Machine Learning Aided Reliability Analysis for Spatially Varying Slopes in 2D and 3D

J.S. Vermeer

Delft University of Technology

TU Delft

Machine Learning Aided Reliability Analysis for Spatially Varying Slopes in 2D and 3D

by

J.S. Vermeer

in partial fulfillment to obtain the degree of

Master of Science

at the Delft University of technology
Faculty of Civil Engineering and Geosciences

to be defended publicly on Wednesday April 17, 2024 at 11:00 AM.

Student number:	4712447	
Project duration:	August, 2023 – April, 2024	
Thesis committee:	Prof. dr. ir. M.A. Hicks,	TU Delft, supervisor, chair
	Dr. ir. G. Rongier,	TU Delft
	W. Huang, MSc.	TU Delft

TU Delft

Preface

This document not only marks the end of my master's in civil engineering but also marks the end of my time as a student in Delft. Reflecting on the past 6.5 years, I have learned a lot, both academically and about myself. What do I enjoy? And what do I not enjoy? Reflecting on these questions led me to take a detour last year and pursue another master's degree in business analytics in Rotterdam. This study sparked the topic of this thesis, combining my interests in machine learning with geotechnical engineering.

I would like to take this opportunity to thank my thesis committee. First of all, I would like to thank Prof. Michael Hicks for his valuable comments during this project, for his guidance, and for his research articles that shaped the basis for this work. I would also like to thank Dr. Guillaume Rongier for his extensive feedback and for giving new perspectives to the project, which were of great value. Additionally, I would like to thank Wei Huang for always being available for a discussion, for giving detailed feedback comments, and for our weekly meetings, which I always enjoyed.

Finally, I would like to express my gratitude to my friends and family for the relaxing moments and their support during this time. I would like to thank my parents in particular, for always giving me the space to do what I want to do. Thank you!

J.S. Vermeer
Delft, April 2024

Summary

The Random Finite Element Method (RFEM) is a robust stochastic method for slope reliability analysis that incorporates the spatial variability of soil properties. However, the extensive computational time associated with the direct Monte Carlo simulation limits its practical application. To overcome this problem, this study investigates the use of machine learning (ML) models as surrogate models for the RFEM in both 2D and 3D contexts. It investigates the performance of two combined ML models– PCA-SVR and PCA-RF– and a Convolutional Neural Network (CNN) in predicting slope stability by means of the *factor of safety* (FoS) based on a generated random field of the undrained shear strength. Additionally, a data augmentation technique is employed to improve performance. The models' performance is assessed for various slope cases, characterised by varying spatial variability.

Two surrogate modeling approaches are employed: semi-surrogate modeling and full-surrogate modeling. In the semi-surrogate modeling approach, a small number of RFEM simulations are conducted for a specified case. The machine learning models are trained using the generated random fields as input data and the calculated factors of safety as output data. The mathematical models are then used to predict outcomes of FoS for a large number of random fields for the same specific slope case. In the full-surrogate modeling approach, many RFEM simulations are conducted for the training set, covering a range of spatial correlation lengths. Once trained, the full-surrogate models are ready for application to another different slope case without the need for any additional numerical simulation.

The results show that the best-performing semi-surrogate ML model, in both 2D and 3D contexts, varies depending on the user's objective. For accurate prediction of the factor of safety, PCA-SVR, when combined with data augmentation, exhibits the best performance. In terms of accurately predicting the probability of slope failure, the CNN outperforms the others. Among the full-surrogate models, the CNN consistently shows the best performance for both objectives, in both 2D and 3D contexts.

The results also indicate that the prediction accuracy of the ML models typically decreases for slope cases with smaller scales of fluctuation. Nonetheless, the FoS predictions by the best-performing semi-surrogate model are highly consistent with the results from RFEM simulations for the whole range of considered slope cases. In terms of predicting the probability of failure for 2D-modeled slopes, the accuracy is high, with relative errors within 10% across the cases considered. This level of accuracy is achieved using no more than 13% of the total number of realisations needed for RFEM analysis. Consequently, the computational time for reliability analysis involving 4000 realisations reduces from 67 hours using the RFEM to between 4 and 8 hours using a semi-surrogate model, with the time increasing as the spatial correlation length decreases. Predicting the p_f for 3D slopes using a semi-surrogate model showed larger errors, indicating a need for improvement.

The full-surrogate models prove to be accurate for testing cases characterised by spatial correlation lengths within the training set's range. Notably, the best-performing full-surrogate model in 3D predicted the p_f within a relative error of 10% for two slope cases. This model performs a stochastic analysis of 4000 simulations within seconds, compared to 83 days of computational time required for RFEM reliability analysis.

Contents

1

Introduction

1.1. Background and Motivation

In the field of geotechnical engineering and environmental safety, slope stability analysis assesses the potential for ground movement or failure in terrains with inclined surfaces, acting as a preventive measure to safeguard structures, infrastructure, and human lives against the harmful impacts of landslides or other slope-induced failures.

Slope stability analysis

Traditionally, slope stability is indicated by the factor of safety (FoS), which is the ratio between the available shear strength and the acting shear stress along a potential sliding surface. It is either calculated using limit equilibrium methods (e.g. [27, 54]) with a predefined slip surface (circular, plane, logarithmic etc.) or numerical analysis methods (e.g. [20, 47]), such as the finite element method (FEM). In these deterministic analyses, properties of a soil type are modeled using a fixed set of parameters, thereby omitting the random spatial variability of these properties natural soils have [39]. Doing so, only a single value of the FoS is obtained, which does not reveal the risks associated with the geotechnical problem in a comprehensive way. As a result, engineers usually choose very conservative soil property values during design and assessment, leading to a FoS » 1, and thereby leading to overconservative and uneconomic designs.

A modeling solution that reveals the associated risks and reliability of a slope and that has arisen in recent years is the random finite element method (RFEM). This method combines the random field theory [45] with the FEM. Vanmarcke [45]'s random field theory characterizes the spatial variability in geotechnical properties by treating them as continuous random fields. A random field is described by a certain statistical distribution and a spatial correlation function, which collectively capture the inherent uncertainty in a spatially variable property. By applying this theory, it becomes possible to incorporate the spatial randomness inherent in natural materials directly into the FEM, offering a more realistic representation of the underlying physical systems.

Using the Monte Carlo simulation (MCS) technique, with each FEM simulation having a different realisation of random field(s), one can obtain the distribution of the FoS and subsequently assess the probability of failure (p_f) [16].

For a meaningful reliability analysis, many realisations in the MCS are needed, especially when one deals with a slope that has a small probability of failure. Specifically, the coefficient of variation (COV) of the p_f across multiple MCSs can be approximated by a function of the number of required realisations N_{MC} in each MCS and the mean p_f value [5]:

$$COV[p_f] = \sqrt{\frac{1 - p_f}{p_f N_{MC}}} \tag{1.1}$$

A reasonable consistent probability of failure is often considered to have a COV of 0.1 [34, 28, 49]. This requires the number of simulations to be on the order of $100/p_f$. Given that each simulation might take several minutes depending on the specific scenario, the total computational time needed to carry

out an RFEM analysis can be problematic. This drawback of the RFEM technique is one of the biggest reasons that it is not yet widely adopted for realiability analysis.

Surrogate modeling

To overcome the computational demand of the RFEM, researchers have made efforts to reduce the number of simulations (eg. subset-sampling [33, 28, 26] and importance sampling [36, 53]) or speed up each simulation. For the latter, much attention has concentrated on so-called surrogate modeling. Here, a mathematical approximation function of the system response is sought, which can then replace the physics-based time-consuming simulations in an MCS.

Researchers working on surrogate modeling have mostly adopted the so-called polynomial response surfaces [29], the Spectral Stochastic Finite Element Method (SSFEM) [14], and Machine Learning (ML) methods [19, 18, 3, 4, 25, 49, 35].

Recently, ML methods have been increasingly researched as a surrogate model for RFEM because the amount of data and computational power have grown. It has been shown that ML models can be a promising surrogate model for a select number of slope reliability assessment cases (e.g. [50, 19, 18]). However, three key gaps remain:

1. While a systematic evaluation has been conducted across a range of anisotropy levels for ML-aided slope reliability analysis in two dimensions (2D), it only employed the stable/fail classification [4]. This way, there remains a gap in understanding how ML models perform when the target is the FoS, for various slopes cases characterised with different spatial variability levels. This emphasis on the FoS as the target variable is relevant because a post-processing technique can then be applied to fit the predicted FoS distribution, to gauge the lower end of the distribution accurately. This is needed for analyses of slopes with a low probability of failure, which are of particular interest to engineers.
2. Prior research on ML-aided reliability analysis mainly focused on 'semi-surrogate' modeling. That is, a subset of simulations on a specific case is used to train a ML model, after which the ML model is then used to predict the remainder of simulations. In practice, this means that one still needs to perform time-consuming numerical simulations. Therefore, there is an interest in understanding the performance of a 'full-surrogate' model. Such a model, once established, can be directly used for application to a broader range of slope scenarios.
3. To the best of the author's knowledge, prior research has only focused on the use of ML methods as (semi-)surrogate models for reliability analyses of spatially varying slopes in 2D. Slope reliability analyses in 2D are often preferred over those in three-dimensions (3D) due to their simplicity and lower computing power requirements.

   However, the foundational assumption for 2D analysis, the plane strain assumption, implies that deformation occurs uniformly along the length of the slope, which is generally not the case. In reality, failure mechanisms tend to follow the path of least resistance, which may not extend uniformly across the entire slope in three dimensions. Consequently, the reliability of 2D and 3D slope analyses can be notably different. Moreover, the reliability analysis in 3D reveals the potential deformation and slide volume from a more practical perspective.

   Additionally, as pointed out earlier, the computational power and time required for 3D analysis are much greater than for 2D analyses. A multiple of the number of elements used in 2D slope modeling is needed in a 3D simulation, amplifying the computational demand. If the performance of ML (semi-)surrogate models is comparable for 3D as compared to those previously applied in 2D analyses, the total amount of computing power and time could then be reduced more significantly.

   Taken together, there's a need for research on ML-aided reliability analysis of spatially varying slopes in 3D, which has also been addressed in recent literature reviews (published in 2023) of machine learning in geotechnical reliability analyses by Kumar et al. [32], Xu et al. [51] and by Zhang et al. [52].

1.2. Objectives

This research has two objectives:

1. **Assess the performance of ML-aided reliability analysis for 2D slopes, considering various slope cases characterised with different scales of fluctuation.** This assessment will use the minimum, mean, and maximum scales of fluctuation for the strength property, as encountered in practice and documented by Phoon and Kulhawy [39].

   First, it starts with semi-surrogate modeling using various ML models. By directly comparing the performance of various ML models on several slope cases with varying spatial variability levels, a better understanding of ML semi-surrogate modeling in RFEM slope reliability analysis is achieved.

   Subsequently, a full-surrogate model will be developed. This model is designed for immediate application to broader scenarios not covered in the training set, and its performance will be evaluated across various cases.
2. **Explore and evaluate how ML models perform as semi-surrogate and full-surrogate models for the RFEM in 3D slope reliability analysis.** This involves investigating various ML models to predict the FoS using generated random fields of the strength property. Both semi-surrogate and full-surrogate modeling will be investigated. The performance of these surrogate models will be assessed by comparing them with complete MCSs for various slope cases, characterised by different scales of fluctuations in the strength property.

1.3. Research Scope and Simplifications

This research specifically targets 2D and 3D slopes with basic geometry, excluding the phreatic surface, soil layering, and the use of advanced soil models. This focused approach serves two main purposes. First, it allows for a direct comparison with earlier 2D studies that integrated ML-aided reliability analysis and typically shared these characteristics. In past literature on this topic, either only the undrained shear strength was used as the main soil strength input parameter [3, 4, 18, 48], or both cohesion and friction angle were considered for drained behavior [18, 50, 19]. Secondly, due to the extensive computational demands of 3D RFEM simulations, these streamlined conditions allow for a focused investigation into how the spatial variability influences the efficacy of each ML model as a surrogate model.

While there are numerous ML models available, this study will only assess the most promising ones due to time constraints. For the 2D reliability assessment, previously-used ML models for aided-reliability analysis are examined, as well as a new method. For 3D ML-aided reliability analysis, which is a new contribution from this work, the 2D ML models will be modified to handle 3D inputs.

1.4. Thesis Outline

This thesis is divided into the following chapters:

Chapter 2 discusses the fundamental concepts that underpin this thesis. This includes a brief description of the RFEM used for reliability analyses of slopes, and the basic concepts of machine learning, as well as the conceptual fundamentals of the selection of ML models used.

Chapter 3 provides a literature review on (semi-)surrogate modeling for the RFEM using ML models.

Chapter 4 assesses the potential of ML models to be a semi-surrogate and full-surrogate model for the RFEM, applied to a slope modeled in 2D. This way, the soil spatial variability in third dimension is ignored, and consequently, the influence of it on the system response. Several slope cases, each characterised by different spatial variability, are used to explore the performance of the ML (semi-)surrogate models. Additionally, a data augmentation technique is employed to potentially improve ML performance.

Chapter 5 extends the slope reliability analysis to 3D. Just as in Chapter 4, several slope cases, each characterised by different spatial variability, are used to explore the performance of the ML (semi-)surrogate models. Additionally, a new strategy for creating a full surrogate model is investigated.

Chapter 6 summarises the main conclusions of this thesis, and gives recommendations for future research.

2

Background Fundamentals

This chapter delves into the foundational concepts that underpin this thesis. The basic theory of the Random Finite Element Method (RFEM) in geotechnical engineering is introduced and discussed in Section 2.1. Next, Section 2.2 describes the basic working principle of the machine learning algorithms used in this research.

2.1. Random Finite Element Method

The Random Finite Element Method (RFEM), introduced by Fenton and Griffiths [12], was developed to account for the spatial variability of soil [39]. The RFEM involves generating random fields of soil properties with certain point- and spatial variability, which is mapped onto a finite element mesh. Afterwards, the finite element method (FEM) is used to compute the variables of interest. This process is repeated multiple times using a Monte Carlo approach. Once a sufficient number of simulations are performed, one can obtain the distribution of the variable of interest. Each of these main components, i.e. the random field, point- and spatial variability, FEM, and Monte Carlo Simulation is described next.

2.1.1. Random fields

The Random Field Theory (RFT), as proposed by Vanmarcke [46], has become an important tool in addressing the uncertainties inherent in soil properties. These uncertainties stem from two primary sources. First, soil investigations, aimed at determining soil properties, are conducted at limited locations. It is impractical and uneconomical to test at every possible location. Secondly, measurement and testing errors can occur, which reduces the reliability of the data. Consequently, analyses and decision-making usually proceed with incomplete knowledge of the site, highlighting the importance of modeling the medium as a random field.

A random field assigns a random value to each point within a continuous domain, encompassing an infinite set of random variables. However, from a computational standpoint, it is needed to represent the random field with a finite set of variables. RFT accomplishes this by discretizing the continuum of a soil body into a series of correlated random variables. In practical terms, this means that a continuous spatial domain (e.g. a slope) is first transformed into a discretized grid or mesh. Each node or element on this grid is then assigned a soil property value consistent with predefined point and spatial statistics. This approach not only allows for the representation of localized variations in properties but also effectively captures the intrinsic spatial correlations between them. A visualisation of four Gaussian random fields using the same input statistics is shown in Figure 2.1.

Several techniques exist for generating random fields. These can be classified into point discretisation methods, average discretisation methods, and series expansion methods [43]. From the average discretisation methods, the Local Average Subdivision method is highlighted as it will be used in this research.

