geoenviron. Eng.*, 134(6), 790-803. https://doi.org/10.1061/(ASCE)1090-0241(2008)134:6(790)

Shao, H., and X. Deng. 2018. "AdaBoosting neural network for short-term wind speed forecasting based on seasonal characteristics analysis and lag space estimation." *Comput. Model. Eng. Sci.*, 114(3), 277-293.

Song, J., and A. Rodriguez-Marek. 2015. "Sliding displacement of flexible earth slopes subject to near-fault ground motions." *J. Geotech. Geoenviron. Eng.*, 141(3): 04014110. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001233

Song, J., G. Y. Gao, A. Rodriguez-Marek, and E. M. Rathje. 2016. "Seismic assessment of the rigid sliding displacements caused by pulse motions." *Soil. Dyn. Earthquake Eng.*, 82, 1-10. https://doi.org/10.1016/j.soildyn.2015.11.014

Wang, G. 2012. "Efficiency of scalar and vector intensity measures for seismic slope displacements." *Front. Struct. Civ. Eng.*, 6(1), 44-52.

Wang, G., and W. Du. 2013. “Spatial cross−correlation models for vector intensity measures (PGA, I_a, PGV, and SAs) considering regional site conditions.” *Bull. Seismol. Soc. Am.*, 103(6), 3189-3204. https://doi.org/10.1785/0120130061

Wang, M. X., X. S. Tang, D. Q. Li, and X. H. Qi. 2020. “Subset simulation for efficient slope reliability analysis involving copula-based cross-correlated random fields.” *Comput. Geotech.*, 118, 103326. https://doi.org/10.1016/j.compgeo.2019.103326

Wang, Y., and E. M. Rathje. 2018. “Application of a probabilistic assessment of the permanent seismic displacement of a slope.” *J. Geotech. Geoenviron. Eng.*, 144(6), 04018034. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001886

Xiao, T., D. Q. Li, Z. J. Cao, S. K. Au, and K. K. Phoon. 2016. “Three-dimensional slope reliability and risk assessment using auxiliary random finite element method.” *Comput. Geotech.*, 79, 146-158. https://doi.org/10.1016/j.compgeo.2016.05.024

Xiao, T., D. Q. Li, Z. J. Cao, and L. M. Zhang. 2018. “CPT-based probabilistic characterization of three-dimensional spatial variability using MLE.” *J. Geotech. Geoenviron. Eng.*, 144(5), 04018023. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001875

Youssef, A. M., H. R. Pourghasemi, Z. S. Pourtaghi, and M. M. Al-Katheeri. 2016. “Landslide susceptibility mapping using random forest, boosted regression tree, classification and regression tree, and general linear models and comparison of their performance at Wadi Tayyah Basin, Asir Region, Saudi Arabia.” *Landslides*, 13(5), 839-856. https://doi.org/10.1007/s10346-015-0614-1

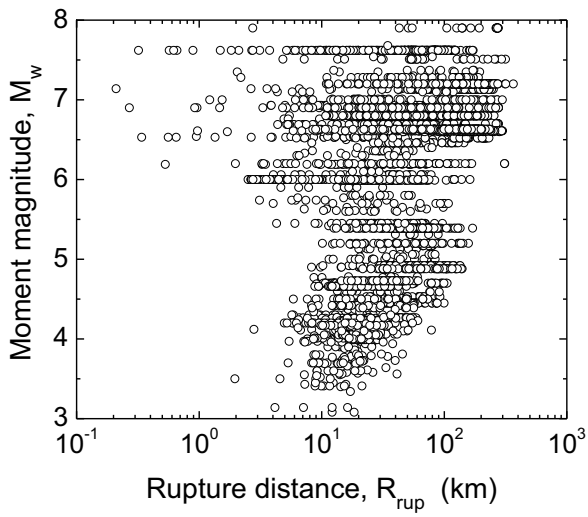


Fig. 1. Distribution of used earthquake recordings in terms of M_w and R_{rup}.

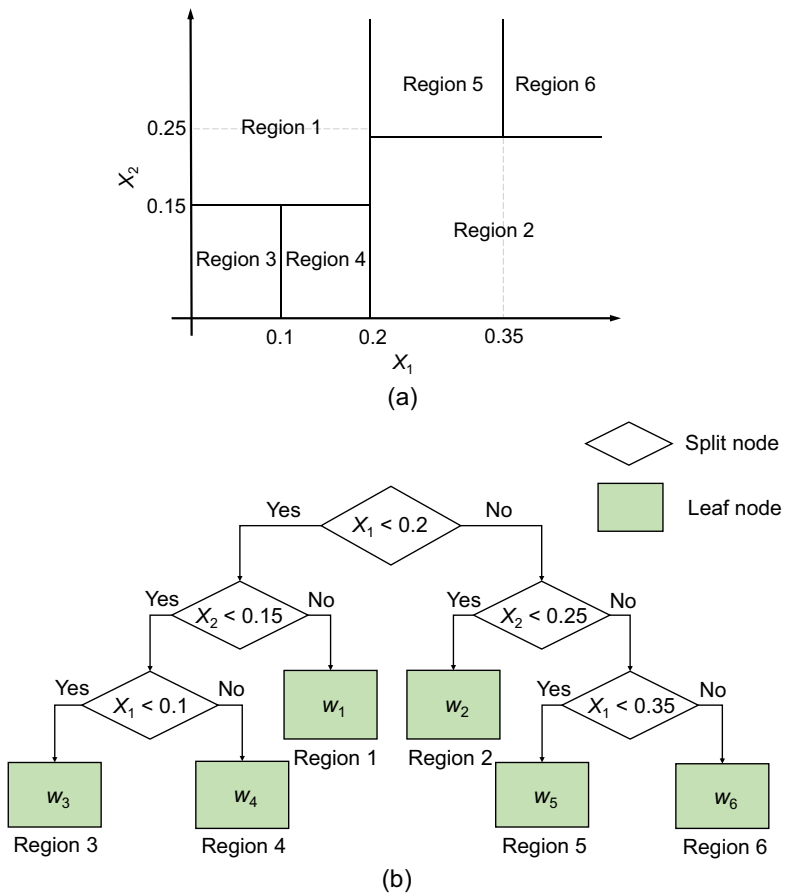


Fig. 2. Individual regression tree: (a) partition of a two-dimensional predictor space; and the associated (b) tree structure.

