# MACHINE LEARNING-BASED PROCEDURES FOR ESTIMATING SEISMICALLY-INDUCED LANDSLIDES IN SUBDUCTION TECTONIC SETTINGS

Jorge Macedo* Chenying Liu†

U.S. Geological Survey Final Technical Report
Award No. G21AP10264
Award Term: April 2021 - April 2022

Principal Investigator:

Jorge Macedo, Ph.D.
Georgia Institute of Technology
School of Civil and Environmental Engineering
Atlanta, GA 30332
P: 1-510-277-6177, E: jorge.macedo@ce.gatech.edu

May 15 2022

Acknowledgment of Support and Disclaimer: This material is based upon work supported by the U.S. Geological Survey under Grant No. G21AP10264. However, the views and conclusions contained in this document are those of the authors and should not be interpreted as representing the opinions or policies of the U.S. Geological Survey. Mention of trade names or commercial products does not constitute their endorsement by the U.S. Geological Survey. We thank Dr. Yousef Bozorgnia, Dr. Silvia Mazzoni, and Dr. Norman Abrahamson for providing access to the intensity measures calculated as part of the NGA-Sub project. We also thank Dr. Silvia Mazzoni for providing support in generating displacement realizations with the NGA-Sub dataset. Lastly, we thank Dr. Ellen Rathje and Dr. Yubing Wang for providing data for regional-based assessments of landslides in Alaska.

---

[*] Georgia Institute of Technology, School of Civil and Environmental Engineering, 790 Atlantic Drive Atlanta, GA 30332-0355. Phone:1-510-277-6177, Email: jorge.macedo@ce.gatech.edu

[†] Georgia Institute of Technology, School of Civil and Environmental Engineering, 790 Atlantic Drive Atlanta, GA 30332-0355. Phone:1-650-300-9778, Email: cliu662@gatech.edu

# Abstract

Seismically-induced landslides (SIL) have caused severe damage to infrastructure and numerous fatalities after recent earthquakes. Thus, proper planning and hazard mitigation in the U.S. require an adequate evaluation of the potential for SIL and the associated consequences. This evaluation often involves the estimation of the associated seismically-induced slope displacements ($D$), which are used as a performance index to assess the impacts caused by SIL. Several robust models (i.e., models developed with a large number of recordings and for different slope conditions) for estimating $D$ have been proposed for shallow crustal tectonic settings. However, the authors are aware of only one robust model for subduction tectonic settings, such as the U.S. Pacific Northwest (i.e., the Bray et al. (2018) model), which has been developed for subduction interface earthquakes. Hence, developing more models for subduction interface tectonic settings is warranted. Moreover, models for subduction intraslab tectonic settings are also lacking. This study uses machine learning (ML) based procedures such as ridge regression, random forest, partial least square regression, gradient boosting decision tree, and neural networks combined with ML-based feature selection techniques to develop a set of new models to evaluate the seismic performance of slope systems in the context of seismically-induced landslides. In addition, traditional models (i.e., models based on fixed polynomial forms) are also developed. We use intensity measures from the robust NGA-Sub ground motion database, which has ground motion information from subduction type earthquakes (including both interface and intraslab tectonic settings).

Using the previously referred ML-based procedures, we identify efficient features to estimate seismically induced slope displacements in subduction tectonic settings, including intensity measures (IMs) and properties of the sliding mass. In particular, we find that the yield coefficient ($k_y$), the fundamental period of the sliding mass ($T_s$ - taken as zero for shallow landslides that often dominate regional assessments), the earthquake magnitude ($M$), the degraded spectral acceleration at 1.3 $T_s$, $Sa(1.3T_s)$ (which becomes the peak ground acceleration, $PGA$, for shallow landslides) and the peak ground velocity ($PGV$) are efficient features for estimating $D$. Our findings on efficient features suggest value in implementing/refining capabilities for a rapid assessment of $PGV$ and spectral accelerations using USGS tools (e.g., ShakeMap, Ground Failure) in subduction tectonic settings such as the U.S. Pacific Northwest. These implementations, which could be performed in future efforts, would be helpful in the rapid assessment of the damage potential of seismically-induced landslides.

The performance of the developed machine learning-based models is assessed considering their prediction capabilities in test sets, trends, and case histories (when there is available information). Our assessments suggest that the following models have a good performance: ridge regression, random forest, gradient boosting decision tree, support vector regression, and residual neural network models. In contrast, the principal component regression and partial least square models have inferior performance. Moreover, the recommended machine learning-based models also show better statistical performance than their traditional counterparts.

The developed models are used in preliminary performance-based probabilistic implementations, which produce $D$ hazard curves (i.e., curves that relate different $D$ thresholds with their annual rate of exceedance), considering the treatment of epistemic uncertainties. Specifically, we consider a typical logic tree approach with Monte Carlo simulation where different models are assigned with weights and a recent novel procedure that uses the polynomial chaos theory, which proves to be more computationally efficient.

Finally, we illustrate the use of the developed models considering (1) pseudo-probabilistic based assessments of $D$ in the U.S Pacific Northwest; (2) performance-based probabilistic assessments of $D$ in the U.S. Pacific Northwest considering the contribution of subduction interface and intraslab seismic sources; (3) the treatment of epistemic uncertainties considering logic trees and the polynomial chaos framework; and (4) a regional-based assessment of seismically-induced landslides in the Alaska region.

# Table of Contents

# 1 Introduction

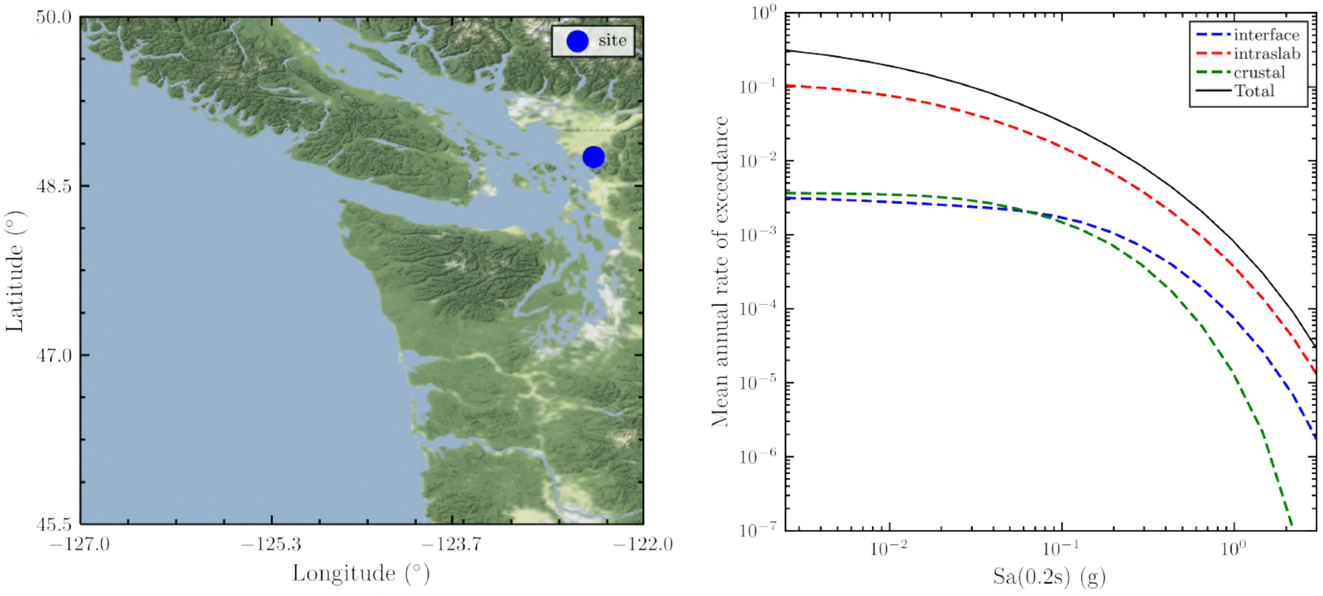
Seismically-induced landslides (SIL) have caused severe damage to our infrastructures as well as the loss of numerous lives after recent earthquakes. Notable SIL events in previous earthquakes include those during the 1964 Alaska (USA), 1989 Loma Prieta (USA), 1994 Northridge (USA), 1999 Chi-Chi (Taiwan), 2004 Nigata (Japan), 2008 Wenchuan (China), 2015 Gorkha (Nepal), and the 2016 Kaikoura (New Zealand) earthquakes. According to Marano et al. (2009), 5% of fatalities related to earthquake-induced damage are associated with SIL, situating SIL as the third-largest contributor to fatalities. There are numerous examples of the devastating damage that SIL can cause. For example, in the 1964 Alaska earthquake, SIL led to more than half of the earthquake-induced damage, responsible for an estimated 56% of the total cost of damage. In Japan, more than half of all deaths in large ($M > 6.9$) earthquakes between 1964 and 1980 were caused by seismic landslides (Kobayashi, 1981). SIL have also caused significant disruptions to lifelines, impeding emergency response efforts (Bird and Bommer, 2004). More recently, SIL triggered during the 2008 Wenchuan, and the 2015 Gorkha earthquakes caused around 20,120 fatalities (Collins and Jibson, 2015), and the 2016 Kaikoura earthquake generated more than 10,000 landslides over a total area of about 10,000 km$^2$ (Massey et al., 2018). Figure 1 shows examples of SIL-Induced damage in previous earthquakes. SIL events in future scenarios similar to those in previous earthquakes have the potential to cause dramatic impacts in areas of the U.S. affected by earthquakes. Thus, proper planning for hazard mitigation requires a proper evaluation of the potential for seismic landslides and the associated consequences.

Figure 1: Examples of damage caused by seismically-induced landslides (SIL) after recent earthquakes. Left: April 2016 Kumamoto earthquake; Right: January 2001 El Salvador earthquake.

The evaluation of the potential for SIL-induced damage often involves an estimation of the associated seismically-induced displacements ($D$), which is the focus of this study. $D$ is often used as a performance-index to assess the damage potential of SILs (Wang and Rathje, 2015; Jibson and Tanyaş, 2020; Rathje and Saygili, 2008; Tsai and Chien, 2016; Hsieh and Lee, 2011; Jibson, 2007). $D$ is commonly estimated using semi-empirical models, which take as inputs slope properties (e.g., the yield coefficient - $k_y$, the fundamental period of the sliding mass - $T_s$), earthquake characteristics (e.g., earthquake magnitude - $M$), and ground motion intensity measures (IMs) and provide $D$ estimates (median estimates and standard deviation).

There are several existing $D$ models developed in previous studies, but most of them have been focused on shallow crustal tectonic settings. For example, Bray and Travasarou (2007) used a ground motion database of 688 recordings to develop $D$ models that are applicable to rigid (i.e., a slope with a $T_s$ value that is nearly zero) and flexible slopes (i.e., a slope with a $T_s$ value greater than zero). Saygili and Rathje (2008) used a larger ground motion database (i.e., 2000 ground motion recordings) to develop $D$ models for rigid slopes; Rathje and Antonakos (2011) then extended the models from Saygili and Rathje (2008) to flexible slopes. Later, Bray and Macedo (2019) updated the model developed by Bray and Travasarou (2007) using a more comprehensive database with 6711 ground motion recordings (i.e., the NGA-West2 Database (Bozorgnia et al., 2014)). More recently, Wang et al. (2020), Cho (2020a), and Macedo et al. (2021) have also proposed additional $D$ models for shallow crustal tectonic settings. Of note, in most cases, $D$ models with $T_s$ equal to zero, associated with shallow landslides have been used for the regional-based assessment of SIL, as commonly shallow landslides dominate in a regional context (Wang and Rathje, 2015; Jibson and Tanyaş, 2020; Rathje and Saygili, 2008; Tsai and Chien, 2016; Hsieh and Lee, 2011; Jibson, 2007). This implies that the sliding mass is assumed rigid. For example, Wang and Rathje (2015) and Rathje (2014), used the Saygili and Rathje (2008), Bray and Travasarou (2007), Jibson (2007), and Rathje and Saygili (2009) models formulated for rigid slopes to calculate regional-based $D$ values to assess the potential for SIL in Alaska. They found that incorporating epistemic uncertainty of $D$ models can increase the area of high seismic landslide hazard by a factor of 2 to 3 as compared to analyses without any epistemic uncertainty.

In contrast to the availability of $D$ models for shallow crustal tectonic settings, $D$ models for subduction tectonic settings are scarce. To our knowledge, the model proposed by Bray et al. (2018) is the only robust (i.e., developed with a large number of ground motions and applicable to both rigid and flexible slopes) $D$ model developed for subduction interface tectonic settings, and we are not aware of $D$ models developed for subduction intraslab tectonic settings. However, subduction intraslab seismic sources can be of significant importance for the seismic design of slope systems in some regions like the Cascadia subduction zone in the Pacific Northwest of the United States. To illustrate this, Figure 2 shows the results of a probabilistic seismic hazard assessment (PSHA) for a site located in Bellingham, Washington, USA, where there is a contribution from shallow crustal and subduction type seismic sources. The PSHA was conducted for $Sa$ (spectral acceleration) at 0.2 s using the USGS unified hazard tool (Frankel et al., 2000). Figure 2 (b) shows that the intraslab seismicity dominates the hazard at this location. This is associated with the tectonic setting at the selected location, where the intraslab seismic source is "beneath" the Bellingham site at relatively shorter distances than interface seismic sources. Thus, formulating $D$ models for subduction intraslab tectonic settings is warranted. Moreover, given the scarcity of $D$ models in subduction interface tectonic settings, more models for this tectonic setting are also warranted to better account for epistemic uncertainties.


(a)

(b)

Figure 2: (a) Evaluation site in Bellingham, Washington for probabilistic seismic hazard analysis (b) $Sa(0.2s)$ hazard curves deaggregated by tectonic settings from USGS unified hazard tool.

Although several semi-empirical $D$ models are available in engineering practice (most of them for shallow crustal tectonic settings, as previously discussed), they have often been formulated under traditional statistical approaches. For example, in terms of the selection of efficient features (i.e., slope properties and IMs) that can explain the trends in $D$, most of the existing models use features that have been selected based on a fixed functional form where different features are tested until getting a standard deviation that is considered acceptable (i.e., low). Another limitation results from the uncertainty in the functional forms of $D$ models. Most of the existing $D$ models have been formulated by assuming fixed parametric forms (usually second- or third-order polynomials), which may limit their capacity to capture more complex relationships between the input features and $D$. With the current momentum in machine learning (ML), aided by the steady increase in computational power, the application of ML-based procedures in the formulation of $D$ models is promising and desired, which is also the case for earthquake engineering and seismology in general (e.g., Xie et al. (2020), Kong et al. (2019)). Our study contributes to that direction. ML procedures enable the selection of important features in a robust and systematic manner. They can also be used to formulate parametric and non-parametric $D$ models that can provide a balance between model interpretability and predictive performance (Friedman et al., 2001). In this context, recent efforts have been oriented to develop ML-based models to estimate $D$. For example, Wang et al. (2020) used the extreme gradient boosting (i.e., XGboost, Chen and Guestrin (2016)) algorithm to develop a semi-empirical $D$ model, which was shown to be superior to traditional counterparts by providing a lower standard deviation. More recently, Cho et al. (2022) developed multi-layer perceptron networks that improved the estimation of $D$, when compared against the Saygili and Rathje (2008) $D$ model. Most of these previous studies have focused on a single selected ML procedure. An exception is the Macedo et al. (2021) study that assessed the performance of 19 different ML procedures for developing $D$ models for shallow crustal tectonic settings. They found that $D$ models based on random forest, second-order polynomials, and kernel regression were efficient and outperformed traditional $D$ models. To our knowledge, all previous efforts for formulating $D$ models using ML-based procedures have been focused on shallow crustal tectonic settings, and we are unaware of previous efforts to develop ML-based $D$ models for subduction type tectonic zones, which is the focus of this study.

This report is organized as follows, after a brief introduction in Section 1; Section 2 describes the ground motion database used in this study and the generation of $D$ realizations; Section 3 describes the development of $D$ models using traditional statistical methods. Section 4 describes the selection of features that explain $D$ using ML-based techniques and the development of ML-based $D$ models. Section 5 presents a performance evaluation of the formulated models. Section 6 describes the treatment of epistemic uncertainty in performance-based procedures. Section 7 discusses the implementation of the developed models. Section 8 presents examples of using the developed models in performance-based assessments, including site-specific and regional assessments. Finally, Section 9 presents the conclusions of this study.

## 2 Ground motion database and generation of $D$ realizations

This study uses the NGA-sub ground motion database (Bozorgnia and Stewart, 2020; Kishida et al., 2018) developed by the Pacific Earthquake Engineering Research (PEER) center to generate $D$ realizations that are later used to develop $D$ models. The entire NGA-Sub database consists of 71343 three-component recordings from 1883 earthquakes between 1937 and 2016 from subduction zones around the globe. The selection criteria in Kuehn et al. (2020) are used to select a subset from this database, including (1) earthquakes with magnitudes greater than 4.0 are selected; (2) recordings with $R_{rup}$(rupture distance) $\leq$ 800 km and $R_{rup} \leq R_{max}$ are selected ($R_{max}$ is the maximum distance for each event to avoid biasing by non-triggered recordings, Contreras et al. (2020)); (3) recordings with unrealistic IMs (e.g., $PGA$ (peak ground acceleration) $>$ 10g according to Kuehn et al. (2020)) are excluded; and (4) recordings with time histories from multiple events, or non-free-field stations are removed. More details on the selection criteria can be found in Kuehn et al. (2020). The selected subset contains 6240 two-component ground motion recordings from 174 interface earthquakes with magnitudes from 4 to 9 and 8299 two-component ground motion recordings from intraslab earthquakes with magnitudes from 4 to 7.8. The distribution of earthquake magnitude ($M$) and rupture distance ($R_{rup}$) for the interface and intraslab earthquake recordings used in this study is shown in Figure 3.

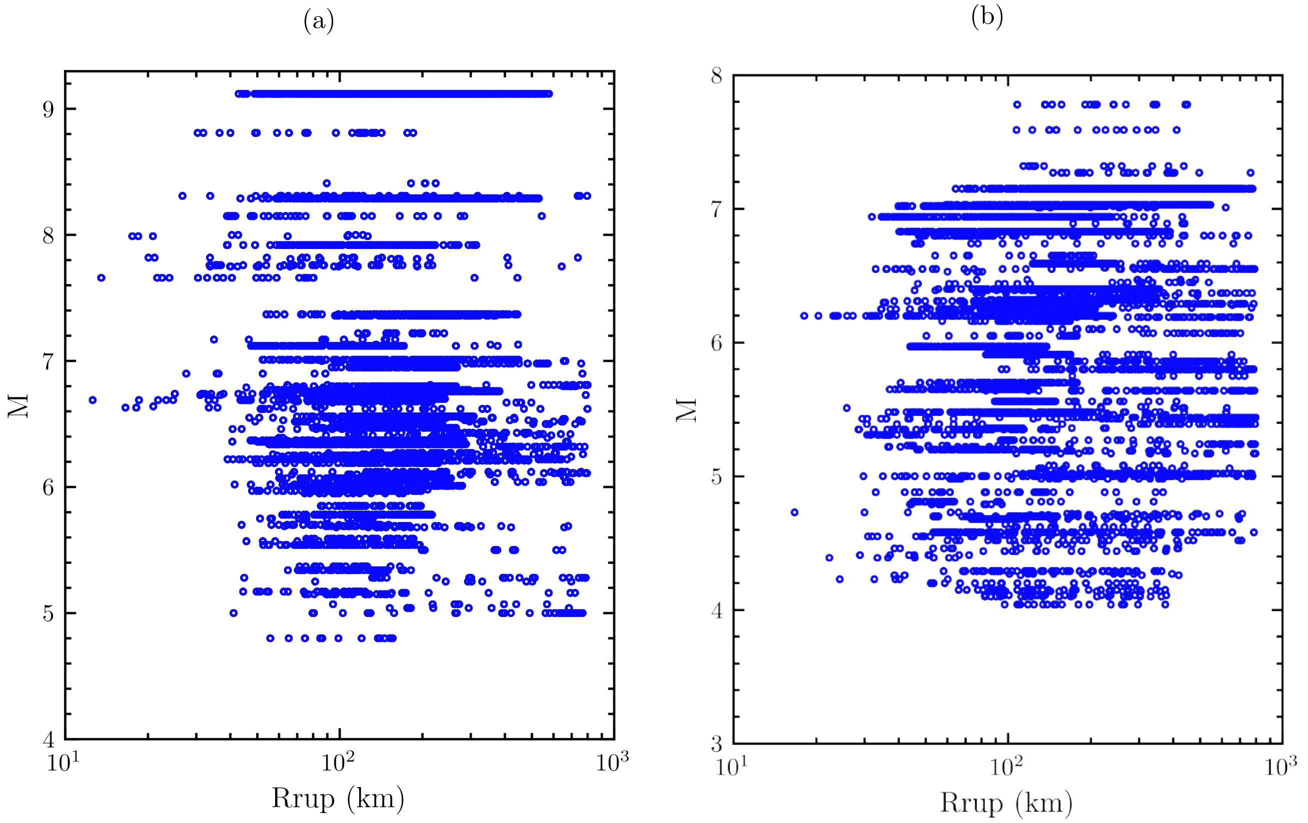


Figure 3: Magnitude and distance distribution for the ground motion recordings used in this study for (a) interface and (b) intraslab earthquakes.

We use the selected subset of the NGA-sub ground motions and the fully coupled, nonlinear, and deformable stick-slip sliding model from Rathje and Bray (2000) and modified by Macedo et al. (2017) to generate $D$ realizations. This model has also been used in previous studies to generate $D$ realizations that are then used to develop $D$ models (Bray and Travasarou, 2007; Bray et al., 2018; Bray and Travasarou, 2007). The model requires as inputs $k_y$, $T_s$, the unit weight of the sliding mass, the shear modulus curves and material damping curves of the sliding mass, and an acceleration time history. The unit weight of the sliding mass is set to 19 kN/m$^3$, and the shear modulus reduction and material damping curves are set according to the values in Darendeli (2001) for 1 atm and PI = 15. The fully coupled stick-slip sliding model has yield coefficients ($k_y$) ranging from 0.01 to 0.8 (i.e., 0.01, 0.02, 0.035, 0.05, 0.075, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, and 0.8), and initial fundamental period ($T_s$) ranging from 0 to 2 s (i.e., 0.0, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0, 1.4, and 2.0 s). The overburden stress-corrected shear wave velocity ($V_{s1}$) is set to 270 m/s for the baseline sliding block, and the sliding block's shear wave velocity profile follows the relationship that shear wave velocity ($V_s$) is proportional to the fourth root of the vertical effective stress. The sliding block height (H) is increased until the specified value of $T_s$ is obtained. For nonzero $T_s$ values, H is varied between 3 and 100 m, and $V_{s1}$ is varied between 200 and 450 m/s. As a result, the cases considered represent a wide range of natural earth slopes, earth dams, and solid-waste landfills. Once the properties for each model realization are defined, each ground motion recording in the selected subset (considering the two horizontal components) is applied to the base of the model to generate $D$ realizations. The estimated $D$ values are averaged over the two horizontal components of the ground motion. The opposite polarity of the components is then applied to compute an average seismic displacement for the other side of the recording. Finally, for each ground motion recording, the $D$ value is estimated as the maximum average $D$ values for each polarity. The procedures used to generate $D$ realizations in this study are consistent with those used by Bray et al. (2018). The final database used for developing semi-empirical $D$ models contains 1,510,093 and 1,855,516 displacements for interface and intraslab earthquakes. Figures 4 and 5 show the trends of $D$ realizations against $k_y$, $T_s$, $M$, $R_{rup}$, $PGV$, and $Sa(1.3T_s)$ for interface and intraslab earthquakes, respectively. Although the variability in $D$ is significant, the trends of $D$ against the slope properties, earthquake characteristics, and ground motion IMs are still noticeable. In general, $D$ decreases as $k_y$ increases or $R_{rup}$ increases, and $D$ tends to increase with the increase of $M$, $PGV$, and $Sa(1.3T_s)$, for both interface and intraslab earthquakes.

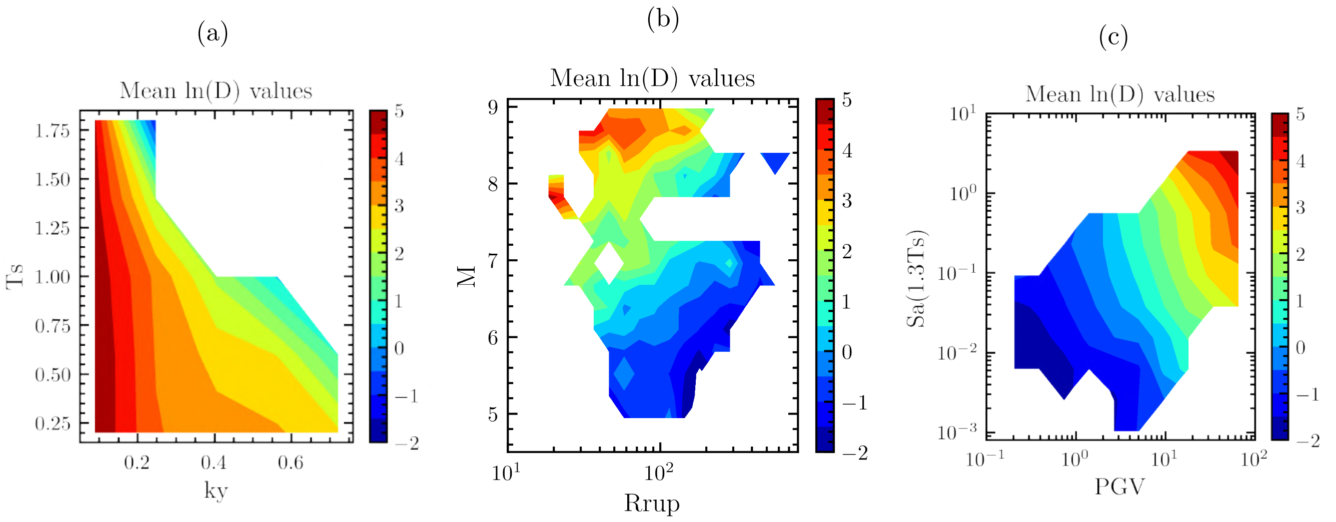


Figure 4: Trends of displacement (cm) against a subset of candidate features for interface earthquakes. The $D$ contours represent the mean $\ln D$ values in selected bins.

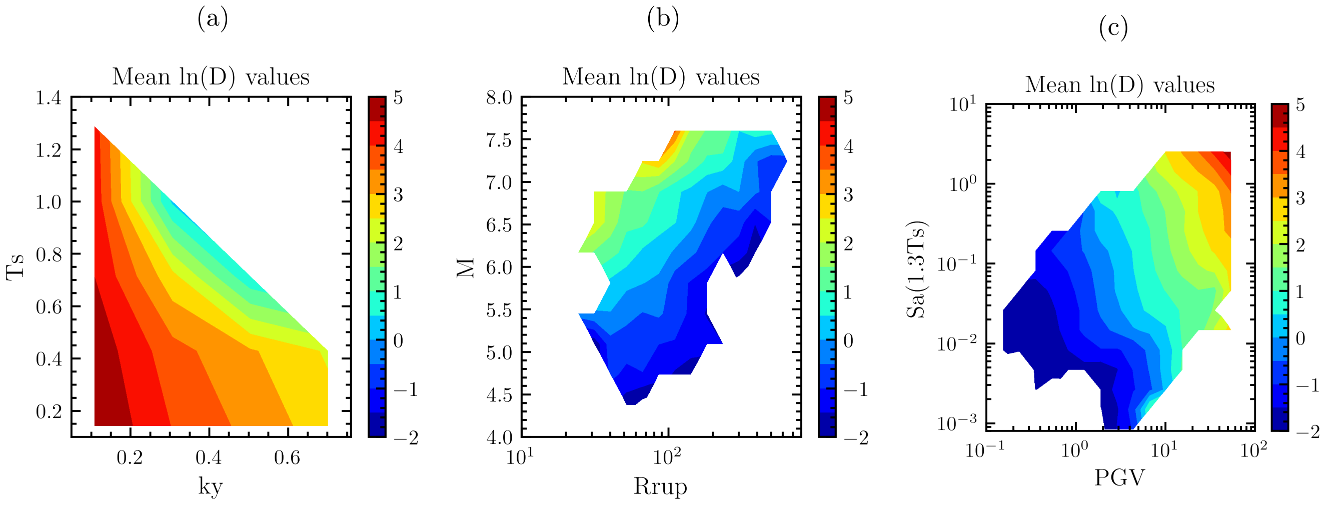


Figure 5: Trends of displacement (cm) against a subset of candidate features for intraslab earthquakes. The $D$ contours represent the mean $\ln D$ values in selected bins.

The database used in this study for interface tectonic settings is over six times greater than that utilized by Bray et al. (2018)), referred to hereinafter as BMT18. We found it instructive to assess the performance of the BMT18 against the generated displacements in our database. This assessment is presented in Figure 6, which shows the BMT18 model residuals (i.e., $Ln(D_{observed}) - Ln(D_{estimated})$). One can observe only a few instances where the residuals are slightly different from zero. This highlights the robustness of the BMT18 model for interface tectonic settings despite being developed with a much smaller (and independent) database than the one considered in this study.