Local Average Subdivision

The LAS method, as developed by Fenton and Vanmarcke [13], represents a discretised random field in each mesh element as an average of the original field over that element. Constructing a random field

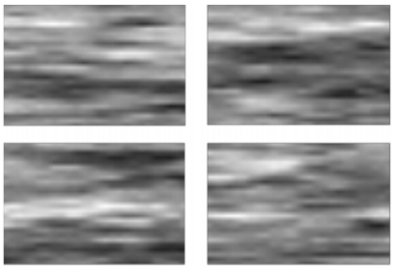


Figure 2.1: Four generated Gaussian random fields of a soil parameter based on the same input statistics. From Hicks and Samy [23].

using the LAS technique follows a top-down approach, as illustrated in Figure 2.2 for 1D discretization:

1. **Stage 0:** A global average Z_1^0 is generated from a predefined distribution model.
2. **Stage 1:** The domain is divided into two equally sized regions, with local averages Z_1^1 and Z_2^2. The two values are determined such that their average equals the parent value, they are appropriately correlated with one another, and the correct variance according to local averaging theory is maintained.
3. **Stage 2:** Each region from Stage 1 is further split into two equal parts. The same criteria as in Stage 1 are upheld. Additionally, the random variables Z_1^2 and Z_2^2 should be properly correlated with Z_3^2 and Z_4^2.
4. **Subsequent stages:** These stages follow the same procedure as Stage 2.

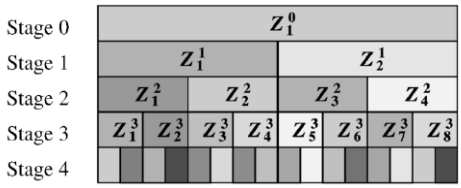


Figure 2.2: Top-down approach to LAS construction in 1D. From Fenton and Vanmarcke [13].

In this procedure, a single material property is mapped onto a FEM mesh. However, geotechnical problems often require the description of multiple material properties. To address this, one can either use a multivariate approach or a reduced-variate approach [20]. In the multivariate approach, a separate random field is generated for each property. In this approach, random fields can also be generated to be cross-correlated to each other. For example, cohesion and friction angle have been reported to be negatively correlated (eg. [11] [8]). In the reduced-variate approach, the total number of random fields generated is reduced, and other parameters are back-figured. For example, the permeability can be backfigured from a generated random field of porosity.

Advantages of the LAS method include its simplicity, the ability to produce scale-dependent realisations, and the absence of aliasing and symmetric covariance problems commonly encountered with other methods [13]. However, a limitation of this method is that it is difficult to use probability density functions other than Gaussian to describe the random variables [37].

2.1.2. Point- and spatial variability

Soil is a complex material shaped through an interplay of diverse geologic, environmental, and physicochemical processes. Many of these processes remain active and can alter the soil in its natural state. Due to these ongoing processes, soil properties exhibit variations both vertically and horizontally [39]. In geotechnical problems, these characteristics of a site are gauged on the basis of geotechnical tests, including laboratory and in-situ tests. To assess the vertical variation of a soil parameter, the soil parameter profile obtained from a geotechnical test is first decomposed into a trend function and a fluctuation component:

$$\zeta(z) = \mu(z) + w(z) \tag{2.1}$$

where, $\zeta(z)$ is the in-situ soil property at depth z, $\mu(z)$ the trend function, and $w(z)$ the fluctuating component. The standard deviation of the fluctuating component is defined by:

$$\sigma = \sqrt{E[w^2]} = \sqrt{\int_{-\infty}^{+\infty} w^2 f(\xi) dx} \tag{2.2}$$

where $f(\xi)$ is the probability density of the in-situ soil property. Since measurements are finite, one can obtain the standard deviation by the discretized equation:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (w(z_i))^2} \tag{2.3}$$

where, n equals the number of measurement points, and $w(z_i)$ is the fluctuation at depth z_i. More often, the inherent point variability is described by the dimensionless coefficient of variation (COV), which is defined as the standard deviation divided by the mean:

$$COV = \frac{\sigma}{\mu} \tag{2.4}$$

A common metric that is used to describe the inherent spatial variability is the scale of fluctuation, also known as the spatial correlation length. This metric describes the distance in space over which the soil parameter is similar or correlated. It is defined as the area under the autocorrelation function ρ [46]:

$$\theta = \int_{-\infty}^{+\infty} \rho(\tau) d\tau \tag{2.5}$$

where τ is the distance between two points in space. The autocorrelation function describes the correlation between the values of the same soil parameters measured at two different locations. Researchers try to estimate the theoretical correlation model using various techniques, such as the method of moments or the maximum-likelihood estimation. This thesis will not delve into the specifics of these techniques, as they are beyond its scope. A visualization of the scale of fluctuation and the point statistics is given in Figure 2.3.

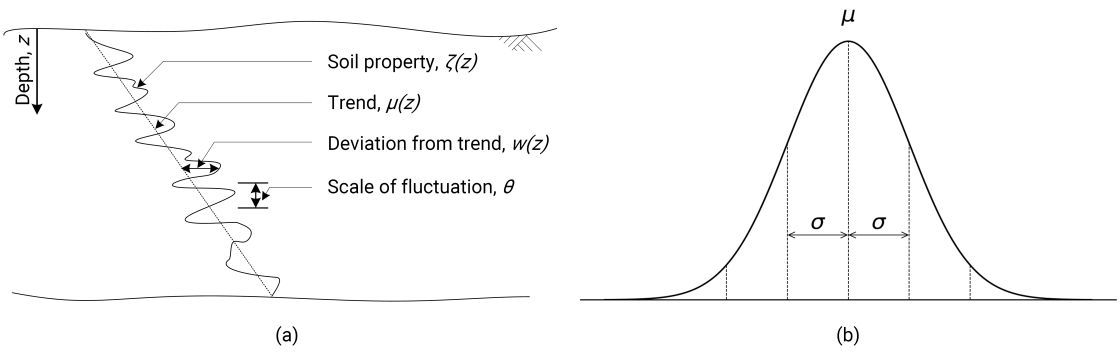


Figure 2.3: a): Inherent soil variability with depth. Adapted from Phoon and Kulhawy [39]. b) Point statistics under the assumption of a Gaussian distribution.

The vertical scale of fluctuation (θ_v) can be estimated from in-situ tests, such as the cone penetration test (CPT).

To express the anisotropy of a certain domain, researchers often use the anisotropy ratio, defined by:

$$\xi = \frac{\theta_h}{\theta_v} \tag{2.6}$$

To estimate the horizontal scale of fluctuation (θ_h), multiple closely-spaced in-situ tests need to be compared at different depths. If θ approaches zero, all points within the domain become uncorrelated, resulting in an infinitely rough field. Conversely, as θ increases, the soil property field smoothens, displaying less spatial variability and ultimately approaching uniformity as θ tends towards infinity. An illustration of this is given in Figure 2.4.

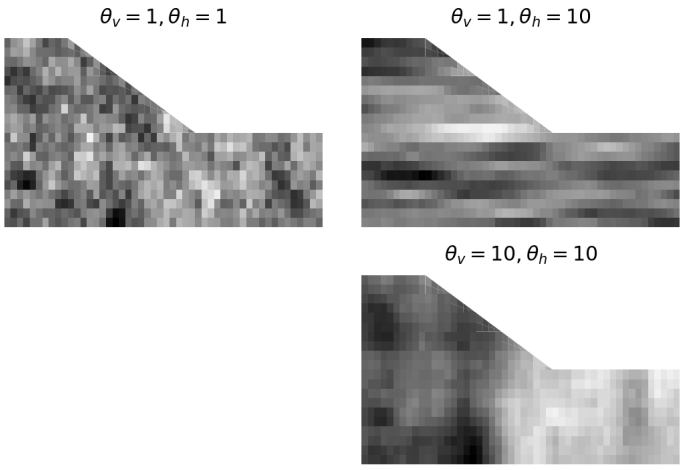


Figure 2.4: Influence of the vertical and horizontal scale of fluctuation (θ_v and θ_h) in meters on the spatial distribution of a soil parameter. Darker zones represent higher values.

2.1.3. Finite Element Method

The Finite Element Method (FEM) is a powerful numerical technique for solving partial differential equations (PDEs) that describe physical phenomena. It subdivides a large system or domain into smaller, simpler parts known as *elements*. By approximating the solution locally within each element and then assembling them globally, FEM transforms the complex differential problem into a more manageable algebraic system. This method has found wide-ranging applications in engineering, physics, and other disciplines, especially for problems involving complex geometries, boundary conditions, and material behaviors. For the details of the working principle, the reader is referred to Reddy [40].

Determination of the factor of safety

For the specific application of calculating the factor of safety (FoS) of a slope using the FEM, the Shear Strength Reduction Method (SSRM) is used. Initially, gravity loading is applied to the soil body to generate in-situ stresses. The shear stresses are then checked against the Mohr-Coulomb failure criterion and excess stresses are iteratively redistributed throughout the model. The shear strength parameters are reduced in a subsequent step by a factor, called Shear Strength Reduction Factor (SRF), and the loading-equilibrium process is repeated. The factored shear strength parameters are given by:

$$\begin{aligned} c_f &= \frac{c}{SRF} \\ \tan\phi_f &= \frac{\tan\phi}{SRF} \end{aligned} \tag{2.7}$$

where, c_f is the reduced cohesion, and ϕ_f is the reduction friction angle.

The lowest SRF to trigger failure is taken as the factor of safety (FoS) of the slope for that realisation. In this study, failure is indicated whenever the FEM solution isn't able to converge within a predetermined number of stress-redistribution iterations set by the user. This implies that a stress distribution that simultaneously meets the Mohr-Coulomb failure criterion and global equilibrium cannot be found.

The FoS at the moment of failure can be represented using a modified version of Eq. 2.7 as:

$$\begin{aligned} c_f &= \frac{c}{FoS} \\ tan\phi_f &= \frac{tan\phi}{FoS} \end{aligned} \tag{2.8}$$

Upon rewriting Eq. 2.8, the equation becomes:

$$\frac{tan\phi_f}{c_f} = \frac{tan\phi}{c} \tag{2.9}$$

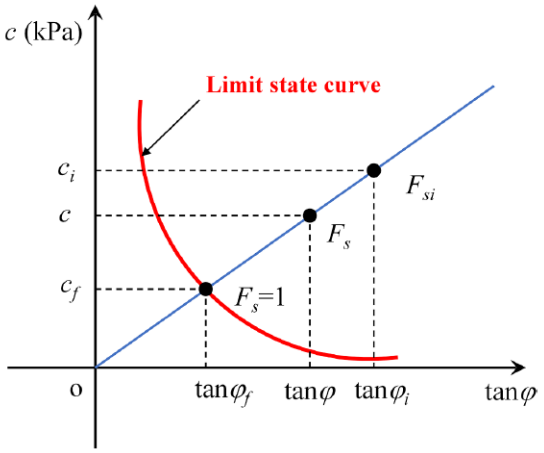


Figure 2.5: Theoretical relationship between the strength reduction factor with the soil shear strength parameters. From Jiang et al. [30].

This equation suggests that within $c - \phi$ space, there exists a linear relationship between original and reduced shear strength parameters. This association is illustrated in Figure 2.5. Every point on the theoretical linear line can be derived for any given factor of safety FoS_i by:

$$\begin{aligned} c_f &= \frac{c_i}{FoS_i} \\ tan\phi_f &= \frac{tan\phi_i}{FoS_i} \end{aligned} \tag{2.10}$$

Combining Eqs. 2.10 and 2.8 gives an equation that can calculate the shear strength parameters c_i and ϕ_i for any given factor of safety FoS_i:

$$\begin{aligned} tan\phi_i &= \frac{FoS_i}{FoS}\, tan\phi \\ c_i &= \frac{FoS_i}{FoS}\, c \end{aligned} \tag{2.11}$$

The given equation shows that one can determine the shear strength parameters for any chosen factor of safety by using a known set of shear strength parameters and its related factor of safety. This method allows for the creation of new sets of shear strength parameters and their corresponding factors of safety without additional numerical calculations. Eq. 2.11 forms the mathematical basis of the data augmentation technique used in this study, which was originally proposed by Jiang et al. [30].

2.1.4. Monte Carlo Simulation

The Monte Carlo Simulation (MCS) is a computational technique that uses random sampling to obtain numerical results for problems that are deterministic in principle. The RFEM uses the Monte Carlo approach, wherein soil parameters in each simulation are described as random fields, accounting for the spatial variability present at the considered site.

As the locations of weak and strong zones vary in each random field, each simulation leads to a different FEM solution. In reliability analysis, the primary variable of interest is the probability of failure p_f, which is calculated as the total number of failed slopes divided by the total number of simulations:

$$p_f = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} I[FoS_i < 1] \tag{2.12}$$

where N_{MC} represents the total number of simulations, FoS_i denotes the factor of safety computed by the FEM for the ith realisation, and $I[\cdot]$ is an indicator function determining slope failure. If FoS_i is

below a threshold of 1, $I[\cdot] = 1$; otherwise, $I[\cdot] = 0$. Conversely, reliability is computed as the chance of not failing:

$$R = 1 - p_f \tag{2.13}$$

The coefficient of variation of the probability of failure can then be estimated without having to do many MCSs by [6]:

$$COV[p_f] = \sqrt{\frac{1 - p_f}{p_f N_{MC}}} \tag{2.14}$$

The number of simulations in a full MCS should be sufficient to ensure that the value for the p_f has converged. Past studies have typically considered $COV[p_f]$ values around 0.1 to be sufficient [34, 28, 49].

In order to quantify and compare the efficiency of different probabilistic methods to the full MCS, the unit coefficient of variation (Unit COV) is often calculated hereafter. In essence, it is a measurement of the accuracy achieved and the computational effort spent. Smaller values of the Unit COV correspond to higher computational efficiency. It is calculated by:

$$Unit\ COV = COV[p_f] * \sqrt{N_{sim}} \tag{2.15}$$

where, N_{sim} equals the number of model simulations.

2.2. Machine Learning Methods

Having introduced the fundamentals of the RFEM, this section discusses potential ML surrogate models. Specifically, the aim is to develop a surrogate model that accurately predicts the FoS, referred to as the *label*, by processing the input values, known as *features*. In this research, the features in the RFEM analysis for a 'simple' undrained slope stability assessment consist of the strength parameter for every random field cell in space. The goal is to discover a function $\hat{f}$ that best approximates the label y using the features x, which can be mathematically represented as:

$$\hat{f} = \underset{f}{\text{argmin}}\ ||f(\vec{x}) - y|| \tag{2.16}$$

In ML, the approximation function in this optimisation problem is sought on a training set comprising feature-label pairs. Once the approximation function is identified, it can be used to forecast the label for new feature values.

This section describes the fundamental working principles of a selection of ML algorithms, which are identified as potential (semi-)surrogate models for the RFEM. The Principal Component Analysis is covered in Section 2.2.1, the Support Vector Regression in Section 2.2.2, the Random Forest regressor in Section 2.2.3 and the Convolutional Neural Network in Section 2.2.4.

2.2.1. Principal Component Analysis

Principal Component Analysis (PCA) is a linear method used to reduce the dimensionality of large datasets, increasing interpretability while minimizing information loss. It does so by identifying the principal components, which are new, uncorrelated variables that capture most of the variance in the data. These components are found by first computing the covariance matrix of the dataset, and then determining its eigenvectors and eigenvalues. The eigenvectors represent the directions of maximum variance, and the eigenvalues indicate their magnitudes. In practice, PCA involves standardizing the data, computing the covariance matrix, and then selecting the top *n* number of eigenvectors based on their eigenvalues. This process transforms the dataset into a simplified form with *n* dimensions, making it easier to analyze without losing significant information.

2.2.2. Support Vector Regression

The Support Vector Regression (SVR) is the regression variant of the Support Vector Machine (SVM), introduced by Boser et al. [7]. While the SVM is designed to find the optimal hyperplane that best separates data points of different classes, the SVR operates differently. Instead of classification, SVR focuses on finding a hyperplane that best captures a continuous relationship. The objective of SVR is

to construct a 'tube' around the regression line such that the errors for most of the data points remain within a specified threshold, $\mathcal{E}$. This means that errors falling inside this tube are considered insignificant and are thus ignored. Outside of this tube, however, the errors are penalized. Figure 2.6 displays the SVR tube for a non-linear and linear regression line.