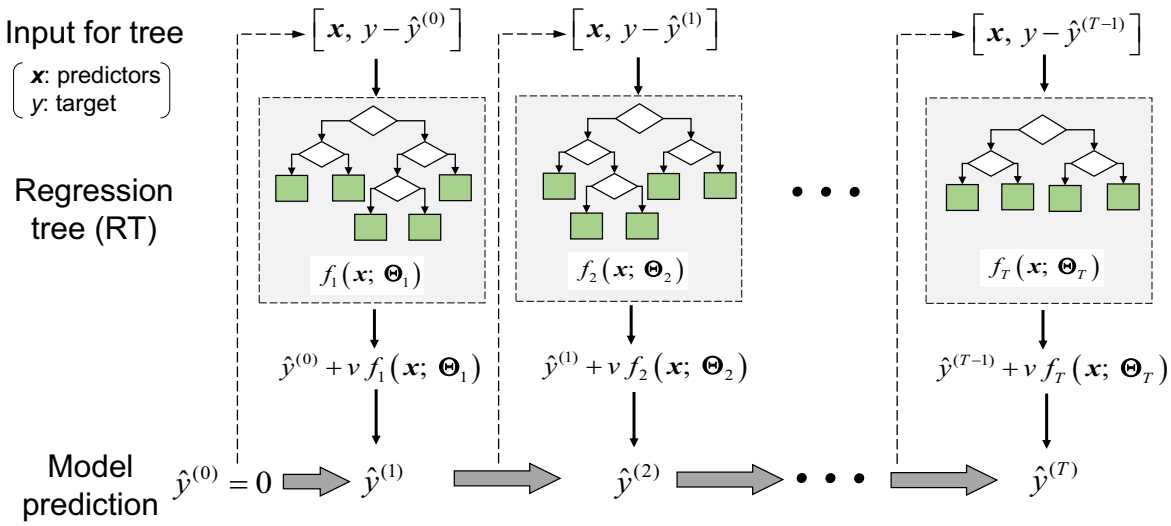


Fig. 3. Schematic diagram of the gradient boosted regression tree.

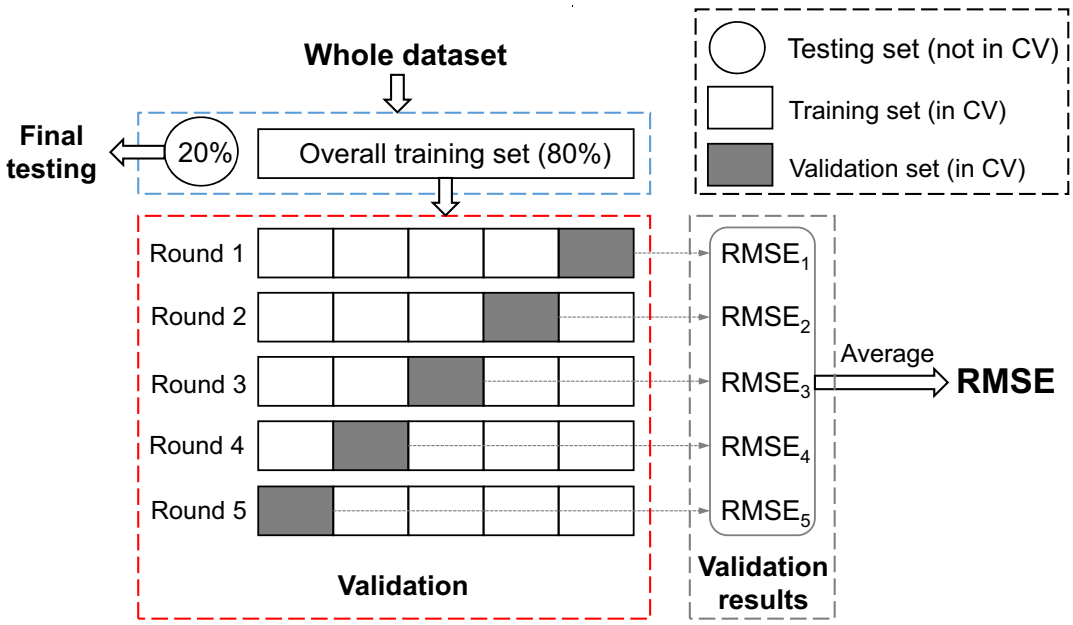


Fig. 4. Schematic diagram of dataset division and 5-fold cross validation (CV).