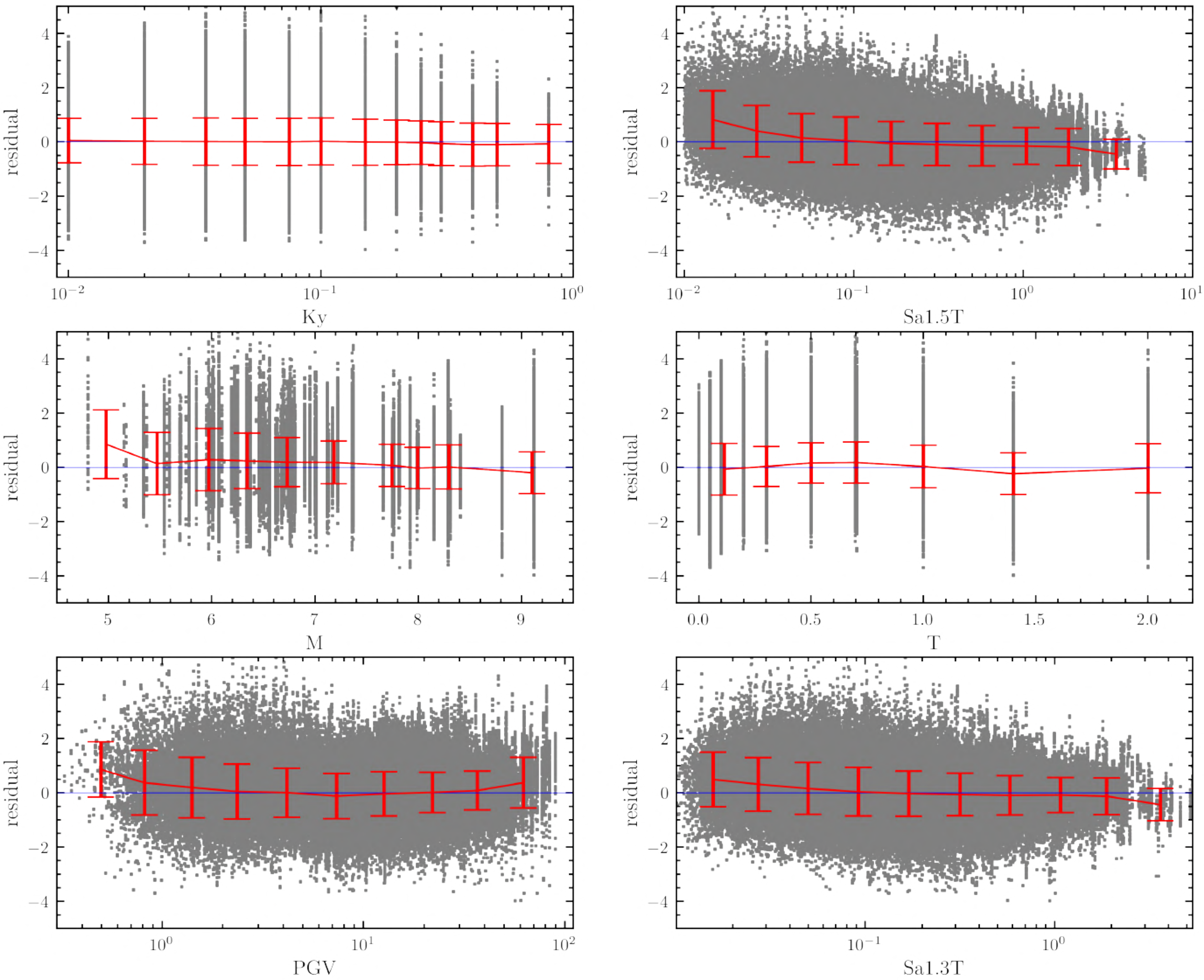


Figure 6: Residuals between the $D$ realizations obtained for interface earthquakes and estimated values from the BMT18 model.

Even though the BMT18 model is robust, updated $D$ models that reduce the small biases of the BMT18 for some scenarios (e.g., large $Sa(1.3T_s)$, and large $PGV$) are formulated by using the more extensive seismic displacement database developed in this study. In addition, considering that engineers often use the BMT18 model (formulated for subduction interface) in subduction intraslab tectonic settings, Figure 7 shows the residuals of the BMT18 model when applied to our subduction intraslab database. In this case, there are significant trends in slope properties and ground motion IMs, which motivates the development of new $D$ models for intraslab tectonic settings.

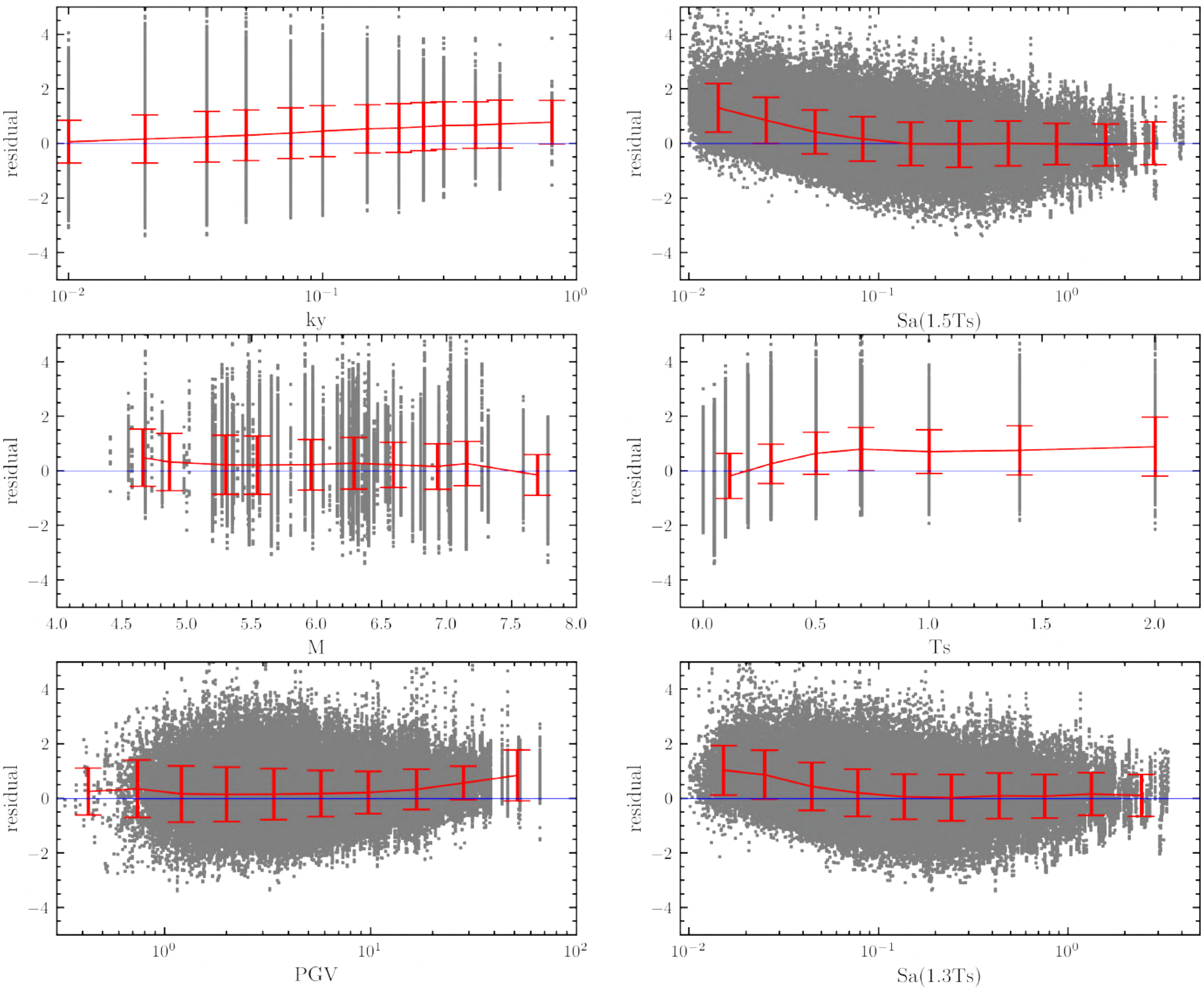


Figure 7: Residuals between the $D$ realizations obtained for intraslab earthquakes and estimated values from the BMT18 model.

# 3 Development of traditional $D$ models

## 3.1 Estimation of the probability of negligible $D$

In this study, we define as negligible (i.e., zero) the $D$ values smaller than 0.5 cm, as small $D$ values are of little interest and can be considered negligible for all practical purposes. This definition is consistent with that used in previous studies (e.g., Bray et al. (2018)). In addition, we use the framework of Bray and Travasarou (2007) and model $D$ as a mixed random variable according to Equation 1. This framework has also been considered for the development of $D$ models in previous studies (e.g., Bray and Macedo (2019), Bray et al. (2018)).

$$f_D(d) = \bar{P}\delta(d - 0.5) + (1 - \bar{P})\bar{f}_D(d) \tag{1}$$

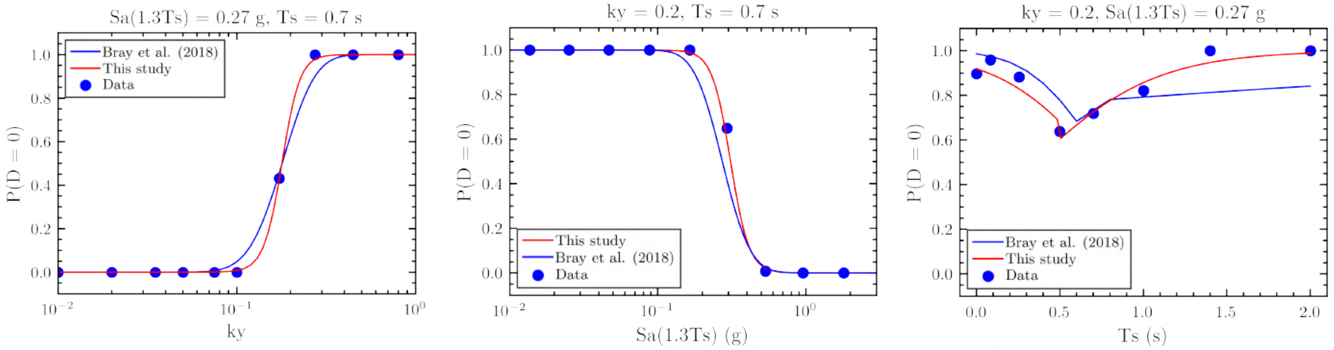
In Equation 1, $f_D(d)$ is the probability density function (PDF) of $D$, $\delta(d - 0.5)$ is the Dirac delta function that gives a value of 1 when $D = 0.5$ and 0 otherwise, $\bar{P}$ is the probability of a negligible $D$ (i.e., $D < 0.5$), and $\bar{f}_D(d)$ is the PDF of $D$ for $D > 0.5$. $f_D(d)$ has a finite probability mass to account for negligible $D$ values and has a PDF for

$D > 0.5$ that models the distribution of non-negligible $D$ values.

Consistent with the mixed random variable characterization of $D$, we first develop a model for estimating the probability of negligible displacements (i.e., $P(D = 0)$). Such a model is expected to depend on the slope's strength and stiffness and the seismic demand (Bray et al., 2018), which can be represented by $k_y$, $T_s$, and an IM ($Sa(1.3T_s)$ was the considered IM after inspecting different degraded Sa options). Under these considerations, we use a logistic regression (Hosmer Jr et al., 2013) to develop a model that estimates $P(D = 0)$ as a function of $k_y$, $T_s$, and $Sa(1.3T_s)$. The functional form of the predictive models is shown in Equation 2.

$$\ln \frac{P(D=0)}{1-P(D=0)} = c_1 + c_2 \ln k_y + c_3(\ln k_y)^2 + c_4 T_s \ln k_y + c_5 T_s + c_6 \ln Sa(1.3T_s) \tag{2}$$

where $c_1$ to $c_6$ are coefficients, which are shown in Table 1. Figures 8 and 9 compare the $P(D = 0)$ developed in this study against the $P(D = 0)$ model in Bray et al. (2018) and $P(D = 0)$ values estimated directly from the data considering subduction interface and intraslab tectonic settings, respectively. Of note, the $P(D = 0)$ model from Bray et al. (2018) is not strictly applicable to subduction intraslab settings, but it is used here for reference. Figures 8 and 9 show how $P(D = 0)$ increases with the increase of $k_y$, and decreases with the increase of $Sa(1.3T_s)$. In addition, when $T_s$ increases, $P(D = 0)$ decreases initially, and then starts to increase as $T_s$ keeps increasing. This is because the sliding mass approaches a resonance condition when $T_s$ increases from zero, leading to a lower $P(D = 0)$; then, as $T_s$ keeps increasing, the sliding mass deviates from a resonance condition and $P(D = 0)$ increases again. The trends aforementioned for $P(D = 0)$ are consistent with the previous findings in Bray et al. (2018), Bray and Travasarou (2007), and Bray and Macedo (2019). In general, both the BMT18 model and the model developed in this study are consistent. However, the model developed in this study fits better the $P(D = 0)$ derived directly from the data, especially in terms of $T_s$ variations.



(a) (b) (c)

Figure 8: Comparison of the probability of negligible $D$ estimated from the models developed in this study and by Bray et al. (2018) with the $D$ realizations for the interface database

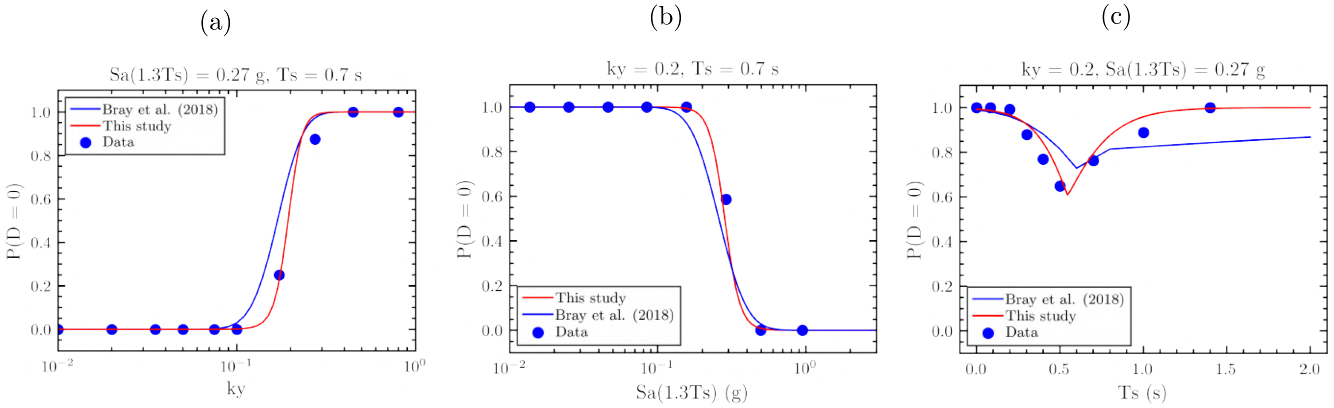


Figure 9: Comparison of the probability of negligible $D$ estimated from the models developed in this study and by Bray et al. (2018) with the $D$ realizations for the intraslab database

Table 1: Estimated coefficients in Equation 2 for interface and intraslab earthquakes

| coefficient | interface | intraslab |
|---|---|---|
| $c_1$ | 3.46 for $T_s < 0.6$; 3.57 for $T_s \geq 0.6$ | 5.22 for $T_s < 0.6$; 2.92 for $T_s \geq 0.6$ |
| $c_2$ | 5.05 for $T_s < 0.6$; 9.39 for $T_s \geq 0.6$ | 6.55 for $T_s < 0.6$; 14.72 for $T_s \geq 0.6$ |
| $c_3$ | 0.15 for $T_s < 0.6$; 0.55 for $T_s \geq 0.6$ | 0.43 for $T_s < 0.6$; 2.24 for $T_s \geq 0.6$ |
| $c_4$ | 1.41 for $T_s < 0.6$; 1.64 for $T_s \geq 0.6$ | 4.73 for $T_s < 0.6$; 5.45 for $T_s \geq 0.6$ |
| $c_5$ | -1.08 for $T_s < 0.6$; 5.37 for $T_s \geq 0.6$ | -0.87 for $T_s < 0.6$; 14.82 for $T_s \geq 0.6$ |
| $c_6$ | -5.13 for $T_s < 0.6$; -7.00 for $T_s \geq 0.6$ | -6.50 for $T_s < 0.6$; -8.47 for $T_s \geq 0.6$ |

### 3.2 Predictive models for estimating non-negligible $D$

Now we develop a model estimating nonzero $D$ values. Towards this end, we use a functional form that is similar to that used by Bray et al. (2018):

$$\begin{aligned}\ln D = a_0 + a_1 \ln k_y + a_2(\ln k_y)^2 + a_3 \ln k_y \ln Sa(1.3T_s) + a_4 \ln Sa(1.3T_s) \\ +a_5(\ln Sa(1.5T_s))^2 + a_6 T_s + a_7 (T_s)^2 + a_8 M + a_9 \ln PGV + \epsilon\end{aligned} \tag{3}$$

in which $a_0$ to $a_9$ are model coefficients and $\epsilon$ is a Gaussian random variable with zero mean and standard deviation of $\sigma$. $Sa(1.3T_s)$ is selected as it leads to the lowest standard deviation compared to other $Sa$, and $PGV$ is added to the functional form as we observed residuals for large $PGV$ values when excluding this term. A truncated regression (Greene, 1981) using the principle of maximum likelihood is used to evaluate the coefficients in Equation 3. Table 2 presents the estimated model coefficients. The values of the coefficients $a_0$, $a_6$, $a_7$, and $a_9$ are modeled as dependent on the values of $T_s$ and $PGV$ based on residual analyses. It can be observed that most of the coefficients in the $D$ models for subduction interface and intraslab earthquake zones show differences, highlighting the different scaling of $D$ across these different tectonic settings. In addition, the standard deviation of the intraslab $D$ model is smaller than that of the interface model, and both models have a standard deviation that is lower than that of the BMT18 model (i.e., 0.73 in ln units). In the case of the interface models, the decrease of the standard deviation is partially attributed to the use of an additional IM, i.e., $PGV$. Figures 10 and 11 show the residuals of Equation 3 (i.e., the difference between $\ln D$ in the data and $\ln D$ predicted by Equation 3) in terms of parameters of interest for both interface and intraslab tectonic settings. The residuals show negligible bias and no significant trends, highlighting the appropriateness of the developed $D$ models.

Table 2: Estimated coefficients in Equation 3 for interface and intraslab earthquakes

| coefficient | interface | intraslab |
|---|---|---|
| $a_0$ | -5.62 for $T_s < 0.1$; -6.20 for $T_s \geq 0.1$ | -5.91 for $T_s < 0.1$; -6.34 for $T_s \geq 0.1$ |
| $a_1$ | -3.26 | -2.36 |
| $a_2$ | -0.36 | -0.22 |
| $a_3$ | 0.48 | 0.26 |
| $a_4$ | 2.62 | 1.97 |
| $a_5$ | -0.12 | -0.02 |
| $a_6$ | -5.25 for $T_s < 0.1$; 2.06 for $T_s \geq 0.1$ | -3.89 for $T_s < 0.1$; 2.31 for $T_s \geq 0.1$ |
| $a_7$ | 0 for $T_s < 0.1$; -0.73 for $T_s \geq 0.1$ | 0 for $T_s < 0.1$; -0.90 for $T_s \geq 0.1$ |
| $a_8$ | 0.19 | 0.38 |
| $a_9$ | 0.62 for $PGV < 10$; 0.73 for $PGV \geq 10$ | 0.68 for $PGV < 30$; 0.72 for $PGV \geq 30$ |
| $\sigma$ | 0.65 | 0.53 |

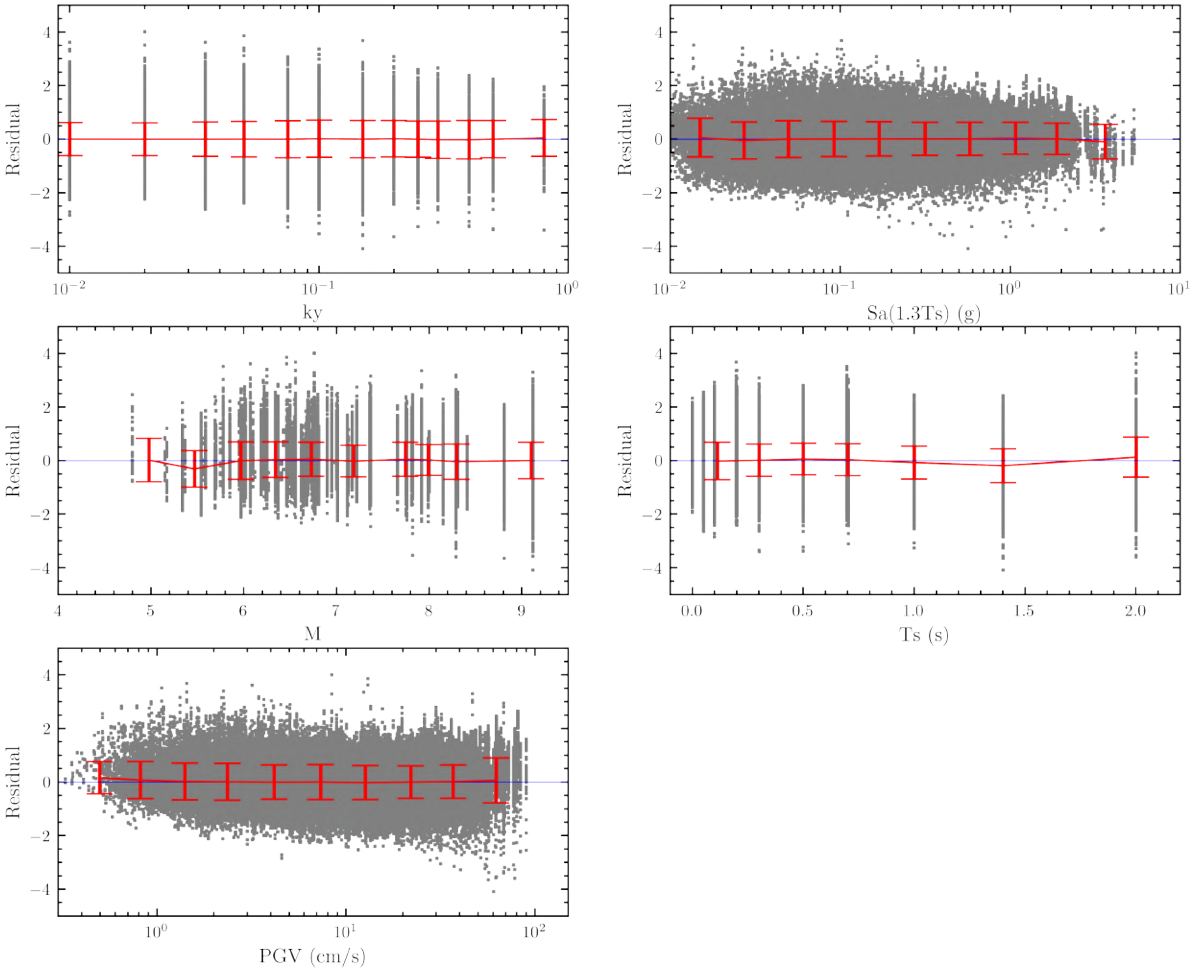


Figure 10: Residuals of the semi-empirical $D$ model for interface earthquakes against $k_y$, $Sa(1.3T_s)$, $M$, $T_s$, and $PGV$

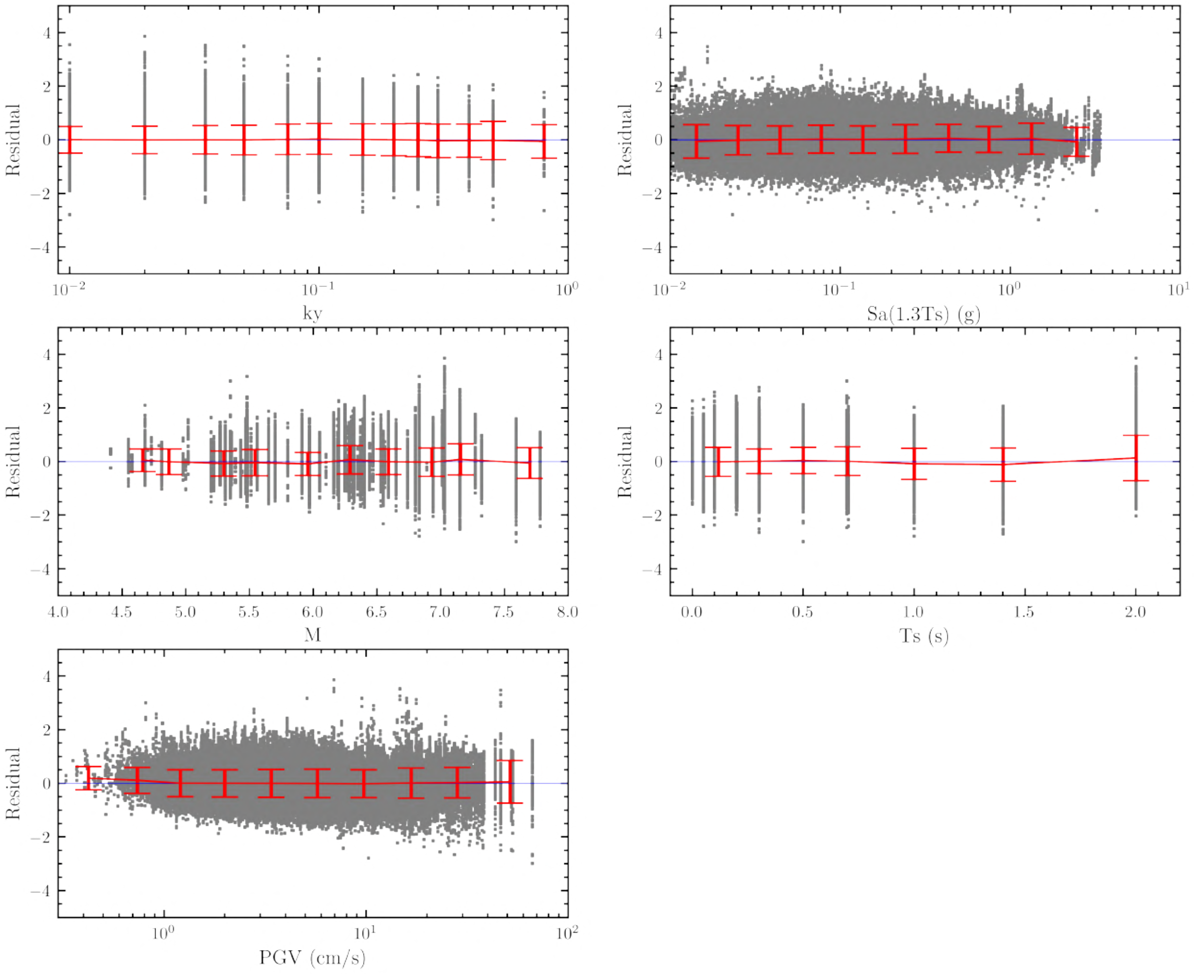


Figure 11: Residuals of the semi-empirical $D$ model for intraslab earthquakes against $k_y$, $Sa(1.3T_s)$, $M$, $T_s$, and $PGV$.

## 3.3 Model Trends

The trends of the developed $D$ models are compared with the BMT18 model trends in Figure 12 considering only the interface tectonic settings as the BMT18 model is not applicable to intraslab settings. Figure 13 also shows general trends for the intraslab model developed in this study. The earthquake characteristics and site conditions considered in these plots are $M = 9$, $R_{rup} = 35$ km, $Z_{TOR} = 30$ km (depth to the top of the earthquake rupture), and $V_{s30} = 760$ m/s for interface earthquakes, and $M = 8$, $R_{rup} = 100$ km, $Z_{TOR} = 100$ km, and $V_{s30} = 760$ m/s for intraslab earthquakes. $Sa(1.3T_s)$ is estimated using the ground motion model (GMM) developed by Abrahamson et al. (2018) and $PGV$ is estimated using the conditional GMMs developed by Liu and Macedo (2021) for subduction zone earthquakes. For Figures 12(c), 12(d), 13(c), and 13(d), $R_{rup}$ is adjusted such that $Sa(1.3T_s)$ and $PGV$ estimated from the GMMs are consistent with the values indicated in these figures. The trends of the models developed in this study are indicated by the solid curves, while the dashed curves represent the trends of the BMT18 model. The general trends in subduction interface tectonic settings are: $D$ increases initially with $T_s$, reaches a maximum value, and then decreases as $T_s$ keeps increasing; $D$ decreases when $k_y$ increases, as expected; and $D$ increases with the increase of $Sa(1.3T_s)$, $PGV$, and $M$. The trends between the subduction interface $D$ model developed in this study and the BMT18 models are generally consistent, but there are also some differences. For example, for $T_s > 0.5$ s and $M = 9$, the BMT18 model produces lower $D$ estimates. In addition, when $Sa(1.3T_s)$ is large (e.g., $Sa(1.3T_s) = 1.4$ g), the BMT18 model produces lower estimates of $D$, while for $M = 9$ and $T_s = 0.1$, it generates higher $D$ estimates.