The formal objective of SVR is to minimize the complexity of the model while ensuring that the prediction errors are within the $\mathcal{E}$-tube. Given a training set $(x_1, y_1), ..., (x_n, y_n)$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, SVR tries to find a function $f(x)$ with the maximum of $\mathcal{E}$ deviation from the actual targets y_i for all the training data, while also ensuring that the function is as flat as possible. In its linear form, this function can be written as:

$$f(x) = \langle \vec{w}, \vec{x} \rangle + b \tag{2.17}$$

where $\langle \vec{w}, \vec{x} \rangle$ denotes the dot product between the vectors $\vec{w}$ and $\vec{x}$, and b is a bias.

The SVR optimization problem is formulated as:

$$\underset{\vec{w},b,\xi,\xi^*}{min} ||\vec{w}||^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \tag{2.18}$$

subject to:

$$y_i - \langle \vec{w}, x_i \rangle - b \leq \mathcal{E} + \xi_i$$
$$y_i + \langle \vec{w}, x_i \rangle + b - y_i \leq \mathcal{E} + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0, \forall i$$

where:

- $||\vec{w}||^2$ is the squared norm of the weight vector w, emphasizing the flatness of the function $f(x)$.
- C is a regularization parameter which determines the trade-off between achieving the flatness of $f(x)$ and the extent to which deviations greater than $\mathcal{E}$ are tolerated. C can take up any positive value (C>0).
- ξ_i and ξ_i^* are slack variables introduced to manage the $\mathcal{E}$-insensitive loss. It is the amount by which the prediction falls outside the $\mathcal{E}$-tube.

For non-linear SVR applications, a kernel function is introduced to map the input space to a higher-dimensional feature space, and the aforementioned principles apply in this transformed space (Figure 2.6).

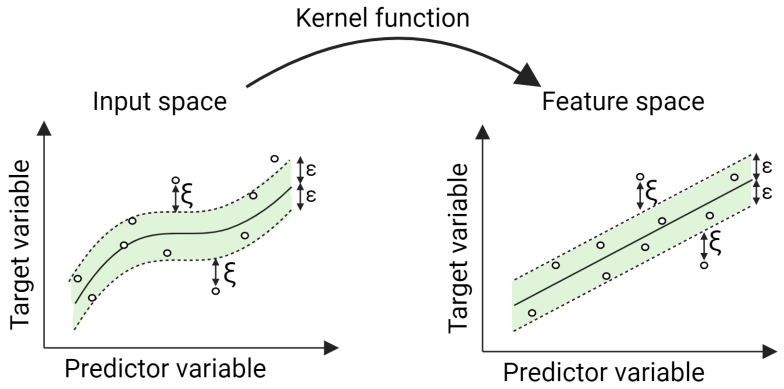


Figure 2.6: Illustration of the Support Vector Regression (SVR) applied to a non-linear input-space. Data is first transformed from an input-space, where non-linear seperation is unfeasible, to a higher-dimensional feature space using the kernel function. Hereafter, data can be separated in the feature space by a linear hyperplane.

2.2.3. Random Forest regressor

The random forest (RF) is a ML model that combines multiple decision trees, a technique known as ensemble learning. A decision tree is a machine learning model used for both classification and regression tasks. It works by splitting a dataset into smaller subsets based on different criteria, and at each level of the tree, it makes a decision that leads to a certain outcome or further splits (Figure 2.7). The tree consists of a root node, internal nodes, and leaf nodes, where each path from the root to a leaf represents a decision pathway.

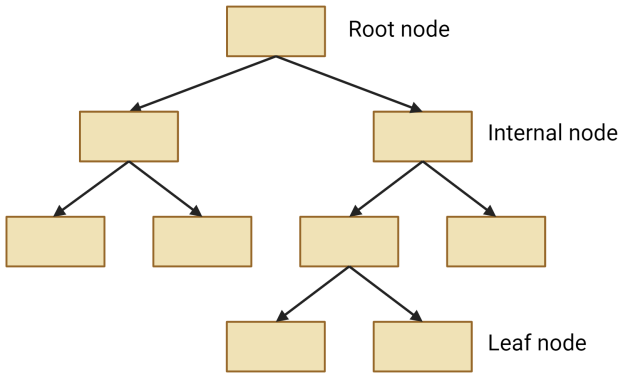


Figure 2.7: Illustration of a decision tree.

Random forests enhance generalisability and robustness by aggregating predictions from multiple decision trees, offering a significant advantage over relying on a single decision tree.

RF employs two key strategies for ensemble learning: Bootstrap Aggregation (often called 'bagging') and Feature Randomness.

- **Bootstrap Aggregation (Bagging)**: This technique involves creating multiple subsets of the dataset through sampling with replacement, meaning the same data points can appear in a subset more than once. In RF, each decision tree is trained on a different subset. For example, from an original dataset [1,2,3,4,5], a bootstrapped sample might be [2,4,5,1,2].
- **Feature Randomness**: The RF can enhance diversity by training each decision tree on a random subset of features. This approach promotes variability among the trees, further improving the model's robustness.

In a regression task, the final prediction of the RF model is the average of the predictions from all individual trees.

2.2.4. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a category of deep neural networks specially designed to process data with a structured grid-like topology, such as an image (2D) or volume (3D). As they build upon the principles of the feedforward neural network, often referred to as a multi-layer perceptron (MLP), this is discussed first. Hereafter, the typical convolutional and pooling operations of CNNs are discussed.

Multi-layer perceptron

The multi-layer perceptron is an ML model inspired by a human brain, designed to learn and predict complex functions and patterns from data. It consists of a network of interconnected artificial neurons arranged in layers: an input layer, one or more hidden layers, and an output layer. Figure 2.8 shows an example of a neural network with two hidden layers. Each neuron within a layer typically accepts multiple inputs, performs a weighted linear combination of the inputs, adds a bias term, and employs an activation function to the summed inputs (to introduce non-linearity) before passing its output to its subsequent layer. Mathematically, the neuron operation is expressed by:

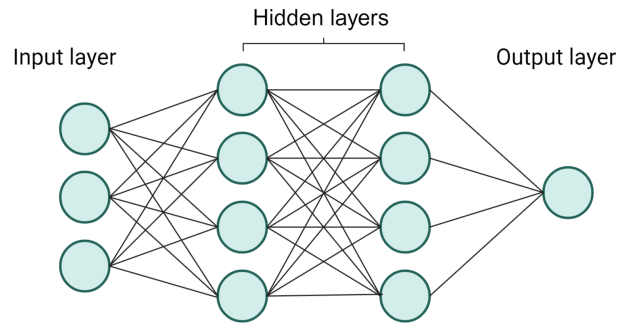


Figure 2.8: Schematic representation of a feedforward neural network, also known as a multi-layer perceptron.

$$\hat{y} = a\left(\sum_{i=0}^{n} w_i x_i + b\right) = a\left(\vec{w}^T \vec{x} + b\right) \tag{2.19}$$

where, $\hat{y}$ represents the output, $\vec{x}$ the input, $\vec{w}$ the associated weights, and a denotes the activation function.

Convolutional layer

The convolutional layer in a CNN 'filters' out features from the input, by convolving the input with a kernel to a feature map. The convolution operation in a CNN architecture is always performed on the input before feeding its output to the MLP. Typically, multiple kernels are used, each responsible for computing a distinct feature map. To illustrate, this process for a 2D input can be visualized as sliding an n×n window both horizontally and vertically across the input data with a specified stepsize, called *stride*. At every window position, a dot product operation is performed with the corresponding n×n subarray of the input. The dot product result is then summed up, and a bias is added to produce an entry in the output matrix. Mathematically, the feature value at position (i, j) in the *k*th feature map of layer *l* is:

$$z^l_{i,j,k} = w^{l^T}_k x^l_{i,j} + b^l_k \tag{2.20}$$

where, w^l_k and b^l_k represent the weight vector and bias term of the *k*th filter of the *l*th layer respectively. $x^l_{i,j}$ refers to the input patch centered at location (i, j) of the *l*th layer. A visualization of the operation done in the convolutional layer is given in Figure 2.9.

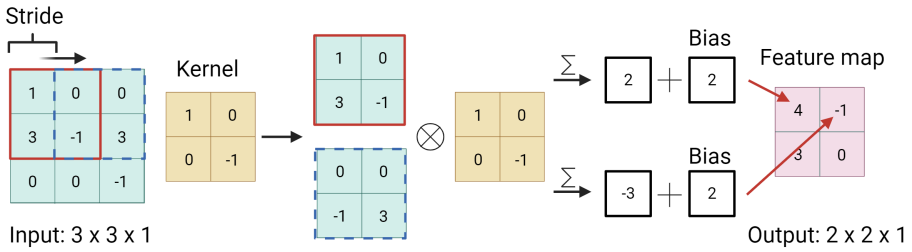


Figure 2.9: Illustration of the convolution operation. Adapted from Wang et al. [49].

Pooling layer

The pooling layer, typically positioned after a convolutional layer, reduces the spatial resolution of the feature maps. This reduction decreases the number of trainable parameters and enhances the model's robustness to spatial variations of features [17]. The pooling operation involves sliding a two-dimensional filter over each feature map and summarising the features lying within the region covered

by the filter. Mathematically, the pooling operation $pool(\cdot)$ that is performed on every feature map is described by:

$$y^l_{i,j,k} = pool(a^l_{m,n,k}), \forall(m,n) \in \mathbb{R}_{ij} \tag{2.21}$$

where, $\mathbb{R}_{ij}$ is a local neighbourhood around location (i,j). The variables *m* and *n* are spatial indices that navigate through this local neighborhood, and *k* denotes the specific feature map being processed. Common pooling operations include Maximum Pooling and Average Pooling, where the maximum and average value for each patch in every feature map is selected, respectively. A visualisation of the pooling operations is shown in Figure 2.10.

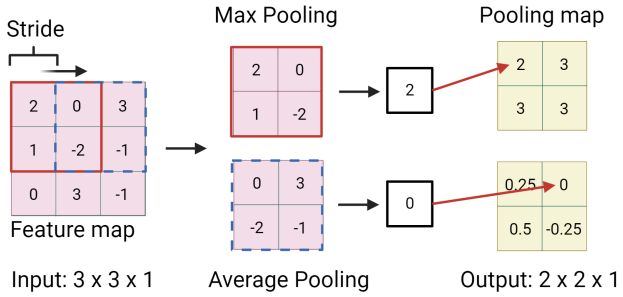


Figure 2.10: Illustration of the pooling operation. Adapted from Wang et al. [49].

A visualisation of stacked convolution, pooling and fully connected layers is shown in Figure 2.11.

Training

In the previous sections, the MLP and the convolution operation were discussed, highlighting their parameters that need adjustment to fit the target function. This adjustment involves measuring how close the model's output is to the desired output, a concept known as 'loss' in machine learning. The function aimed to be minimized to reduce this loss is called the loss function.

An algorithm known as the 'optimizer' adjusts the network's parameters to minimize the loss. This process begins by passing data through the network. Next, it involves calculating the gradient of the loss function with respect to the parameters through a process called 'backpropagation,' which moves backwards through the network. Finally, the parameters are updated by adjusting them in the direction opposite to the gradient of the loss function.

The size of these updates is determined by a factor known as the learning rate. Selecting an appropriate learning rate is critical because a rate that is too high can cause the optimizer to overshoot the optimal point, whereas a rate that is too low can lead to slow convergence towards the minimum.

To address challenges associated with the learning rate selection, the Adam optimizer introduces a momentum term [31], which helps to dynamically adjust the learning rate. This adjustment has been

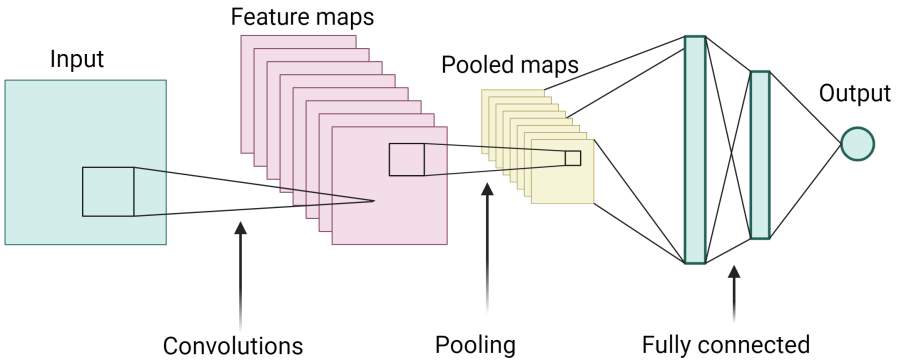


Figure 2.11: Schematic structure of a Convolutional Neural Network (CNN).

shown to converge faster compared to other optimizers and has the ability to escape local minima, making it the most used optimizer nowadays [2].

In practice, updating the parameters is often done after a small batch of data is fed into to model, instead of passing all data. This procedure enhances training speed and introduces a level of stochasticity that can help prevent the optimization process from becoming stuck in local minima, thereby facilitating more robust convergence towards the global minimum.

3

Literature review

This chapter reviews the current literature on ML surrogate modeling for the RFEM. First, studies that used a semi-surrogate modeling technique applied to slope reliability analysis and other geotechnical problems are summarized. Then, it highlights two papers where a full-surrogate model was used. Note that all studies described here dealt with geotechnical problems in two dimensions (2D).

Most research on ML-aided slope reliability analysis focuses on 'semi-surrogate' modeling. That is, ML models are trained on a small subset of a full MCS, and are then used to efficiently predict outcomes such as 'stand/fall' or FoS for the remainder of random fields. In general, the main procedure of such semi-surrogate modeling consists of the following steps [52] (Figure 3.1):

1. Use an MCS approach to generate multiple random field samples of geotechnical soil properties.
2. Undertake the FEM for a subset of random fields to compute the corresponding outcomes.
3. Feed the random field samples into an ML model with the computed outcomes as the targets. Post-training, the ML model captures the relationship between the input samples and outcomes.
4. Utilize the trained ML model to efficiently and accurately compute outcomes for the remaining random field samples.

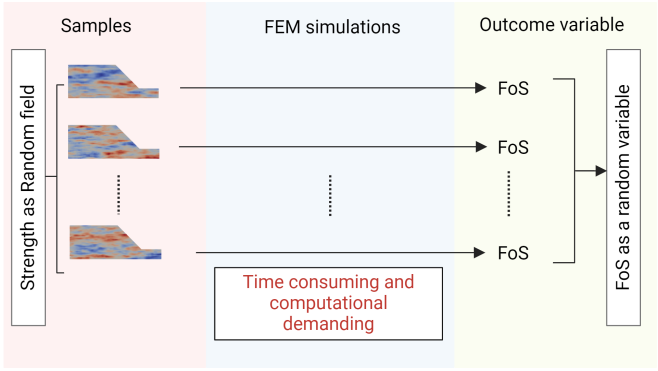
Aminpour et al. [3] used the stable/fail indicator as the outcome from a slope stability analysis. They chose this target variable to bypass the computational expenses related to the strength reduction method, which is needed to obtain the FoS. They showed that using a Support Vector Classifier trained on only 1% of the total simulations, the mean error in p_f can be limited to a 0.7% margin of that determined by the exhaustive 'brute force' MCS. However, the p_f associated with their investigated slopes was as large as 44%, which is an unrealistic value in practical cases.