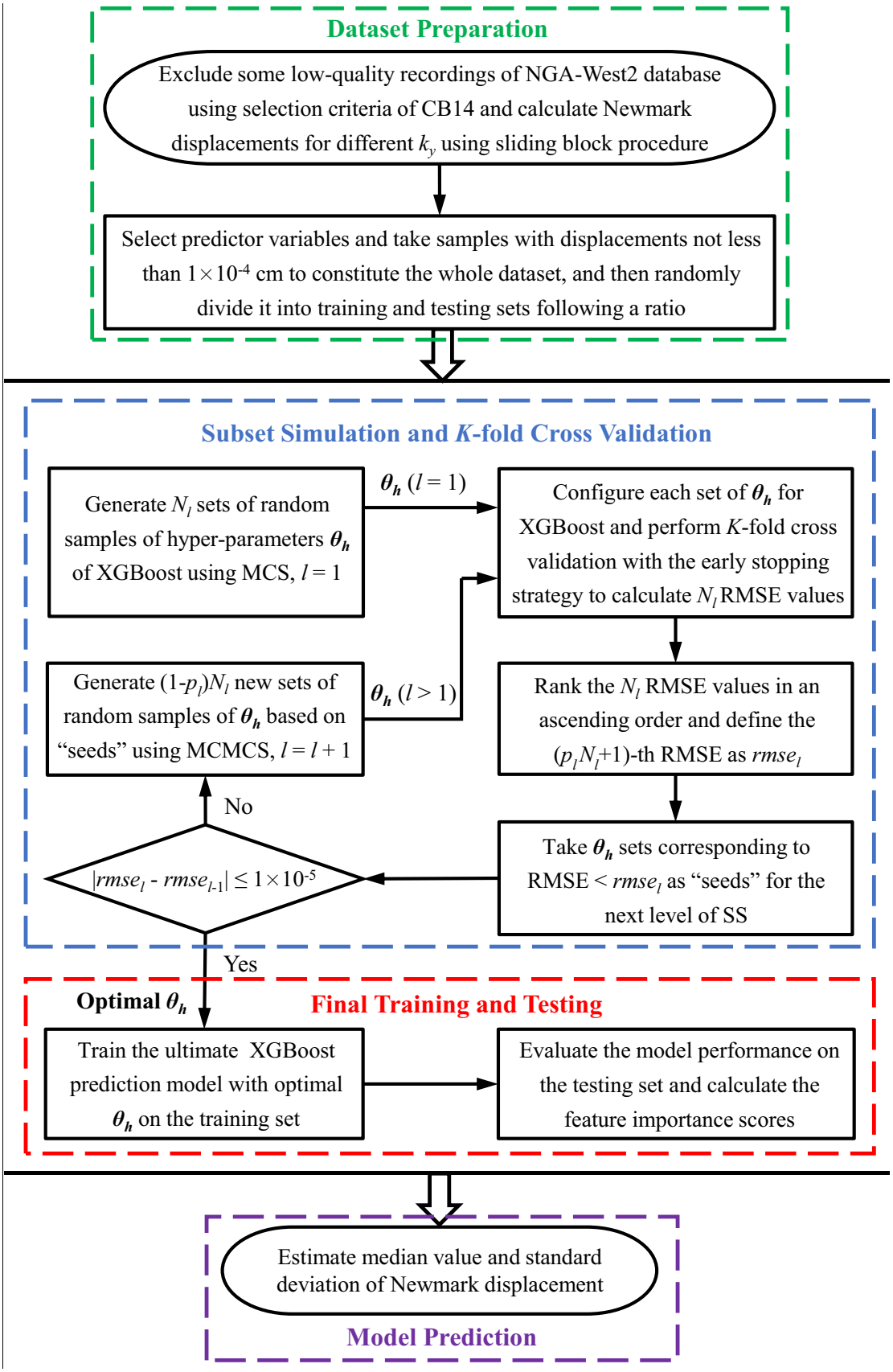


Fig. 5. Flowchart of the SS-XGBoost framework.

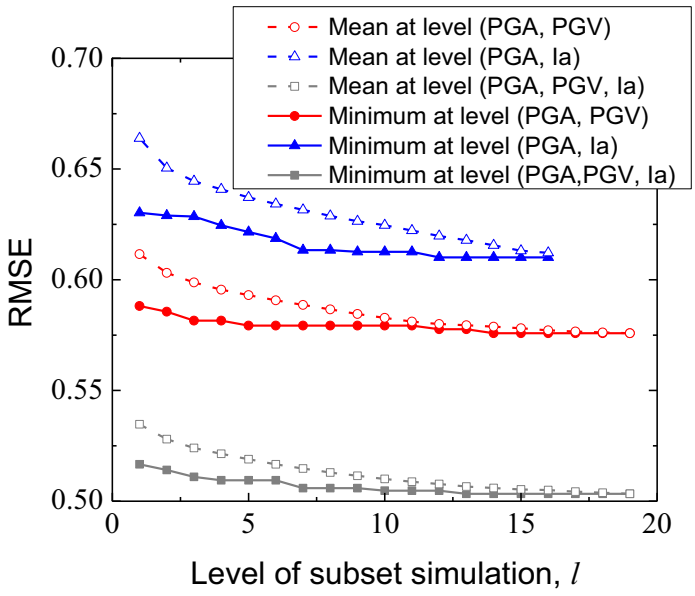


Fig. 6. RMSE versus simulation level in hyper-parameter tuning process for three XGBoost models.

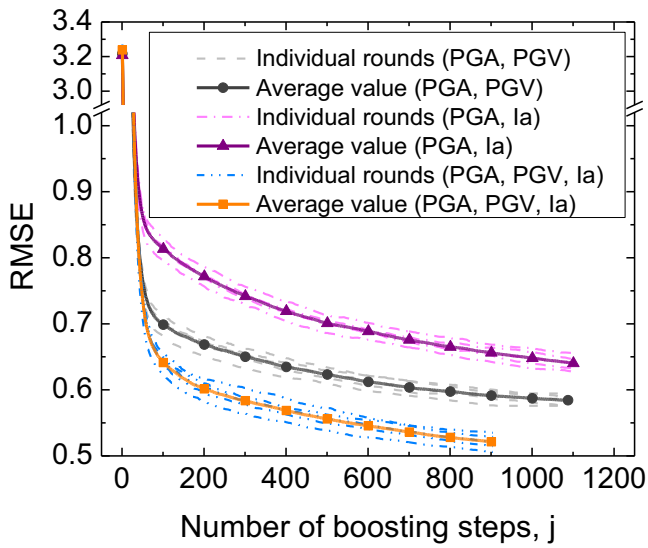


Fig. 7. RMSE versus boosting step in 5-fold cross validation for three XGBoost models with the optimal hyper-parameters.