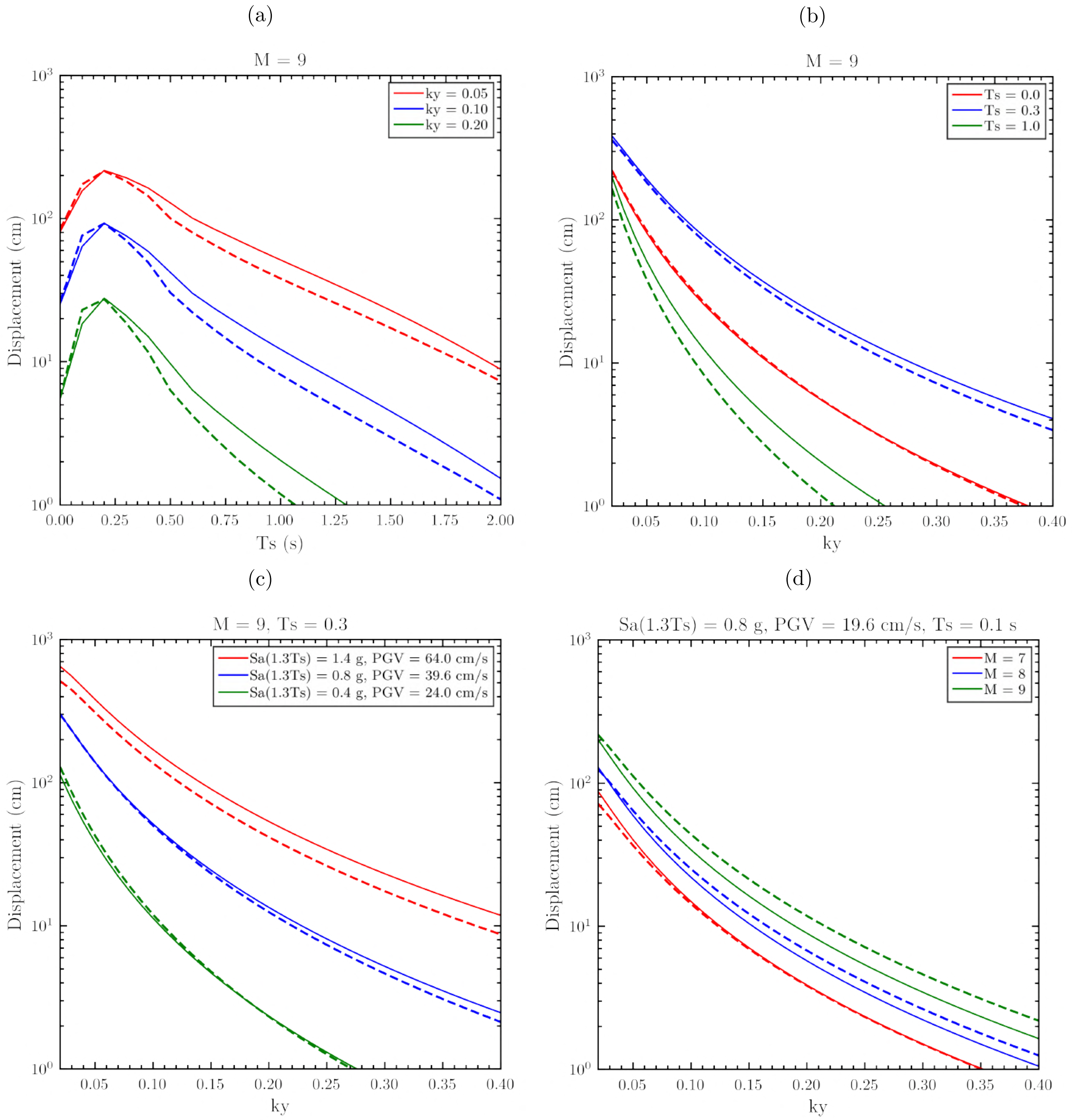


Figure 12: Comparison of the model developed in this study (solid curves) against the BMT18 model (dashed curves) considering interface tectonic settings and variations in $k_y$, $T_s$, $Sa(1.3T_s)$, $PGV$, and $M$.

The general trends for $D$ in terms of $k_y$, $T_s$, $M$, and $S(1.3T_s)$ discussed before for subduction interface settings also apply for subduction intraslab settings (i.e., Figure 13). However, the $D$ estimates from the $D$ model are significantly different from those from the BMT18 model if the same inputs were used (not shown in Figure 13 to avoid confusion). These differences result from the fact that the BMT18 model was developed based on the ground motion recordings from interface earthquakes. This observation has a practical implication as engineers often assume that the BMT18 model is also applicable in subduction intraslab tectonic settings. However, as discussed in this section, this is not the case (see also the residuals in Figure 7). Hence, we recommend using the subduction intraslab $D$ model developed in this study in subduction intraslab tectonic settings.

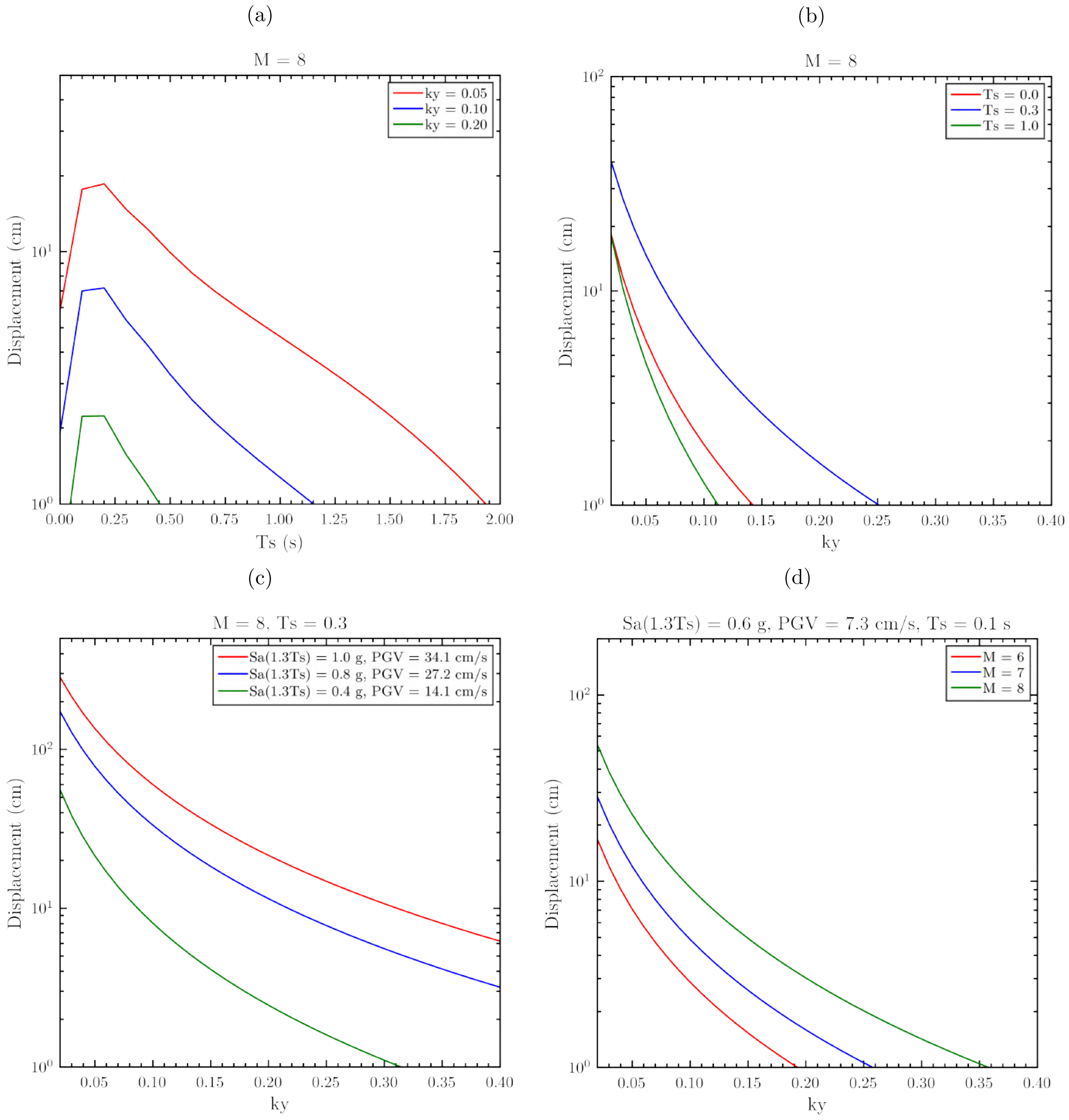


Figure 13: Trends of the model developed in this study considering intraslab tectonic settings and variations in $k_y$, $T_s$, $Sa(1.3T_s)$, $PGV$, and $M$.

## 4 Development of ML-based $D$ models

In this section, we develop several $D$ models using ML approaches, including ridge regression, principal component regression, partial least square regression, random forest, gradient boosting decision tree, support vector regression, and residual neural network. We consider 23 candidate features from slope properties, earthquake parameters, and ground motion IMs for the estimation of $D$, which are presented in Table 3.

Table 3: Features considered in this study for the development of semi-empirical $D$ models*

| Feature | Definition |
|---|---|
| $\boldsymbol{k_y}$ | Slope's yield coefficient |
| $\boldsymbol{T_s}$ | Initial fundamental period of the sliding mass (s) |
| $\boldsymbol{M}$ | Earthquake moment magnitude |
| $\boldsymbol{R_{rup}}$ | Closest distance from the station to the earthquake rupture (km) |
| $\boldsymbol{V_{s30}}$ | Time-averaged shear-wave velocity in the top 30 m of soil (m/s) |
| $T_m$ | Mean period of a ground motion |
| $I_a$ | Arias intensity (m/s) |
| $\boldsymbol{D595}$ | Significant duration (s) of the ground motion |
| $CAV$ | Cumulative absolute velocity (m/s) |
| $\boldsymbol{PGA}$ | Peak ground acceleration (g) |
| $\boldsymbol{PGV}$ | Peak ground velocity (cm/s) |
| $\boldsymbol{Sa(nT_s)}$ | Spectral acceleration (g) at a degraded period equal to $nT_s$ (n = **1.0**, 1.1, 1.2, **1.3**, 1.4, 1.5, 1.6, **1.7**, 1.8, **2.0**, **2.5**, and **3.0**) |

*As illustrated in subsequent sections, all the features are used in forward stepwise regression. Only boldfaced features are used in LASSO and random forest.

We follow the conventional approach in ML to pre-process the database described in Section 2 before the development of $D$ models. We transform the $D$ values and some of the features (i.e., $k_y$, $R_{rup}$, $V_{s30}$, and ground motion IMs) into natural log space and standardize all features to have 0 means and unit standard deviations to facilitate the use of feature selection and regression algorithms. Then we split the database into a training-validation set and a test set. The training-validation set (which contains 85% of the entire database) is used to train the parameters of a given model, whereas the test set (15% of the data) is used to evaluate the model's performance. The training-validation set is further split into ten cross-validation folds for hyperparameter optimization. More details on the cross-validation are provided in the section "Hyperparameter Optimization and Cross-Validation".

## 4.1 Feature selection

We use the forward stepwise regression (FSR, Friedman et al. (2001)), least absolute shrinkage and selection operator (LASSO, Tibshirani (1996)), and random forest (Breiman (2001)) algorithms to select the efficient features from the 23 candidate features listed in Table 3. The selection of features is performed in two steps: first, all features are inspected using FSR, and then a subset is considered for LASSO and random forest, as discussed subsequently. The performance of the feature selection algorithms is evaluated using the mean squared error (MSE) obtained using 10-fold cross-validation on the training-validation set. A brief introduction to these algorithms and the feature selection results is presented in the subsequent sections.

### 4.1.1 Forward Stepwise Regression

FSR is a greedy search algorithm that is commonly used for feature selection. This algorithm starts with a model of zero features and includes one of the remaining features at each step according to some predefined criteria. The feature whose inclusion yields the largest statistically significant improvement to explain $D$ is added to the model. This process is repeated until none of the remaining features improves the explanation of $D$ to a statistically significant extent. Compared to exhaustive search algorithms with optimal solutions that are computationally intractable (often discussed in terms of the non-deterministic polynomial-time hardness, NP-hard, see details in Hochba (1997)), FSR generates a suboptimal solution that is more computationally efficient. Note that the standard FSR algorithm keeps adding a feature to the feature subset obtained from previous steps. To increase the flexibility of the feature selection process, we use a modified floating FSR algorithm, in which features can also be removed if the removal improves the fit, allowing a more robust selection of features (Raschka, 2018). Figure 14 shows the results obtained using the FSR algorithm. The prediction accuracy is represented by the negative MSE averaged over ten cross-validation folds. Interestingly, for both interface and intraslab tectonic settings, the prediction accuracy increases with the addition of features but soon reaches a threshold of -0.14 after approximately five features. This threshold corresponds to potential data "noises" (e.g., outliers, randomness, etc.) and nonlinearity in the database. Using the results in Figure 14 from FSR, we select five features since the improvement of the prediction accuracy is not significant for additional features. These features are $PGV$, $k_y$, $T_s$, $Sa(1T_s)$, and $M$ for interface earthquakes, and $k_y$, $PGV$, $Sa(1.3T_s)$, $Sa(1T_s)$, $T_s$ for intraslab earthquakes.

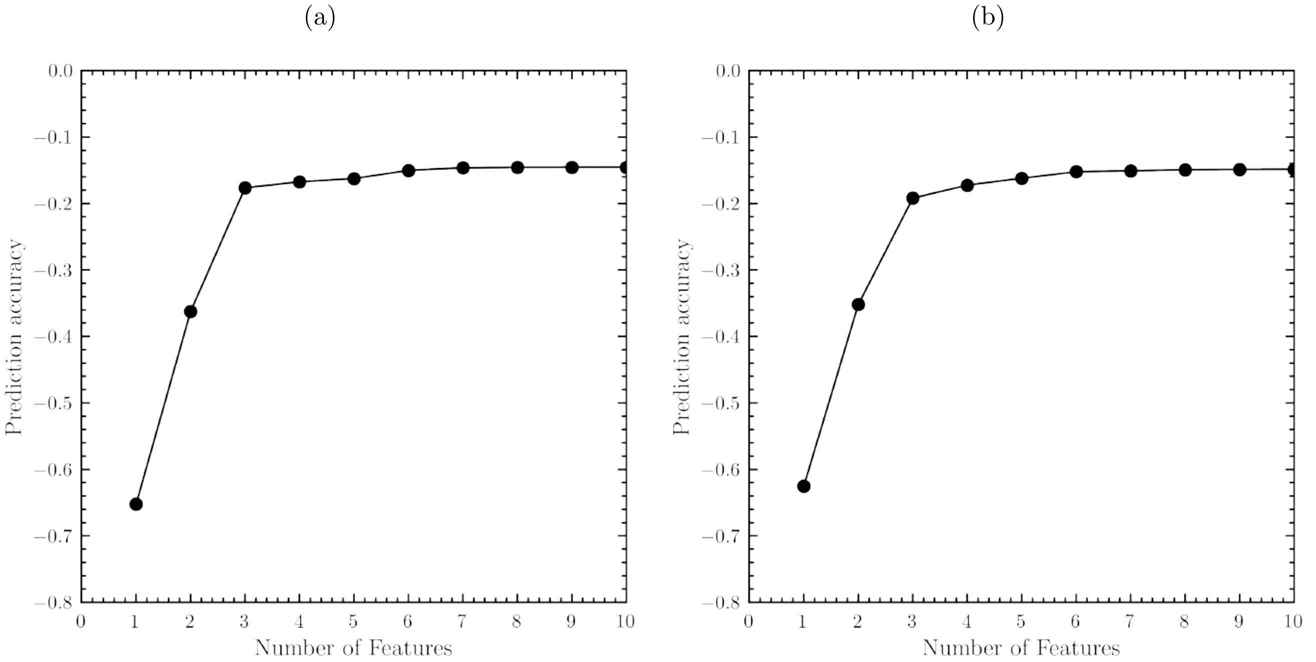


Figure 14: Prediction accuracy as a function of number of features in the FSR algorithm for (a) interface and (b) intraslab tectonic settings

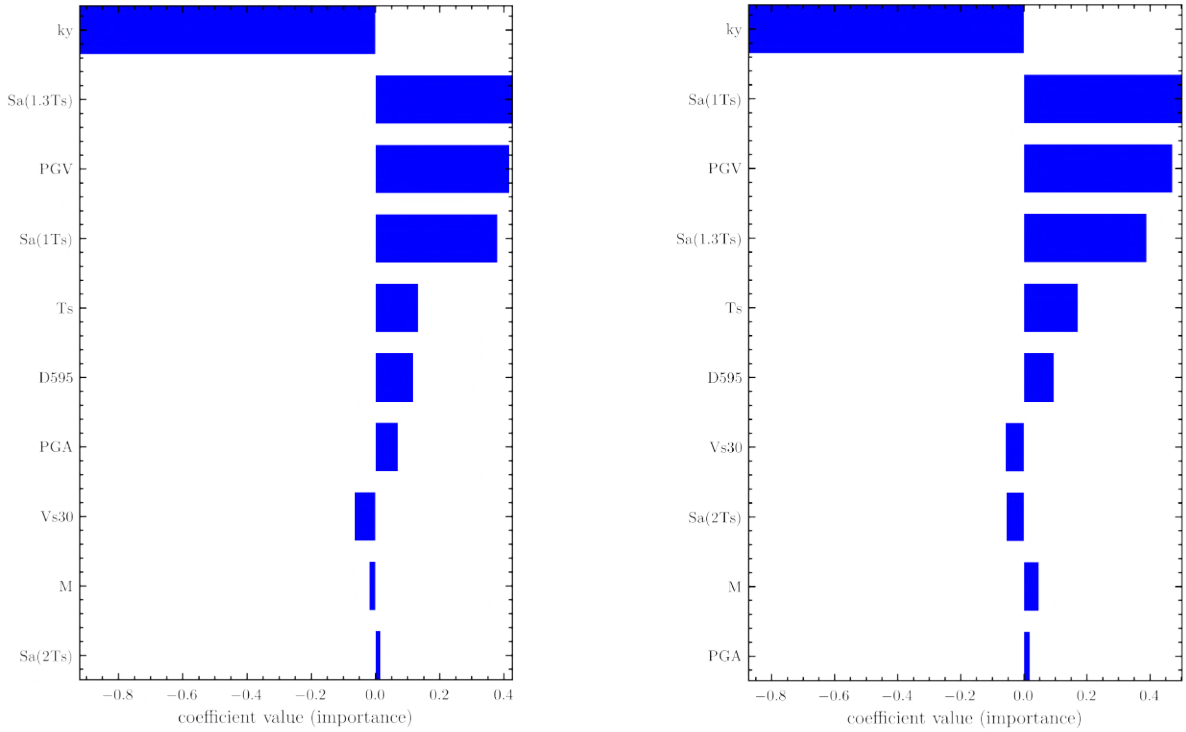
### 4.1.2 LASSO

LASSO is a linear regression algorithm with an applied penalty on its coefficients to learn a model with sparse features. Because LASSO prefers solutions with fewer non-zero coefficients, it can reduce the number of features used in a regression model. Mathematically, LASSO is a modification of the standard linear model with L1 regularization on the objective function as shown in Equation 4.

$$\min \frac{1}{2n}||X\theta - y||_2^2 + \alpha||\theta||_1 \tag{4}$$

where $n$ is the total number of data, $X$ is an $n$ by $d$ matrix with $d$ features, $y$ is an $n$-dimensional column vector of the response variable (i.e., $D$), $\theta$ is a $d$-dimensional row vector of coefficients, and $\alpha$ is a hyper-parameter, which controls the magnitude of the coefficients. $||\theta||_1$ indicates the $\ell_1$ norm of vector $\theta$ and the term $\alpha||\theta||_1$ is called the L1 regularization. LASSO adjusts the value of $\alpha$ to produce the most parsimonious model (model with least features), constrained by minimizing the residual sum of squares. Increasing $\alpha$ leads to a sparser model with fewer coefficients as some coefficients become zero and are eliminated from the model. In this study, $\alpha$ is optimized through a grid search method and the value corresponding to the minimum mean squared error (MSE) averaged across all folds in a 10-fold cross validation is selected. As discussed before, a subset of features (highlighted in Table 3 are used in LASSO as several of them are correlated (see the discussion in Section - Discussion on selected features). Figure

15 shows the top 10 features with the largest absolute values of the coefficients selected by LASSO for the interface and intraslab databases. Because the input features are standardized, their coefficients represent their importance in LASSO. The five features with the highest importance in the interface database are $k_y$, $Sa(1.3T_s)$, $PGV$, $Sa(1T_s)$ and $T_s$, whereas in the case of the intraslab database these features are $k_y$, $Sa(1T_s)$, $PGV$, $Sa(1.3T_s)$ and $T_s$.


(a)

(b)

Figure 15: Importance of the top 10 candidate features estimated using the LASSO algorithm for (a) interface and (b) intraslab earthquakes

### 4.1.3 Random Forest

Random forest randomly splits the data and feature space into several subsets to generate multiple decision trees that reduce the overall variance and return the average results of the constructed trees. In this algorithm, the importance of each feature can be represented by the decrease in the impurity of the split in each tree. Since the random forest algorithm contains several hyperparameters, we use a cross-validated randomized search method to find the combination of hyperparameters that give the minimum MSE averaged over all cross-validation folds (see section "Hyperparameter Optimization and Cross-Validation" for details on randomized search). More details on random forest are provided in Breiman (2001) and later sections.

Similar to LASSO, a subset of features is also considered in random forest for feature selection. Figure 16 shows the top 10 features with the highest importance selected by random forest for interface and intraslab databases. The highest five features are $PGV$, $k_y$, $Sa(1T_s)$, $Sa(1.3T_s)$, and $T_s$, and $k_y$, $PGV$, $Sa(1T_s)$, $Sa(1.3T_s)$ and $T_s$ for interface and intraslab tectonic settings, respectively.

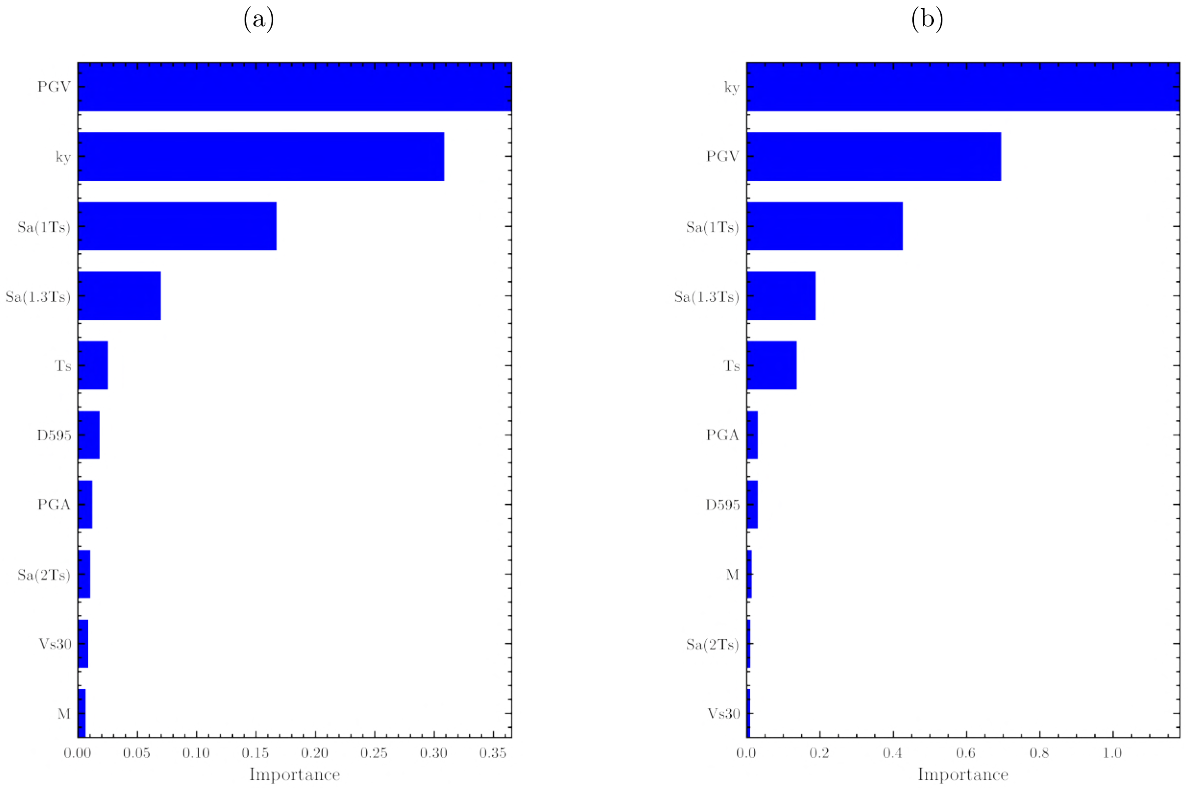


Figure 16: Importance of the top 10 candidate features estimated using the random forest algorithm for (a) interface and (b) intraslab earthquakes

The five most important features selected by the three algorithms to explain $D$ are generally consistent. For both interface and intraslab earthquakes, $k_y$, $PGV$, $Sa(1.3T_s)$, $Sa(1T_s)$, and $T$ are selected by LASSO and random forest, and FSR also selects $M$ as an important feature. Thus there are six potential features to be considered; however, as previously discussed, FSR indicates that there is not a significant increase in performance after five features. Considering that $Sa(1.3T_s)$ and $Sa(1T_s)$ are highly correlated (Macedo and Liu, 2021), it is more efficient to include only one of them as the IMs are expected to carry similar information to explain $D$. Thus, we keep $Sa(1.3T_s)$ instead of $Sa(1T_s)$ and also consider $M$. $Sa(1.3T_s)$ is kept as it is expected to better represent the likely degradation of the initial fundamental period due to soil nonlinearity (Bray and Macedo, 2019). Thus, the final set of selected features is $k_y$, $T_s$, $M$, $Sa(1.3T_s)$, and $PGV$. In terms of prediction accuracy, the selection of these five features has basically the same performance as the features selected directly by the three ML-based methods discussed in this section.

## 4.2 Discussion on selected features

As previously mentioned, in the case of LASSO and random forest, we do not use features that carry similar information to avoid multicollinearity, which follows the same philosophy used in previous studies (e.g., Saygili and Rathje (2008)). In particular, $I_a$, $CAV$, and $T_m$ are not considered. $I_a$ is excluded because it is highly correlated with $PGA$ and because duration ($D595$) is already in the subset. Another motivation for excluding $I_a$ is that its predictability is inferior compared to other IMs (Bray and Macedo (2017)) as its standard deviation in ln units is in the order of 1. $CAV$ is not considered for the same reasons (i.e., high correlation with $PGA$ and the inclusion of $D595$); in addition, as discussed in Macedo et al. (2020a) and Liu and Macedo (2021), $CAV$ is a high frequency parameter, similar to $PGA$. Lastly, $T_m$ is not considered as $PGV$ and spectral acceleration IMs, which also provide information on frequency content (Saygili and Rathje, 2008; Bray and Travasarou, 2007), are already considered. Of note, $T_m$ has been used in some previous studies but mainly in the context of modifying rigid-based models to incorporate the flexibility of a sliding mass in the context of uncoupled assessments (e.g., Rathje and Antonakos (2011), Tsai and Chien (2016)). The models developed in this study are based on coupled analyses.

The final selected features are spectral acceleration at a degraded period (i.e., $Sa(1.3T_s)$), $PGV$, $M$ and slope properties ($k_y$ and $T_s$). As discussed by Bray and Travasarou (2007), a spectral acceleration at a degraded period does directly capture the important ground motion characteristics of intensity and frequency content in relation to the fundamental period of the potential sliding mass, and it indirectly partially captures the influence of duration in that it tends to increase as earthquake magnitude (i.e., duration) increases. In addition, $M$ and $PGV$ bring additional information on frequency content and duration. It is also worth highlighting that IMs based on spectral accelerations and $PGV$ have the advantage that there are available robust ground-motion models (GMMs) for their estimation in subduction zones. For instance, the NGA-Sub project (Bozorgnia et al. (2021)) has developed a suite of models for spectral accelerations and $PGV$ that can facilitate performance-based applications. Lastly, the selected IMs are amenable to be implemented in performance-based engineering assessments using vector PSHA (e.g., Liu et al. (2021)).

## 4.3 ML-based models for estimating non-negligible $D$

In this section, we propose seven different ML-based predictive $D$ models. The ML-based procedures discussed in this section are selected based on their popularity in the ML community (Bonaccorso, 2018) and also based on the recommendations by Macedo et al. (2021). The five features selected in the previous section are used as inputs to the considered ML-based procedures, which are described in the following sections. Specifically, the ML procedures include ridge regression, principal component regression, partial least square regression, random forest, gradient boosting decision trees (GBDT), support vector regression (SVR), and residual neural network (ResNet). It is essential to highlight here that some of these procedures (i.e., principal component regression, partial least square regression, GBDT, and random forest) have been used before in shallow crustal tectonic settings (Wang et al., 2020; Macedo et al., 2021). In contrast, we are not aware of previous efforts using ridge regression or SVR to develop $D$ models. In addition, the ResNet model used in this study is highly customized and tailored (e.g., residual mapping, batch normalization), as discussed subsequently.

### 4.3.1 Hyperparameter Optimization and Cross Validation

The ML procedures used in this study contain one or more hyperparameters that need to be tuned. To facilitate a robust and consistent process for optimizing hyperparameters, we use two methods - grid search and randomized search - that aid in the optimization. In the grid search method, grids of different hyperparameter values are created, and the performance of each grid point is evaluated. This method is applied to ML procedures that are computationally efficient and contain few hyperparameters (e.g., ridge regression, principal component regression, partial least square regression, and support vector regression). In the case of ML procedures that use a large number of hyperparameters or are computationally expensive, we adopt the randomized search method. Like the grid search, the randomized search also creates grids of different hyperparameters values. However, instead of evaluating the performance of all the grid points, only the performance of a fixed number of uniformly sampled grid points is evaluated. In addition, we use 10-fold cross-validation to assess the performance of different hyperparameters. Specifically, the training-validation set is first split into ten folds with similar amounts of data. Then for each set of hyperparameters, the ML model is trained on nine folds of the data and validated on the remaining fold to calculate the MSE. This process is repeated 10 times such that the model is tested on each fold, and the MSE averaged over all folds is used as the performance indicator. The hyperparameter values corresponding to the lowest averaged MSE are selected for the ML model. Details of hyperparameter settings for specific algorithms are summarized in each subsequent section and in Table 4.

### 4.3.2 Ridge Regression with Polynomial Feature Expansion

Most previous studies have extensively used polynomial models as the functional forms that develop semi-empirical $D$ models. However, the selection of polynomial terms has typically been based on semi-manual procedures and residual analyses, which are, to some extent, qualitative. As opposed to manually selecting polynomial terms, in this study, we keep all second-order polynomial terms for a $D$ model (i.e., constant, interaction, squared terms, etc.) and use the ridge regression technique to reduce the risk of overfitting. We first transform the five features into 20 polynomial features to form a linear model, then we standardize each feature and apply the ridge penalty (L2-regularization) to solve the regression coefficients. The ridge regression is quite similar to LASSO and uses the objective function in Equation 5.

$$\min \frac{1}{2n}||X\theta - y||_2^2 + \alpha||\theta||_2 \tag{5}$$

where $||\theta||_2$ indicates the $\ell_2$ norm of vector $\theta$ and the term $\alpha||\theta||_2$ is called the L2 regularization. It can be observed from Equation 5 that the only difference between LASSO and ridge regression is the regularization term ($\alpha||\theta||_2$). In LASSO, the $\ell_1$ norm of $\theta$ is used while the $\ell_2$ norm is used in the ridge regression. The $\alpha$ term in the ridge regression is known as the complexity parameter, and it controls the amount of shrinkage in the coefficients. When $\alpha$ increases, the coefficients shrink and the resulting model is less prone to overfitting even when some features are correlated (e.g., $PGV$ and $Sa(1.3T_s)$).