In a subsequent study by the same authors, a *probability summation* method was proposed, where the outcome of the ML model is the probability of failure-class instead of a single stable/fail outcome [4]. This approach led to a reduction in the mean error in the probability of failure prediction, which was then limited to 0.46% for a large range of anisotropy values ($1<\xi<24$) and point variability levels ($0.1<COV<0.5$). Doing so, the CPU computational time was reduced from 306 days for a full MCS to only 3 days using their method, which is a 100-times faster analysis. It should be pointed out that the level of p_f was again as large as 47%.

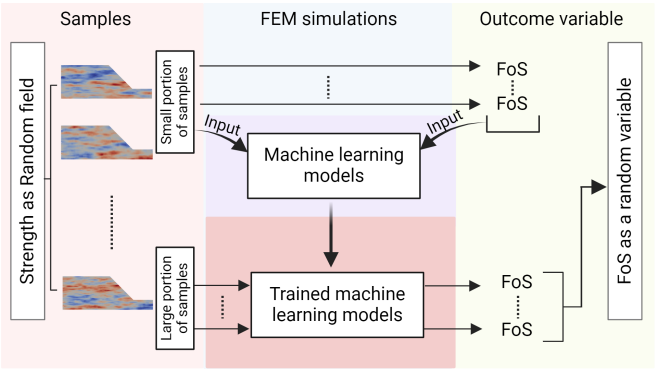
Works that focus on predicting the FoS have used the Support Vector Regression (SVR) [3, 19], the Artificial Neural Network (ANN)[19, 3], and the Convolutional Neural Network (CNN)[50, 49, 48].

He et al. [19] showed that both the ANN and the SVR can be deployed on multilayered slopes to accurately predict the FoS at a fraction of the time a full MCS using the RFEM would cost. A promising accuracy on the test set was obtained when training on only 200-300 samples along with their corresponding FoSs. They reported that especially the SVR model needs extensive hyperparameter optimisation.

Wang and Goh [50] proposed a CNN as a semi-surrogate model and showed that the proposed method compares favorably against other metamodel-based methods (spectral stochastic finite element method (SSFEM) and the Multi quadratic response surface method) in terms of computational efficiency and accuracy for the Congress Street case, involving a multi-layered soil system. It should



(a) Slope reliability analysis using a full Monte Carlo Simulation



(b) Semi-surrogate machine learning method

Figure 3.1: Comparison between the conventional method and the approach enhanced by machine learning for slope reliability analysis. Adapted from He et al. [19].

be noted that they used a vertical scale of fluctation of 10.63 meters, which is an unrealistic value from a practical perspective.

Wang et al. [49] used a ML semi-surrogate modelling technique for the RFEM, applied to different geotechnical problems. They applied the CNN to an excavation problem and a surface footing problem. They reported that the CNN is better at handling sites that exhibit high heterogeneity in comparison with the SSFEM technique in the sense that a significantly lower computational cost is required.

Research that has explored a 'full-surrogate' ML model is scarce. Such a model, once trained, can be applied directly to another different slope case without the need for any additional numerical calculations. In this area, He et al. [18] trained a deep CNN on a dataset comprising more than 12.000 simulations of a bearing capacity problem. The pre-trained model is accurate on a wide range of new cases without the need for retraining. They showed that full reliability analysis with $N_{MC} = 2 * 10^5$ predictions can be done within minutes, whereas a 'brute force' MCS reliability analysis with $N_{MC} = 1000$ would take up to eight hours. High accuracy of the CNN predictions can be achieved, given that the parameters of the new problems are well within the limits on which the model was trained. A drawback of using a CNN is the need for a uniform mesh. In numerical calculations, however, nonuniform meshes—some parts having coarser and others finer discretization—are often preferred to get accurate results.

In a very recent work, Xu et al. [51] created a full-surrogate model for the RFEM applied to a slope stability problem, with the FoS as outcome variable. Their presented model can handle slopes with varying slope shapes, and is accurate with a mean-absolute-percentage-error of about 6% for the testing dataset. They compared three kinds of artifical neural networks and noted the best results for a locally connected network. This network comprises both convolutional layers and a locally connected layer. Unlike a convolutional layer, where weights are shared across the entire input, a locally connected layer applies a unique set of filters to each specific patch of the input, allowing for distinct processing at each location.

However, using ML surrogate models for the RFEM in slope reliability analysis has its drawbacks. Aminpour et al. [3] highlighted that ML methods can struggle to make accurate predictions on the slope

stability status in instances of high heterogeneity and high anisotropy. This limitation is significant, especially since RFEM is preferred in these scenarios over deterministic methods [15].

Another challenge, highlighted by Wang and Goh [48], Aminpour et al. [3], and He et al. [18], arises when slopes with small probabilities of failures are dealt with. It's possible that the training set for the ML model does not contain any instances of a failed slope. Consequently, the ML model may not be adequately trained to make predictions in the failure region. This means that the accuracy of predictions for this region largely hinges on the model's ability to extrapolate. This reliance on extrapolation can compromise the accuracy of the ML model's predictions. This problem is relevant as engineers are mainly interested in slope instances that are close to failure. To address this problem, Wang and Goh [48] introduced a post-processing approach to gauge the lower end of the FoS distribution using the maximum entropy method. Using this approach, probabilities of failure in the order of 10^{-4} were predicted with a 10% error using only a fraction of the total number of MC simulations that would have been needed without this approach.

4

2D Slope Reliability Analysis

In this chapter, the investigation of machine learning-aided reliability analysis in two dimensions (2D) is presented. The methodology is first explained, encompassing a description of setting up the RFEM model, three investigated slope cases with varying point statistics and spatial correlation lengths, and the utilisation and optimisation of three different ML models to make predictions of the FoS and probability of failure.

After going through the methodology, the results of the ML models as (semi-)surrogate models for the 2D RFEM are presented. This starts by assessing the performance of the ML models to be a semi-surrogate model for the 2D RFEM applied to the slope cases. Hereafter, a full-surrogate model for the RFEM in 2D is evaluated, which eliminates the need for additional numerical simulations when applied to a broader range of scenarios.

To conclude the analysis in 2D, the main insights are presented.

4.1. Methodology

4.1.1. Initialization of the RFEM

Figure 4.1 shows the idealised, fictive slope on a Cartesian *x*-*z* plane that is investigated in this research. It is characterized by an angle of 45°and a height of 4.8 meters. The foundation of the slope is 2 meters deep. The choice for this geometry is made to allow for the full development of possible slip surfaces while limiting the computation demand to an acceptable level.

A rigid boundary is modeled to represent a hard rock bed at the bottom of the foundation, constraining all movements. Horizontal movements are constrained at the vertical boundaries.

The slope is discretized into 596, 8-node, quadrilateral elements, each of size 0.4×0.4 meters, except for those along the slope face, that have been distorted to fit the slope geometry. Figure 4.1 displays the mesh of the slope. Each element has 2×2 Gaussian integration points. For optimal utilisation, random field (cell) values are mapped onto these points instead of the elements themselves. In this configuration, four random field cells cover the same area as one finite element, leading to 2384 random field cells.

The slope consists of clay modeled by a linear elastic, perfectly plastic stress-strain behaviour using the Mohr-Coulomb failure criterion and a non-associated flow rule (dilation angle = 0°). The elastic component of this model is defined by Young's Modulus E = 100.000 kPa, and Poisson's ratio ν = 0.3. The unit weight of the clay γ_{sat} = 20 kN/m^3. The undrained shear strength s_u is a stochastic parameter variable that is modeled by a random field, which is generated with the LAS technique (refer to Section 2.1.1), with an underlying Gaussian distribution. The point statistics and the spatial correlation lengths of the shear strength are subject to change across several cases. Negative undrained shear strength values are avoided by setting a truncating threshold of 10^{-5} kPa.

A maximum number of 500 iterations in the numerical calculations is used to indicate when failure has occurred, which was previously identified by Hicks and Spencer [24] to be sufficient to define failure. For the SSRM, a constant stepsize of 0.01 of the Safety Reduction Factor is used. This way, the FoS is assessed with a 0.01 resolution.

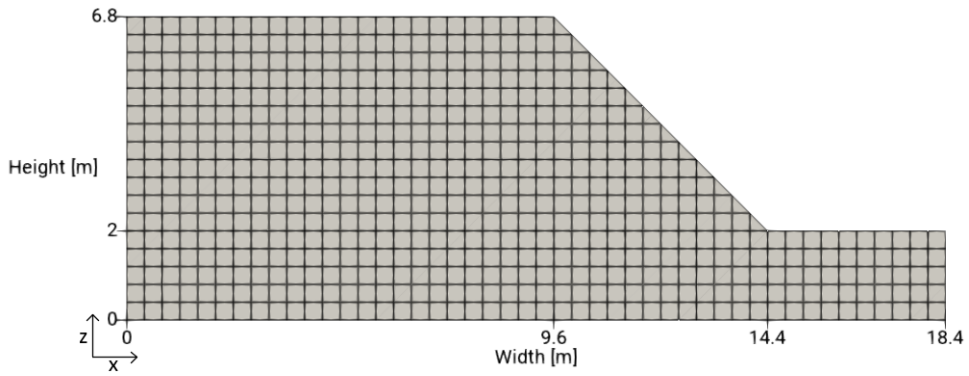


Figure 4.1: Slope geometry and element mesh of the FE model.

Table 4.1: Parameters that are being held constant in this thesis.

Parameter	Symbol	Value	Unit
Saturated unit weight	γ_{sat}	20	kN/m^3
Young's Modulus	E	100.000	kPa
Poisson's ratio	ν	0.3	[-]
Dilation angle	ψ	0	$^\circ$

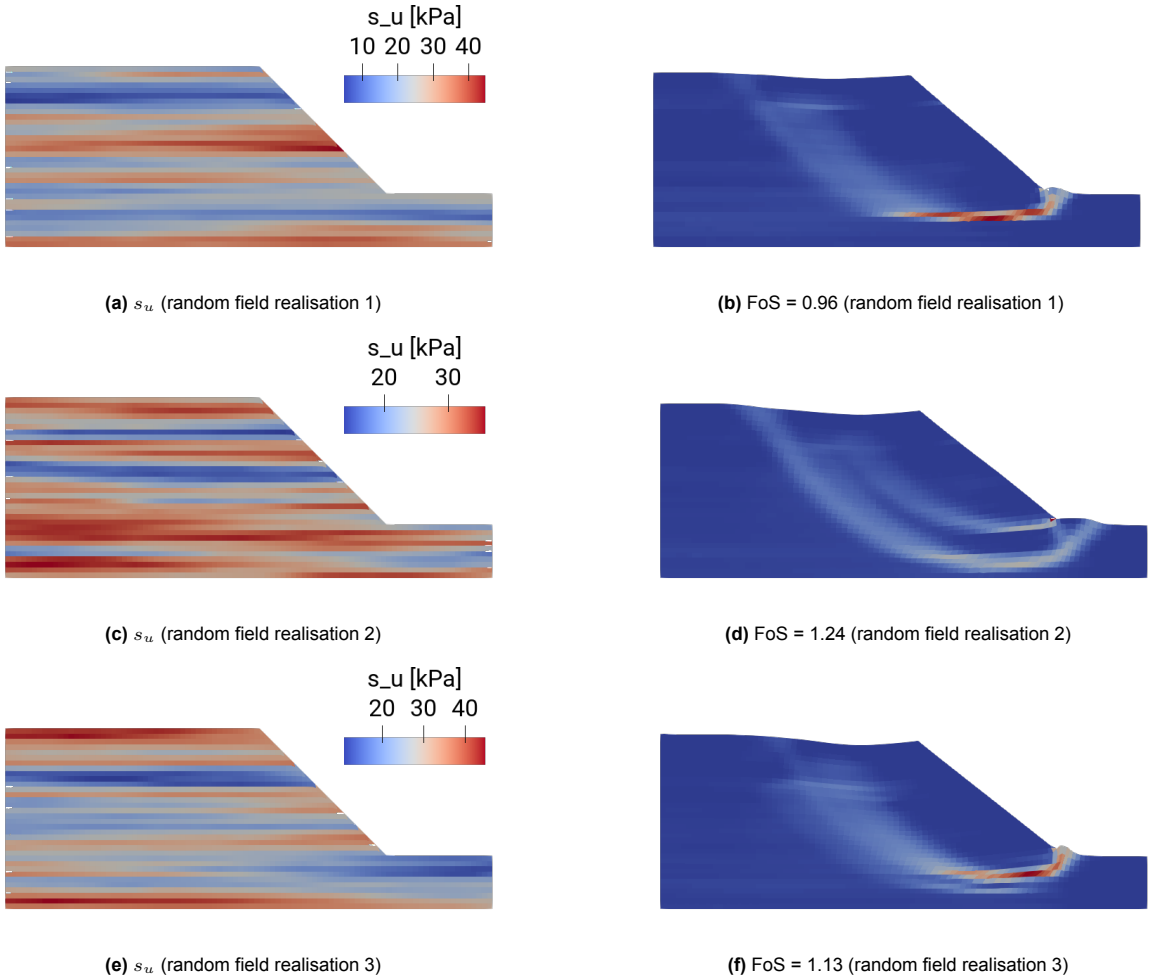
4.1.2. Investigated cases

Table 4.2: The point statistics and scales of fluctuation for the cases considered.

Case	Undrained shear strength			Scale of fluctuation		Extra information
	Mean [kPa]	COV	Distribution	θ_v [m]	θ_h [m]	
U1	27.76	0.3	Normal	6.1	60	Max scale of fluctuation values for s_u of clay, Phoon and Kulhawy [39]
U2	26.56	0.3	Normal	2.5	50.7	Mean scale of fluctuation values for s_u of clay, Phoon and Kulhawy [39]
U3	24.34	0.3	Normal	0.8	46	Min scale of fluctuation values for s_u of clay, Phoon and Kulhawy [39]

In this study, three slopes with varying point statistics and spatial correlation lengths are investigated, as detailed in Table 4.2. These slopes, labeled Case U1, U2, and U3, use the maximum, mean, and minimum of the scales of fluctuation respectively that are found in practice for the undrained shear strength of clay, according to a literature review by Phoon and Kulhawy [39]. The input statistics for the slopes used in other studies, such as those by Wang and Goh [50] and Wang et al. [49], who also explored ML-aided slope reliability analysis in 2D focusing on the Factor of Safety (FoS), are not considered in this research. This is because their studies involve slopes with different characteristics, like soil layering and geometry irregularities, making direct comparisons challenging. Including such features would overcomplicate the problem and purpose of this thesis.

The mean undrained shear strength is assumed constant in each case and has been determined to ensure that the p_f derived from 4000 FE simulations is within the range of 2.8 - 3.2%. The level of p_f is determined arbitrarily to strike a balance between the total number of simulations needed to achieve a consistent p_f and a practical value. In addition, considering slopes with the same probability of failure ensures consistency and allows for a fair comparison between the errors in the ML predicted p_f and the p_f obtained using 4000 FE simulations. The coefficient of variation (COV) of the undrained shear strength is set constant to 0.3, which is within the typical range used in practice [39]. This value, in combination with the selected mean undrained shear strengths, also ensures that the chance of getting close to zero strength values is minimal. Figure 4.2 displays three examples of generated random fields of the undrained shear strength s_u (a, b, c) along with the associated failure mechanisms deduced from the computed shear strains at failure (d, e, f) for case U3, while Figures B.1 and B.2 in the appendix show the corresponding examples for cases U1 and U2, respectively. Generally, there is a deep-seated failure pattern with a near-circular failure surface. This aligns with the usual failure modes seen in undrained slopes.



(a) s_u (random field realisation 1)

(b) FoS = 0.96 (random field realisation 1)

(c) s_u (random field realisation 2)

(d) FoS = 1.24 (random field realisation 2)

(e) s_u (random field realisation 3)

(f) FoS = 1.13 (random field realisation 3)

Figure 4.2: Three examples of random fields of the undrained shear strength s_u (a, c, e) and the deviatoric shear strains at failure (b, d, f) computed using the FEM for the slope case U3.