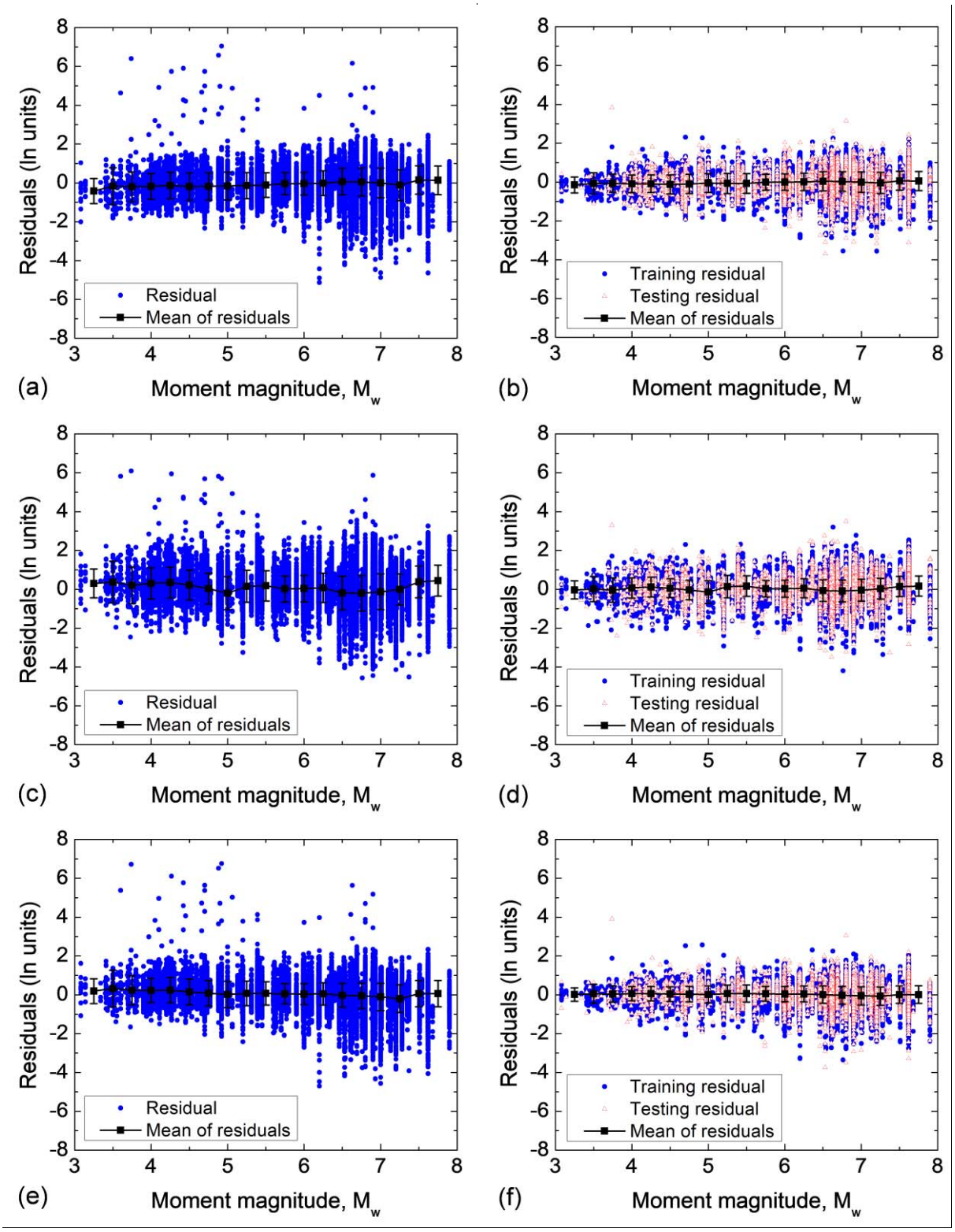


Fig. 8. Distributions of residuals with respect to M_w for different models: (a) updated SR08 (PGA, PGV); (b) XGBoost (PGA, PGV); (c) updated SR08 (PGA, I_a); (d) XGBoost (PGA, I_a); (e) updated SR08 (PGA, PGV, I_a); and (f) XGBoost (PGA, PGV, I_a). The error bar represents $\mu \pm \sigma$.

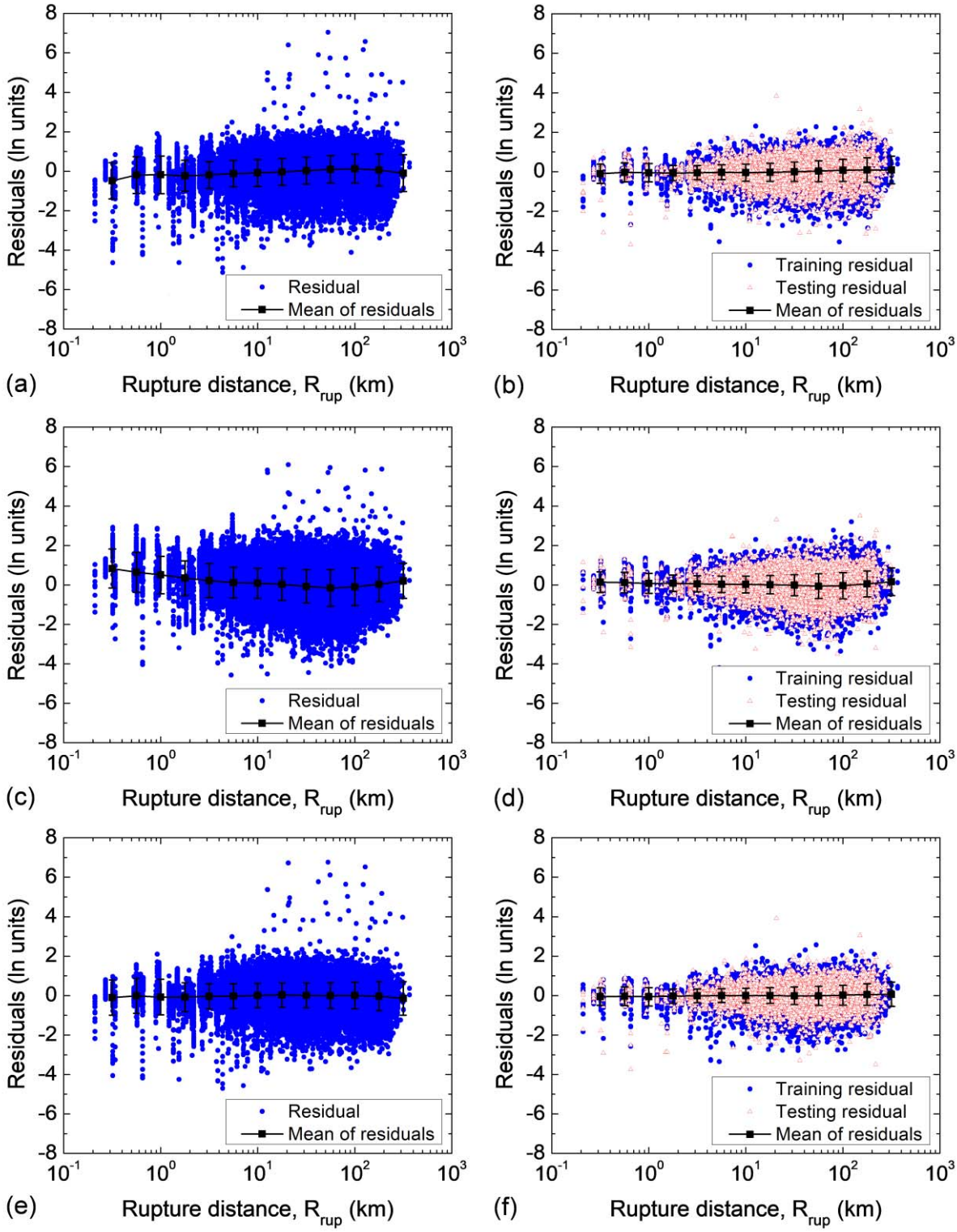


Fig. 9. Distributions of residuals with respect to R_{rup} for different models: (a) updated SR08 (PGA, PGV); (b) XGBoost (PGA, PGV); (c) updated SR08 (PGA, I_a); (d) XGBoost (PGA, I_a); (e) updated SR08 (PGA, PGV, I_a); and (f) XGBoost (PGA, PGV, I_a). The error bar represents $\mu \pm \sigma$.