We developed two $D$ models using the ridge regression technique: one with five selected input features and another with four features (i.e., excluding $PGV$). Both models have polynomial functional forms and hence can be used to represent traditional $D$ models using one or two IMs. We optimized the hyperparameter $\alpha$ using the grid search method previously discussed. For this purpose, a grid of $\alpha$ values ranging from $10^{-8}$ to 1 with a $10^{-1}$ increment was considered. The best $\alpha$ value estimated from 10-fold cross-validation for interface and intraslab tectonic settings is presented in Table 4.

### 4.3.3 Principal Component Regression

Principal component regression (PCR, Jolliffe (1982)) is a method used to model a response variable when the number of input variables is large and/or when those variables are highly correlated. PCR performs dimensionality reduction on the data using the principal component analysis (PCA). Specifically, principal components are first constructed by projecting the input and response variables into a new space. Then, linear regression is performed between those principal components and the response variable. PCR contains a hyperparameter that determines the number of principal components used in PCA. We apply the grid search method to optimize the number of principal components. A grid of values (i.e., 1, 2, 3, 4, and 5, since the maximum number of features is five) is considered, and the results indicate that four components are optimal for both the interface and intraslab databases.

### 4.3.4 Partial Least Square Regression

Partial least square regression (PLSR) (Geladi and Kowalski, 1986) is a method like PCR, which also performs dimensionality reduction on the data using PCA. Similarly, PLSR also uses linear regression to model the relationship between the principal components constructed using PCA and the response variable. We apply the grid search method with values of 1, 2, 3, 4, and 5 to optimize the number of principal components to be used. The optimized number of components is 3 and 2 for the interface and intraslab databases, respectively. It can be seen that PLSR requires a smaller number of principal components than PCR. This difference between PLSR and PCR originates from the methodologies for constructing principal components. PCR explores options for the appropriate principal components that reflect the variability in the input variables; however, PLSR constructs the components to explain the variations in both the input and the response variables. Since PLSR takes into account the variability in the response variable, it often leads to a more parsimonious model with fewer components (e.g., Geladi and Kowalski (1986)).

### 4.3.5 Random Forest

Random forest is an ensemble learning method that combines the prediction of multiple estimators (decision trees) to decrease the variance of the resulting model without increasing its bias. The random forest algorithm utilizes a modified bootstrap aggregating algorithm to generate multiple tree learners. For each tree learner, a subset of the data is randomly sampled with replacement (bootstrapping) for training, then at each split of the tree, a subset of the features is randomly selected. Finally, the prediction results are averaged over all tree learners (aggregating). This process is schematically illustrated in Figure 18 for our data. Because the data and features are randomly sampled, the tree learners are uncorrelated, and the random forest algorithm is less sensitive to the noise in the training set.

Figure 17

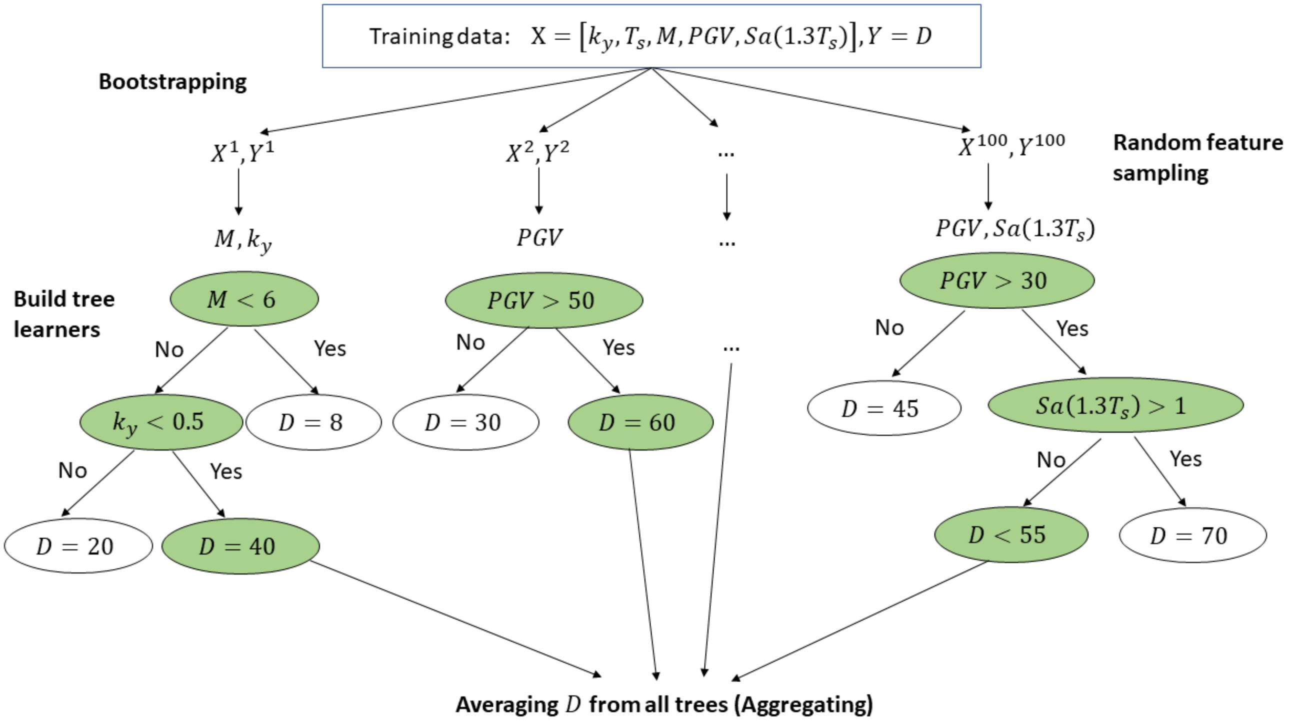


Figure 18: Schematic illustration of the random forest algorithm with 100 trees. The training data is firstly sampled 100 times with replacement; then for each sample data, a subset of features are randomly sampled (bootstrapping); next, 100 decision trees are built for the 100 data samples independently; finally the predictions of $D$ values from the 100 trees are averaged (aggregating).

The random forest algorithm contains a dozen of hyperparameters to tune. In practice, only three of them are considered crucial to the performance of random forest (Probst et al., 2019), namely, the number of trees, the maximum depth of each tree, and the maximum number of features to consider when splitting a node (more detailed definitions on these hyperparameters are available in Probst et al. (2019)). We create a grid with the number of trees ranging from 50 to 400 with interval increments of 10; the maximum depth ranging from 20 to 100 with interval increments of 5 (we also included the option of unlimited depth); and the maximum number of features ranging from 1 to 5 with an interval increment of 1. Under these considerations, we applied the randomized search method previously described with 100 samples, finding that the best combination of hyperparameters for the subduction interface and intraslab tectonic settings were 320 and 350 trees, maximum depths of 65 and 50, and a maximum number of features of 2 and 3, respectively.

### 4.3.6 Gradient Boosting Decision Trees

Gradient boosting decision trees (GBDT, Friedman (2001)) is another ensemble learning method that is similar to a random forest. The significant difference between GBDT and random forest is the training procedure of the trees. GBDT combines multiple weak tree learners into a single strong learner in an iterative manner. The GBDT model starts with a weak tree learner. A new tree learner is added at each iteration using the residual between the previous learner and the observed response variable. The algorithm terminates when reaching a prescribed maximum number of iterations or when the loss function (see Chen and Guestrin (2016)) does not decrease significantly. Finally, the predictions are summed up by all tree learners.

We use the randomized search method to optimize the hyperparameters in the GBDT method. The considered hyperparameters are the maximum depth of each tree, the boosting learning rate, the subsample ratio, the subsample ratio of columns when constructing each tree, the subsample ratio of columns for each level, and the number of trees (for more details on the definition of these hyperparameters, refer to Chen and Guestrin (2016); Friedman (2001)). The corresponding optimal values for these parameters are 35, 0.1, 0.6, 0.6, 0.7, and 650; and 30, 0.1, 0.7, 0.6, 0.6, and 700 for interface and intraslab tectonic settings, respectively.

### 4.3.7 Support Vector Regression with Kernel Approximation

The support vector regression (SVR, Smola and Schölkopf (2004)) is an extension of the support vector machine method (Cortes and Vapnik, 1995) often used in classification problems. In SVR, a hyperplane is fitted to the data after defining an acceptable error threshold. In contrast to other conventional regression algorithms that use the MSE as the loss function, SVR's objective is to minimize the model coefficients. At the same time, the error term is imposed as a minimization constraint (Smola and Schölkopf, 2004). The optimization problem solved by SVR is shown in Equations 6 and 7.

$$\min \frac{1}{2}||\theta||_2^2 + C\sum_{i}^{n} |\xi_i| \tag{6}$$

$$\text{subject to } |y_i - \theta x_i| \leq \epsilon + |\xi_i| \text{ for } i = 1 : n \tag{7}$$

where $\theta$ is a $d$-dimensional vector with $d$ being the total number of features, $n$ is the total number of observations in the data, $y_i$ is the $i$-th observation of the response variable, and $x_i$ is a vector that contains the features that go along with the $i$-th observation. $\epsilon$ is a hyperparameter that controls the maximum error margin between the observed response variable and the fitted hyperplane. $\xi_i$ is the slack variable for the $i$-th observation that represents its deviation from the error margin. $C$ is a regularization hyperparameter representing the tolerance for observations outside the error margin $\epsilon$. The input features for SVR are often transformed into a high-dimensional space through a kernel trick (Amari and Wu, 1999) or reproducing the kernel Hilbert space (Wahba et al., 1999). However, a kernel trick (especially for a nonlinear kernel) is not scalable for a large database like the one used in this study. Hence, we use the Nystroem method (Williams and Seeger, 2001), which is a general method for low-rank approximations of kernels, with a linear kernel SVR to approximate the nonlinear kernel SVR. The Nystroem method uses a subset of the large training data as the basis to approximate a kernel mapping from the basis onto the original large training data (more details are in Williams and Seeger (2001)). Hence, the computational cost of SVR is affected by the size of the subset, which is much smaller than the original training data. The Nystroem approximation uses three hyperparameters: the kernel, gamma (kernel-specific parameter), and the target dimension of the transformed features. After an extensive examination of different combinations of hyperparameter values, we find the 2nd-order polynomial kernel with a gamma (which is related to the radius of the influence of a single training data point) of 0.2 and 300-dimensional transformed features are efficient for both interface and intraslab tectonic settings. The hyperparameters $\epsilon$ and $C$ are optimized using the grid search method. We consider a grid of $\epsilon$ ranging from 0 to 1 with a 0.1 interval and a grid of $C$ ranging from 0.1 to 10 with a 0.2 interval. The optimal values of $\epsilon$ and $C$ found by the grid search method are 0.4 and 0.7; and 0.3 and 0.7 for the interface and intraslab tectonic settings, respectively.

### 4.3.8 Residual Neural Network

Artificial neural networks have been extensively used in classification problems with unstructured databases in computer vision and natural language processing (LeCun et al., 1995, 2015; He et al., 2016; Mikolov et al., 2010). However, the application of neural networks in regression problems is less frequent than in classification problems. We are only aware of the study by Cho et al. (2022) that examined the use of neural networks (by using a multi-layer perceptron) for estimating $D$. In this study, we use the fully connected (FC) residual neural network (ResNet, He et al. (2016)) to develop regression models for estimating $D$, and we also discuss differences with the Cho et al. (2022) study. The proposed architecture of the neural network used in this study is shown in Figure 19.

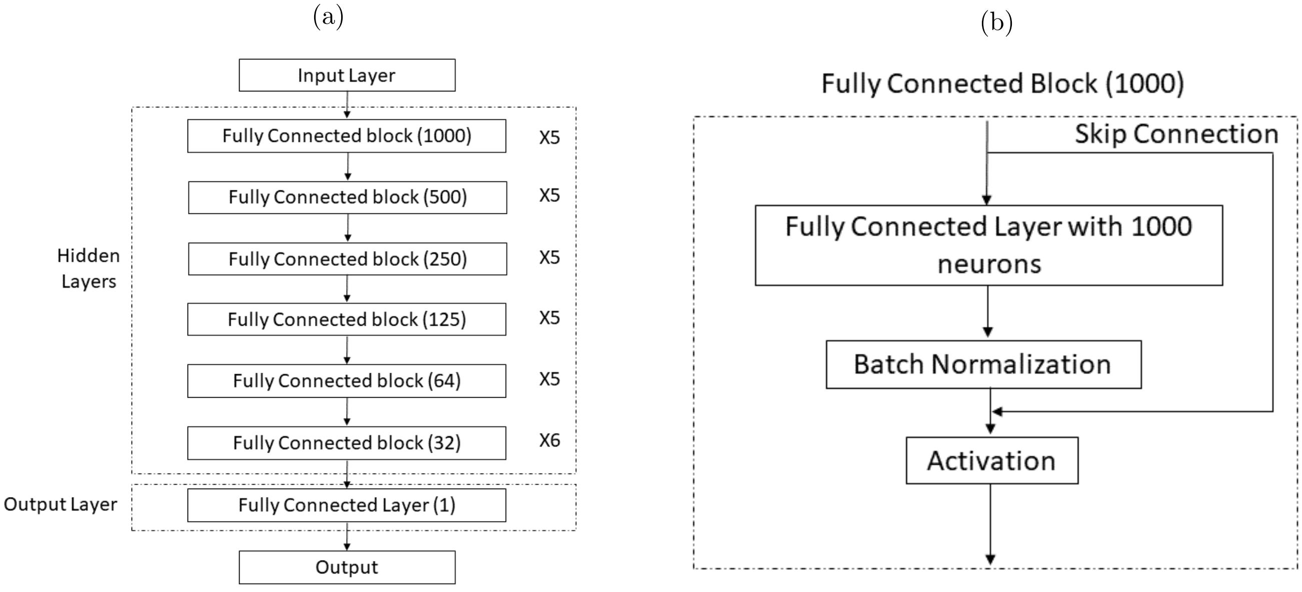


Figure 19: Illustration of the architectures of the residual neural network (‘X5’ and ‘X6’ in (a) indicate that block is duplicated by 5 and 6 times, respectively).

In contrast to a vanilla FC neural network such as the multi-layer perceptron (e.g., Cho et al. (2022)), we introduce several modifications to the neural network to improve its prediction performance. The ResNet contains one input layer, 31 hidden layers, and one output layer, as shown in Figure 19(a). The input layer represents the five features previously selected, the output layer is an FC layer that combines the output from the hidden layers and outputs a scalar (i.e., $D$ values), and the hidden layers contain 31 FC blocks. Figure 19(b) illustrates the architecture of an FC block with 1000 neurons. This block consists of an FC layer, a batch normalization (BN) layer (Ioffe and Szegedy, 2015), a skip connection or shortcut, and an activation function. The FC layer has 1000 neurons, and each neuron combines the output from the previous layer with its weight parameters and generates a scalar; hence, the output of this FC layer is a 1000-dimensional vector. Then the BN layer normalizes (re-center and re-scale) the 1000-dimensional vector before it goes through the activation function. The activation function is used to add nonlinearity to the neural network. As its name suggests, the skip connection is used to skip the FC and BN layers such that the FC block can learn the residual mapping. To illustrate this point, suppose we denote the input as $x$, the function represented by the FC and BN layers as $F(\cdot)$, and the hypothesis that needs to be learned from the FC block as $H(\cdot)$. A FC block without a skip connection aims to find $F(\cdot)$ that is as close as possible to $H(\cdot)$ (loosely speaking $H(x) = F(x)$). In contrast, the skip connection directly feeds the input $x$ into the activation function and the FC block learns an $F(\cdot)$ that satisfies $H(x) = F(x) + x$. Hence, $F(x) = H(x) - x$, and the FC and BN layers in an FC block learn from the residuals between $H(x)$ and $x$, which is denominated as the residual neural network (ResNet) that we are using. If the input $x$ has a different dimension than the output of the FC layer (e.g., the first block has an input dimension of 5 and an output dimension of 1000), a linear projection can be used to match the dimensions according to:

$$H(x) = F(x) + W_s x \tag{8}$$

where $W_s$ is a matrix that acts as a dimension reduction agent to match the dimension of $x$ with $F(x)$.

To complete the ResNet architecture, we examine different layouts and choices for the activation function using our database. We divide the training-validation set into a training set (80% of the data) and a validation set (20% of the data) and evaluate the performance of different ResNet architectures based on the MSE on the validation set. We find that: (1) a deep neural network performs better than a shallow neural network (e.g., one based on perceptrons); (2) adding the skip connection helps to stabilize the training process; (3) the use of BN layers significantly improves the performance; (4) the use of BN layers before the activation layer has better predictive performance than the option of adding BN layers after the activation in estimating $D$, (5) the use of dropouts (randomly drop some of the neurons to prevent the co-adaptation of neurons) results in a degradation of the network performance possibly due to the probabilistic nature of dropouts, (6) after examining different activation functions (sigmoid, tanh, rectified linear unit - ReLU, and leaky ReLU), we found that ReLU has the best performance.

The most significant differences between the neural network developed in this study and that developed in Cho et al. (2022), which is the only previous study we are aware of using neural networks for estimating $D$, are three-fold. First, our neural network is deep. In contrast to perceptrons with few layers, our neural network contains 33 layers with 6.7 million parameters. By including more layers, the neural network can learn features at various levels of abstraction and can approximate more complicated nonlinear functions, enhancing its generalization ability (LeCun et al., 2015; Bishop, 2006). In contrast, a neural network with few layers requires much more (exponentially larger) neurons in each layer to achieve similar performance as compared with a deep neural network, which is computationally inefficient. Second, we include a batch normalization layer before each activation function. This mitigates the internal covariate shift often observed in feed-forward neural networks (e.g., Ioffe and Szegedy (2015)), which occurs when the output of the previous layer is used as the input of the next layer. Hence, when the weight parameters of a previous layer are updated, the distribution of their output is also changed, affecting the output distribution of subsequent layers and making the training process unstable and slow. Batch normalization helps mitigate these issues, stabilizing and accelerating the training process of a neural network. In addition, batch normalization makes the neural network less sensitive to the learning rates and initialization of weights, thus, simplifying the hyperparameter tuning process.

Lastly, we include skip connections in each layer. It has been well recognized that the training process becomes much more difficult when the number of layers is large in a neural network due to the vanishing gradient problem (Hochreiter, 1998), in which the update of weight parameters is significantly slower due to very small gradients (i.e., partial derivatives of the loss function against weight parameters). Very deep neural networks such as the ResNet network used in this study could show a non-decreasing training error if skip connections are not considered. Skip connections enable the neural network layers to learn identity mapping (i.e., skip some layers) and mitigate the vanishing gradient problem.

The previous study by Cho (2020b) pointed out that the performance of neural networks was insensitive to the number of layers. To investigate this observation, we have considered deep and single-layer networks with the same number of neurons (100+, 1000+, and 10000+) but excluding skip connections and batch normalization (similar to the networks used by Cho (2020b)). The performance (MSE on the validation set) we observed was not as robust as the ResNet model. In addition, we did not observe a significant difference in the performance for the deep and shallow networks, which is consistent with the findings in Cho et al. (2022) where neither skip connections nor batch normalization were considered. The comparable performance of deep and shallow networks observed by Cho (2020b) may result from potential challenges in the training process caused by the internal covariate shift and vanishing gradient problem. It is worth emphasizing again the value of the batch normalization and skip connections used in this study to improve the performance of deep networks.

We use the grid search method to optimize the hyperparameters of the ResNet model. Because the training of ResNet is computationally expensive, instead of performing 10-fold cross-validation, we split the training-validation set into a training set and validation set. We train the ResNet on the training set and evaluate the performance of the hyperparameters based on the MSE on the validation set. The considered grids of hyperparameters include a learning rate (the tuning parameter in the optimization algorithm that determines the step size at each iteration while moving towards minimizing the loss function) ranging from $10^{-1}$ to $10^{-6}$ with a $10^{-1}$ interval, a batch size in $[30, 300, 500, 1000, 5000, 10000, 30000]$, and different optimizers (e.g., stochastic gradient descent, Adam, and RMSProp, see LeCun et al. (2015) for more details). We found that a batch size of 1000 and the Adam optimizer with a learning rate of 0.01 yield the best performance (lowest MSE on the validation set) for interface and intraslab earthquakes. To facilitate the training process at larger epochs (one epoch is when the entire training data is passed forward and backward through the neural network once), the learning rate is decayed by 10% every 100 epochs. Rather than training the ResNet for a fixed number of epochs, we apply an early stopping criterion of 100 epochs, in which the ResNet terminates training if its performance is not significantly improved over the last 100 epochs. Under these considerations, the training of ResNet terminates at 1210 epochs and 1510 epochs in the interface and intraslab databases, respectively.

## 4.4 Overfitting Prevention

Overfitting usually occurs when a model contains more adjustable parameters than an optimal model. (Bishop, 2006). It is nontrivial to directly compare the complexity of a model with the optimal model, as the latter is not known a-priori. However, some measures can be taken to prevent overfitting. We adopt three major strategies during the training of the proposed ML models to minimize the risk of overfitting: regularization, hyperparameter optimization, and early stopping. As discussed in previous sections, for the ridge regression and SVR models, we apply regularization to control the complexity of the derived models. The regularization parameters in these models are optimized based on the values that yield minimal cross-validation error. For the principal component and partial least square regression, we select the best number of principal components based on their best performance in cross-validation. In terms of the random forest and GBDT models, we reduce their potential for overfitting by tuning hyperparameters, e.g., number of trees, maximum tree depth, etc. These hyperparameters affect the model complexity and are optimized to the values producing minimal cross-validation error. In addition, we also allow the option of automatically pruning some trees to reduce the model complexity further. Lastly, for the ResNet model, we apply early stopping criteria (see Section - Residual Neural Network) to prevent the overtraining of the neural network. We have also examined the dropout approach to reduce the number of neurons, which seemed unnecessary (see Section - Residual Neural Network). As a final check of overfitting, we have monitored the training and validation error of the models during cross-validation and their test errors. Table 4 summarizes the settings of hyperparameters considered for each ML-based $D$ model.

Table 4: Considered range and optimal values of the hyperparameters for different ML-based $D$ models developed in this study

| Model | Hyperparameter range (interface and intraslab) | Optimized hyperparameters |
|---|---|---|
| Ridge Regression | $\alpha = \{10^{-8}, 10^{-7}, \ldots, 10^{-1}\}$ | $10^{-6}$ for both interface and intraslab |
| Ridge Regression with one IM ($Sa(1.3T_s)$) | $\alpha = \{10^{-8}, 10^{-7}, \ldots, 10^{-1}\}$ | $10^{-6}$ (interface) and $10^{-5}$ (intraslab) |
| PCR | number of components = $\{1, 2, 3, 4, 5\}$ | 4 (interface) and 4 (intraslab) |
| PLSR | number of components = $\{1, 2, 3, 4, 5\}$ | 3 (interface) and 2 (intraslab) |
| Random Forest | number of trees = $\{50, 60, 70, \ldots, 400\}$,<br>maximum depth = $\{20, 25, 30, \ldots, 100, \infty\}$,<br>maximum number of features = $\{1, 2, 3, 4, 5\}$ | number of trees: 320 (interface) and 350 (intraslab),<br>maximum depth: 65 (interface) and 50 (intraslab),<br>number of features: 2 (interface) and 3 (intraslab) |
| GBDT | maximum depth = $\{5, 10, \ldots, 50\}$,<br>learning rate = $\{10^{-5}, 10^{-4}, \ldots, 10^{0}\}$,<br>subsample ratio = $\{0.1, 0.2, \ldots, 1\}$,<br>subsample ratio for column (each level) = $\{0.1, 0.2, \ldots, 1\}$,<br>subsample ratio for column (each tree) = $\{0.1, 0.2, \ldots, 1\}$,<br>number of trees = $\{100, 150, \ldots, 800\}$ | maximum depth = 35 and 30,<br>learning rate = 0.1 and 0.1,<br>subsample ratio = 0.6 and 0.7,<br>subsample ratio for column (each level) = 0.6 and 0.6,<br>subsample ratio for column (each tree) = 0.7 and 0.6,<br>number of trees = 650 and 700, for interface and intraslab, respectively |
| SVR | $\epsilon = \{0, 0.1, 0.2, \ldots 1\}$,<br>$C = \{0.1, 0.3, 0.5, \ldots, 10\}$ | $\epsilon = 0.4$ (interface),<br>$\epsilon = 0.3$ (intraslab),<br>$C = 0.7$ (interface and intraslab) |
| ResNet | learning rate = $\{10^{-6}, 10^{-5}, \ldots, 10^{-1}\}$,<br>batch size = $\{30, 300, 500, 1000, 5000, 30000\}$ | learning rate = 0.01 and batch size = 1000 for both interface and intraslab |

# 5 Performance of ML-based $D$ models

In this section, we investigate the residuals, predictive performance, computational cost, trends, and case history validation performance of the different ML-based and traditional models previously developed. Figure 20 shows the residuals (in ln units) of the GBDT model against $k_y$, $M$, $T_s$, $Sa(1.3T_s)$, and $PGV$ for the interface database. Figure 21 shows similar residual plots of the GBDT model for the intraslab database. It can be observed that there are no trends in the residuals. Similar residual plots for the traditional $D$ models were already presented in section 2 (Figures 10 and 11). Additional residuals plots for other ML models are shown in Appendix A.

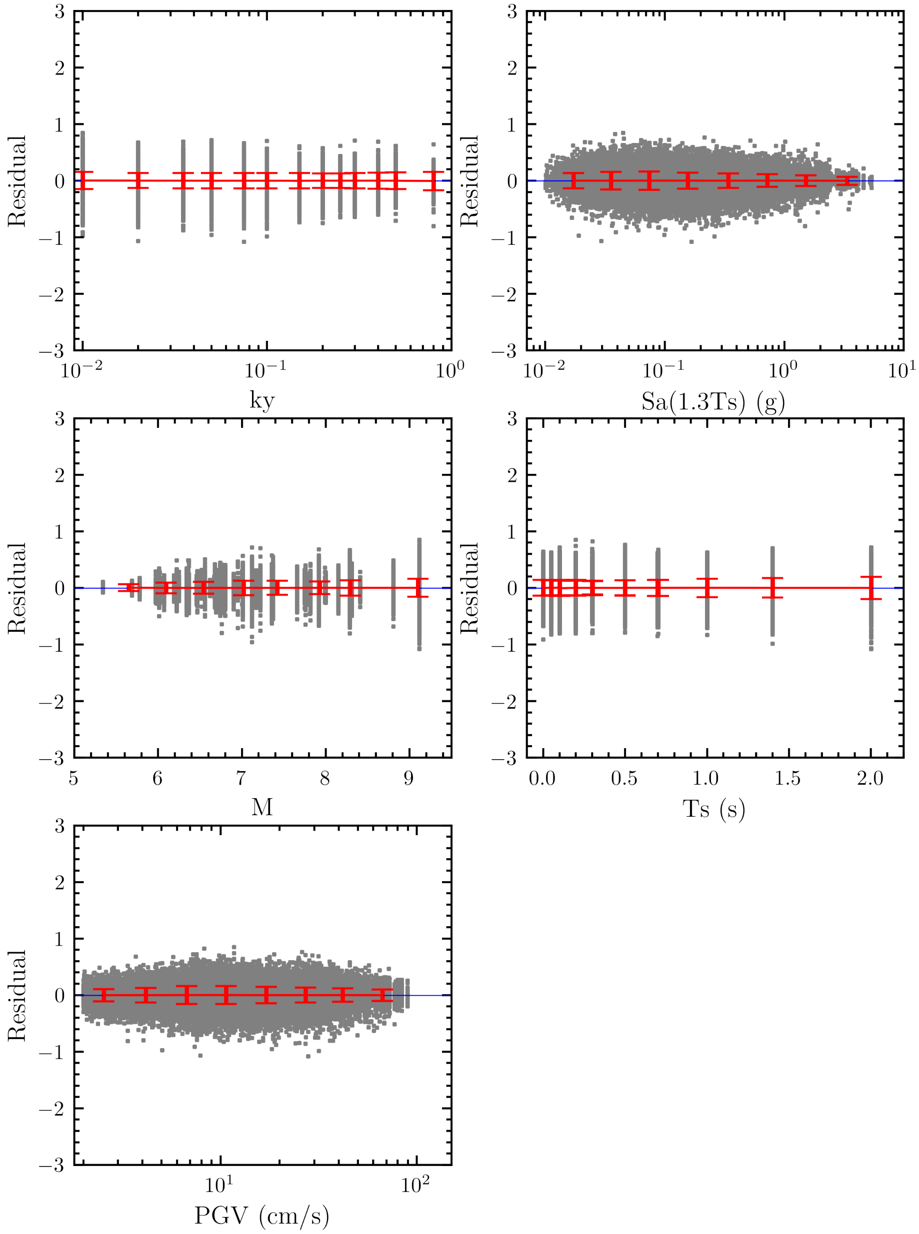


Figure 20: Residuals (ln units) of the GBDT models against $k_y$, $M$, $T_s$, $Sa(1.3T_s)$, and $PGV$ for the interface database (The error bars show the mean values ± one standard deviation of the residuals in each bin).