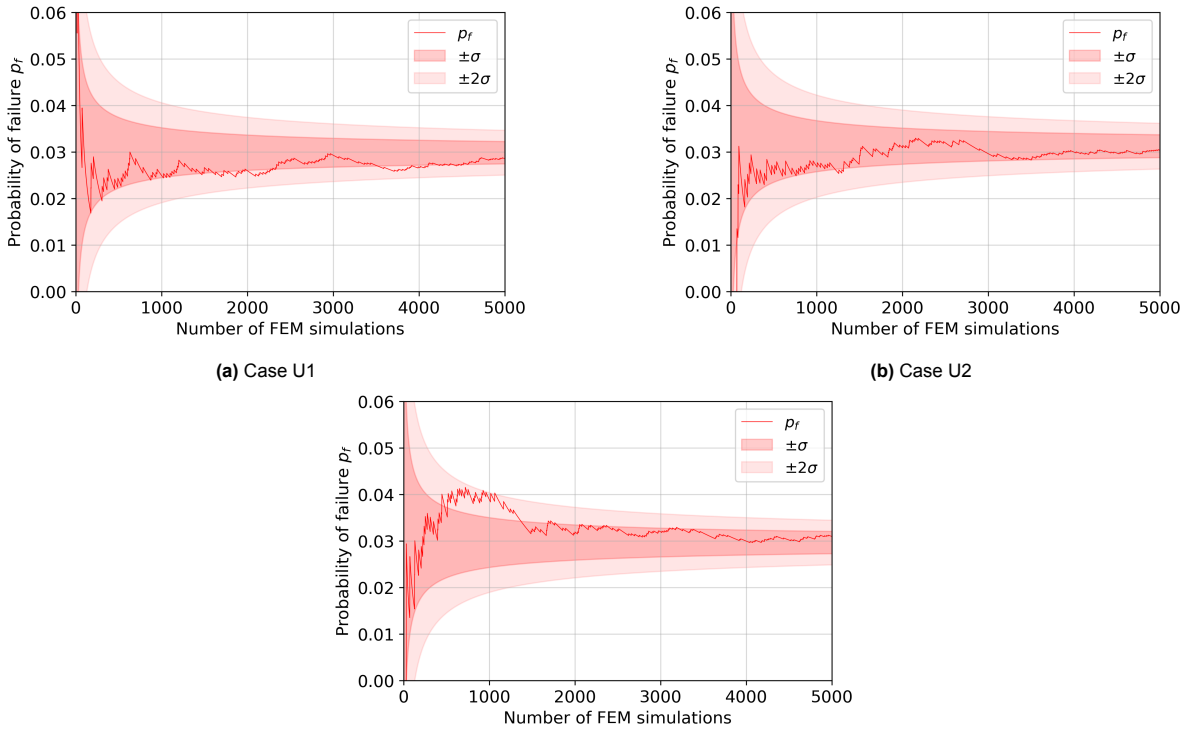
The choice to use 4000 realisations in the RFEM analysis is based on the fact that the COV of the p_f is well below 0.1 when the slope is at a p_f of 2.8%, according to Eq. 1.1. The stable behaviour of p_f is further illustrated in Figure 4.3, where the p_f is plotted against the number of samples for every considered case. These plots emphasize the evolution of p_f with an increasing number of simulations and include shaded areas that represent one and two standard deviations, illustrating the range of potential p_f values across multiple MCSs, as calculated by combining Eq. 1.1 with $\sigma = COV[p_f] \times \mu[p_f]$. Here $\mu[p_f]$ is the mean probability of failure calculated from a total of 6400 simulations due to computational limits. Ideally, this value should be calculated from an infinitely large sample size. The plots indicate that a stable value of p_f is reached after approximately 3500 simulations in each case, aligning with the estimated $COV[p_f]$ of 0.1. Additionally, the plots indicate that the p_f can take a wide range of values when the number of simulations is small.

Histograms of the obtained factors of safety for each RFEM case using 6400 FE simulations are shown in Figure 4.4. A normal distribution is fitted on each histogram and the mean and standard deviation are displayed in the top left corner, denoted as $\mu[FoS_R]$ and $\sigma[FoS_R]$, respectively. The FoS obtained using a deterministic analysis based on the mean undrained shear strength is denoted by the red dashed line and is denoted as FoS_D. Consistent with prior studies (e.g., [24, 21]), the mean FoS obtained by the RFEM differs from the deterministic FoS, highlighting the importance of accounting for spatial variability of the soil. This discrepancy is larger for slope cases with smaller spatial correlation lengths, as strong and weak zones within each random field alternate more frequently, and consequently, the failure path can go through weak zones relatively easily.

Additionally, the histograms show a smaller range of obtained FoSs for slope cases with smaller

spatial correlation lengths. This is because the failure surface for each realisation passes through more diverse random field cells, leading to more strength averaging over the failure surface. Conversely, slope cases with larger spatial correlation lengths exhibit a broader FoS distribution as their failure surfaces traverse random field cells that are more uniform, either being more consistently strong or weak.

Note that some slopes fail at an SRF close to zero. After inspection of these instances, a local weak zone near the slope surface is observed on each of these slopes, where the undrained shear strength is close to zero.

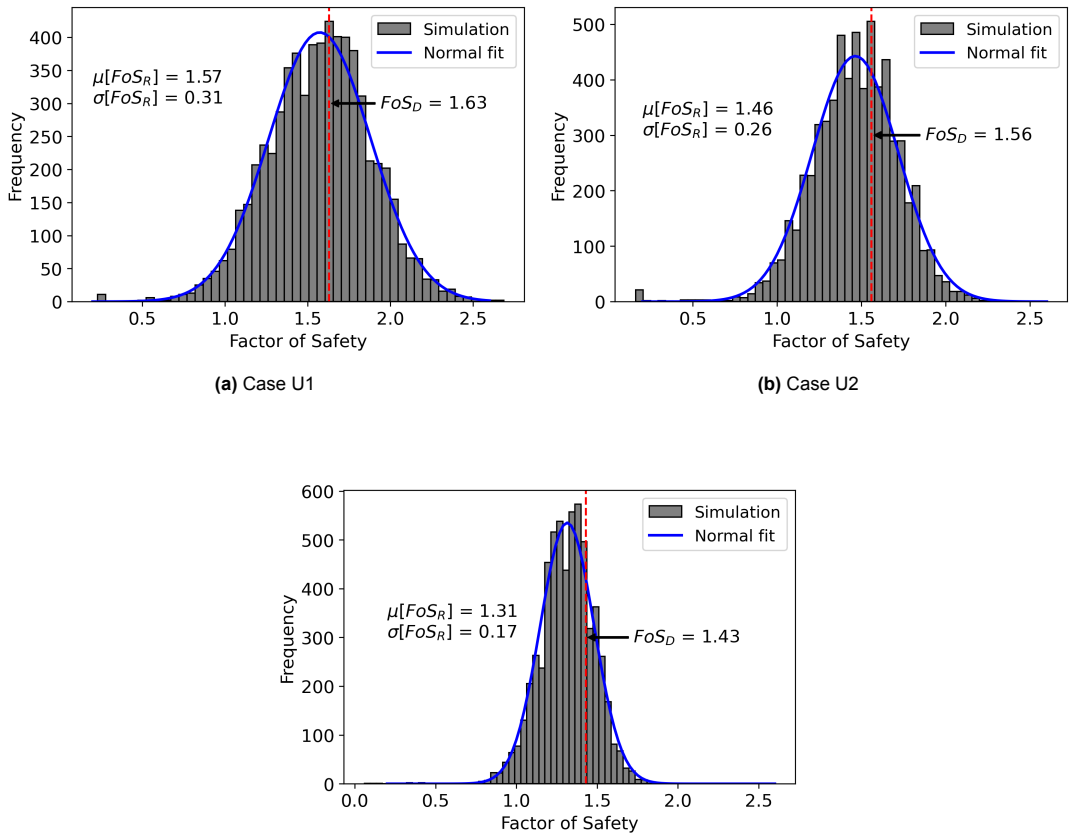


(a) Case U1

(b) Case U2

(c) Case U3

Figure 4.3: Relationship between the number of simulations in the RFEM analysis and the probability of failure p_f for the three considered cases. The shaded areas represent one and two standard deviations, illustrating the of range of potential p_f values across multiple MCSs.



(a) Case U1

(b) Case U2

(c) Case U3

Figure 4.4: Histograms of the realised Factors of Safety using the RFEM analysis (FoS_R) for the three considered cases. The Factor of Safety based on the mean undrained shear strength is denoted by FoS_D.

4.1.3. Splitting data

Before training the ML models, the data, containing random fields with the associated Factors of Safety, is split into a training batch and a testing set, for every individual case. After this initial split, 20 random training sets of sample size m are drawn from the training batch. This approach allows for the training of 20 independent versions of the ML models. This way, the variation in predicted values of the ML models can be assessed across 20 separate training and prediction cycles. Adopting this strategy helps mitigate the risk of relying on a single training set, which might be favourable or non-favourable, leading to skewed or non-representative results. Figure 4.5 illustrates this data-splitting process. The training set size m is subject to change in this study.

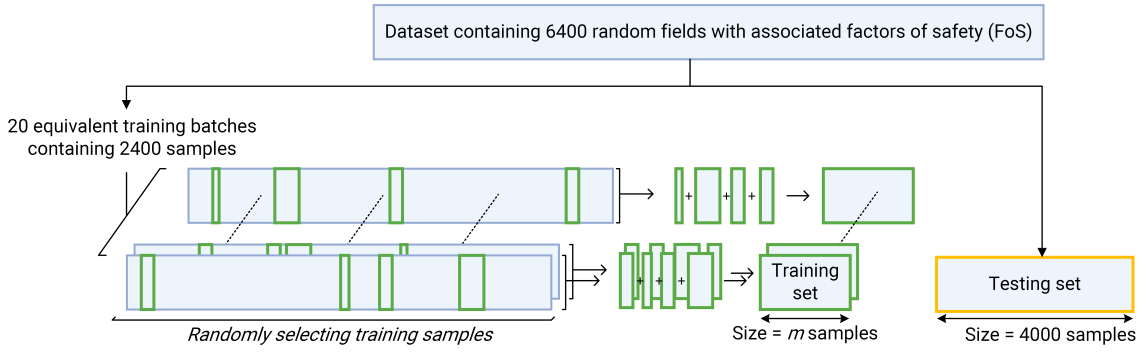


Figure 4.5: Process of splitting up the data for each slope case into training set and testing set. The size of the training set m is part of investigation in this study.

Using the same testing set for every ML model allows for a valid comparison between the ML models. The testing set comprises 4000 realisations, as this number ensures that a reliable p_f is obtained (refer to Section 4.1.2). Since this is the number of simulations needed when a traditional MCS is performed, a fair comparison in terms of computational time with the suggested ML-aided reliability analysis can be made.

4.1.4. Data augmentation technique

Data augmentation is a common strategy to enhance the learning capability of an ML model, particularly when dealing with a limited dataset. It artificially increases the size of the dataset by creating new data points from existing ones through methods such as rotation, flipping, or modifying the input features. The underlying principle is that by diversifying the training data, the model can more effectively learn the relationship between input features and their corresponding labels.

Following this concept, Jiang et al. [30] proposed a data augmentation technique specifically for RFEM simulations. This study replicates this technique and applies it selectively across the cases, ensuring to mention its usage wherever applicable.

The technique is based on the theory explained in Section 2.1.3. It expands the training dataset by using the relationship between random fields and their corresponding FoS already present in the training set, without having to perform any additional numerical simulations.

Suppose that a single random field is represented by $x_0 = (x_0^1, x_0^2, \ldots, x_0^D)^T$, where D denotes the total number of random field cells. Given its associated FoS_0, a new random field sample $x_{M_g} = (x_{M_g}^1, x_{M_g}^2, \ldots, x_{M_g}^D)^T$ can be derived for a different specified FoS_{M_g}. The (random field) cell values for the undrained shear strength can be calculated using:

$$\text{Random field } s_u : \begin{cases} s_{u,M_g}^1 = s_{u,0}^1 * \dfrac{FoS_{M_g}}{FoS_0} \\ s_{u,M_g}^2 = s_{u,0}^2 * \dfrac{FoS_{M_g}}{FoS_0} \\ \qquad \vdots \\ s_{u,M_g}^D = s_{u,0}^D * \dfrac{FoS_{M_g}}{FoS_0} \end{cases} \tag{4.1}$$

where, M_g represents the number of new samples generated from the original sample. As an illustrative example, consider a slope with an undrained shear strength random field $s_0 = (s_0^1, s_0^2, \ldots, s_0^{100})^T$. If the FE simulation determines an FoS of 1.8, a new random field sample for a given Factor of Safety F_{M_g} of 0.6 can be generated using Equation 4.1, resulting in $s_{u,F_{M_g}=0.6} = (\frac{s_0^1 * 0.6}{1.8}, \frac{s_0^2 * 0.6}{1.8}, \ldots, \frac{s_0^{100} * 0.6}{1.8})^T$. By applying this process for various FoS_{M_g}, a diverse range of new random field samples can be created, without performing additional numerical simulations. For validation of this method, the reader is referred to Jiang et al. [30].

In this study, a set of 15 extra samples is generated (M_g = 15) for each feature-label pair from a FE simulation by applying the data augmentation technique. The chosen FoS_{M_g} values are evenly distributed between the minimum and maximum observed FoS of the training set. However, from several experiments conducted, it is noted that the exact values for the lower and upper FoS_{M_g} do not influence the performance of the ML models much. Hence, in this study, for all cases considered, the range of the FoS_{M_g} is between 0.6 and 2.0, i.e FoS_{M_g} = 0.6, 0.7, 0.8, ...2.0. Consequently, for an initial training sample size of 100, this approach yields an addition of 1500 random field samples, each paired with a corresponding FoS.

4.1.5. Machine learning models for FoS prediction

In this study, the Principal Component Analysis - Support Vector Regression (PCA-SVR) combination, the Principal Component Analysis Random Forest regression (PCA-RF) combination, and the Convolutional Neural Network (CNN) are used for FoS prediction. The choice for these three ML (combined) models is twofold:

1. The prior application of a PCA-SVR and CNN to 2D slopes offers a relevant benchmark for this work. Aminpour et al. [4] used the SVR in combination with a Principal Component Analysis (PCA) to predict the FoS. Other studies only considered CNN architectures [18, 50, 48, 49]. By proposing and adding a new ML combination (PCA-RF) to the comparison, a new ML surrogate model for the RFEM is explored in this study.
2. The models are well known for their ability to handle high-dimensional data effectively. This ability is needed as the number of random field (cell) values in the slope problem considered is very large, and is likely to be in the same order of magnitude for other RFEM problems.

Recognizing that each model needs to be treated and used differently, they are covered individually in the next sections. These sections detail the inputs and hyperparameters specific to each model and include a description of the training procedure for the CNN. For foundational information on each model, please refer to Chapter 2.

4.1.5.1 PCA-SVR

Input

The input for the PCA-SVR model, when used for FoS prediction, comprises a set of standardised random fields generated by the RFEM program. Standardisation is needed to ensure that all features contribute equally to the PCA, preventing features with larger scales from dominating the first components. Each random field is represented by an array: $x_0 = (x_0^1, x_0^2, ..., x_0^D)^T$. Here, x_0^i denotes the random field value at each Gauss integration point and D is the total number of random field values. When addressing problems characterized by multiple random fields, such as a drained slope with spatially varying cohesion c and friction angle ϕ, it is necessary to stack the random field arrays and scale each random field individually prior to passing them into the PCA-SVR model. This way, the input dimension for the PCA-SVR is *m*×(*D*·*p*), where *m* is the number of samples in the training set, and *p* represents the number of variables described by random fields. For the slopes considered in this study, there are 2384 random field values of the undrained shear strength, making the size of the input *m*×2384.