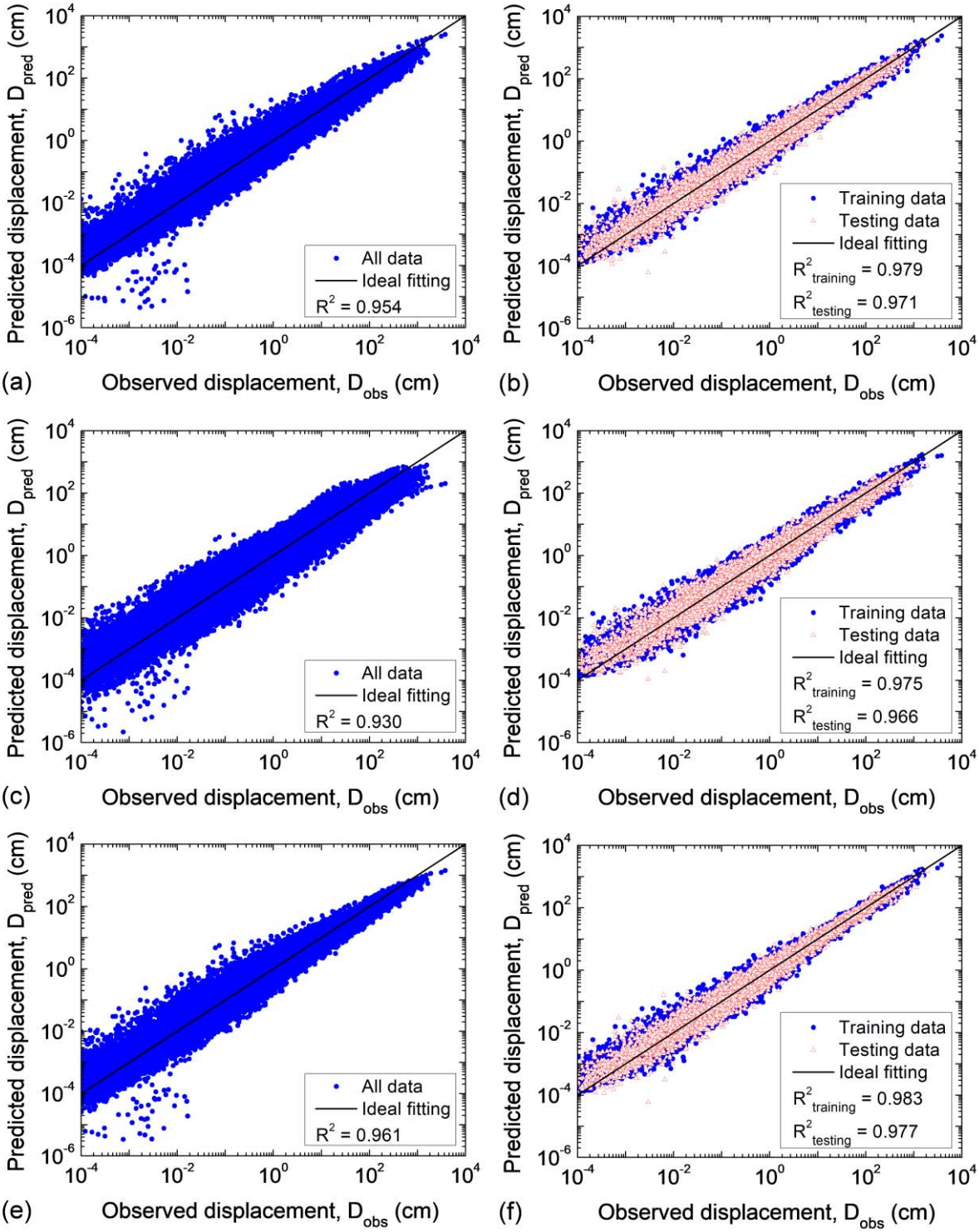
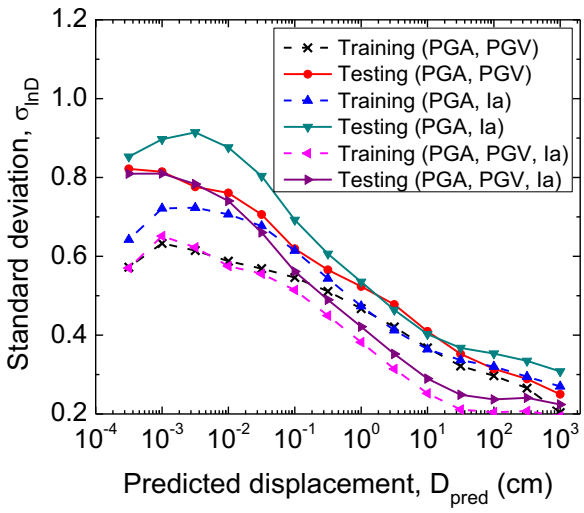


Fig. 10. Distributions of D_{pred} with respect to D_{obs} for different models: (a) updated SR08 (PGA, PGV); (b) XGBoost (PGA, PGV); (c) updated SR08 (PGA, I_a); (d) XGBoost (PGA, I_a); (e) updated SR08 (PGA, PGV, I_a); and (f) XGBoost (PGA, PGV, I_a).



Fig, 11. Standard deviation versus predicted displacement on training and testing datasets for three XGBoost models.

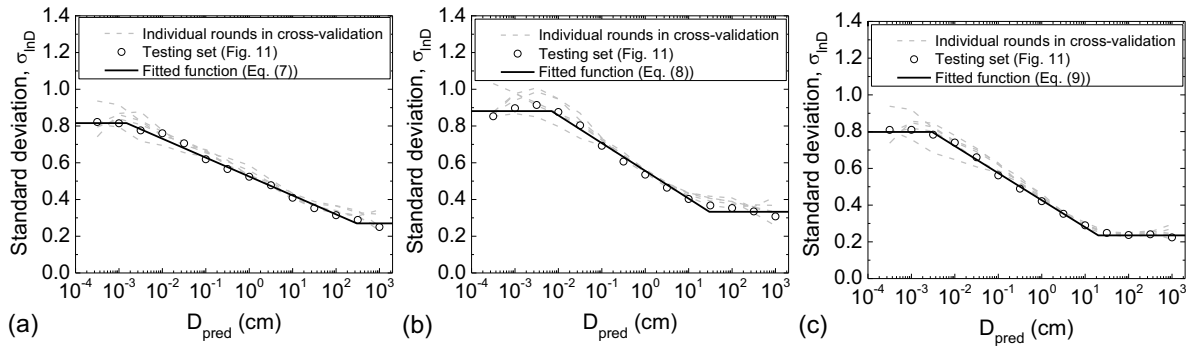


Fig. 12. Standard deviation versus predicted displacement on validation and testing datasets for (a) XGBoost (PGA, PGV); (b) XGBoost (PGA, I_a); and (c) XGBoost (PGA, PGV, I_a) models.