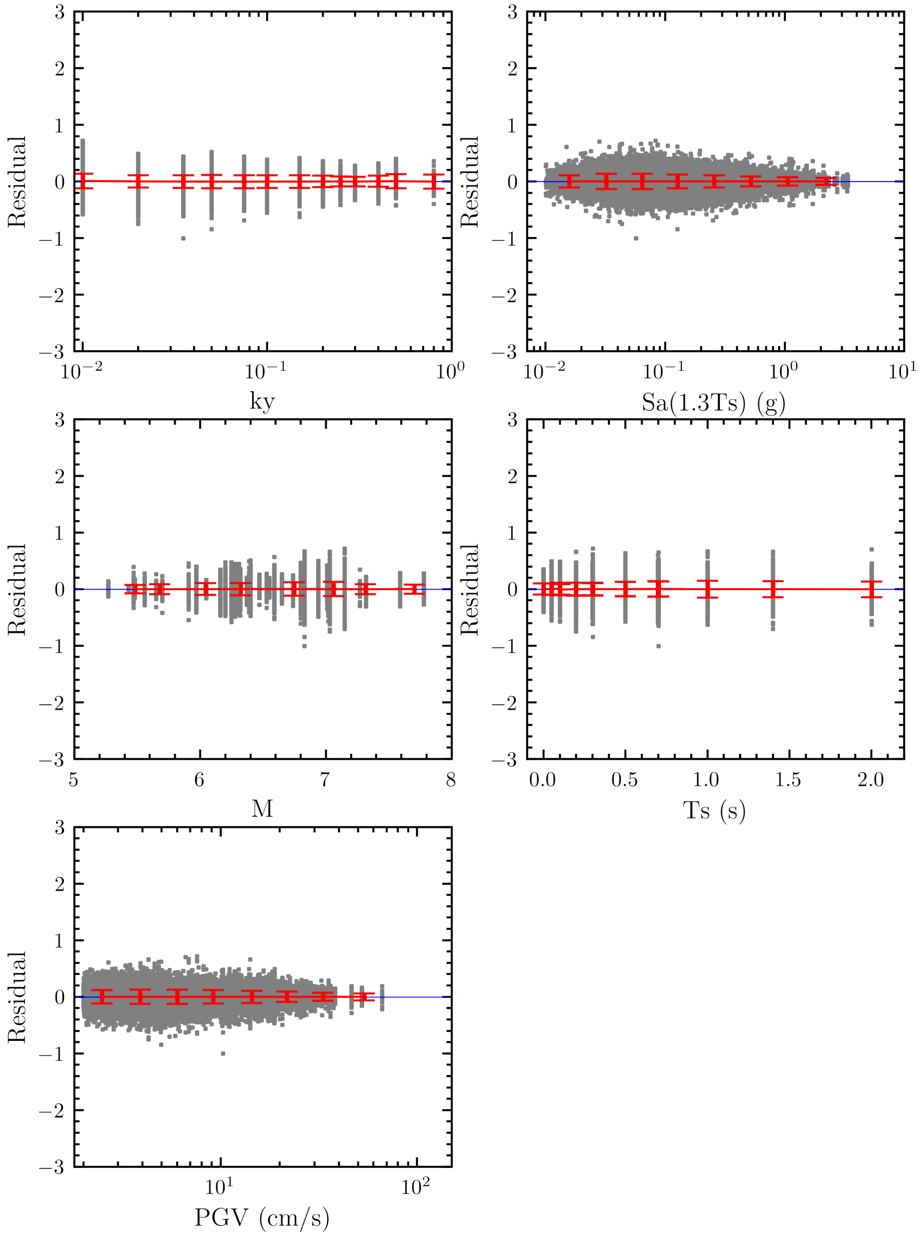


Figure 21: Residuals (ln units) of the GBDT models against $k_y$, $M$, $T_s$, $Sa(1.3T_s)$, and $PGV$ for the intraslab database (The error bars show the mean values ± one standard deviation of the residuals in each bin).

The predictive performance of the different ML and traditional models implemented in the previous sections is represented by their root mean squared error (RMSE) on the test set. The RMSE can be calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - f_i)^2}{n}} \tag{9}$$

where $n$ is the total number of data in the test set, $f_i$ is the $D$ value for the $i$th data point predicted by the $D$ models, and $y_i$ is the actual $D$ value for the $i$th data point. Table 5 presents the comparison of the predictive performance of the derived ML models. The ridge regression models (with one or two IMs, as discussed in Section 4) are used to represent generic traditional $D$ models as they have polynomial functional forms. Compared to the ridge regression models, most of the other developed ML-based models are able to capture better the complex relationship between $D$, the slope properties, and IMs, showing a better predictive performance on test tests (i.e., lower RMSE values, see Table 5); hence, they outperform the expected performance of traditional models. The RMSE values of the BMT18 model and the traditional models developed in this study (denoted as ML22) are also included in Table 5 as a reference. BMT18's RMSE is somewhat larger than the ridge regression model using two IMs. This is attributed to the fact that the BMT18 model used only one IM and was developed using a different database. However, the performance of the BMT18 model is comparable to the ridge regression model using one IM. The RMSE difference between the BMT18 and the ridge regression model with two IMs is also comparable to the reduction observed in other $D$ models that use one or two IMs (e.g., Saygili and Rathje (2008)). However, other ML-based models offer a further reduction in the RMSE. The GBDT model exhibits the best performance for both interface and intraslab databases by showing the lowest RMSE. This is consistent with previous findings that highlight the good performance of the GBDT model in different problems with structured data, showing a high prediction accuracy (Bishop, 2006; Wang et al., 2020).

The performance of ResNet and random forest is similar, with random forest performing better on the interface database and ResNet performing better on the intraslab database. This is consistent with the no free lunch theorem, which states that there is no single best model for all databases (problems) in ML. One interesting fact to be noted is that although the ResNet model contains a larger number of parameters (we do not observe overfitting of ResNet during training and validation), its performance is inferior to that of GBDT for the problem discussed in this study. There are several potential reasons to explain this observation. First, the number of input features for ResNet is relatively small. Neural networks are most effective for problems with unstructured data (e.g., images, signals, etc.) that contain a large number of features. Hence, the structured database with a relatively small number of features used in this study may not fully exploit the performance of ResNet. This observation would also apply to the previous study by Cho et al. (2022) for shallow crustal tectonic settings. In contrast, Jha et al. (2019) showed that ResNet was better than other ML models in material discovery problems where 200+ features are used. Second, the performance of neural networks scales with the number of data; hence, the size of the database used in this study may not be large enough compared to other deep learning problems with millions or even billions of observations. One potential solution is to use a smaller neural network with fewer layers and parameters; however, we observed degradation in the predictive performance compared to a deeper neural network. A larger database could be used in future studies to inspect the performance of the neural network used in this study. Lastly, the number of training epochs (i.e., training time) may also affect the performance of the ResNet network. Without much dependence on the learning rate, the performance of the ResNet model increases rapidly in the first few hundred epochs but soon slows down in the following epochs. In our study, the training process is terminated when the decrease in validation RMSE is less than 1% in the last 100 epochs. With more computational resources (the specifications of the computer we used are described shortly), much larger training epochs could be investigated in future studies to evaluate if the performance of the ResNet model is improved.

In terms of the SVR model, it achieves intermediate performance among the considered models by using the Nystroem kernel approximation. We also tested the performance of a vanilla linear SVR without any approximation, being inferior to the derived SVR model, confirming the kernel approximation's effectiveness. However, it is essential to highlight that the kernel approximation represents a compromise between computational efficiency and prediction accuracy as the direct application of SVR with nonlinear kernels is computationally intractable in most modern computers. Also, note that the Nystroem approximation is one of the few kernel approximation methods currently available, and there is active ongoing research on kernel approximations for SVR. Thus, the performance of SVR could be further improved when more sophisticated kernel approximation methods are available in the future.

Although both PCR and PLSR perform linear regression on dimension-reduced data, the performance of the PLSR model is significantly better than the PCR model, despite using fewer principal components. The performance of the PCR is even slightly worse than the traditional ML22 model. This is because the PLSR model constructs principal components that capture the variation of both the input features and the response variable. In contrast, the PCR model constructs four components that only capture the variation of the five input features. In addition, the four components constructed by PCR are unable to capture all the information contained in the original five features.

Lastly, the standard deviation of $D$ estimates is an input for performance-based implementations (e.g., Liu et al. (2021), Macedo et al. (2020b)). Traditionally, this standard deviation is estimated as the standard deviation of the model residuals in the training data. However, this estimation could be biased and not reflect the uncertainty of the prediction of future unseen data (Wang et al., 2020). Hence, in this study, we follow the recommendations in Wang et al. (2020) and use the standard deviation of the model residuals in the test set, which are expected to better represent the uncertainty of $D$ in prediction problems. In this context, it can be seen from Equation 9 that the

RMSE value of a model is mathematically equivalent to the standard deviation of its residuals. Hence, the RMSE values in Table 5 are recommended as the standard deviations of the ML models for their use in performance-based slope displacement assessments.

Table 5: Predictive performance (model standard deviations) and training cost of different $D$ models developed in this study and previous studies

| Model | RMSE (interface) | Training time (interface) | RMSE (intraslab) | Training time (intraslab) |
|---|---|---|---|---|
| Ridge Regression | 0.587 | 20 s | 0.510 | 19.2 s |
| Ridge Regression with one IM | 0.741 | 19 s | 0.655 | 18 s |
| PCR | 0.778 | 2.3 s | 0.636 | 2.2 s |
| PLSR | 0.564 | 6.3 s | 0.412 | 5.7 s |
| Random Forest | 0.367 | 7501 s | 0.310 | 6701 s |
| GBDT | 0.280 | 4012 s | 0.249 | 3919 s |
| SVR | 0.516 | 9313 s | 0.448 | 10113 s |
| ResNet | 0.373 | 213102 s | 0.299 | 197121 s |
| ML22 | 0.698 | 1.2 s | 0.584 | 1.1 s |
| BMT18 | 0.730 | - | - | - |

In addition to evaluating the performance of the ML-based models using the RMSE on the test sets, it is also informative to compare the training and prediction cost for each model. The training time (including hyperparameter optimization) of each model is presented in Table 5. The models developed in this study are trained on a machine with a 10-core CPU (i.e., Intel Core i9-10900K) and 64 GB RAM, except for the ResNet model, which is trained on the Tesla P100 GPU with 12 GB memory. The prediction time on the test set is negligible for all models, and it is not reported. In general, more complex models have a longer training time. For example, the ResNet model, which has the larger number of parameters, has the longest training time. The model with the best performance (i.e., GBDT) has an intermediate training time among all the models. Considering the trade-off between training time and predictive performance, the random forest, GBDT, PLSR, and SVR models can be considered for estimating $D$.

In the following, we illustrate typical trends for the GBDT model considering both subduction interface (Figure 22) and intraslab (Figure 23) earthquakes. Similar figures considering the other ML-based models previously discussed are shown in Appendix B and trends for the traditional models were presented in Figures 12, and 13. The earthquake characteristics and site conditions considered for generating the trends are: $Z_{TOR}$ (depth to the top of the rupture model) = 30 (interface) and 60 (intraslab) km, $V_{s30} = 760$ m/s, and $R_{rup} = 30$ km. The $Sa(1.3T_s)$ value is estimated using the BC Hydro ground motion model (GMM) (Abrahamson et al., 2018) and the $PGV$ value is estimated using the conditional $PGV$ GMM developed by Liu and Macedo (2022). For Figures 22(c)(d) and 23(c)(d), we adjust the $R_{rup}$ values such that $Sa(1.3T_s)$ estimated from the BC Hydro GMM is consistent with the values shown in the figures. Because the tree-based models (i.e., GBDT and random forest) provide discrete $D$ estimates given continuous input features, we use the Savitzky-Golay filter (Press and Teukolsky, 1990) to smooth the $D$ trends, as the motivation is to show general trends. In general, all the ML-based models derived in this study show similar trends: (1) $D$ increases with $T_s$ and then decreases when $T_s$ keeps increasing (2) $D$ decreases with the increase of $k_y$ (3) $D$ increases with the increase of $M$ or $Sa(1.3T_s)$, when other features are fixed. These observations are also consistent with previous studies (e.g., Bray et al. (2018)).

Finally, we evaluate the performance of the developed models against the case histories compiled in Bray et al. (2018) for subduction interface tectonic settings. We also searched for case histories after recent subduction intraslab tectonic settings. However, we did not find cases in the public domain with the information required (i.e., ground motion information, material properties, and observed performance) to assess the proposed models.Table 6 presents the details of the case histories, which are associated with the observed seismic performance of different slope systems that were affected by subduction interface earthquakes. We have used the mean absolute error (MAE) to evaluate the performance of the ML and traditional models on the case histories, which is calculated as:

$$MAE = \frac{\sum_{i=1}^{N_{ch}} |D_i - D_i'|}{N_{ch}} \tag{10}$$

where $N_{ch}$ is the total number of case histories (i.e., 12 for interface). $D_i$ and $D_i'$ (both in units of cm) are the observed and predicted slope displacement for an ML model, respectively.

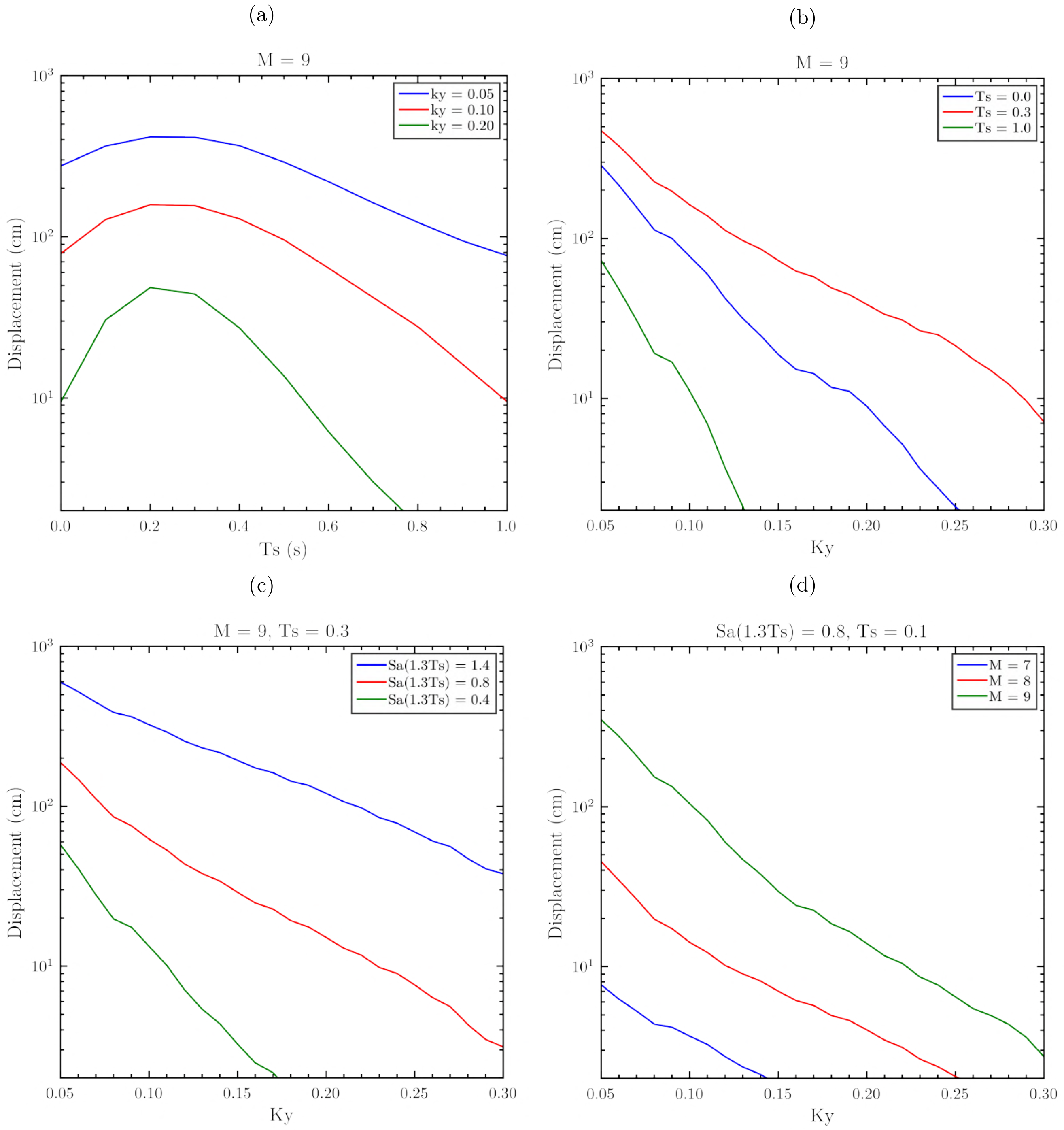


Figure 22: Trends for the variations of $k_y$, $T_s$, $Sa(1.3T_s)$, and $M$ for GBDT for interface earthquakes

We use the same GMMs used in evaluating the model trends to estimate $PGV$ and $Sa(1.3T_s)$, required for estimating

$D$. Table 7 presents the MAEs for the different ML and traditional models considered in this study when they are evaluated against case histories. Figures 24 and 25 show the comparison of predicted $D$ values from the ML and traditional models with observed $D$ values in the case histories from interface earthquakes (the bars in Figures 24 and 25 correspond to 95% confidence intervals). All the ML models have MAE values that vary between 23 and 31. The GBDT, random forest, Resnet, SVR, and ridge regression models predict a good seismic performance (small $D$) when the observed performance is good, and they also predict potential damage (large $D$) when the observed $D$ values are large. The ML22 model shows a comparable MAE compared with most of the aforementioned ML models, but as discussed before, the statistical performance of most ML models is superior. Lastly, the PCR and PLSR models have quite large MAE values associated with poor performance for most case histories with small observed $D$ values.

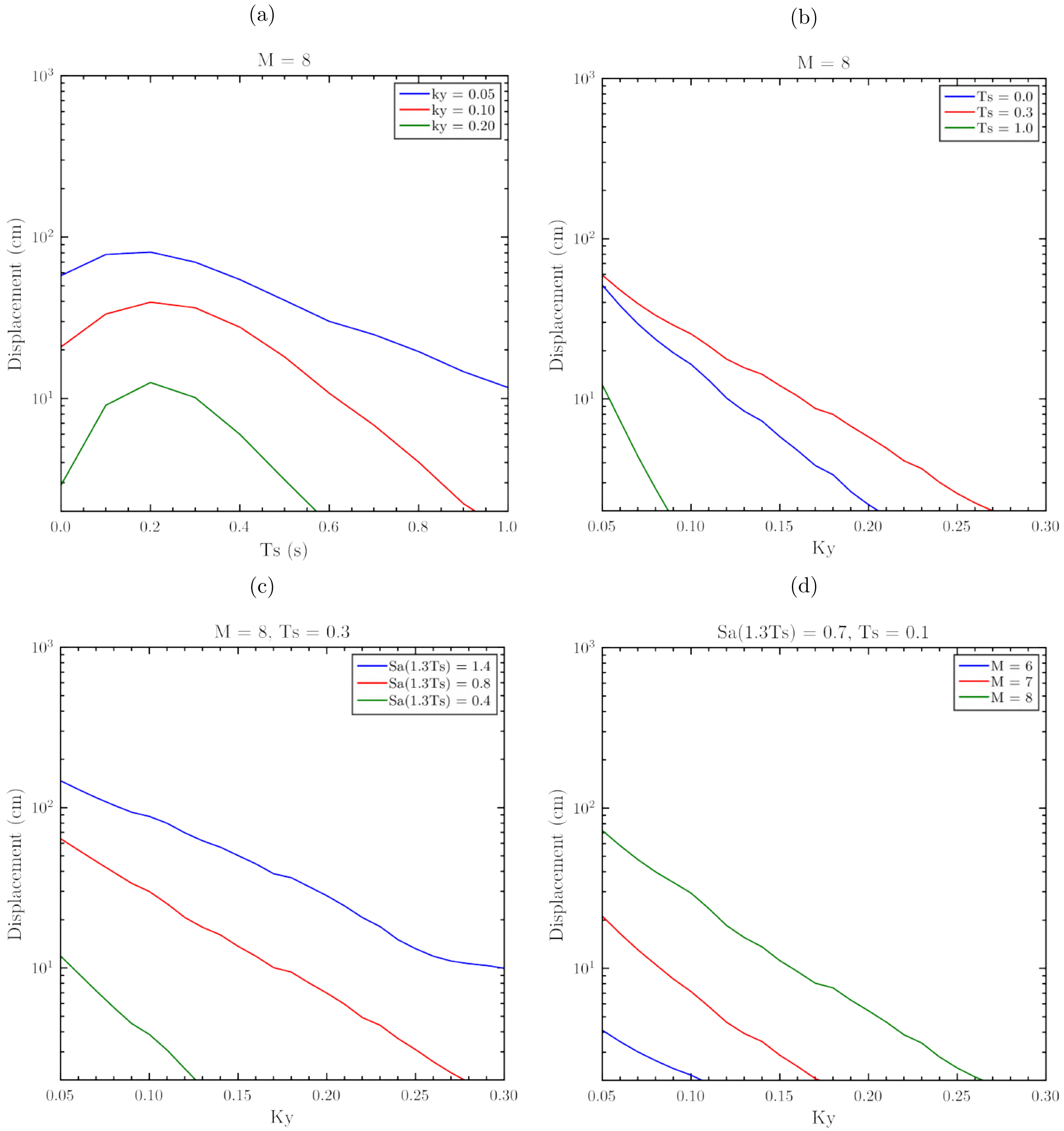


Figure 23: Trends for the variations of $k_y$, $T_s$, $Sa(1.3T_s)$, and $M$ for GBDT for intraslab earthquakes

Table 6: Characteristics of the case histories of observed slope displacements ($D_{obs}$) affected by subduction interface earthquakes. PP = 2007 Peru/Pisco; PM = 2001 Peru/Moquegua; EM = 2016 Ecuador/Muisne; CM = 2010 Chile/Maule; JT = Japan/Tohoku; S3, S4, and S5 are from Elgamal et al. (1990)

| Slope system | Earthquake | $k_y$ | $T_s$ | $M$ | $V_{s30}$ | $R_{rup}$ | $Sa(1.3T_s)$ | $PGV$ | $D_{obs}$ |
|---|---|---|---|---|---|---|---|---|---|
| Coastline Slope Peru | PP | 0.1 | 0.6 | 8 | 760 | 37 | 0.308 | 25.201 | 6 |
| Yuracmayo Dam | PP | 0.27 | 0.45 | 8 | 400 | 125 | 0.158 | 9.539 | 0 |
| La Villita dam | S3 | 0.2 | 0.6 | 7.4 | 360 | 80 | 0.227 | 14.023 | 1 |
| La Villita dam | S4 | 0.2 | 0.6 | 7.2 | 360 | 23 | 0.402 | 30.528 | 1.4 |
| La Villita dam | S5 | 0.2 | 0.6 | 8 | 360 | 40 | 0.482 | 33.824 | 4 |
| Torata dam | PM | 0.13 | 0.65 | 8.4 | 760 | 100 | 0.269 | 18.841 | 5 |
| Esperanza dam | EM | 0.24 | 0.4 | 7.8 | 400 | 45 | 0.510 | 24.62 | 0 |
| Tutuven dam | EM | 0.39 | 0.15 | 8 | 400 | 40 | 0.802 | 28.518 | 0 |
| Nishigo dam | JT | 0.26 | 0.15 | 9 | 350 | 90 | 1.514 | 25.315 | 40 |
| Shitoki dam | JT | 0.29 | 0.4 | 9 | 760 | 52 | 0.483 | 27.605 | 0 |
| Surikamigawa dam | JT | 0.3 | 0.68 | 9 | 500 | 90 | 0.257 | 21.235 | 0 |
| Coihueco dam | CM | 0.1 | 0.25 | 8.8 | 400 | 65 | 1.463 | 31.543 | 350 |

Table 7: Predictive performance of different ML and traditional $D$ models in case histories from interface earthquakes

| Model | MAE (interface) |
|---|---|
| Ridge Regression | 24.52 |
| Ridge Regression with one IM | 22.11 |
| PCR | 30.34 |
| PLSR | 29.01 |
| Random Forest | 23.47 |
| GBDT | 24.23 |
| SVR | 23.91 |
| ResNet | 23.46 |
| ML22 | 24.88 |

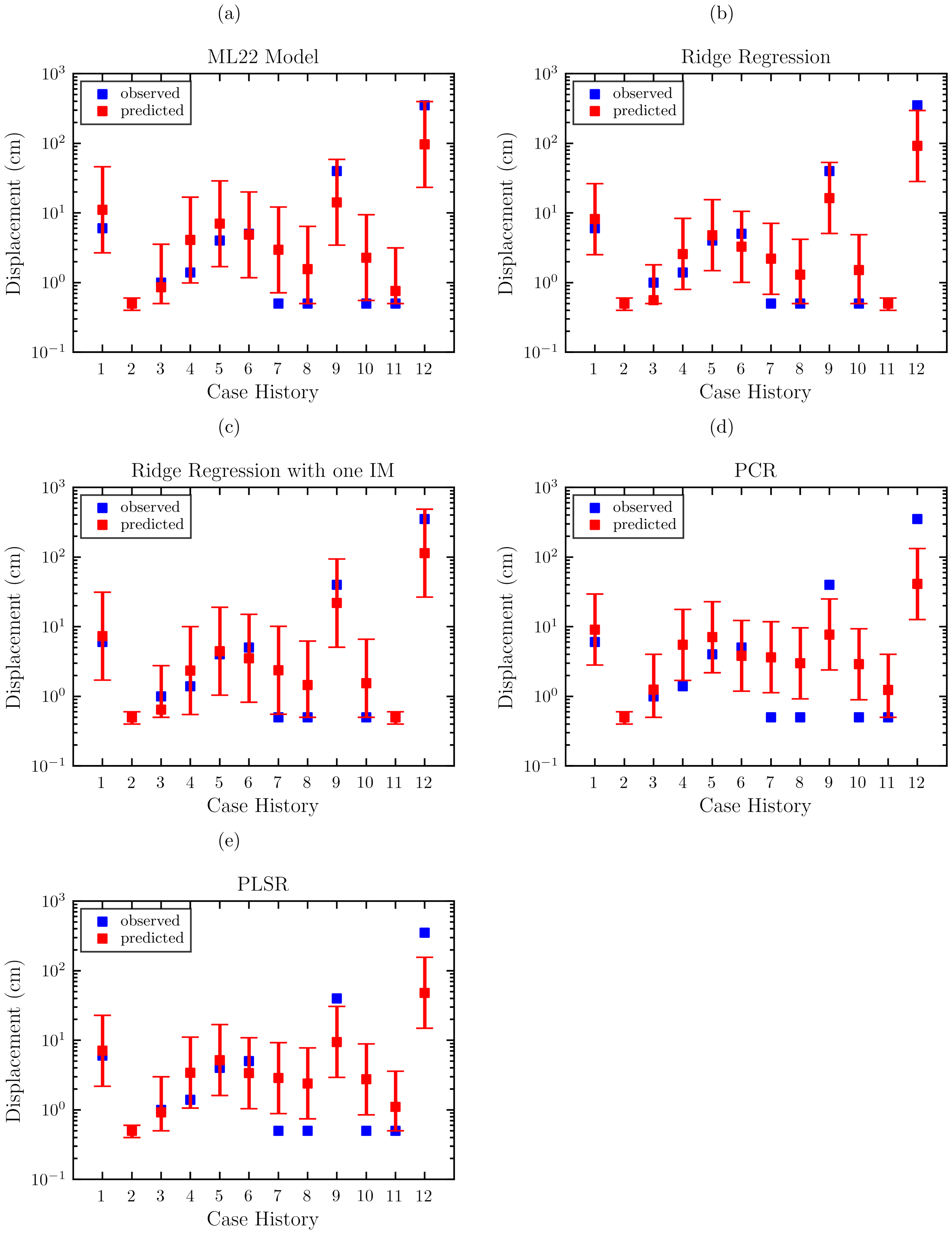


Figure 24: Performance of the ML models on interface case histories (see Table 6 for the case histories details). Note that zero $D_{obs}$ values are shown as 0.5 cm for plotting on log scale.

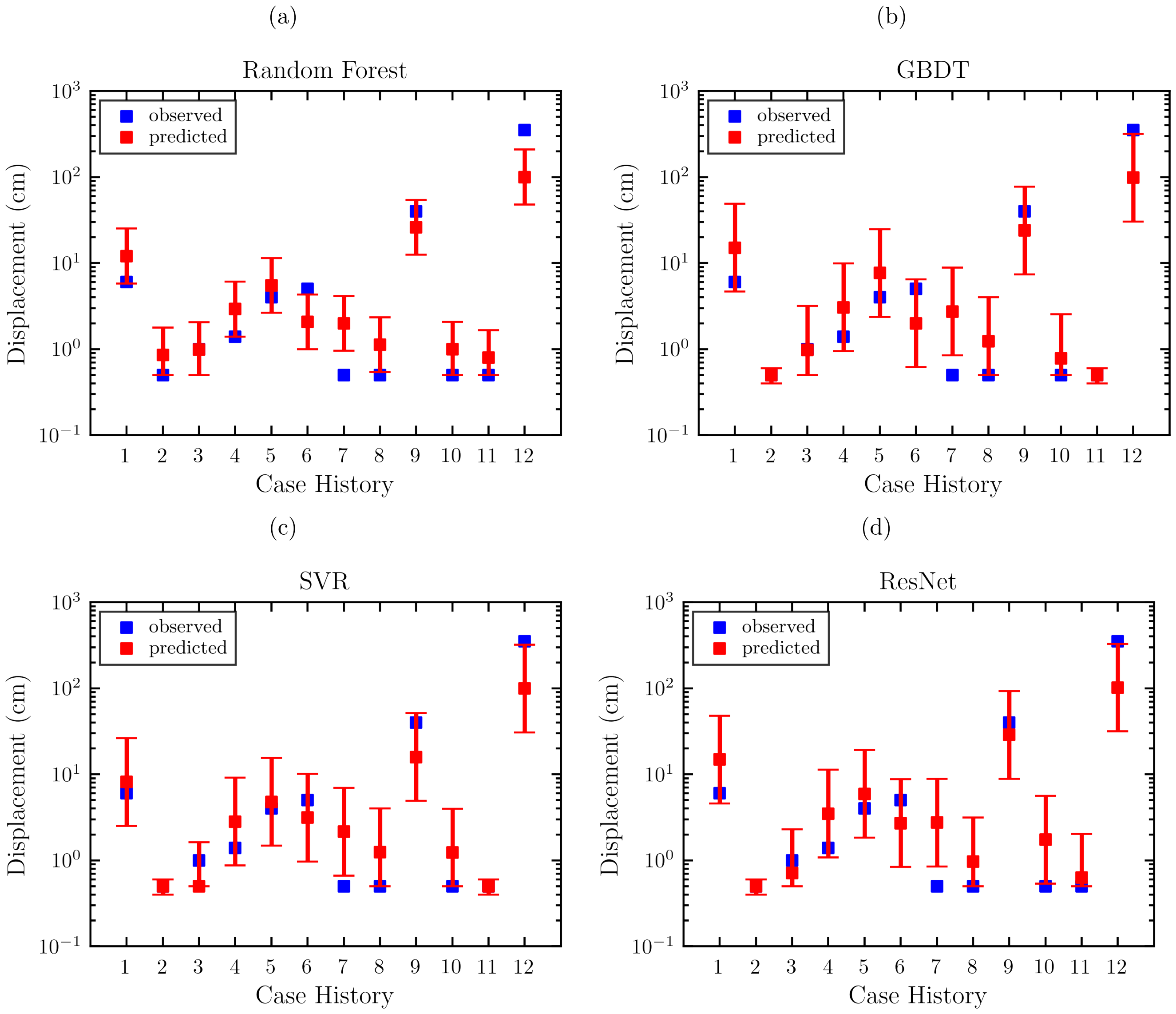


Figure 25: Performance of the ML models on interface case histories (see Table 6 for the case histories details). Note that zero $D_{obs}$ values are shown as 0.5 cm for plotting on log scale.