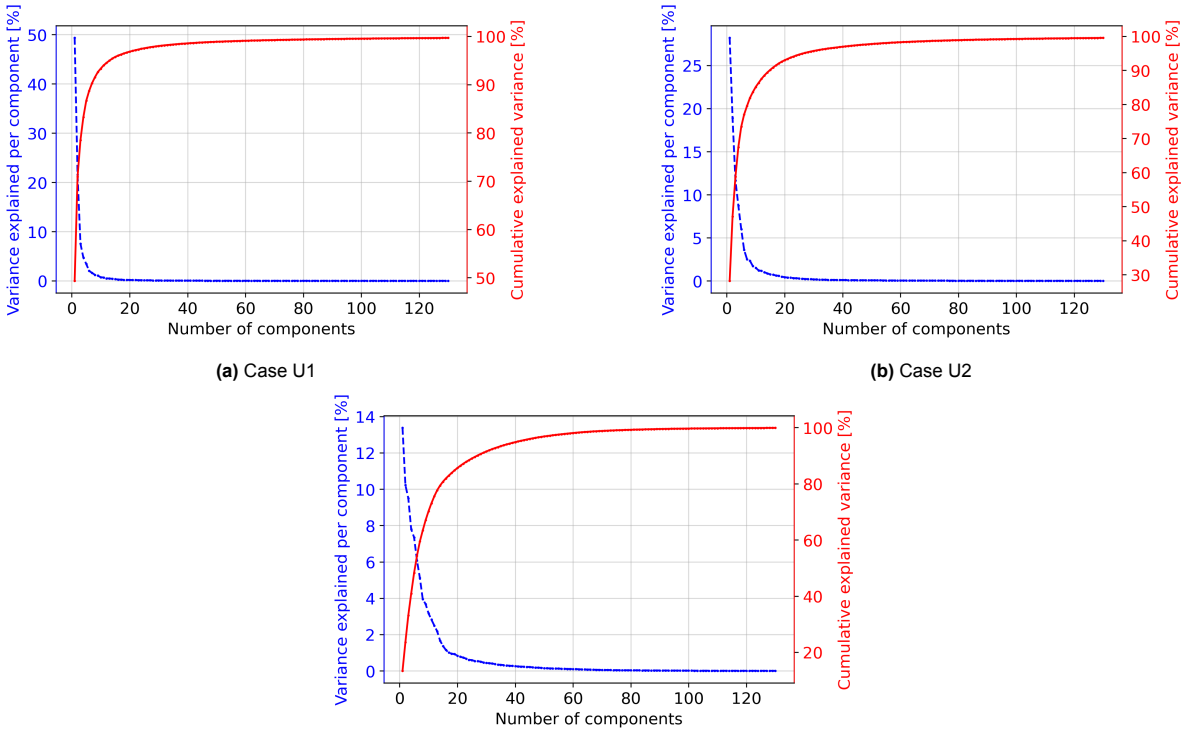
Figure 4.6 presents the scree plots resulting from the PCA conducted on the training sets for the cases under study. It shows the percentage of total variance captured per principal component (left axis) and the cumulative explained variance as a function of the number of components (right axis). It shows that almost all the variance in the random fields is explained by using a limited number of components. For slopes with smaller spatial correlation lengths, the same cumulative variance is explained by more principal components, as expected.

Hyperparameter optimisation

For hyperparameter optimisation, the full training set (2400 realisations) for each investigated case is used, combined with 5-fold cross-validation. For details on cross-validation, the reader is referred to Section A.1.

Given the PCA-SVR's relatively quick training times and its limited number of hyperparameters, a grid search is implemented for optimisation. This approach systematically evaluates a predefined array of hyperparameter values by training the PCA-SVR model with every possible combination of these values, thereby thoroughly examining the parameter space. The primary advantage of using grid search in this context is its systematic nature, which ensures the identification of the best possible model configuration within the explored grid. The hyperparameter grid tested for the PCA-SVR is detailed in Table 4.3. The effectiveness of each hyperparameter combination is assessed using the mean squared error (MSE), which is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(FoS_i - \hat{FoS}_i)^2 \tag{4.2}$$



(a) Case U1

(b) Case U2

(c) Case U3

Figure 4.6: Scree plots for the three considered cases.

Table 4.3: Hyperparameter range for optimization of the Support Vector Regression.

Parameters	Kernel	Min	Max	Type	Steps	Scale
C	linear, RBF, polynomial	0.001	10	Real	10	logarithmic
$\mathcal{E}$	linear, RBF, polynomial	0.001	1	Real	4	logarithmic
degree	polynomial	1	5	Integer	1	linear
PCA components	linear, RBF, polynomial	20	160	Integer	20	linear

where $\hat{FoS}$ is the ML predicted FoS and FoS the FoS obtained using the FEM.

The best hyperparameter combination appeared to be the same for every case considered: the radial basis function (rbf) kernel, $C = 0.25$, $\mathcal{E} = 0$, and *PCA components* = 100. Consequently, these settings are used in training the PCA-SVR for all cases considered. Note that by setting $\mathcal{E}$=0, the error insensitive 'tube' around the regression line is essentially removed.

With 100 components used, more than 99% of the variance in the random fields is captured for all cases considered, as shown in Figure 4.6.

4.1.5.2 PCA-RF

Input

The input for the Principal Component Analysis - Random Forest regressor (PCA-RF) model is the same as the 1D arrays containing the random field cell values, as for the PCA-SVR.

Hyperparameter optimisation

Due to the PCA-RF requiring considerably more computational time compared to the PCA-SVR, a random-search, rather than a grid-search, is utilized along with 5-fold cross-validation on the entire training set for each case under investigation. Using this method, 200 random combinations of hyperparameters in the parameter space are explored.

The ranges of the random-search hyperparameters are detailed in Table 4.4. For the description of each hyperparameter, the reader is referred to Section A.2. The mean squared error (Eq. 4.2) determined the best hyperparameter combination.

Table 4.4: Hyperparameter range for optimization of the PCA-RF.

Parameters	Min	Max	Type
Min_samples_split	2	10	Integer
Min_samples_leaf	1	11	Integer
Max_features	-	-	None, sqrt, log2
PCA components	5	200	Integer

For all cases considered, the PCA-RF demonstrated the lowest MSE with the following hyperparameters: *PCA components* = 10, *Min_samples_split* = 2, *Min_samples_leaf* = 1, and *Max_features* = *None*. Setting *Max_features* to *None* means that all 10 PCA components are utilised in each decision tree within the forest. This configuration essentially drops the 'Feature Randomness' aspect of Random Forests, as every tree has access to the full set of features (PCA components) rather than a random subset for making splits. The number of trees in the forest that showed consistent performance was found to be 200.

With 10 principal components used, between 70% (for Case U3) and 94% (for Case U1) of the total variance in the data is captured, as shown in Figure 4.6.

4.1.5.3 CNN

Input

The input for the CNN consists of digital images of random fields, represented as rectangular 2D matrices. These matrices are created using the 'closest pixel' procedure, which involves several steps. Initially, a 2D matrix of size *m*×*n* is constructed, where *m* and *n* represent the maximum number of random field cells in the *z* and *x* directions, respectively. Coordinates are assigned to the pixels, aligning them with the range of the random field. Then, each random field cell value corresponding to a Gauss point is mapped to the pixel within the 2D matrix that has the closest Euclidean distance to it. As an illustrative example, Figure 4.7 shows the associated 2D digital image of the slope realisation depicted in Figure 4.2a.

There is a direct correspondence between the original random field and the 2D image: each random field cell in Figure 4.2a is directly related to the spatially coincident pixel in Figure 4.7. The irregularities along the slope face stem from translating the distorted random field cells into the square pixels of the digital image.

Additionally, to ensure that the CNN receives a complete two-dimensional matrix as input, areas of the matrix not depicting any part of the slope are filled with zeros. This step is a standard procedure in image pre-processing. In the case of the two-dimensional slope investigated in this study, the input

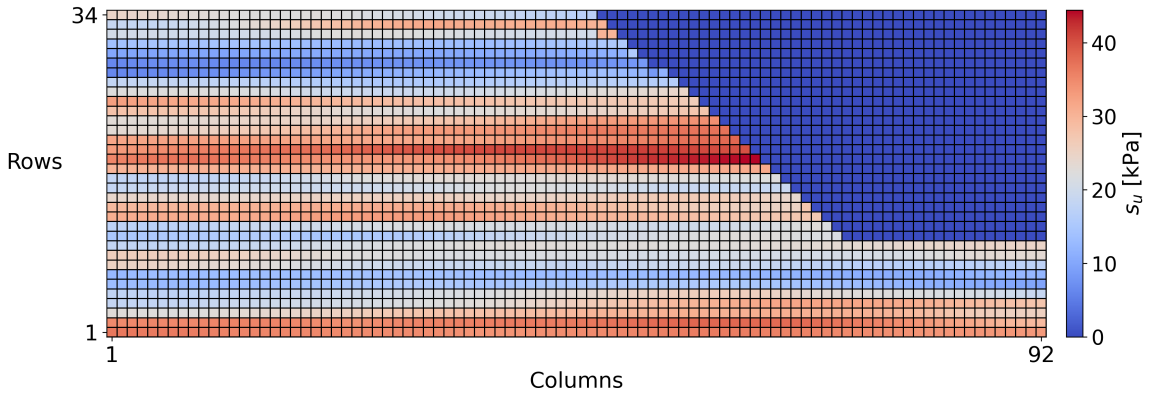


Figure 4.7: 2D digital image of a realised random field of the undrained shear strength.

images are structured with dimensions of 34×92 pixels. This number equals the number of elements in each direction, multiplied by two, as 2×2 Gaussian integration is used.

Note that for the slope considered, only the undrained shear strength is described by a random field. If one intends to describe more than one random soil property, the 2D matrices should be 'stacked'. For example, if a slope's spatially varying properties include cohesion and friction angle, the input size to the CNN would be *m*×*n*×*2*.

Architecture

The CNN architecture is subject to hyperparameter tuning by iteratively adjusting the hyperparameter until the model becomes complex enough to have good performance while maintaining a low number of trainable parameters to reduce training time. Experiments were done using either one or two stacked convolutional layers, and either one or two fully connected (FC) layers. Table 4.5 displays an overview of the hyperparameters tested.

Table 4.5: Hyperparameters tested for the CNN model.

Layer	Hyperparameters	Settings tested
Conv2D layer 1	Kernelsize	3x3, 5x5, 7x7
Conv2D layer 1	Filters Conv2D layer	5, 10, 20
Conv2D layer 2	Kernel size Conv2D layer	3x3, 5x5
Conv2D layer 2	Filters Conv2D layer	5, 10, 20
Pooling layer 1 & 2	Pooling size	2x2
FC layer 1	Neurons	64, 128, 256
FC layer 2	Neurons	64, 128, 256
Dropout layer	Dropout rate	0, 0.1, 0.2, 0.3, 0.4, 0.5
-	Batch size	128, 256
-	Learning rate	0.01, 0.001, 0.0001
-	Optimizer	Adam
-	Loss function	Mean squared error

A single convolutional layer, MaxPooling layer, and a single FC layer appear to work best for all cases (lowest validation loss). Specifically, the convolutional layer uses a kernel size of 5×5 and generates 20 feature maps. The MaxPooling layer uses a filter size of 2×2 with a stride of 2, which is the most common form in CNNs. After the convolutional and MaxPooling layers, the feature maps are flattened into a one-dimensional vector, which is a necessary step for transitioning to the FC layer. To prevent overfitting, a dropout layer is included, which randomly sets a fraction of the output units from the FC layer to zero during training. The dropout rate that works best is found to be 0.4, meaning that 40% of the output units in the FC layer are randomly dropped during training. By dropping out different sets of neurons during training, dropout forces the network to not rely on any single neuron, thereby promoting redundancy in the network. Consequently, neurons learn to function effectively even when some are inactive, enhancing the network's ability to generalise to new data. Given that the task at hand is a regression problem, the output layer consists of a single neuron. To introduce non-linearity, the convolutional and FC layers use the ReLU activation function. For more information on each layer,

please refer to Section 2.2.4.

Training

During CNN training, 30% of the training data is allocated for validation, a technique known as a validation split. This approach reserves 30% of the data for validation, enabling the evaluation of the model's performance on unseen data during the training process. For example, when a training set with a size of 200 is used, 140 simulations are used for actual training, and 60 simulations are used for validation. After each training epoch (a single forward and backward pass of the training dataset), the model's performance is assessed using the loss function on both the training and validation datasets. Typically, the loss on the training data decreases over time. However, monitoring the loss on the validation set is crucial to ensure it also decreases, indicating good generalization of the model. In this study, the mean squared error (MSE) is used as the loss function. The MSE is a widely used loss function for regression tasks due to its effectiveness in heavily penalizing larger errors more than smaller ones, encouraging the model to minimize significant inaccuracies.

To mitigate the risk of overfitting, an early stopping mechanism is employed. This technique halts the training process when there is no improvement in the validation loss after a specified number of epochs—in this case, 500 epochs. When this condition is met, the model restores the weights from the epoch where the validation loss was at its lowest, ensuring optimal generalization performance.

The learning rate is set to 10^{-4}, and the Adam optimizer is used. This optimizer is effective in managing sparse gradients and adapting the learning rate for each parameter, enhancing the training process. Training is performed on batches of 256 samples. Table 4.6 provides an overview of all training hyperparameters, as well as the hyperparameters of the architecture of the CNN used in this study.

Figure 4.8 displays an example of the evolution of the loss function (MSE) during training of the CNN. It shows that both the training and validation losses decrease rapidly, with the validation loss stabilizing after approximately 200 epochs. The training loss remains higher than the validation loss over time, which can be attributed to the use of dropout during training; this involves randomly omitting some neurons, whereas, during validation, all neurons are utilised.

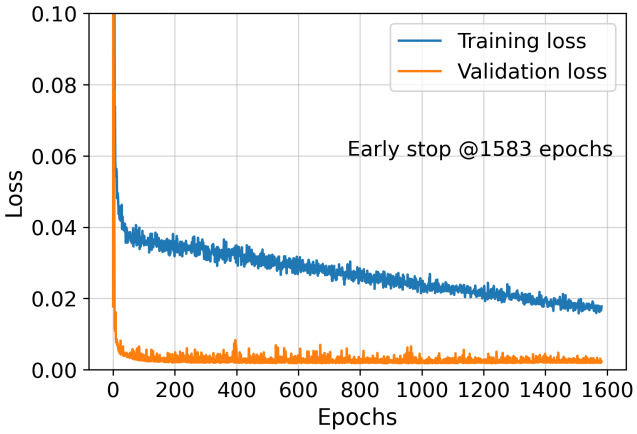


Figure 4.8: Evolution of the losses (MSE of the FoS prediction) during training of the CNN model.

Table 4.6: Hyperparameters used for the CNN model.

Layer	Hyperparameters	Settings used
Conv2D layer	Kernelsize	5x5
Conv2D layer	Filters Conv2D layer	20
Pooling layer	Pooling size	2x2
FC layer	Neurons	128
Dropout layer	Dropout rate	0.4
-	Batch size	256
-	Learning rate	0.0001
-	Optimizer	Adam
-	Loss function	Mean squared error

4.2. Results and Discussion

This section shows the results of the ML models applied to the cases considered. Furthermore, the performance of a full-surrogate model is discussed.

The performance criteria of each ML (semi-)surrogate model encompass two aspects: the total computational time needed and the accuracy of the predictions. Computational time includes the time needed to perform the FE simulations, the time required for training the ML model, and the time taken to make FoS predictions on the random fields using the ML model. The accuracy of the predictions is quantified by three metrics: the coefficient of determination (r^2), the root mean squared error (RMSE), and the relative error in the failure probability (p_f) predictions.