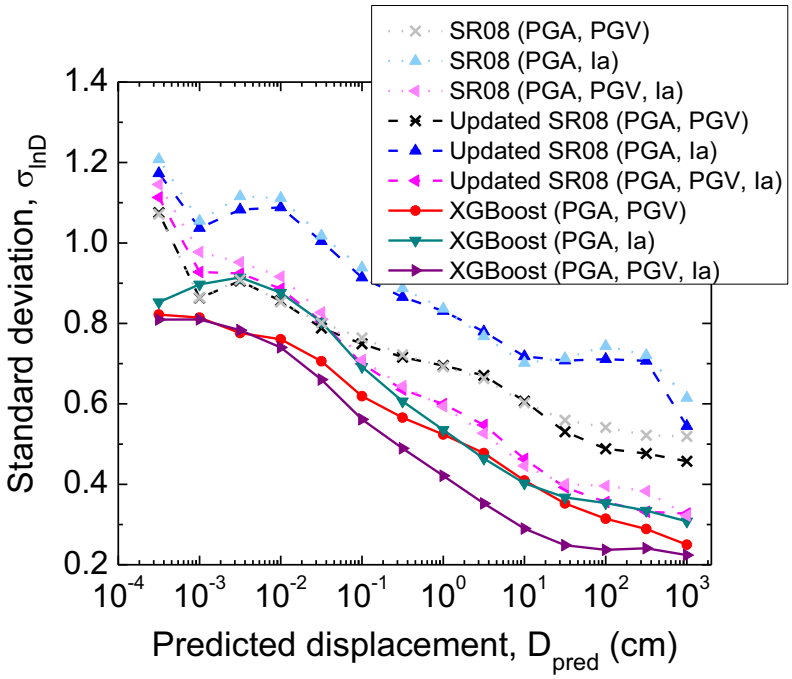


Fig. 13. Comparison of standard deviations on the testing set for different models.

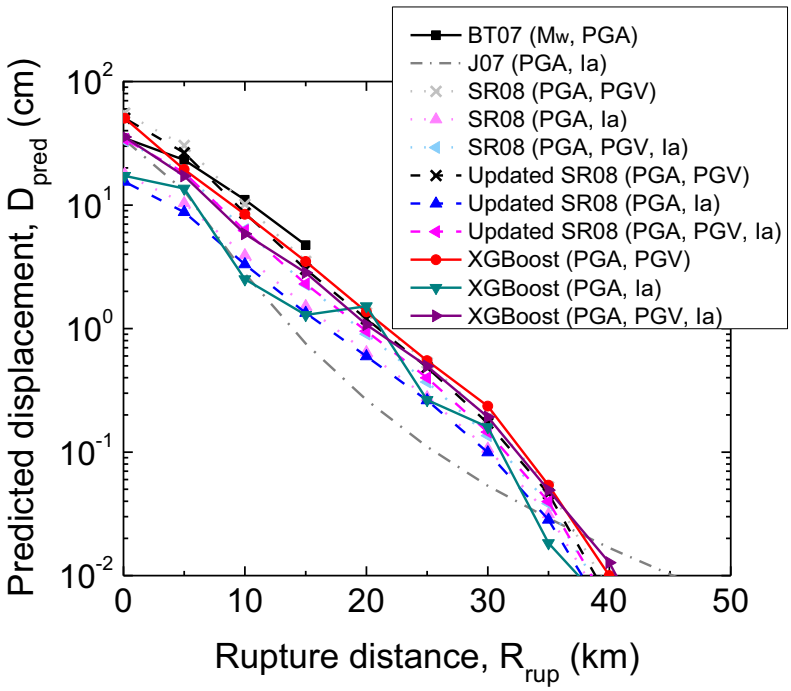


Fig. 14. Median predicted displacements associated with different models for deterministic earthquake scenario but varying slope conditions.