We have ranked the performance of ML models according to three criteria: their errors on the test set (RMSE), the case histories (MAE, when case histories are available), and their $D$ trends. The models with the best performance are assigned with scores of 2 while those with the worst performance are assigned with scores of 0. Intermediate models receive an intermediate score. Tables 8 and 9 present the final ranking of all the ML models for interface and intraslab tectonic settings, respectively. Based on the ranking, in terms of the developed ML-based models in this study, we recommend the ridge regression, random forest, GBDT, SVR, and ResNet models for their use in performance-based slope displacement procedures.

Table 8: Rating ML models according to their performance for interface tectonic settings

| Model | Trends | Test Error | Case Histories | Cumulative |
|---|---|---|---|---|
| Ridge Regression | 2 | 1 | 1 | 4 |
| Ridge Regression with one IM | 2 | 0 | 1 | 3 |
| PCR | 2 | 0 | 0 | 2 |
| PLSR | 2 | 1 | 0 | 3 |
| Random Forest | 1 | 2 | 2 | 5 |
| GBDT | 1 | 2 | 2 | 5 |
| SVR | 2 | 1 | 1 | 4 |
| ResNet | 0 | 2 | 2 | 4 |

Table 9: Rating ML models according to their performance for intraslab tectonic settings

| Model | Trends | Test Error | cumulative |
|---|---|---|---|
| Ridge Regression | 2 | 1 | 3 |
| Ridge Regression with one IM | 2 | 0 | 2 |
| PCR | 2 | 0 | 2 |
| PLSR | 2 | 0 | 2 |
| Random Forest | 1 | 2 | 3 |
| GBDT | 1 | 2 | 3 |
| SVR | 2 | 1 | 3 |
| ResNet | 0 | 2 | 2 |

# 6 Treatment of epistemic uncertainties

In this section, we first discuss the performance-based probabilistic assessments of $D$, which requires a performance-based implementation of $D$ models. The outcome of a performance-based probabilistic assessment is a displacement hazard curve ($DHC$), which relates different $D$ thresholds with their annual rate of exceedance. An important aspect of estimating $DHC$ is the assessment of epistemic uncertainties. Epistemic uncertainties are related to the scientific uncertainties in modeling a process due to limited data and knowledge. Hence, in estimating $D$, the epistemic uncertainty results from the uncertainties in the functional forms of alternative $D$ models as well as the uncertainties in the slope characteristics (e.g., $k_y$ and $T_s$). After discussing the details of performance-based probabilistic assessments, we describe the traditional approach used to capture epistemic uncertainties in the $D$ estimation using a logic tree approach. Then we introduce an alternative, albeit more efficient approach that utilizes polynomial chaos expansion. The procedures discussed in this section are later illustrated in Section 8, where different examples are presented.

## 6.1 Performance-based probabilistic assessment

In the performance-based probabilistic assessment of $D$, first, a PSHA is performed to estimate the hazard curves of the IM of interest. Then the entire IM hazard curve obtained from a PSHA is convolved with a $D$ model, and the result is a $D$ hazard curve that relates different $D$ thresholds and their annual rate of exceedance. Using a $D$ hazard curve, $D$ can be directly estimated for the hazard design level (or return period) of interest; thus, the $D$ hazard level is known. The performance-based assessment of $D$ is consistent with performance-based earthquake engineering concepts and should be preferred in practice (e.g., Macedo et al. (2020b); Rathje and Saygili (2008); Liu et al. (2021)),

In this subsection, we illustrate the use of the developed $D$ models in this study for the performance-based probabilistic assessment of slope systems affected by subduction zone earthquakes. As previously discussed, in a performance-based probabilistic assessment, a $D$ hazard curve (i.e., a curve that relates different $D$ thresholds to their annual rate of exceedance) is estimated through the convolution of the IM (or IMs) hazard and a $D$ model according to Equation 11 (Macedo et al., 2020b).

$$\lambda_D(z) = \sum_{i=1}^{nk_y} \sum_{j=1}^{nT_s} \int_{M_{min}}^{M_{max}} \int_{\mathbf{IM}} w_i w_j P(D > z | \mathbf{IM}, M, k_y^i, T_s^j) f(M|\mathbf{IM}) \Delta\lambda(\mathbf{IM}) d(\mathbf{IM}) d(M) \tag{11}$$

where $\mathbf{IM}$ is a vector of ground motion IMs (e.g., $PGV$ and $Sa(1.3T_s)$ in the case of the models developed in this study), and $\lambda_D(z)$ is the mean annual rate of $D$ exceeding the threshold $z$. $\Delta\lambda(\mathbf{IM})$ is the joint annual rate of occurrence of $\mathbf{IM}$ and $P(M|\mathbf{IM})$ is the conditional probability of $M$ given $\mathbf{IM}$, which can be estimated using standard vector PSHA and magnitude deaggregation, respectively. $nk_y$ and $nT_s$ are the number of different $k_y$ and $T_s$ values considered for the slope system to account for the uncertainty in the slope properties. $k_y^i$ and $T_s^j$ are the $i$-th and $j$-th realizations of $k_y$ and $T_s$ for a given slope with weighting factors $w_i$ and $w_j$, correspondingly. $P(D > z|\mathbf{IM}, M, k_y^i, T_s^j)$ is the conditional probability of $D$ exceeding $z$ given the values of $\mathbf{IM}$, $M$, $k_y^i$, and $T_s^j$, which can be estimated according to Equation 1 as:

$$\begin{aligned} P(D > z) &= (1 - P(D = 0|\mathbf{IM}, M, k_y^i, T_s^j))P(D > z|D > 0) \\ &= (1 - P(D = 0|\mathbf{IM}, M, k_y^i, T_s^j))(1 - P(D \leq z|D > 0)) \\ &= (1 - P(D = 0|\mathbf{IM}, M, k_y^i, T_s^j)) \left(1 - \Phi\left(\frac{\ln z - \ln \mu(\mathbf{IM}, M, k_y^i, T_s^j)}{\sigma}\right)\right) \end{aligned} \tag{12}$$

where $\Phi$ is the cumulative distribution function of the standard normal distribution and $\mu(\mathbf{IM}, M, k_y^i, T_s^j)$ and $\sigma$ are the median value and standard deviation of $D$ which can be estimated using the developed $D$ models given $\mathbf{IM}$, $M$, $T_s$, and $k_y$. $P(D = 0)$ can be estimated using Equation 2, the developed ML-based classification models, or previously developed models (e.g., Bray and Macedo (2019) in the case of shallow crustal tectonic settings. Equation 11 can be applied separately to different tectonic settings. For instance, when considering the subduction interface, subduction intraslab, and shallow crustal tectonic settings, three different annual rate of exceedance curves can be evaluated for each tectonic setting (i.e. $\lambda_D^{interface}$, $\lambda_D^{intraslab}$, and $\lambda_D^{crustal}$), which can then be combined to estimate the total annual rate of exceedance $\lambda_D^{total}$ as per Equation 13. Importantly, when this is performed, adequate $D$ models should be applied to each tectonic setting to evaluate Equation 11 (please refer to Macedo et al. (2020b) for more details).

$$\lambda_D^{total} = \lambda_D^{interface} + \lambda_D^{intraslab} + \lambda_D^{crustal} \tag{13}$$

In performance-based engineering, it is important to identify the earthquake scenarios of most interest that control the $D$ hazard curves, which can be done by deaggregating the $D$ hazard curves. Considering a combination of $nM$ earthquake magnitude, $nR$ site-to-source distances, and $n\epsilon$ epsilon values (i.e., the number of standard deviations above the median IM), we can obtain a total number of ground motion scenarios $nScen = nMnRn\epsilon$. In addition, if we consider $nx$ levels of $\mathbf{IM}$ and $ny$ levels of $D$; then, the $\mathbf{IM}$ hazard can be stored in a matrix $\lambda(IMT)$ of $nScen$ by $nx$. For a fixed $D$ level, $k_y$, and $T_s$ values, we can evaluate Equation 11 for each ground motion scenario and $\mathbf{IM}$ level, resulting in a matrix $PS\lambda_D$, that contains the partial annual rate of exceedance sorted by scenarios and IM values. The $D$ hazard contribution from all $\mathbf{IM}$ levels can be computed by summing up all columns of $PS\lambda_D$, resulting in a vector $S\lambda_D = [S\lambda_D^1, \cdots, S\lambda_D^{nScen}]$. This vector can be used as a proxy to perform the deaggregation of the $D$ hazard curves as follows (please refer to Macedo et al. (2020b) for additional details):

$$Deagg_{Dj} = S\lambda_{D_j}^i / \sum_{p=1}^{nScen} S\lambda_{Dj}^p \quad \text{for} \quad j = 1 : ny \tag{14}$$

whre $Deagg_{Dj}$ contains the earthquake scenarios deaggegrated for the $j$th $D$ level.

### 6.2 Epistemic uncertainty treatment using a logic tree approach

In a logic tree approach, the epistemic uncertainties associated with $D$ models can be represented as the branches in a logic tree. Each branch corresponds to an alternative $D$ model with a weighting factor. For example, one can use the five ML models (i.e., ridge regression, random forest, GBDT, SVR, and ResNet) recommended in the previous section with equal weights of 0.18 and a weight of 0.10 for the traditional model to capture the epistemic uncertainties of $D$ models for subduction zone earthquakes (i.e., interface or intraslab earthquake zones separately). Of note, the same weight for the ML models is considered here as their performance (evaluated in the previous section) was similar, and a higher weight is considered for the ML models as they performed better than their traditional counterparts. In this way, a logic tree with six branches can be constructed, and each branch will yield a $D$ hazard curve (DHC) generated by the five ML $D$ models and the traditional $D$ model. Similarly, to capture the epistemic uncertainties in $k_y$ and $T_s$, one can sample $nk_y$ $k_y$ values and $nT_s$ $T_s$ values from lognormal distributions and create $nk_y * nT_s$ branches in a logic tree to represent each combination of $k_y$ and $T_s$, following the recommendations in Macedo et al. (2018). Moreover the $k_y$, $T_s$, and $D$ logic trees could also be combined with logic trees for the IMs as discussed in detail in Macedo et al. (2020b).

Although the logic tree approach is straightforward to implement, there are some potential issues and limitations regarding this approach. Firstly, the epistemic uncertainties represented by the $D$ model branches may not capture the full range of the alternative $D$ models. For example, the epistemic uncertainties captured by the six previously mentioned $D$ model branches only reflect the range of alternative models expanded by the five ML models and the traditional model developed in this study. As a result, more ML models in the future could improve the estimation of the range/epistemic uncertainties of $D$ models. Secondly, the logic tree approach can become computationally inefficient when the number of branches is large, especially when many realizations of the IM of interest, $k_y$ and $T_s$ are considered.

### 6.3 Epistemic uncertainty treatment using Polynomial chaos

As an alternative to the logic tree approach, we also consider the polynomial chaos expansion (PCE) method to capture the epistemic uncertainties in the estimation of DHCs. The general principle behind the PCE theory is that a random function can be expressed as a linear combination of a family of orthogonal polynomials in which each polynomial is a function of random variables, and the coefficients in the expansion are deterministic. In the context of DHC estimation, if we consider each $D$ model as a sample from a common continuous distribution of $D$ models, and assume that the epistemic uncertainty associated with the median $D$ values from alternative $D$ models is a normal distribution, the range of DHCs resulting from different $D$ models can be represented using PCE. To illustrate the calculations involved in PCE, we consider the ridge regression model with 1 IM - $Sa(1.3T_s)$ (see Section 4 for more details). Then, assuming the median $D$ estimate ($\ln \hat{d}$) is normally distributed with a mean $\mu_{EU}(Sa(1.3T_s), k_y, T_s)$ and a standard deviation $\sigma_{EU}(Sa(1.3T_s), k_y, T_s)$, the probability of exceedance in Equation 12 can be calculated as:

$$\begin{gathered} P(D > z|Sa(1.3T_s), k_y, T_s, \eta) = 1 - \Phi\left(\frac{\ln z - (\ln \mu_{EU}(Sa(1.3T_s), k_y, T_s) + \eta\sigma_{EU}(Sa(1.3T_s), k_y, T_s))}{\sigma}\right) \\ \eta = \frac{\ln \hat{d} - \mu_{EU}(Sa(1.3T_s), k_y, T_s)}{\sigma_{EU}(Sa(1.3T_s), k_y, T_s)} \end{gathered} \tag{15}$$

in which $\mu_{EU}(Sa(1.3T_s), k_y, T_s)$ and $\sigma_{EU}(Sa(1.3T_s), k_y, T_s)$ can be estimated by considering the mean and standard deviation of $\ln \hat{d}$ given a fixed $Sa(1.3T_s)$ and different values of $k_y$ and $T_s$ (see Macedo et al. (2020c); Lacour and Abrahamson (2019) for more details). Since $\ln \hat{d}$ is normally distributed, $P(D > z|Sa(1.3T_s), k_y, T_s, \eta)$ can be expressed as a combination of Hermite polynomials using PCE as follows:

$$P(D > z|Sa(1.3T_s), k_y, T_s, \eta) = \sum_{l=0}^{P} y_l(z, Sa(1.3T_s))\Psi_l(\eta) \tag{16}$$

where $y_l(z, Sa(1.3T_s))$ represents the deterministic coefficients in the expansion, $\Psi_l(\eta)$ is the Hermite polynomial, which is a function of the standard normal variable $\eta$, and $P$ is the number of Hermite polynomials. Usually $P = 4$

is able to achieve a good approximation of $P(D > z|Sa(1.3T_s), k_y, T_s, \eta)$ for PCE. The estimation of $y_l(z, Sa(1.3T_s))$ can be performed by considering the orthogonality between the Hermite polynomial with more details available in Macedo et al. (2020c). Finally, the DHCs using PCE can be estimated by combining Equations 11 and 16 as per Equation 17.

$$\lambda_D(z) = \int_{M_{min}}^{M_{max}} \int_{Sa(1.3T_s)} \sum_{l=0}^{P} y_l(z, Sa(1.3T_s))\Psi_l(\eta) f(M|Sa(1.3T_s))\Delta\lambda(Sa(1.3T_s))d(Sa(1.3T_s))d(M) \tag{17}$$

As it can be observed in Equation 17, the PCE approach avoids intensive calculations involved in the logic tree branches that represent alternative $k_y$, $T_s$, and $D$ models; hence, improving the computational efficiency. Besides, the epistemic uncertainties of alternative $D$ models can be represeted through the mean $(\mu_{EU}(Sa(1.3T_s), k_y, T_s))$ and standard deviation $(\sigma_{EU}(Sa(1.3T_s), k_y, T_s))$ of $\ln \hat{d}$. A flow chart showing the sequence of calculations for implementing the PCE-based assessments of DHCs is presented in Appendix C.

# 7 Implementation of $D$ models

To facilitate the use of the developed $D$ models in performance-based assessments, we have implemented them in the SeismicHazard and PSDA platforms, with details in Candia et al. (2019); Macedo et al. (2020b); Macedo and Candia (2020, 2019b); Candia et al. (2018); Macedo and Candia (2019a).

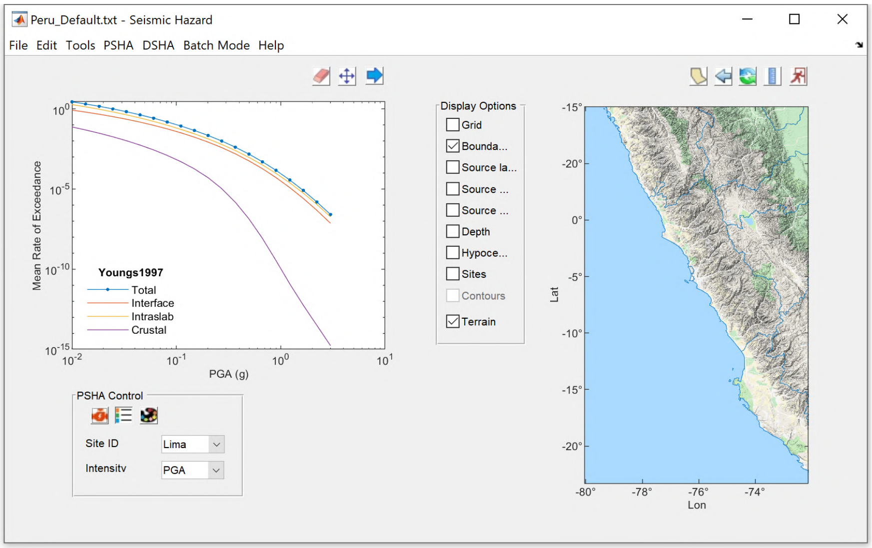
The SeismicHazard platform permits characterization of the intensity, uncertainty, and likelihood of ground motions from subduction zone and shallow crustal earthquakes, considering site-specific and regional-based assessments. It features several state-of-the-art capabilities for probabilistic and deterministic (scenario-based) seismic hazard assessment. The platform integrates the latest developments in performance-based engineering for seismic hazard assessment, including seismic zonation models, GMMs, ground motion correlation structures, and the estimation of design spectra (Uniform Hazard Spectrum and Conditional Mean Spectrum). In addition to these standard capabilities, the platform supports advanced features not commonly found in existing seismic hazard codes, such as the i) vector-hazard PSHA, ii) uncertainty treatment in the median ground motions (i.e., logic trees), iii) regional shaking fields, and iv) estimation of conditional mean spectra considering multiple tectonic settings. The platform has been validated against accepted and well-documented benchmark solutions, including the Pacific earthquake engineering research center (PEER) 2018 Probabilistic Seismic Hazard Analysis Code Verification (Hale et al., 2018). Figure 26 shows the functionalities of the SeismicHazard platform in the context of PSHA.

The PSDA platform interacts with the SeismicHazard platform to extract hazard information to evaluate DHCs. The current PSDA platform contains built-in $D$ models (e.g., Bray and Macedo (2019), Saygili and Rathje (2008), etc) for different tectonic settings. The PSDA platform can perform: (a) estimation of DHCs in the context of scalar PSHA, (b) estimation of DHCs in the context of vector PSHA, (c) estimation of DHC for systems with contributions from multiple tectonic settings, (d) uncertainty treatment on DHCs through a logic tree scheme, (5) deaggregation of earthquake scenarios from DHCs, and 6) uncertainty quantification on DHCs through the PCE theory. In this study, we have implemented the developed $D$ traditional and ML-based models, the treatment of epistemic uncertainties (logic tree, PCE), and the performance-based assessment of DHCs using the developed models in the PSDA platform. Since most of the $D$ models are non-parametric models developed using Python, we use a MATLAB-Python API (application programming interface) to call Python programs in MATLAB. The general workflow of the PSDA platform (including the interaction with the developed $D$ models) is shown in Figure 27.

Figure 28 shows some of the implemented $D$ models in the PSDA platform. The estimated mean values of $D$ as well as the probability of exceedance for various $D$ models given different $k_y$, $T_s$, $M$, and ground motion IMs can be visualized in an efficient manner.

We have also integrated the platform with the existing Unified Hazard USGS tool. In this way, the Unified Hazard USGS tool can be used to retrieve the hazard information from any location in the United States, and then combine this information with the developed ML $D$ models to finally estimate seismically-induced slope displacement hazard curves.

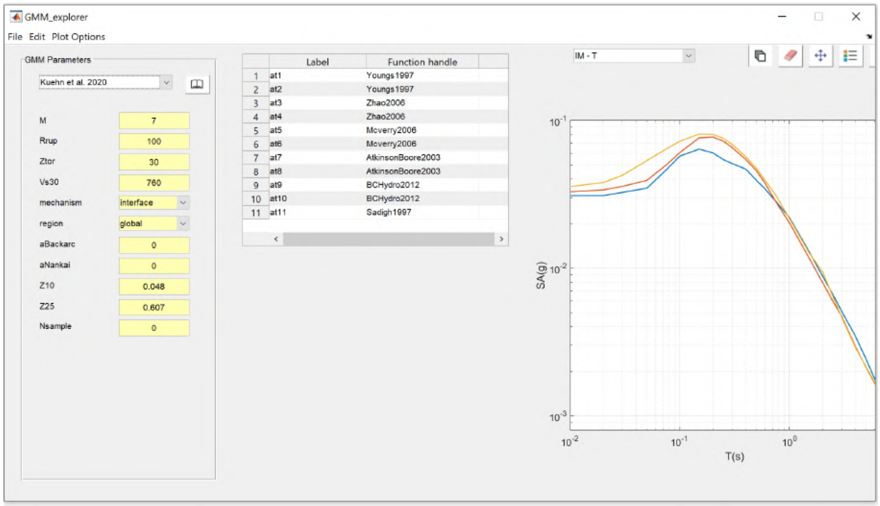
(a)



Site ID: Lima

Intensitv: PGA

(b)

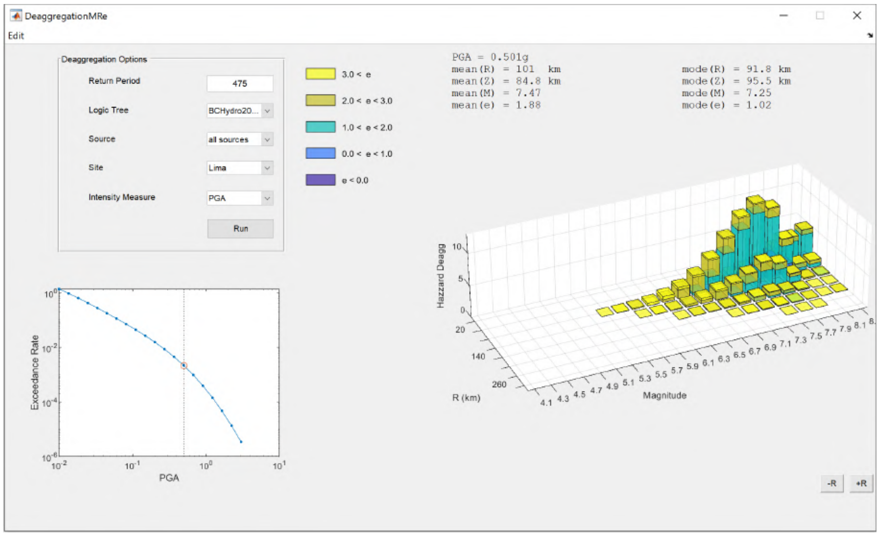


Kuehn et al. 2020

M: 7
Rrup: 100
Ztor: 30
Vs30: 760
mechanism: interface
region: global
aBackarc: 0
aNankai: 0
Z10: 0.048
Z25: 0.607
Nsample: 0

| | Label | Function handle |
|---|---|---|
| 1 | at1 | Youngs1997 |
| 2 | at2 | Youngs1997 |
| 3 | at3 | Zhao2006 |
| 4 | at4 | Zhao2006 |
| 5 | at5 | Mcverry2006 |
| 6 | at6 | Mcverry2006 |
| 7 | at7 | AtkinsonBoore2003 |
| 8 | at8 | AtkinsonBoore2003 |
| 9 | at9 | BCHydro2012 |
| 10 | at10 | BCHydro2012 |
| 11 | at11 | Sadigh1997 |

(c)



Return Period: 475
Logic Tree: BCHydro20...
Source: all sources
Site: Lima
Intensity Measure: PGA

```
PGA = 0.501g
mean(R) = 101  km                  mode(R) = 91.8 km
mean(Z) = 84.8 km                  mode(Z) = 95.5 km
mean(M) = 7.47                     mode(M) = 7.25
mean(e) = 1.88                     mode(e) = 1.02
```

Figure 26: SeismicHazard platform. (a) Main window, (b) GMM editor and viewer, and (c) module for $M - R - \epsilon$ deaggregation of seismic hazard.

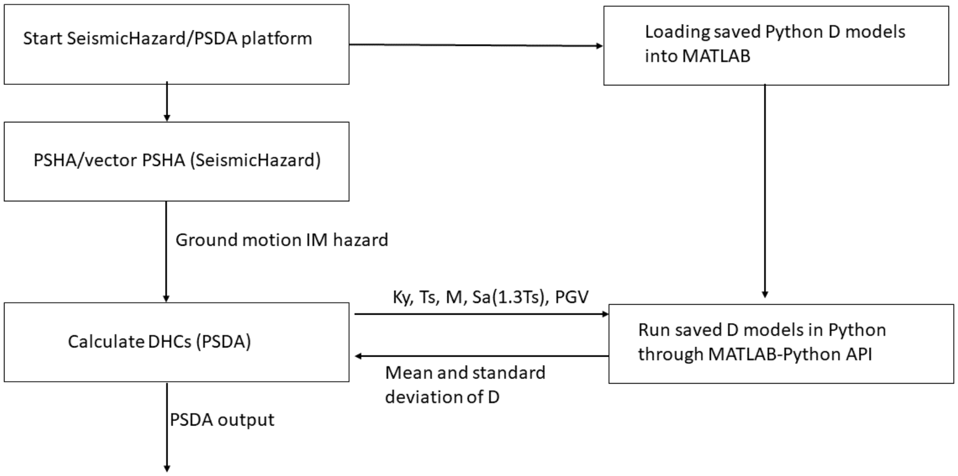
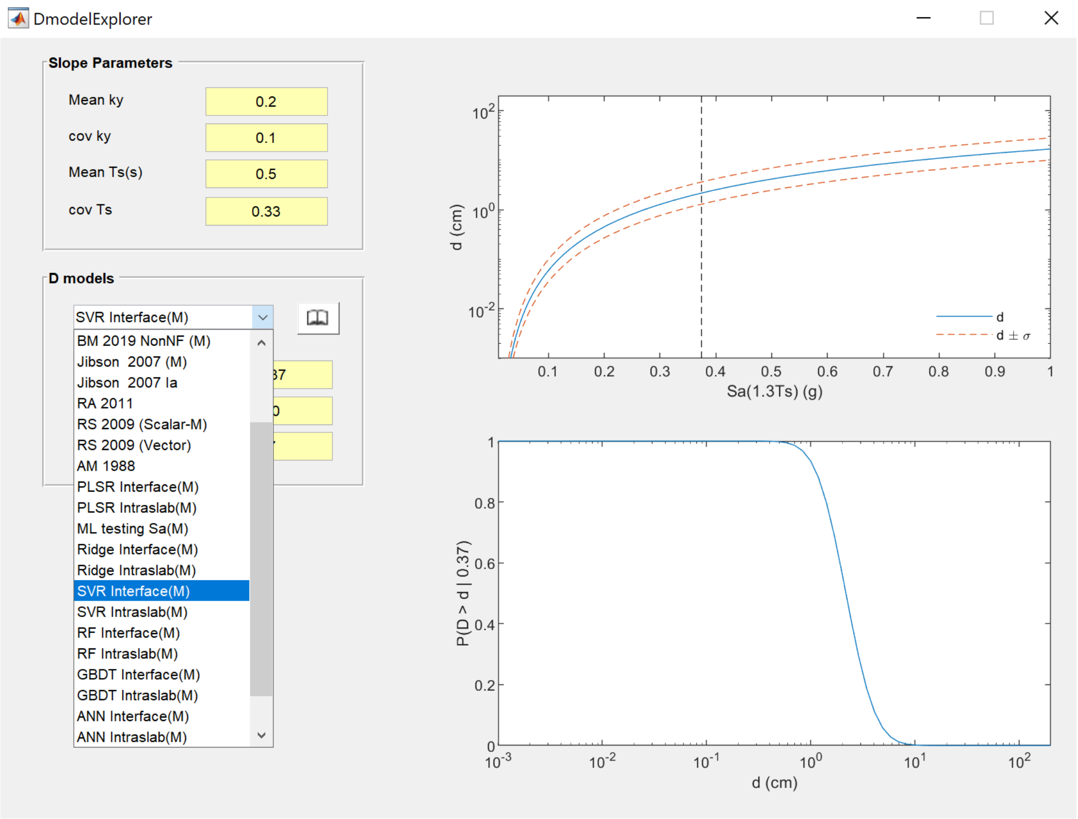


Figure 27: Schematic illustration of the interface between $D$ models and the MATLAB platform



**Slope Parameters**

Mean ky 0.2

cov ky 0.1

Mean Ts(s) 0.5

cov Ts 0.33

**D models**

SVR Interface(M)

Figure 28: Demonstration of the GUI window for the implemented $D$ models in the PSDA plaftorm

# 8 Examples using the developed $D$ models and implementations from this study

In this section, we present several examples of applying the developed ML and traditional $D$ models in performance-based engineering procedures. Firstly, We conduct an estimation of $D$ using the pseudoprobabilistic procedures, which dominate the current engineering practice. Then we implement the performance-based probabilistic assessments previously discussed, which are more consistent with the performance-based design. Specifically, we calculate $D$ hazard curves considering the convolution of IM hazard and the developed $D$ models and use a logic tree to account for the epistemic uncertainty. Next, we conduct a similar performance-based assessment of $D$ hazard curves but use PCE to account for the epistemic uncertainty. Finally, we conduct a regional assessment of seismically-induced landslides in Anchorage, Alaska.