The r^2 indicates the proportion of variance in the dependent variable that can be explained from the independent variable(s). It is calculated as:

$$r^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \tag{4.3}$$

where SS_{res} and SS_{tot} are defined as:

$$\begin{aligned} SS_{\text{res}} &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ SS_{\text{tot}} &= \sum_{i=1}^{n} (y_i - \bar{y})^2 \end{aligned} \tag{4.4}$$

where y_i represents the observed values from FE simulations, $\hat{y}_i$ represents the predicted values by the ML model, $\bar{y}$ is the mean of the FEM-predicted FoSs, and n is the number of observations. A higher value of r^2 indicates that the predicted values by the ML model explain a greater portion of the variance in the FoS values obtained by the FEM. The maximum value of r^2 is 1, which indicates a perfect fit between the predictions of the ML model and the FoS values obtained by the FEM.

The RMSE is a commonly used metric of the differences between values predicted by a model and the values observed. It is defined by:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{4.5}$$

The lower the RMSE, the better the predictions by the ML model.

The relative error of the p_f predictions is defined by:

$$\text{Relative error in } p_f[\%] = \frac{\text{ML-predicted } p_f - \text{RFEM-predicted } p_f}{\text{RFEM-predicted } p_f} * 100 \tag{4.6}$$

To further evaluate the ML models, the consistency of the models' predictions is assessed by training 20 distinct versions of each model for each training set size. This involves using 20 unique training sets for each model, as detailed in Section 4.1.3. To illustrate, 20 CNNs were obtained, each with a different set of 400 simulations as its training sample. The performance of each ML model is evaluated by calculating the mean, standard deviation, minimum, and maximum values of each performance metric across the 20 results.

The open-source library TensorFlow [1], implemented in Python, is used for training the CNN models. The scikit-learn library [38] is used for training the PCA-RF and PCA-SVR. All computations, including the RFEM simulations, are performed on the DelftBlue cluster, equipped with Intel XEON E5-6248R 24C 3.0GHz CPUs [10].

4.2.1. Case U1

For reference, the input statistics of Case U1 are repeated in Table 4.7.

Figures 4.9, 4.10, and 4.11 present the ML-predicted FoS values using the PCA-SVR, PCA-RF, and CNN, respectively, against the FEM-predicted FoS values on the test set. The predictions are made

Table 4.7: The point statistics and scales of fluctuation of the random field for Case U1.

Case	Undrained shear strength			Scale of fluctuation		Extra information
	Mean [kPa]	COV	Distribution	θ_v [m]	θ_h [m]	
U1	27.76	0.3	Normal	6.1	60	Max scale of fluctuation values for s_u of clay, according to Phoon and Kulhawy [39]

by training the ML models on 100, 200, and 400 random FE-simulations. Note that the predictions displayed are made using a single training set (out of 20 sets). The r^2 and RMSE are denoted in the top left corner of every plot. A general observation is that the predictions made by all the models are reasonably accurate, indicated by the close fit to the diagonal line in each subplot. The r^2 is above 0.949 for all three models when using 400 training samples, indicating a strong relationship between ML and FEM FoS predictions.

All three models exhibit improved performance with increased training size. In terms of r^2 and RMSE, the CNN outperforms the other two models at each training sample size. With all the ML models, the predictions made in the lower and upper tails of the distribution are especially off. This is most likely attributed to the low number of instances in these FoS regions that are included in the training dataset.

It is interesting to point out that the tails of prediction distribution by the PCA-SVR are curved, which is due to the choice of the RBF kernel. With increased training sample size, it is observed that these tails become less apparent. Given this, the data augmentation technique (as explained in Section 4.1.4) is employed to investigate its effect on the performance of the PCA-SVR and the other two models.

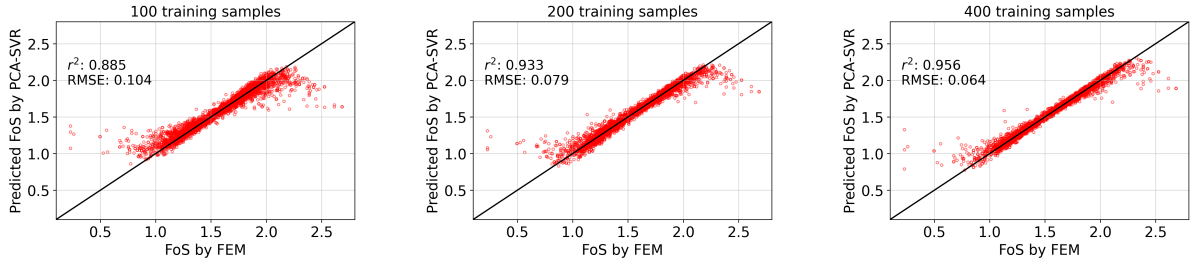


Figure 4.9: FEM vs. PCA-SVR predictions on the FoS for three different training set sizes.

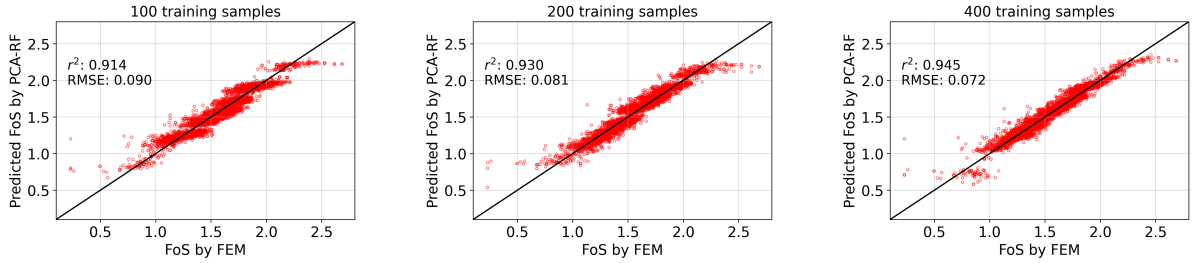


Figure 4.10: FEM vs. PCA-RF predictions on the FoS for three different training set sizes.

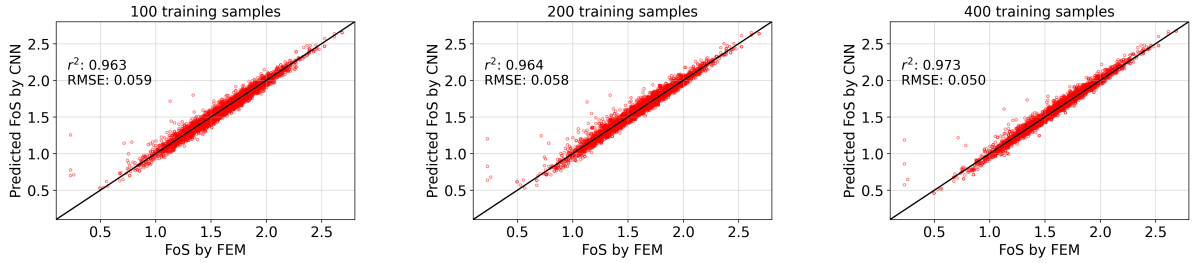


Figure 4.11: FEM vs. CNN predictions on the FoS for three different training set sizes.

Figure 4.12 illustrates the PCA-SVR predictions when the data augmentation technique is used with a training set size of 100, 200, and 400. Notably, the distribution's 'curved tails' are indeed less

prominent, particularly in the lower tail. This improvement is significant, as the most critical predictions – those related to the failure probability that the RFEM analysis targets – are typically located in this lower tail. Furthermore, the performance metrics showed positive changes: the r^2 increased and the RMSE decreased when using the data augmentation technique. These changes confirm the effectiveness of the data augmentation method when integrated with the PCA-SVR. The data augmentation did not lead to any performance improvements for the PCA-RF and CNN, as shown in Figure C.2 and Figure C.3 in the appendix, respectively.

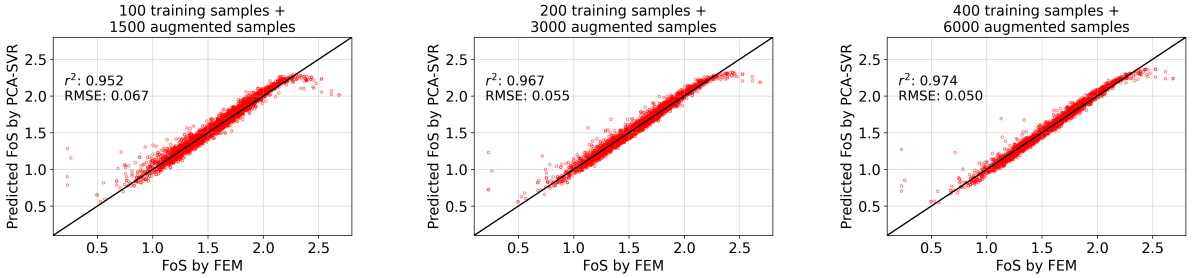
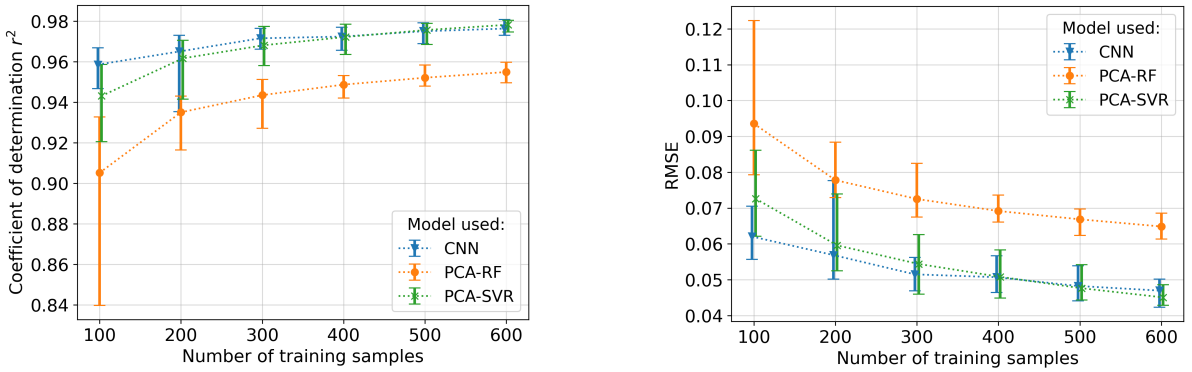


Figure 4.12: FEM vs. PCA-SVR predictions on the FoS for three different training set sizes, with PCA-SVR employing the data augmentation technique.

Figures 4.13a and 4.13b further examine how the RMSE and r^2 evolve as the size of the training set increases. In addition, they help assess the consistency of the ML models by displaying the mean, minimum, and maximum values of the performance metric across the 20 training and prediction cycles, using different training sets. The mean metric values are shown with dots, and the error bars indicate the ranges, each spanning the lowest and highest values from the 20 training and prediction cycles. Note that only the PCA-SVR makes use of the data augmentation technique, as discussed before.

It is observed that the r^2 increases and the RMSE decreases with increasing training sample size. These observations suggest that all three models are effectively learning the relationship between random fields of s_u and associated FoS. In addition, the shrinking range of the error bars suggests that the models become more robust with increasing training sample size.

One can observe that the metrics using the CNN and PCA-SVR in combination with the data augmentation technique are comparable and that the two models outperform the PCA-RF.



(a) r^2 **(b)** RMSE

Figure 4.13: Performance metrics r^2 (a) and RMSE (b) obtained by the three ML models on the test set of Case U1 using different training sample sizes. The dots indicate the mean metric value across 20 independent training and prediction cycles. The error bars show the range, indicating the minimum and maximum values across these 20 cycles.

Figure 4.14 displays the predictions of p_f, calculated based on the ML-predicted FoS datasets, for different training sample sizes. The right axis shows the relative percentage error of the ML p_f predictions to the p_f obtained by the RFEM, which is itself depicted by the black dotted line. Again, the dots in the plot represent the mean computed values, and the error bars indicate the range of the computed p_f. In line with observations from Figure 4.13, the p_f predictions generally improve with increasing training set size for the CNN and PCA-RF models. This does not hold for the PCA-SVR,

whose predictions hover around an 11% relative error for various training set sizes. Combined with an improvement in r^2 and RMSE, this indicates that, with increased training size, the PCA-SVR predictions fit better in the 'body' of the FoS distribution but not in the lower tail.

In addition, the figure shows that the range of the error bars generally decreases for all models with an increased training sample size. This is in line with observations from Figure 4.13 and highlights the increased consistency of the models. Notably, the PCA-RF model shows the largest percentage decrease in error range, suggesting that it particularly benefits from larger training datasets to make p_f predictions.

The PCA-SVR and CNN models demonstrate superior performance in p_f prediction compared to the PCA-RF model, as indicated by the narrow range of predicted values and low mean relative error across 20 training and prediction cycles. Accordingly, detailed results for the PCA-SVR and CNN models are presented in Table 4.8 and Table 4.9, corresponding to Figure 4.14a and Figure 4.14c, respectively.

These tables also include the COV of the ML predicted p_f values obtained across the 20 training and prediction cycles and the Unit COV, computed by a modified version of Eq. 2.15, to account for the training and prediction time of the ML model:

$$Unit\ COV = COV(p_f) * \sqrt{N_{sim} + \frac{t^*}{t}} \tag{4.7}$$

Here, N_{sim} is the number of simulations needed to train the ML model on, t^* is the computational time needed for the ML model to both train and predict the 'remaining' FE simulations, and t is the computational time needed for a single FE simulation. For a traditional MCS, the fraction $\frac{t^*}{t}$ is set to zero, as it doesn't make use of a surrogate modeling technique. A relatively small value of Unit COV indicates relatively high computational efficiency.

For this equation, the average summed training and predicting time for the PCA-SVR was within a few seconds. To compare, for the CNN, the average training time ranged from 3 minutes for a training set size of 100 realisations to 10 minutes for a training set size of 600 realisations. Predictions using the CNN were made within two seconds. For the PCA-RF, the training time lasted on average 50 seconds, and the prediction time was within a second. While the CNN requires on average more training time, it's notable that a single FE simulation for this slope typically takes around 60 seconds, providing context for the training and prediction time by the ML models.

The tables show that the Unit COV can be reduced largely by employing PCA-SVR or CNN as a semi-surrogate model while achieving low variance and error in the p_f.

Table 4.8: Results from the full MCS and PCA-SVR-aided slope reliability analyses for Case U1.

Method	Number of training samples	Probability of failure p_f	Error in p_f (%)	COV of p_f	Unit COV	Reduction in Unit COV (%)
MCS	-	0.0298	0	0.0903	5.786	0
PCA-SVR	100	0.0327	9.83	0.1366	1.357	76.23
	200	0.0338	13.73	0.0680	0.963	83.15
	300	0.0333	11.97	0.0684	1.184	79.25
	400	0.0326	9.45	0.0556	1.111	80.53
	500	0.0360	13.94	0.0375	0.838	85.32
	600	0.0348	10.63	0.0347	0.849	85.13

Table 4.9: Results from the full MCS and CNN-aided slope reliability analyses for Case U1.