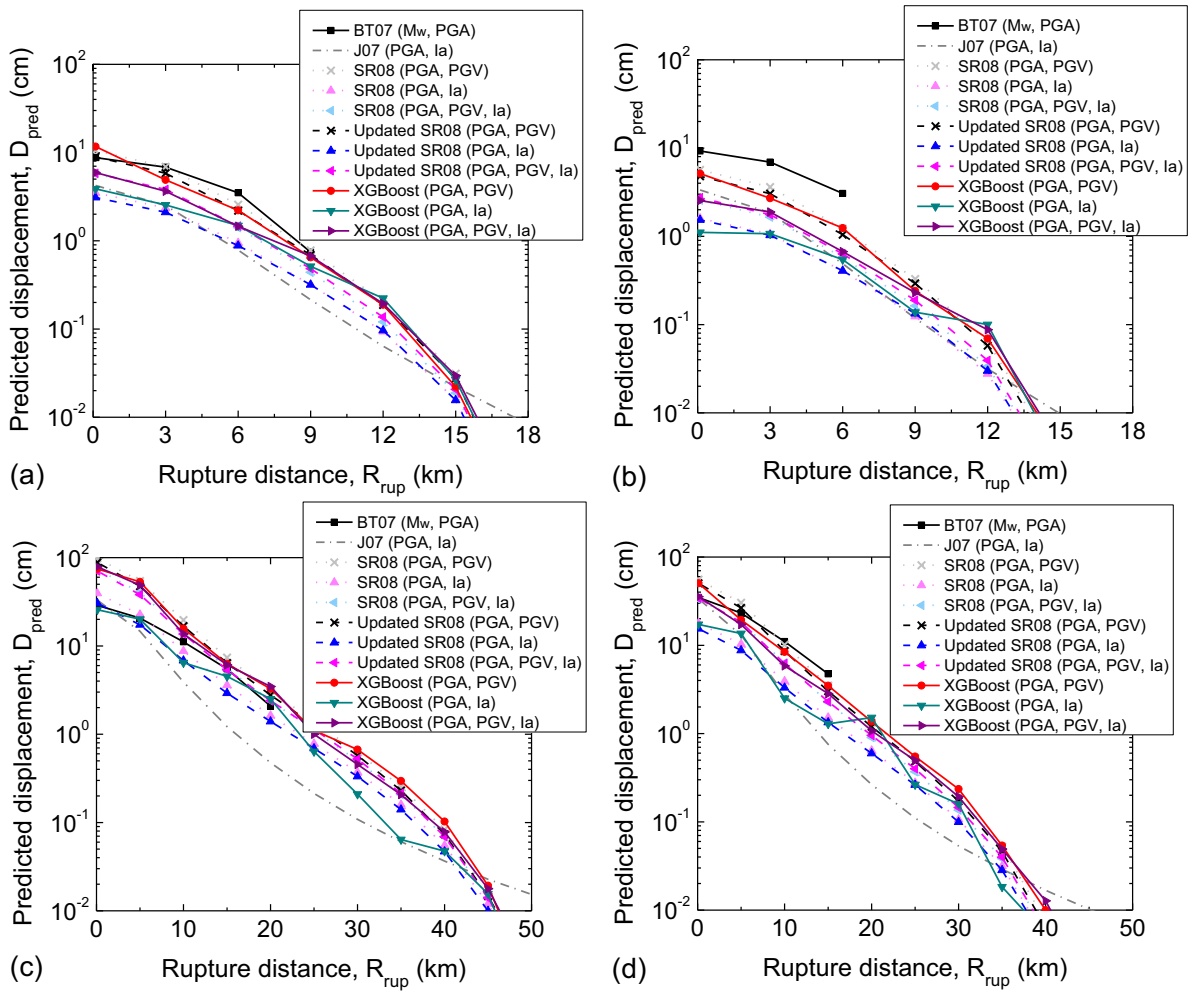


Fig. 15. Median predicted displacements associated with different models for deterministic slope condition (k_y = 0.1 g) but varying earthquake scenarios: (a) M_w = 5.5, V_{s30} = 400 m/s; (b) M_w = 5.5, V_{s30} = 760 m/s; (c) M_w = 7.5, V_{s30} = 400 m/s; and (d) M_w = 7.5, V_{s30} = 760 m/s.

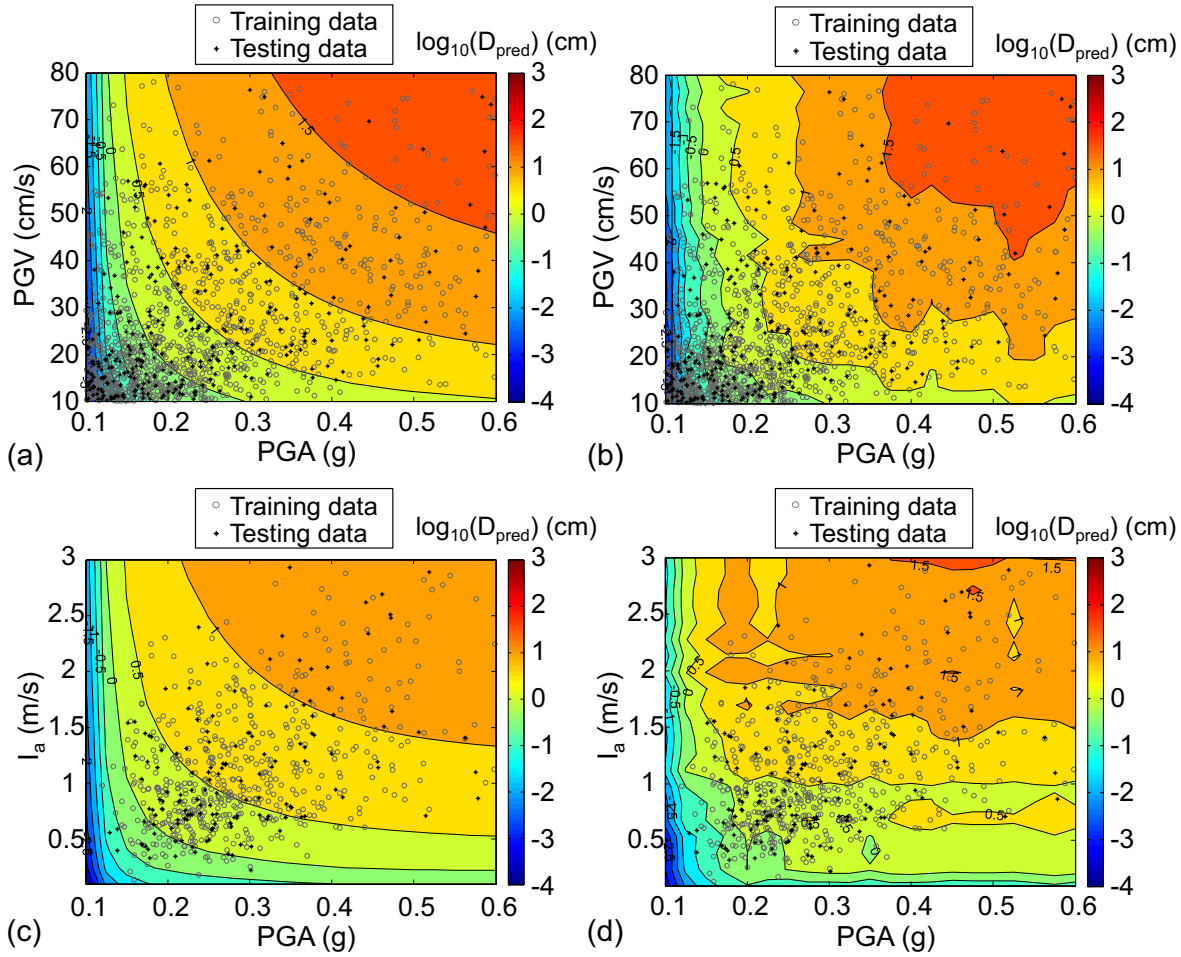


Fig. 16. Comparison of median displacement contours for different models: (a) updated SR08 (PGA, PGV); (b) XGBoost (PGA, PGV); (c) updated SR08 (PGA, I_a); and (d) XGBoost (PGA, I_a) considering $k_y = 0.1$ g.

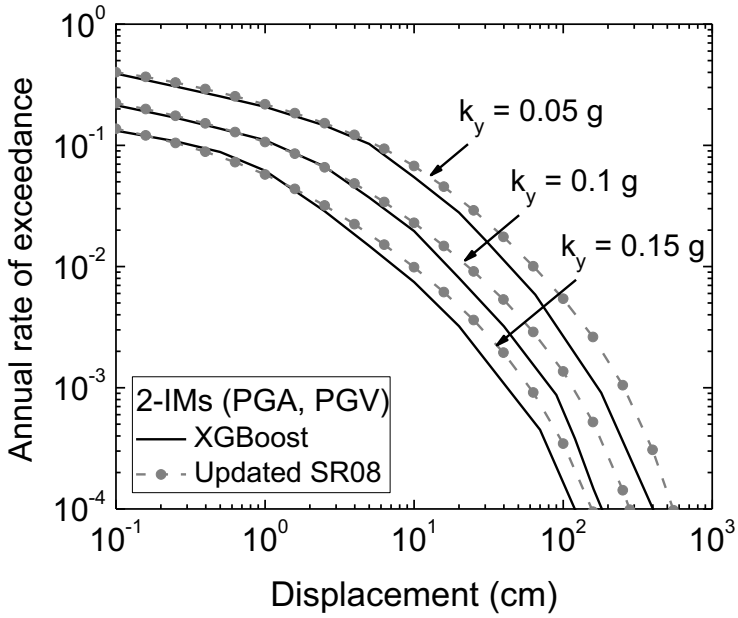


Fig. 17. Comparison of displacement hazard curves associated with updated SR08 model and XGBoost model for the (PGA, PGV) combination.