## 8.1 Estimation of $D$ using pseudoprobabilistic procedures

$D$ models are typically used in pseudoprobabilistic or performance-based approaches, with the former being most popular in current engineering practice. In a pseudoprobabilistic approach, first, a PSHA is performed, and a hazard design level (or return period) is defined, which is then used to estimate the IMs of interest and earthquake-related parameters if needed (e.g., $M$). These parameters along with the slope properties (e.g., $k_y$ and $T_s$), are used to estimate $D$ and its uncertainty through the uncertainty in the $D$ model.

In this section, we demonstrate the use of pseudoprobabilistic procedures to estimate $D$ using the five recommended ML and traditional models. We consider a hypothetical slope located in the vicinity of Seattle, Washington, as shown in Figure 29, which is mainly affected by subduction interface and intraslab earthquakes. The slope is assumed to have a mean $T_s$ of 0.23 s and a mean $k_y$ of 0.1. The hazard curves of $Sa(1.3T_s)$ (i.e., $Sa(0.3s)$) are retrieved using the USGS unified hazard tool at the location of the slope. Figure 30 shows the hazard curves of $Sa(0.3s)$ for interface and intraslab earthquakes. Then we conduct a magnitude ($M$) and distance ($R_{rup}$) deaggregation at a return period of 2475 years using the USGS unified hazard tool. The results indicate that the controlling earthquake scenarios for interface and intraslab tectonic settings are $M = 8.8$, $R_{rup} = 85$ km and $M = 7.1$, $R_{rup} = 66$ km, respectively. Given the earthquake scenarios and $Sa(0.3s)$, we estimate $PGV$ using the conditional mean spectrum (CMS) method (Baker, 2011) with the correlation coefficients between $PGV$ and $Sa(0.3s)$ taken from Macedo and Liu (2021). Finally, we use the five recommend ML and traditional models to estimate $D$ values. The $D$ values estimated from these models are presented in Table 10.

Table 10: Estimated $D$ values (cm) using the pseudoprobabilistic approach.

| model | median $D$ (interface) | 5 to 95 interval (interface) | median $D$ (intraslab) | 5 to 95 interval (intraslab) |
|---|---|---|---|---|
| ML22 | 10.4 | 2.5 - 42.1 | 12.2 | 3.7 - 39.2 |
| Ridge Regression | 10.6 | 3.2 - 34.2 | 12.6 | 4.5 - 34.9 |
| SVR | 10.2 | 3.6 - 28.6 | 17.2 | 7.0 - 42.1 |
| Random Forest | 13.3 | 6.3 - 27.8 | 17.3 | 9.3 - 32.2 |
| GBDT | 18.4 | 10.5 - 32.2 | 17.2 | 10.4 - 28.3 |
| ResNet | 10.7 | 5.1 - 22.6 | 22.1 | 12.1 - 40.3 |

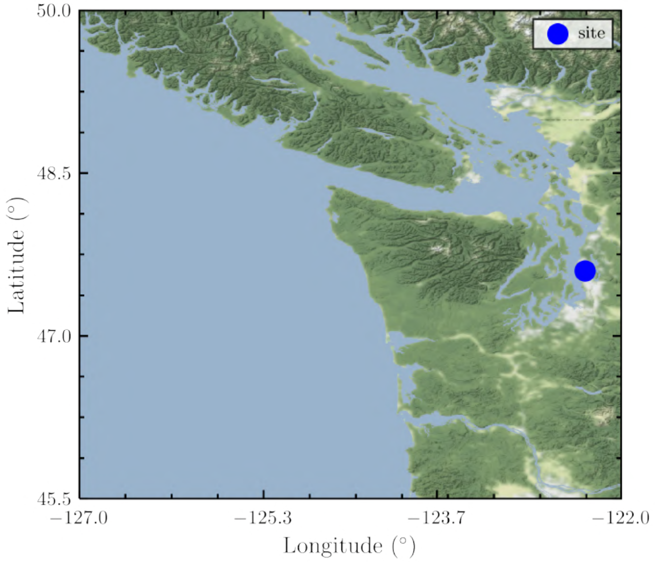


Figure 29: The consider site for the pseudoprobabilistic assessment of landslides in Seattle, Washington.

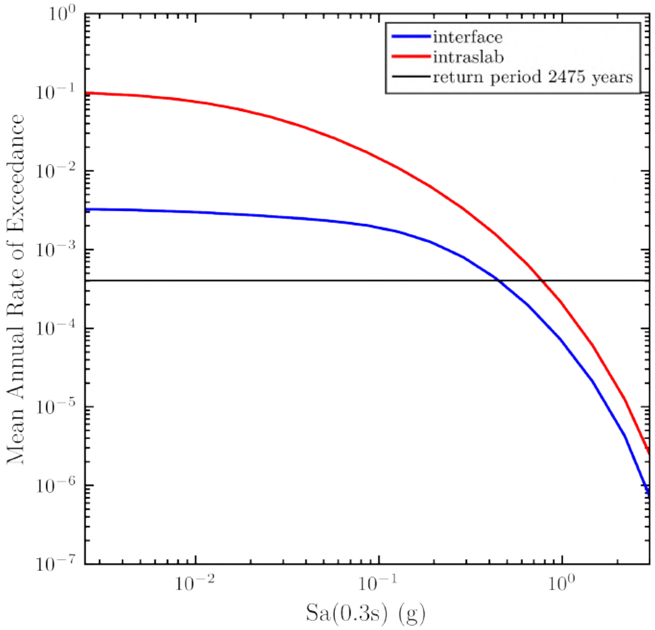


Figure 30: Hazard curves of $Sa(0.3s)$ considered for the pseudoprobabilistic assessment for interface and intraslab earthquakes

In addition to the estimation of $D$, we also compare the scaling of $D$ versus $Sa(0.3s)$ for recommended ML and traditional models. Because the tree-based models (i.e., GBDT and random forest) provide discrete $D$ estimates given continuous input features, we use the Savitzky-Golay (Press and Teukolsky, 1990) filter to smooth the $D$ trends. Figures 31 (a) and (b) show the trends of $D$ versus $Sa(0.3s)$ considering the previous slope and earthquake scenarios obtained from deaggregation for interface and intraslab earthquakes, respectively. $PGV$ is estimated in the same manner using the CMS method as in the pseudoprobabilistic assessment. It can be observed that all the models show generally consistent trends, however, with some variations associated with epistemic uncertainties.

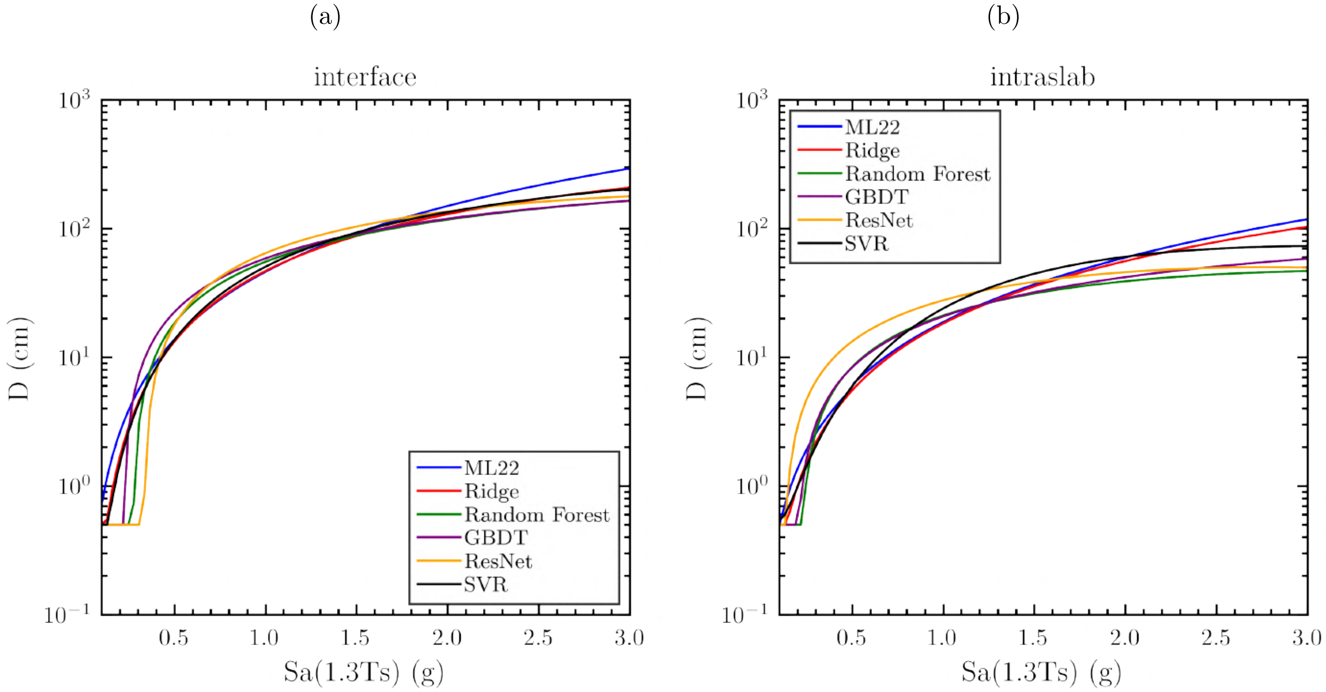


Figure 31: Scaling of $D$ against $Sa(1.3T_s)$ for $D$ models considering the controlling earthquake scenarios from deaggregation for (a) interface and (b) intraslab earthquakes. Negligible displacements are assigned a value of 0.5 cm.

## 8.2 Estimation of $D$ hazard curves using performance-based procedures and a logic tree

To illustrate the performance-based implementations, we consider the same slope system located in Seattle as in the previous section (i.e., Figure 29). In addition to obtain the hazard curves of $Sa(0.3s)$, the hazard curves of $PGV$ also need to be estimated for their use in the convolution of IM hazard curves and $D$ models. Because the USGS unified hazard tool does not provide $PGV$ hazard curves, for each hazard level of $Sa(0.3s)$, we perform deaggregation to identify the controlling earthquake scenarios and use the CMS method to obtain the conditional mean and standard deviation of $PGV$ (the correlation coefficients between $PGV$ and $Sa(0.3s)$ used in CMS are retrieved from Macedo and Liu (2021) for interface and intraslab settings). Then, for each level of $PGV$, we estimate its exceeding probability according to the distribution of $PGV$ conditioned on $Sa(0.3s)$. Finally, the joint rate of exceedance (or occurrence) of $PGV$ and $Sa(0.3s)$ can be estimated by combining the hazard of $Sa(0.3)$, the probability of controlling earthquake scenarios, and the exceeding probability of $PGV$ conditioned on $Sa(0.3)$. Figure 32 shows the hazard curves of $Sa(0.3s)$ and $PGV$ as well as their joint rate of exceedance for interface and intraslab tectonic settings.

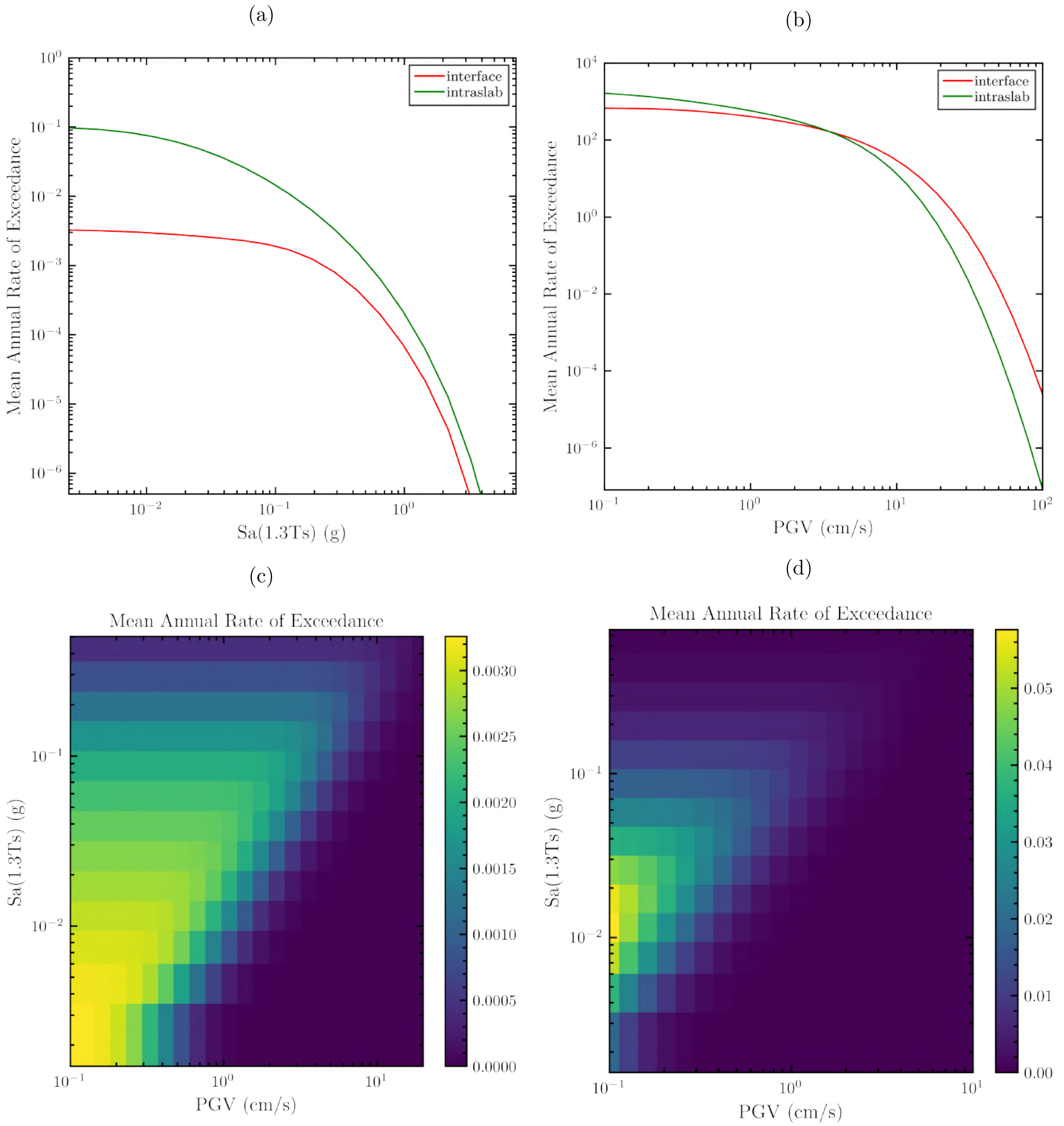


Figure 32: $Sa(1.3T_s)$ and $PGV$ hazard for performance-based assessment using a logic tree. (a) marginal hazard curves of $Sa(1.3T_s)$, (b) marginal hazard curves of $PGV$, (c) joint rate of exceedance for $Sa(1.3T_s)$ and $PGV$ for interface earthquakes, and (d) joint rate of exceedance for $Sa(1.3T_s)$ and $PGV$ for intraslab earthquakes

To account for the epistemic uncertainties of the slope properties, we consider alternative $k_y$ and $T_s$ values in a logic tree scheme. We assume $k_y$ and $T_s$ have mean values of 0.23 and 0.1, respectively, and a coefficient of variation of 0.3. Then we consider 100 realizations of $k_y$ and $T_s$ with their weights assigned according to their probability densities, resulting in 100 branches in the logic tree. The five recommended ML models and traditional models are considered to incorporate the epistemic uncertainty associated with the median $D$ values through the logic tree. Tables 11 summarizes the logic tree branches and weighting factors for the considered $D$ models.

Table 11: Logic tree branches used for $D$ models in performance-based assessment.

| Subduction interface and intraslab | Weight |
|---|---|
| ML22 | 0.1 |
| Ridge Regression | 0.18 |
| SVR | 0.18 |
| Random Forest | 0.18 |
| GBDT | 0.18 |
| ResNet | 0.18 |

Figure 33(a) shows the $D$ hazard curves obtained using the performance-based procedures for different $D$ models (each model has 100 hazard curves corresponding to 100 branches of $k_y$ and $T_s$). In general, all the models show consistent ranges of hazard curves. Differences can be observed for moderate to large displacements (i.e., $D > 10$ cm). Especially, the residual neural network model shows slightly wider ranges of hazard curves than the other models. These differences can be attributed to the epistemic uncertainties in the median $D$ estimates. Figure 33(b) shows the fractiles and mean values of the 600 hazard curves across all the models.

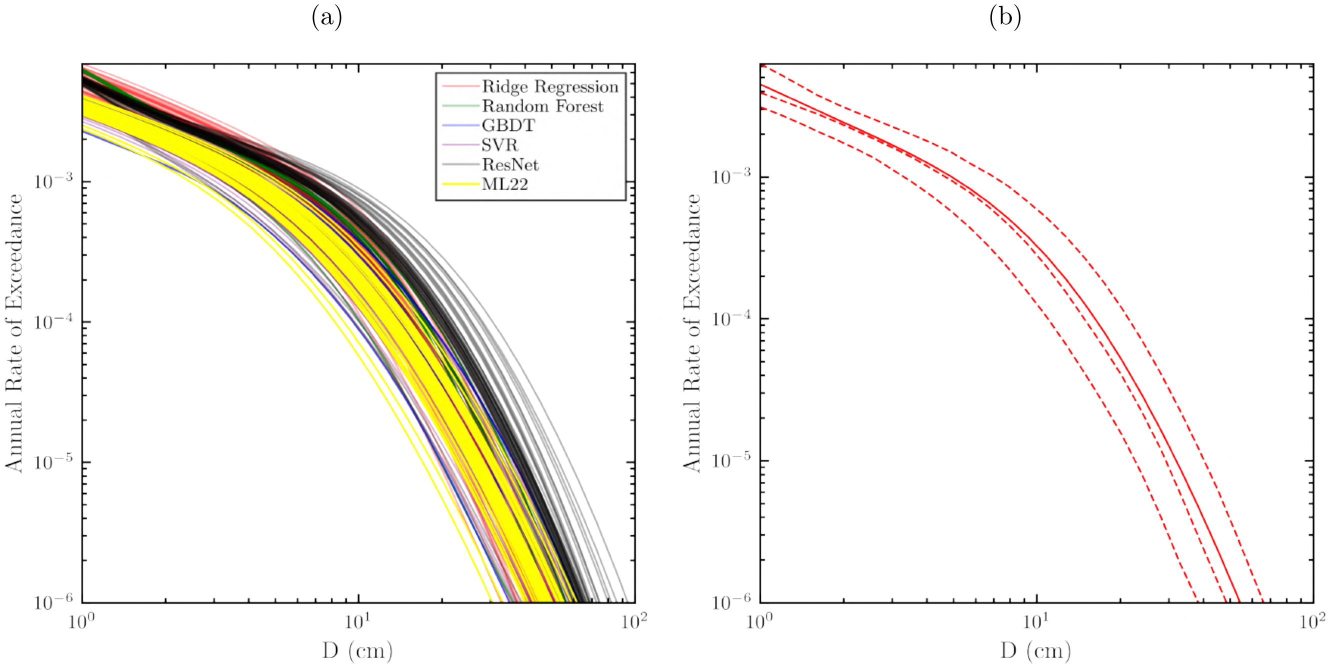


Figure 33: $D$ hazard curves estimated using the performance-base assessment. (a) $D$ hazard branches for different realizations of $k_y$ and $T_s$ for different $D$ models, and (b) mean and fractiles of the $D$ hazard curves across all the models (the dashed curves indicate 5-50-95 percentiles and the solid curve indicates the mean hazard).

## 8.3 Estimation of $D$ hazard curves using performance-based procedures and polynomial chaos expansion

Performance-based assessment of $D$ hazard curves using Monte Carlo-based simulation of slope properties ($k_y$ and $T_s$) can be computationally challenging, especially when there is a large number of $k_y$ and $T_s$ realizations. PCE can be used to reduce the computational cost (Macedo et al. (2020c)). The application of PCE for estimating $D$ hazard curves is illustrated in this subsection. We consider the same slope system and site location described in previous sections. In terms of $D$ models, we use the ridge regression model developed with one IM (i.e., $Sa(1.3T_s)$) as described in Section 3 for illustration purposes and simplicity. In this context, the input parameters for estimating $D$ in the considered ridge regression model are $k_y$, $T_s$, $M$, and $Sa(1.3T_s)$. We also compare the results obtained using the traditional Monte Carlo-based and PCE approaches.

The hazard curves of $Sa(1.3T_s)$ are retrieved using the USGS unified hazard tool as described earlier (i.e., Figure 30). Then for each hazard level, we also obtain the magnitude deaggregation. To account for the epistemic uncertainties associated with slope properties, we consider 1000 realizations of $k_y$ and $T_s$, sampled from their probabilistic distributions (i.e., mean and coefficient of variation) using Monte Carlo simulation. As a result, the logic tree approach contains 1000 branches of $k_y$ and $T_s$. Then we compare the estimated $D$ hazard curves using the Monte Carlo and the PCE approaches as described in Section 6.

Figure 34(a) shows the 1000 hazard curves corresponding to the 1000 realizations of $k_y$ and $T_s$ using the two approaches (i.e., Monte Carlo and PCE). Figure 34(b) shows the corresponding fractiles and means for the two approaches. It can be observed that the hazard curves estimated using PCE are consistent with those estimated using a traditional Monte Carlo approach. However, the computational time for PCE (0.07s) is more than 50 times smaller than that of the Monte Carlo approach (3.71s). Hence, the PCE approach is an attractive alternative for performance-based procedures when the number of branches in a logic tree is large.

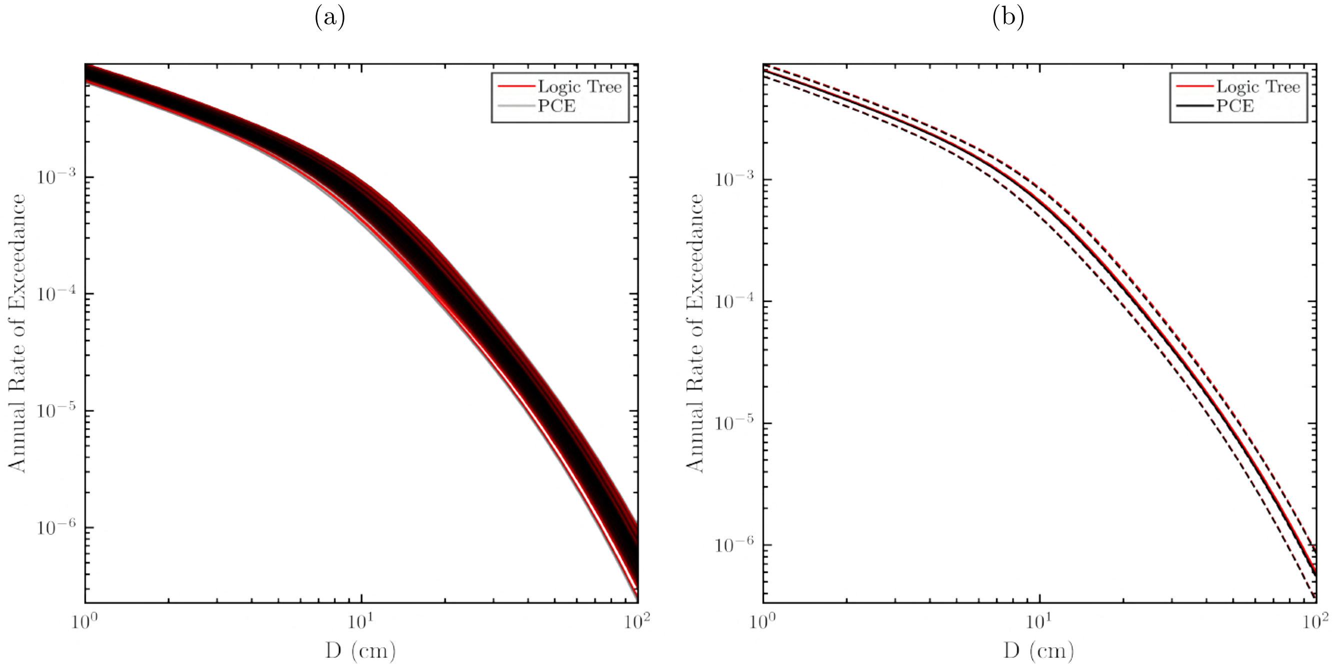


Figure 34: Comparison of $D$ hazard curves obtained using the Monte Carlo and PCE approaches. (a) Hazard curve branches corresponding to 1000 realizations of $k_y$ and $T_s$, (b) 5-50-95 fractiles (dashed curves) and mean hazard curves (solid curves).

## 8.4 Regional assessments of seismically-induced landslides in Alaska using performance-based procedures

Regional assessments of $D$ hazard are often required to produce seismic landslide hazard maps that identify zones prone to earthquake-induced slope failures that are key inputs for mitigation plans. We illustrate in this subsection how the $D$ models developed in this study can be used in the regional assessment of seismically-induced landslides.

We consider a study area of around 301 km$^2$ located in the vicinity of Anchorage, Alaska. Previous studies (e.g., Jibson and Michael (2009), Wang and Rathje (2015), etc.) have also developed seismic landslide hazard maps for the study area, but using only shallow crustal $D$ models. Figure 35 shows the extent of the study area considered in this study, which is similar to the one considered in Wang and Rathje (2015).

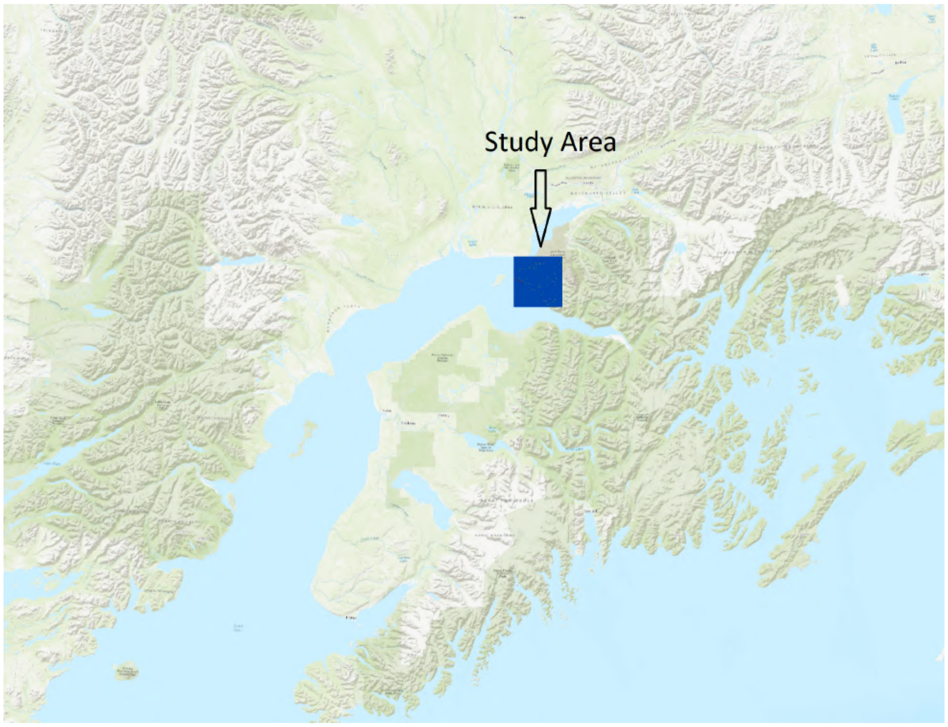


Figure 35: Overview of the study area in Anchorage, Alaska.

In terms of slope properties, we focus on shallow landslides; hence we assume rigid sliding masses with a $T_s$ of zero. The $k_y$ values are estimated using as inputs (1) strength properties (friction and cohesion) for the different geological units, assigned based on geological maps; (2) topographic information to assess the inclination of different slopes; and (3) the regional variation of the water table. The inputs are used to estimate $k_y$, as follows:

$$FS_{static} = \frac{c'}{\gamma t \sin\alpha} + \frac{\tan\phi'}{\tan\alpha}(1 - m\frac{\gamma_w}{\gamma}) \tag{18}$$

$$k_y = (FS_{static} - 1)\sin\alpha \tag{19}$$

in which $\alpha$ represents the slope inclination, $c'$ is the effective cohesion, $\phi'$ is the effective friction angle, $\gamma$ is the material unit weight (considered as 18.8 kN/m$^3$), $\gamma_w$ is the unit weight of water (9.8 kN/m$^3$), $t$ is the slope-normal thickness of the sliding mass, and $m$ is the proportion of the sliding mass thickness that is saturated. The geological and topographical information for the study area, required for Equations 18 and 19, is available in Wang and Rathje (2015) and was kindly shared with us by the authors.

To estimate $k_y$, the study area is divided into 16 million grid cells with a resolution in the order of meters. Then each grid cell is assigned to one of the existing 17 geologic units in the study area. Figure 36(a) shows the spatial distribution of the geological units in the study area. Then the friction angle and effective cohesion for each cell are assigned based on their corresponding geologic unit, as shown in Table 12. The slope angle for each cell is estimated based on a digital elevation model used in Wang and Rathje (2015), as shown in Figure 36(b). The values of $m$ and $t$ are assumed based on the geologic units as well as engineering observations in Anchorage (Jibson and Michael, 2009). To capture the epistemic uncertainties associated with $k_y$, coefficients of variation are assigned to $c'$ and $\phi'$, and a three-point approximation (Keefer and Bodily, 1983) is used to represent the distribution of $c'$ and $\phi'$ in a logic tree, leading to 54 $k_y$ values in each cell. More details regarding the estimation of $k_y$ can be found in Wang and Rathje (2015).

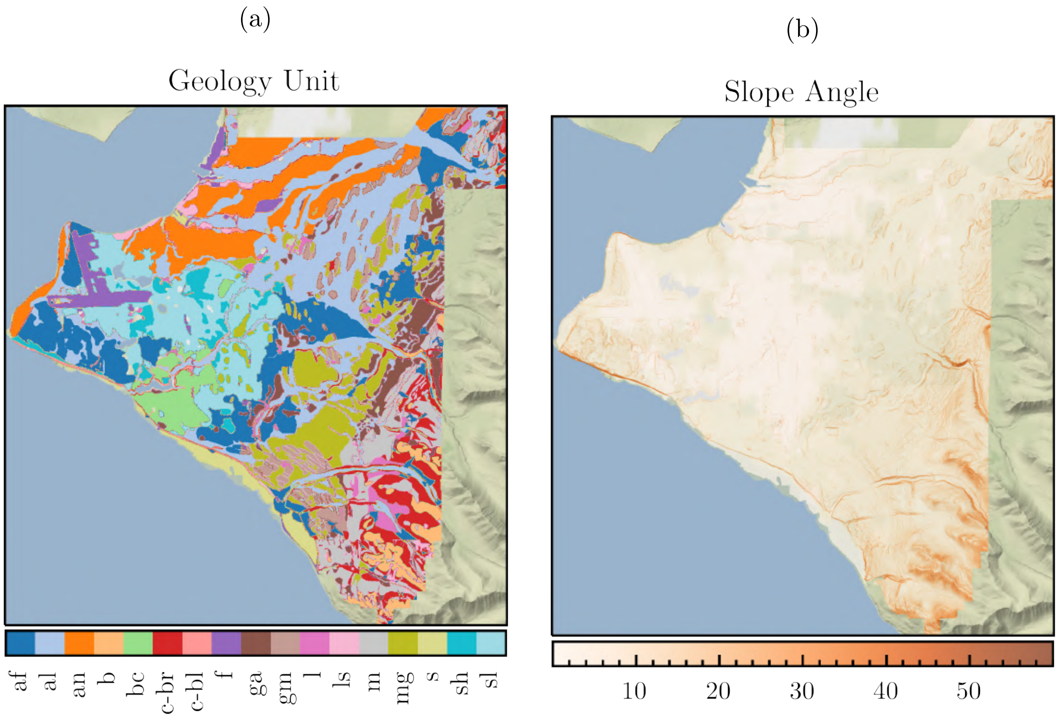


Figure 36: Spatial distribution of (a) slope angles and (b) geologic units in the study area. Geologic unit descriptions are provided in Table 12

Table 12: Geologic units and associated shear strengths.

| Unit | $\phi$ | $c$ (kPa) | Unit description |
|---|---|---|---|
| af | 36 | 24 | Deposits in alluvial fans, alluvial cones, and emerged deltas |
| al | 36 | 19 | Alluvium in abandoned stream channels and in terraces along modern streams |
| an | 36 | 24 | Coarse-grained surficial deposits |
| b | 40 | 192 | Bedrock |
| bc | 0 | 120 | Bootlegger Cove Clay |
| c-br | 38 | 38 | Colluvium derived from bedrock on slopes of the Chugach Mountains |
| c-bl | 0 | 38 | Colluvium derived from glacial materials along coastal bluffs |
| f | 34 | 48 | Manmade fills |
| ga | 32 | 38 | Glacial alluvium in irregular-shaped hills (including kames, eskers, and kame terraces) |
| gm | 38 | 48 | Glacial and (or) marine deposits, typically in elongate hills |
| l | 0 | 144 | Lake and pond deposits |
| ls | 30 | 24 | Landslide deposits, similar to an unit |
| m | 38 | 43 | Morainal deposits, generally in long ridges marking the margins of former glaciers |
| mg | 37 | 38 | Marine, glacial, and (or) lacustrine deposits |
| s | 0 | 72 | Silt |
| sh | 34 | 24 | Sand deposits in broad, low hills, and windblown sand deposits in cliffhead dunes near Point Campbell |
| sl | 34 | 19 | Sand deposits in a wide low-lying belt around Connors Lake |

Regarding ground motion hazard, we obtain the peak ground acceleration ($PGA$) hazard curves ($PGA$ is considered as $T_s$ is equal to zero) for Anchorage (-149.9, 61.22) using the USGS unified hazard tool. The hazard curves are assumed to be applied to the entire study area as its extension is the order of 10 by 10 km, where no significant spatial effects are expected, putting site conditions aside. The probability of magnitude given each level of $PGA$ is obtained from the deaggregation results in the USGS unified hazard tool. The $PGV$ hazard curves are estimated using the same CMS approach described in previous sections. Figure 37 shows the hazard curves of $PGA$ and $PGV$ for interface and intraslab earthquakes, as well as their joint rate of exceedance.

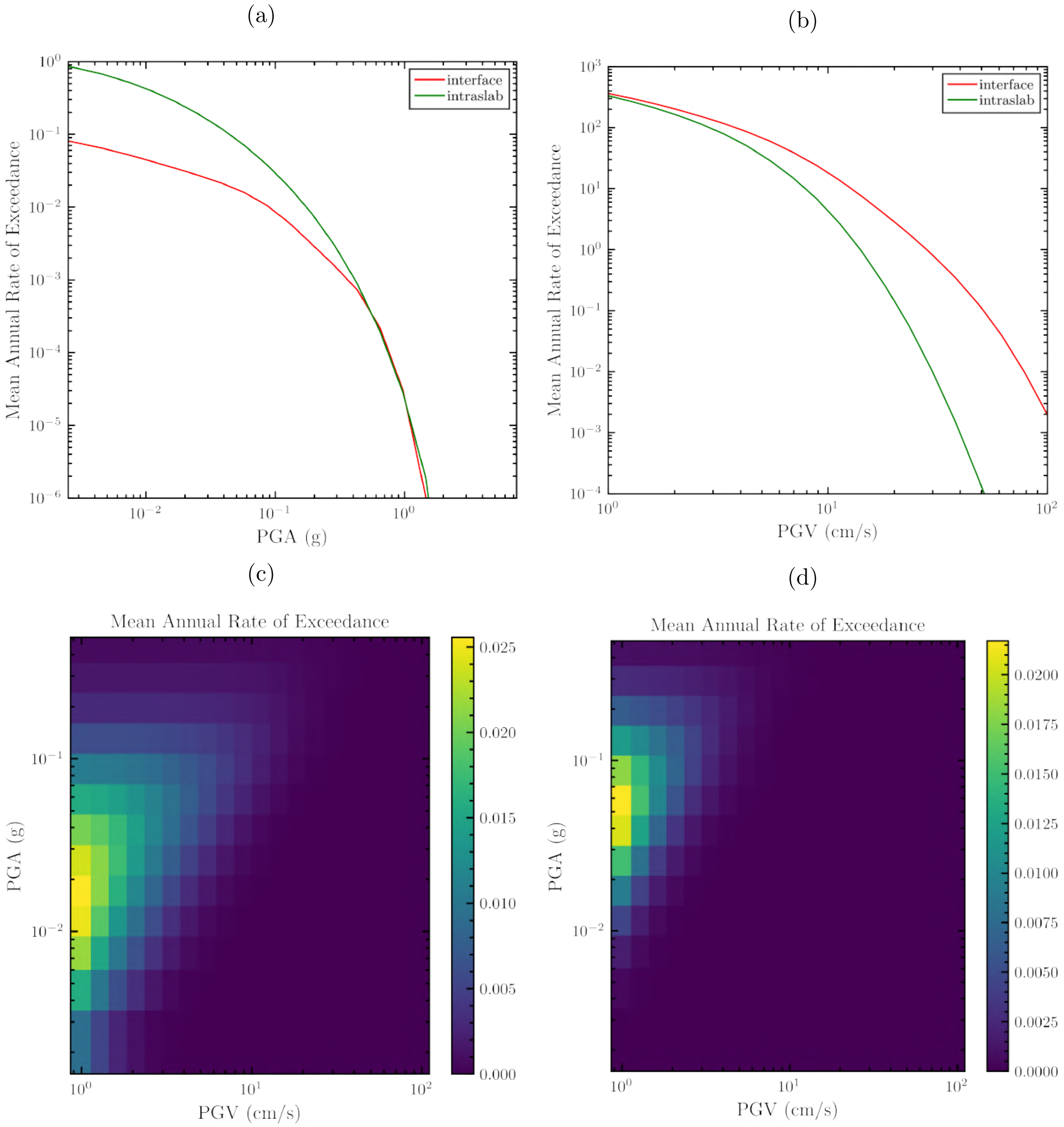


Figure 37: $PGA$ and $PGV$ hazard for the regional assessment in Anchorage, Alaska. (a) marginal hazard curves of $PGA$, (b) marginal hazard curves of $PGV$, (c) joint rate of exceedance for $PGA$ and $PGV$ for interface earthquakes, and (d) joint rate of exceedance for $PGA$ and $PGV$ for intraslab earthquakes

The slope properties and ground motion hazard are used as inputs on the PLSR model to generate seismic landslide maps. We select the PLSR model to illustrate the application of an ML-based model for the regional-based assessment of seismically-induced landslides. The potential of landslide-induced damage is classified into different categories based on the $D$ estimates given a specific hazard level. In this study, we adopt the thresholds used by USGS to produce a seismic landslide map (Wang and Rathje (2015)), in which the landslide hazard is defined as low, moderate, high, and very high if the estimated $D$ with 2% probability of exceedance in 50 years is smaller than 1 cm, between 1 to 5 cm, between 5 to 15 cm, and larger than 15 cm, respectively.

Figure 38 shows the seismic landslide map for the study area in Anchorage for a 2% probability of exceedance in 50 years. It can be observed that only a small fraction of the area has a very high potential for seismically-induced damage. Most areas with high or very high hazard are along coastal bluffs, stream valleys, or mountainous areas. The results in Figure 38 are also consistent with the findings in Wang and Rathje (2015) for the same study area.

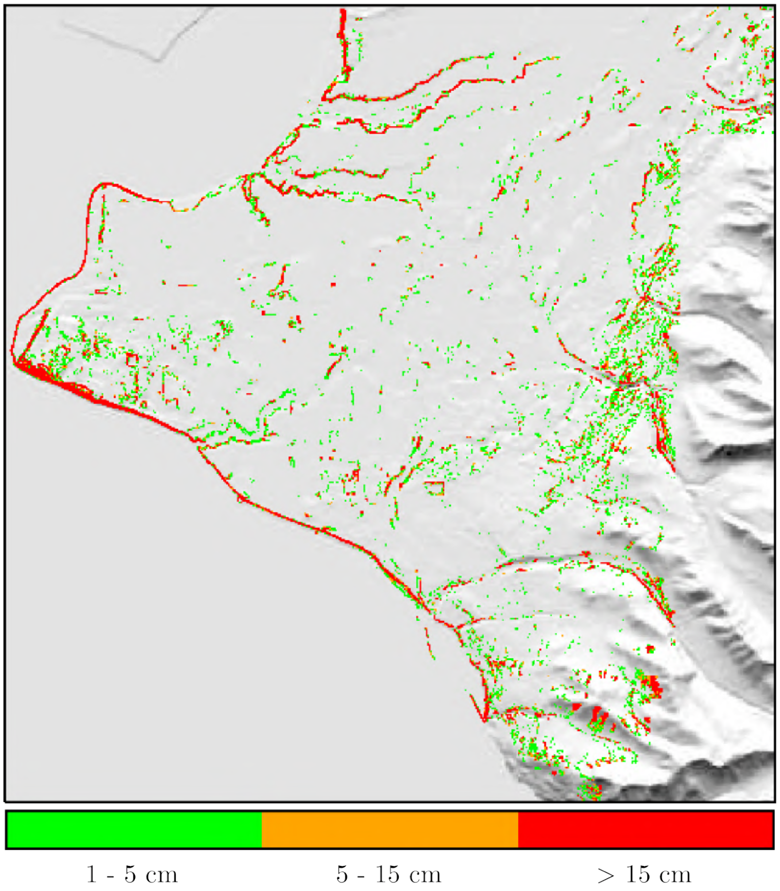


Figure 38: Seismic landslide hazard map of Anchorage at 2% probability of exceedance in 50 years.

## 9 Summary and Conclusions

In this study, we have developed machine-learning-based models for estimating seismically-induced landslides in subduction tectonic settings. The developed models benefit the performance-based assessment of the damage potential of seismically-induced landslides in regions affected by subduction-type earthquakes.

Given the lack of models for estimating seismically-induced slope displacements ($D$) induced by subduction zone earthquakes, we develop traditional and machine-learning-based $D$ models for subduction interface and intraslab earthquakes. It is important to highlight that $D$ is commonly used as a proxy to assess the damage potential of seismically-induced landslides. We use the NGA-Sub ground motion database to generate $D$ realizations with stick-slip coupled Newmark-based analyses that are subsequently used to develop $D$ models. We first develop traditional $D$ models (i.e., models based on polynomial-based fixed functional forms) using functional forms similar to the BMT18 model. The developed traditional model for interface earthquakes shows consistent performance with the BMT18 model, which was also developed for interface earthquakes. However, the traditional intraslab model developed in this study exhibits different trends than the BMT18 model. Moreover, the BMT18 model presents residual trends in intraslab tectonic settings, highlighting the different scaling of $D$ and the need to develop different $D$ models for subduction interface and intraslab tectonic settings.

We also develop machine-learning-based $D$ models using different algorithms, including ridge regression, principal component regression, partial least square regression, random forest, gradient boosting decision tree, support vector regression, and residual neural networks. Before estimating the parameters and hyperparameters in the machine-learning-based $D$ models, we conduct a systematic feature selection among various candidates representing slope properties and ground motion IMs. Specifically, we use forward stepwise regression, LASSO, and random forest algorithms to select efficient features, finding that $k_y$, $T_s$, $M$, $Sa(1.3T_s)$, and $PGV$ are adequate features for estimating $D$. Our findings on efficient features suggest that there is value in implementing/refining capabilities for a rapid assessment of $PGV$ and spectral accelerations using USGS tools (e.g., ShakeMap, Ground Failure) in subduction tectonic settings such as the U.S. Pacific Northwest. These implementations, which could be performed in future efforts, would be useful in the rapid assessment of the damage potential of seismically-induced landslides.

The performance of the machine-learning-based $D$ models is evaluated based on their prediction error on the test set, computational cost, model trends, and case histories (when case histories are available). Based on this assessment, we recommend ridge regression, support vector regression, random forest, gradient boosting decision tree, and residual neural network models as alternative models for estimating $D$ for both interface and intraslab tectonic settings. Of note, in the case of subduction intraslab tectonic settings, we did not find case histories in the public domain with adequate information on ground motions, material properties, and the observed performance. Thus, we encourage the future collection/generation of case histories in this tectonic setting.

We also demonstrate the application of the developed $D$ models to estimate $D$ hazard curves in subduction earthquake zones, which are required in the performance-based assessment of the damage potential of seismically-induced landslides. In addition, we present two approaches to treat the epistemic uncertainties associated with $D$ estimates. The approaches include a logic tree-based treatment of uncertainties, which is commonly used in engineering practice, and a more computationally efficient approach that employs the polynomial chaos framework. Finally, we present several examples to illustrate the application of the machine-learning-based $D$ models developed in this study. First, we conduct a pseudoprobabilistic assessment of $D$ for an area in the Pacific Northwest affected by subduction-type earthquakes. The pseudoprobabilistic approach has been commonly used for assessing the damage potential of seismically-induced landslides due to its simplicity (e.g., Wang and Rathje (2015); Jibson and Tanyaş (2020); Rathje and Saygili (2008); Tsai and Chien (2016); Hsieh and Lee (2011); Jibson (2007)). However, with most applications limited to shallow crustal tectonic settings. Then we illustrate the use of the developed machine learning-based models in performance-based assessments considering two different approaches for the treatment of epistemic uncertainties, namely logic trees, and polynomial chaos expansions. In particular, the computational efficiency of the polynomial chaos approach is highlighted. Lastly, we illustrate the application of one of the developed machine learning-based models in the regional assessment of the damage potential of seismically-induced damage for an area in Anchorage, Alaska that has also been examined in previous studies (Wang and Rathje (2015); Jibson and Michael (2009)), so it is used as a benchmark. The results indicate that only a few areas have a very high damage potential, which is consistent with the findings in Wang and Rathje (2015). The other machine learning models could also be similarly used in regional-based assessments to incorporate additional epistemic uncertainties. In closing, it is essential to highlight that the traditional and machine learning models developed in this study (six models for subduction interface and six models for subduction intraslab) enhance the performance-based assessment of seismically-induced landslide hazards in subduction earthquake zones.

# 10 Project Data

The developed machine-learning-based displacement models have been made readily and permanently available through the DesignSafe data depot (`https://doi.org/10.17603/ds2-d0s9-ew95`). The research results will also be promptly disseminated to the scientific and engineering communities, professional organizations, and the public, in peer-reviewed technical journals and conference proceedings, consistent with USGS expectations.

# References

Abrahamson N, Kuehn N, Gulerce Z, Gregor N, Bozognia Y, Parker G, Stewart J, Chiou B, Idriss I, Campbell K et al. (2018) Update of the BC Hydro subduction ground-motion model using the nga-subduction dataset. *PEER Rept. No* 2: 101.

Amari Si and Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. *Neural Networks* 12(6): 783–789.

Baker JW (2011) Conditional mean spectrum: Tool for ground-motion selection. *Journal of Structural Engineering* 137(3): 322–331.

Bird JF and Bommer JJ (2004) Earthquake losses due to ground failure. *Engineering geology* 75(2): 147–179.

Bishop CM (2006) Pattern recognition and machine learning. *Machine learning* 128(9).

Bonaccorso G (2018) *Machine Learning Algorithms: Popular algorithms for data science and machine learning.* Packt Publishing Ltd.

Bozorgnia Y, Abrahamson NA, Ahdi SK, Ancheta TD, Atik LA, Archuleta RJ, Atkinson GM, Boore DM, Campbell KW, SJ Chiou B et al. (2021) Nga-subduction research program. *Earthquake Spectra* : 87552930211056081.

Bozorgnia Y, Abrahamson NA, Atik LA, Ancheta TD, Atkinson GM, Baker JW, Baltay A, Boore DM, Campbell KW, Chiou BSJ et al. (2014) NGA-West2 research project. *Earthquake Spectra* 30(3): 973–987.

Bozorgnia Y and Stewart J (2020) Data resources for NGA-Subduction project. *PEER Report* 2020/02.

Bray JD and Macedo J (2017) 6th ishihara lecture: Simplified procedure for estimating liquefaction-induced building settlement. *Soil Dynamics and Earthquake Engineering* 102: 215–231.

Bray JD and Macedo J (2019) Procedure for estimating shear-induced seismic slope displacement for shallow crustal earthquakes. *Journal of Geotechnical and Geoenvironmental engineering* 145(12): 04019106.

Bray JD, Macedo J and Travasarou T (2018) Simplified procedure for estimating seismic slope displacements for subduction zone earthquakes. *Journal of Geotechnical and Geoenvironmental Engineering* 144(3): 04017124.

Bray JD and Travasarou T (2007) Simplified procedure for estimating earthquake-induced deviatoric slope displacements. *Journal of geotechnical and geoenvironmental engineering* 133(4): 381–392.

Breiman L (2001) Random forests. *Machine learning* 45(1): 5–32.

Candia G, Macedo J, Jaimes MA and Magna-Verdugo C (2019) A new state-of-the-art platform for probabilistic and deterministic seismic hazard assessment. *Seismological Research Letters* 90(6): 2262–2275.

Candia G, Macedo J and Magna-Verdugo C (2018) An integrated platform for seismic hazard evaluation. In: *11th US National Conference on Earthquake Engineering. Los Angeles, USA*. 11NCEE.

Chen T and Guestrin C (2016) Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. pp. 785–794.

Cho Y (2020a) *Probabilistic assessment of the seismic performance of earth slopes using computational simulation.* PhD Thesis.

Cho Y (2020b) *Probabilistic assessment of the seismic performance of earth slopes using computational simulation.* PhD Thesis.

Cho Y, Khosravikia F and Rathje EM (2022) A comparison of artificial neural network and classical regression models for earthquake-induced slope displacements. *Soil Dynamics and Earthquake Engineering* 152: 107024.

Collins BD and Jibson RW (2015) Assessment of existing and potential landslide hazards resulting from the april 25, 2015 gorkha, nepal earthquake sequence. Technical report, US Geological Survey.

Contreras V, Stewart J, Kishida T, Darragh R, Chiou BJ, Mazzoni S, Kuehn N and Ahdi S (2020) Data resources for nga-subduction project. *PEER Rept. No* 2.

Cortes C and Vapnik V (1995) Support-vector networks. *Machine learning* 20(3): 273–297.

Darendeli MB (2001) *Development of a new family of normalized modulus reduction and material damping curves.* The university of Texas at Austin.

Elgamal AWM, Scott RF, Succarieh MF and Yan L (1990) La villita dam response during five earthquakes including permanent deformation. *Journal of Geotechnical Engineering* 116(10): 1443–1462.

Frankel A, Mueller C, Barnhard T, Leyendecker E, Wesson R, Harmsen S, Klein F, Perkins D, Dickman N, Hanson S et al. (2000) Usgs national seismic hazard maps. *Earthquake spectra* 16(1): 1–19.

Friedman J, Hastie T, Tibshirani R et al. (2001) *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics* : 1189–1232.

Geladi P and Kowalski BR (1986) Partial least-squares regression: a tutorial. *Analytica chimica acta* 185: 1–17.

Greene WH (1981) On the asymptotic bias of the ordinary least squares estimator of the tobit model. *Econometrica: Journal of the Econometric Society* : 505–513.

He K, Zhang X, Ren S and Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* pp. 770–778.

Hochba DS (1997) Approximation algorithms for np-hard problems. *ACM Sigact News* 28(2): 40–52.

Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6(02): 107–116.

Hosmer Jr DW, Lemeshow S and Sturdivant RX (2013) *Applied logistic regression*, volume 398. John Wiley & Sons.

Hsieh SY and Lee CT (2011) Empirical estimation of the newmark displacement from the arias intensity and critical acceleration. *Engineering Geology* 122(1-2): 34–42.

Ioffe S and Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International conference on machine learning.* PMLR, pp. 448–456.

Jha D, Ward L, Yang Z, Wolverton C, Foster I, Liao Wk, Choudhary A and Agrawal A (2019) Irnet: A general purpose deep residual regression framework for materials discovery. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* pp. 2385–2393.

Jibson RW (2007) Regression models for estimating coseismic landslide displacement. *Engineering geology* 91(2-4): 209–218.

Jibson RW and Michael JA (2009) *Maps showing seismic landslide hazards in Anchorage, Alaska.* US Geological Survey Reston, VA, USA.

Jibson RW and Tanyaş H (2020) The influence of frequency and duration of seismic ground motion on the size of triggered landslides—a regional view. *Engineering Geology* 273: 105671.

Jolliffe IT (1982) A note on the use of principal components in regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 31(3): 300–303.

Keefer DL and Bodily SE (1983) Three-point approximations for continuous random variables. *Management Science* 29(5): 595–609.

Kishida T, Contreras V, Bozorgnia Y, Abrahamson N, Ahdi S, Ancheta T, Boore D, Campbell K, Chiou B, Darragh R et al. (2018) NGA-Sub ground motion database. *UCLA Previously Published Works* .

Kobayashi Y (1981) Causes of fatalities in recent earthquakes in japan. *Natural disaster science* 3(2): 15–22.

Kong Q, Trugman DT, Ross ZE, Bianco MJ, Meade BJ and Gerstoft P (2019) Machine learning in seismology: Turning data into insights. *Seismological Research Letters* 90(1): 3–14.

Kuehn N, Bozorgnia Y, Campbell K and Gregor N (2020) Partially Non-Ergodic Ground-Motion Model for Subduction Regions using NGA-Subduction Database. *PEER Rept. No* 2020/04.

Lacour M and Abrahamson NA (2019) Efficient propagation of epistemic uncertainty in the median ground-motion model in probabilistic hazard calculations. *Bulletin of the Seismological Society of America* 109(5): 2063–2072.

LeCun Y, Bengio Y and Hinton G (2015) Deep learning. *nature* 521(7553): 436–444.

LeCun Y, Bengio Y et al. (1995) Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361(10): 1995.

Liu C and Macedo J (2021) New conditional, scenario-based, and non-conditional cumulative absolute velocity models for subduction tectonic settings. *Earthquake Spectra* : 87552930211043897.

Liu C and Macedo J (2022) New conditional, scenario-based, and traditional peak ground velocity models for interface and intraslab subduction zone earthquakes. *Earthquake Spectra* : 87552930211067817.

Liu C, Macedo J and Candia G (2021) Performance-based probabilistic assessment of liquefaction-induced building settlements. *Soil Dynamics and Earthquake Engineering* 151: 106955.

Macedo J, Abrahamson N and Liu C (2020a) New scenario-based cumulative absolute velocity models for shallow crustal tectonic settings. *Bulletin of the Seismological Society of America* .

Macedo J, Bray J, Abrahamson N and Travasarou T (2018) Performance-based probabilistic seismic slope displacement procedure. *Earthquake spectra* 34(2): 673–695.

Macedo J, Bray J and Travasarou T (2017) Simplified procedure for estimating seismic slope displacements in subduction zones. In: *Proc., 16th World Conf. on Earthquake Engineering.*

Macedo J and Candia G (2019a) Performance-based assessment of seismically-induced slope displacements in tailings dams. In: *Tailings and mine waste 2019 conference*, Vancouver. p. Canada.

Macedo J and Candia G (2019b) Performance-based assessment of the seismic pseudostatic coefficient in mining projects. In: *Tailings and mine waste 2019 conference*, Vancouver. p. Canada.

Macedo J and Candia G (2020) Performance-based assessment of the seismic pseudo-static coefficient used in slope stability analysis. *Soil Dynamics and Earthquake Engineering* 133: 106109.

Macedo J, Candia G, Lacour M and Liu C (2020b) New developments for the performance-based assessment of seismically-induced slope displacements. *Engineering Geology* 277: 105786.

Macedo J, Lacour M and Abrahamson N (2020c) Epistemic uncertainty treatment in seismically induced slope displacements using polynomial chaos. *Journal of Geotechnical and Geoenvironmental Engineering* 146(10): 04020111.

Macedo J and Liu C (2021) Ground-motion intensity measure correlations on interface and intraslab subduction zone earthquakes using the nga-sub database. *Bulletin of the Seismological Society of America* 111(3): 1529–1541.

Macedo J, Liu C and Soleimani F (2021) Machine-learning-based predictive models for estimating seismically-induced slope displacements. *Soil Dynamics and Earthquake Engineering* : 106795.

Massey C, Townsend D, Rathje E, Allstadt KE, Lukovic B, Kaneko Y, Bradley B, Wartman J, Jibson RW, Petley D et al. (2018) Landslides triggered by the 14 november 2016 mw 7.8 kaikōura earthquake, new zealandlandslides triggered by the 14 november 2016 mw 7.8 kaikōura earthquake, new zealand. *Bulletin of the Seismological Society of America* 108(3B): 1630–1648.

Mikolov T, Karafiát M, Burget L, Cernockỳ J and Khudanpur S (2010) Recurrent neural network based language model. In: *Interspeech*, volume 2. Makuhari, pp. 1045–1048.

Press WH and Teukolsky SA (1990) Savitzky-golay smoothing filters. *Computers in Physics* 4(6): 669–672.

Probst P, Wright MN and Boulesteix AL (2019) Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9(3): e1301.

Raschka S (2018) Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *Journal of open source software* 3(24): 638.

Rathje E and Bray J (2000) An examination of simplified earthquake-induced displacement procedures for earth structures: Reply. *Canadian Geotechnical Journal* 37(3): 731–732.

Rathje EM (2014) Probabilistic seismic landslide maps: Implementation project. *USGS Final Report* G12AP20083.

Rathje EM and Antonakos G (2011) A unified model for predicting earthquake-induced sliding displacements of rigid and flexible slopes. *Engineering Geology* 122(1-2): 51–60.

Rathje EM and Saygili G (2008) Probabilistic seismic hazard analysis for the sliding displacement of slopes: scalar and vector approaches. *Journal of Geotechnical and Geoenvironmental Engineering* 134(6): 804–814.

Rathje EM and Saygili G (2009) Probabilistic assessment of earthquake-induced sliding displacements of natural slopes. *Bulletin of the New Zealand Society for Earthquake Engineering* 42(1): 18–27.

Saygili G and Rathje EM (2008) Empirical predictive models for earthquake-induced sliding displacements of slopes. *Journal of geotechnical and geoenvironmental engineering* 134(6): 790–803.

Smola AJ and Schölkopf B (2004) A tutorial on support vector regression. *Statistics and computing* 14(3): 199–222.

Tibshirani R (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1): 267–288.

Tsai CC and Chien YC (2016) A general model for predicting the earthquake-induced displacements of shallow and deep slope failures. *Engineering Geology* 206: 50–59.

Wahba G et al. (1999) Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning* 6: 69–87.

Wang MX, Huang D, Wang G and Li DQ (2020) Ss-xgboost: a machine learning framework for predicting newmark sliding displacements of slopes. *Journal of Geotechnical and Geoenvironmental Engineering* 146(9): 04020074.

Wang Y and Rathje EM (2015) Probabilistic seismic landslide hazard maps including epistemic uncertainty. *Engineering Geology* 196: 313–324.

Williams C and Seeger M (2001) Using the nyström method to speed up kernel machines. In: *Proceedings of the 14th annual conference on neural information processing systems*, CONF. pp. 682–688.

Xie Y, Ebad Sichani M, Padgett JE and DesRoches R (2020) The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra* 36(4): 1769–1801.