Table 1. Hyper-parameter tuning ranges and results for the three XGBoost models

Method	Hyper-parameter	Tuning range	Optimal hyper-parameter		
			(PGA, PGV)	(PGA, I_a)	(PGA, PGV, I_a)
Subset simulation	$d_{\max}$	3-6	6	6	6
	v	0.05-0.1	0.075	0.078	0.067
	w_{mc}	1-10	4.790	4.939	5.289
	λ	1-10	6.020	5.619	8.537
	γ	0.1-0.2	0.116	0.123	0.105
Early stopping	T	1-2000	1088	1101	905

Table 2. Model parameters of original and updated SR08 models

Model parameter	SR08 (Saygili and Rathje 2008)			Updated SR08		
	(PGA, PGV)	(PGA, I_a)	(PGA, PGV, I_a)	(PGA, PGV)	(PGA, I_a)	(PGA, PGV, I_a)
a_1	-1.56	2.39	-0.74	-1.18	3.06	-0.48
a_2	-4.58	-5.24	-4.93	-9.99	-11.31	-10.37
a_3	-20.84	-18.78	-19.91	-5.13	0.48	-3.43
a_4	44.75	42.01	43.75	26.44	18.34	24.28
a_5	-30.50	-29.15	-30.12	-23.20	-19.24	-22.28
a_6	-0.64	-1.56	-1.30	-0.66	-1.36	-1.18
a_7	1.55	1.38	1.04	1.56	1.24	1.16
a_8	-	-	0.67	-	-	0.50
$\sigma_{\ln D}$	Fig. 13					

Note: The updated SR08 models are developed based on the whole dataset of this study; and the $\sigma_{\ln D}$ for both the SR08 and updated SR08 models are derived based on the whole dataset of this study.

Table 3. External validation for the three XGBoost models

Metric	Criterion	XGBoost (PGA, PGV)		XGBoost (PGA, I_a)		XGBoost (PGA, PGV, I_a)	
		Training	Testing	Training	Testing	Training	Testing
RMSE	-	0.477	0.564	0.524	0.613	0.425	0.501
$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i \rvert$	-	0.357	0.418	0.385	0.452	0.304	0.354
R	$R > 0.8$	0.989	0.985	0.987	0.983	0.992	0.988
$k = \frac{\sum_{i=1}^{N}(y_i\hat{y}_i)}{\sum_{i=1}^{N}\hat{y}_i^2}$	$0.85 < k < 1.15$	1.006	1.005	1.010	1.006	1.006	1.005
$k' = \frac{\sum_{i=1}^{N}(y_i\hat{y}_i)}{\sum_{i=1}^{N}y_i^2}$	$0.85 < k' < 1.15$	0.973	0.967	0.966	0.960	0.978	0.973
$R_0^2 = 1 - \frac{\sum_{i=1}^{N}\left(\hat{y}_i - y_i^{r_0}\right)^2}{\sum_{i=1}^{N}\left(\hat{y}_i - \bar{\hat{y}}^{r_0}\right)^2}$, where $y_i^{r_0} = k\hat{y}_i$	$r = \left\lvert \frac{R^2 - R_0^2}{R^2} \right\rvert < 0.1$	R_0^2=1.000 (r=0.022)	1.000 (0.030)	1.000 (0.026)	1.000 (0.035)	1.000 (0.017)	1.000 (0.024)
$R_0'^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i^{r_0}\right)^2}{\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2}$, where $\hat{y}_i^{r_0} = k'y_i$	$r' = \left\lvert \frac{R^2 - R'^2}{R^2} \right\rvert < 0.1$	R'^2=0.999 (r'=0.020)	0.999 (0.029)	0.999 (0.025)	0.998 (0.033)	1.000 (0.017)	0.999 (0.023)

Note: y_i = observed value; $\hat{y}_i$ = predicted value; $\bar{y}$ = average of observed values.

Table 4. Comparison of external validation results on the testing set for SR08, updated SR08 and XGBoost models considering a cut-off displacement value of 0.01 cm

Metric	SR08			Updated SR08			XGBoost		
	(PGA, PGV)	(PGA, I_a)	(PGA, PGV, I_a)	(PGA, PGV)	(PGA, I_a)	(PGA, PGV, I_a)	(PGA, PGV)	(PGA, I_a)	(PGA, PGV, I_a)
R^2	0.934	0.901	0.948	0.936	0.903	0.949	0.961	0.955	0.972
MAE	0.529	0.651	0.447	0.517	0.647	0.441	0.391	0.417	0.319
RMSE	0.693	0.849	0.606	0.675	0.834	0.597	0.524	0.561	0.444

Table 5. Relative differences between $\sigma_{\ln D}$ for updated SR08 models and XGBoost models

IM combination	D_{pred} (cm)				
	0.1	1	10	100	1000
2-IM (PGA, PGV)	17.4%	24.7%	32.5%	35.6%	45.4%
2-IM (PGA, I_a)	24.3%	35.6%	44.0%	50.3%	43.5%
3-IM (PGA, PGV, I_a)	19.7%	29.7%	37.5%	33.5%	31.5%

Note: The value is calculated by $(1 - \sigma_{\ln D,\ \mathrm{XGBoost}}/\sigma_{\ln D,\ \mathrm{SR08}}) \times 100\%$.

Table 6. Importance scores of predictor variables for the three XGBoost models

IM combination	PGA	PGV	I_a	k_y
2-IM (PGA, PGV)	14745	12744	-	6507
2-IM (PGA, I_a)	15856	-	13796	5206
3-IM (PGA, PGV, I_a)	9604	7760	7142	6615

Table 7. Comparison of displacement hazard for updated SR08 model and XGBoost model with (PGA, PGV) combination

		k_y = 0.15 g			k_y = 0.1 g			k_y = 0.05 g		
Probability of exceedance in 50 years		10%	5%	1%	10%	5%	1%	10%	5%	1%
D (cm)	Updated SR08	37	59	121	76	114	225	176	254	448
	XGBoost	26	41	93	52	81	145	113	168	310
Hazard reduction (%)		31.6	28.9	35.6	35.8	33.9	30.8	29.7	30.5	23.1

Note: The hazard reduction is calculated by $(1 - D_{\mathrm{XGBoost}}/D_{\mathrm{SR08}}) \times 100\%$.