Method	Number of training samples	Probability of failure p_f	Error in p_f (%)	COV of p_f	Unit COV	Reduction in Unit COV (%)
MCS	-	0.0298	0	0.0903	5.786	0
CNN	100	0.0354	18.95	0.1362	1.383	75.78
	200	0.0329	10.46	0.0850	1.215	78.72
	300	0.0325	9.24	0.0574	1.004	82.42
	400	0.0327	9.75	0.0994	2.003	64.91
	500	0.0319	7.27	0.0699	1.575	72.42
	600	0.0319	7.27	0.0657	1.621	71.61

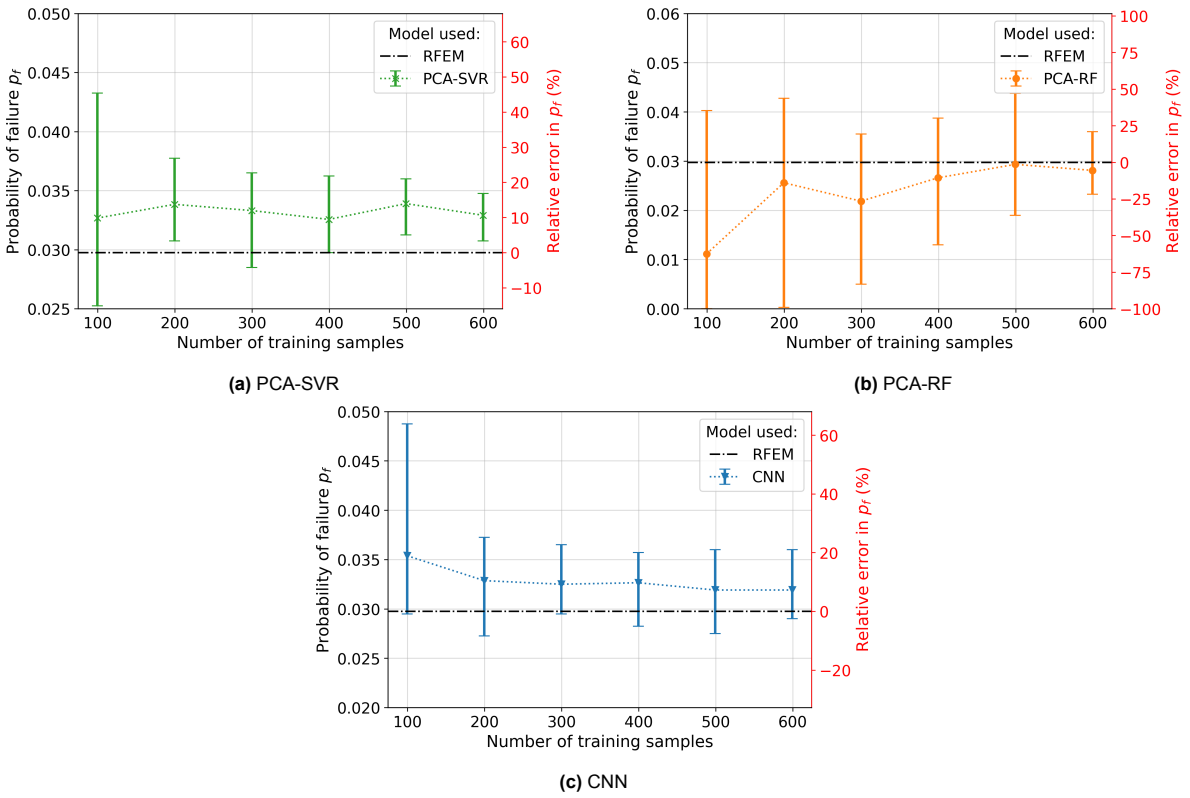


Figure 4.14: The predicted probability of failure p_f on the test set of Case U1 using different training sample sizes by the PCA-SVR (a), PCA-RF (b), and CNN (c). The dots indicate the mean metric value across 20 independent training and prediction cycles. The error bars show the range, indicating the minimum and maximum values across the 20 cycles. The black dotted line indicates the p_f obtained from a full MCS, comprising 4000 FE simulations.

4.2.2. Case U2

For reference, the input statistics of case U2 are repeated in Table 4.10.

Table 4.10: The point statistics and scales of fluctuation of the random field for Case U3.

Case	Undrained shear strength			Scale of fluctuation		Extra information
	Mean [kPa]	COV	Distribution	θ_v [m]	θ_h [m]	
U3	26.56	0.3	Normal	2.5	50.7	Mean scale of fluctuation values for s_u of clay, according to Phoon and Kulhawy [39]

Figures 4.15, 4.16, and 4.17 present the ML-predicted FoS values using the PCA-SVR, PCA-RF, and CNN, respectively, against the FEM-predicted FoS values on the test set. The predictions are made by training the ML models on 100, 200, and 400 random FE-simulations. Again, only the PCA-SVR makes use of the data augmentation technique as the performance of the PCA-RF and CNN did not improve when using it, which is demonstrated in Figure C.5 and Figure C.6.

Compared to case U1, the models performed worse in terms of r^2 and RMSE. It is clear from this figure that all ML models have more difficulty predicting instances that have a low FoS. This is particularly true for instances that fail almost immediately due to very local weak zones.

To improve performance in these challenging lower FoS regions, several approaches were attempted. First, adding extra training data from a slope with different input statistics was experimented with. The hypothesis was that including instances with a lower mean undrained shear strength would enhance the model's ability to learn from low FoS instances, thereby improving performance in the lower region of the FoS distribution of the original slope. However, this approach did not yield significant improvements, possibly because the COV in undrained shear strength in these additional instances was necessarily lowered to 0.1 to minimize the occurence of locally very weak zones in the random field

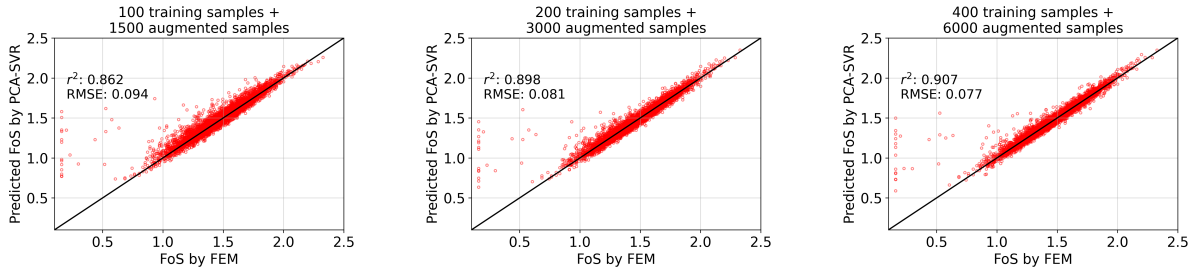


Figure 4.15: FEM vs. PCA-SVR predictions on the FoS for three different training set sizes.

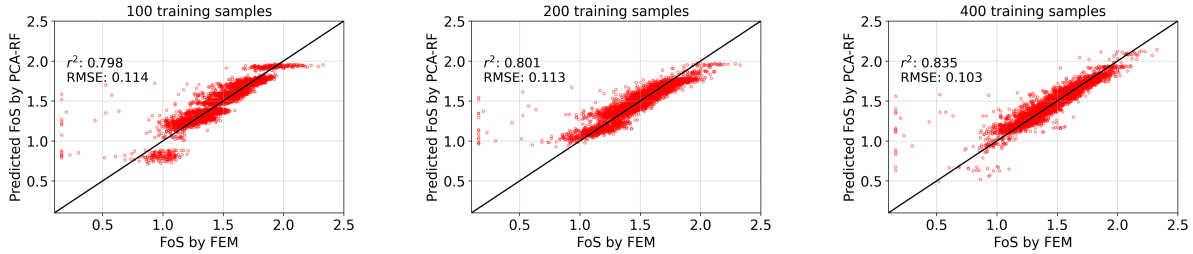


Figure 4.16: FEM vs. PCA-RF predictions on the FoS for three different training set sizes.

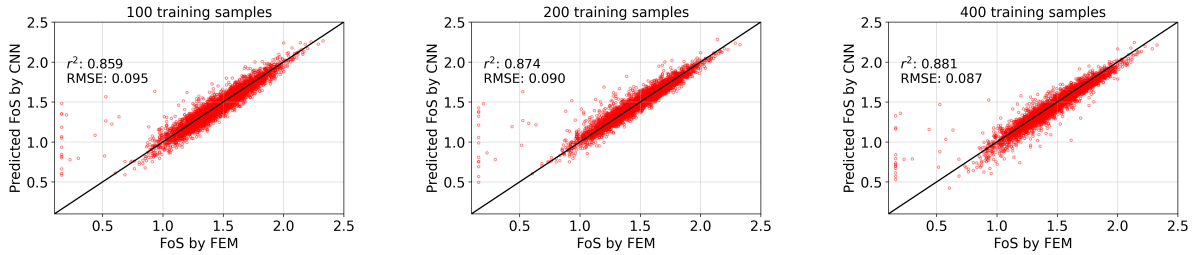


Figure 4.17: FEM vs. CNN predictions on the FoS for three different training set sizes.

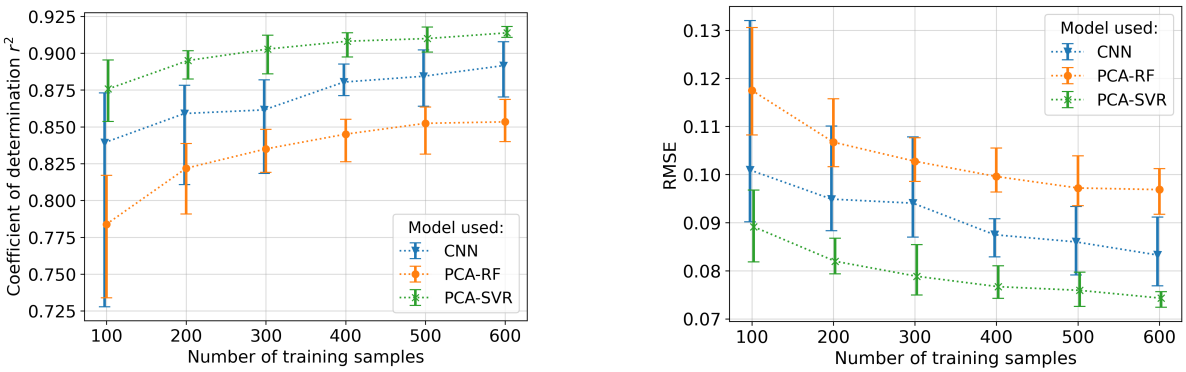
that trigger immediate failure. Consequently, the added instances lacked variation compared to the original training set and the models couldn't effectively learn from them.

Secondly, experiments were conducted with a custom MSE loss function used by the CNN, which penalised instances with low FoS more heavily. The hypothesis was that by better fitting in the lower region, the model would also predict other instances in this region more accurately. However, this modification did not lead to noticeable overall improvements.

Figure 4.18 further displays the evolution of the performance metrics r^2 and RMSE for the PCA-SVR, PCA-RF, and CNN for case U2 trained on different training set sizes. Note that the data augmentation technique is used solely for the PCA-SVR. The absolute metric values are worse in comparison with case U1. This suggests that for slopes with smaller spatial correlation lengths, ML models perform less satisfactorily in predicting the FoS.

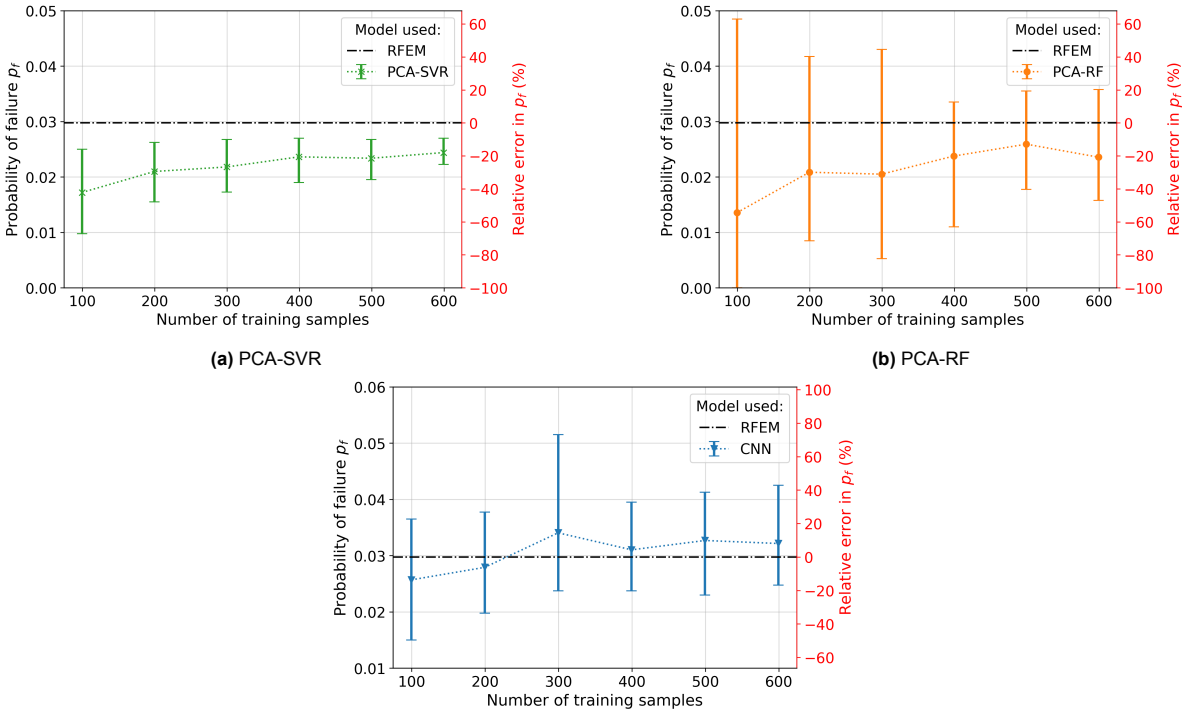
Figure 4.19 displays the predicted p_f by the ML models, calculated based on the ML-predicted FoS datasets, trained on different training set sizes. It shows that the CNN outperforms the PCA-SVR and PCA-RF in terms of the mean predicted value over 20 training and prediction cycles, with a maximum relative error of 9% when 400 realisations or more are used for training. However, it should be noted that the range of predicted p_f by the CNN model remains significant when trained on an equivalent or larger number of realisations, with a maximum of COV[p_f] equal to 0.125. In that light, the PCA-SVR predicts the p_f the most consistently over the 20 training and prediction cycles among the three ML models, having a maximum COV[p_f] of 0.08 when trained on 400 or more realisations.

It is important to note that while the range of ML-aided predictions of p_f can be wide, particularly with fewer training samples, it remains significantly narrower than the range observed in traditional RFEM analysis using an equivalent number of realisations, as shown in Figure 4.3.



(a) r^2 **(b)** RMSE

Figure 4.18: Performance metrics r^2 (a) and RMSE (b) obtained by the three ML models on the test set of Case U2 using different training sample sizes. The dots indicate the mean across 20 independent training and prediction cycles, and the errorbars indicate the minimum and maximum values across these 20 cycles. Note that only the PCA-SVR makes use of the proposed data augmentation technique.



(a) PCA-SVR **(b)** PCA-RF

(c) CNN

Figure 4.19: The computed probability of failure p_f on the test set of Case U2 for different training sample sizes by the PCA-SVR (a), PCA-RF (b), and CNN (c). The dots indicate the mean value across 20 independent training samples. The error bars show the range, indicating the minimum and maximum values across the 20 training samples. The black dotted line indicates the p_f obtained from a full MCS, comprising 4000 FE simulations.

4.2.3. Case U3

For reference, the input statistics of Case U3 are repeated in Table 4.11.

Table 4.11: The point statistics and scales of fluctuation of the random field for Case U3.

Case	Undrained shear strength			Scale of fluctuation [kPa]		Extra information
	Mean	COV	Distribution	θ_v [m]	θ_h [m]	
U3	24.34	0.3	Normal	0.8	46	Min scale of fluctuation values for s_u of clay, according to Phoon and Kulhawy [39]

Figures 4.20, 4.20, and 4.22 present the ML-predicted FoS values using the PCA-SVR, PCA-RF, and CNN, respectively, against the FEM-predicted FoS values on the test set. The predictions are made by training the ML models on 200, 400, and 600 random FE-simulations. Here, the data augmentation technique is also deployed for the PCA-RF (in addition to the PCA-SVR), as the model showed improvements in performance metrics. The enhancements achieved through data augmentation are further detailed in Figure C.7 and C.8 in the appendix. Observations similar to those made for cases U1 and U2 are noted.

Figure 4.23 further displays the evolution of the performance metrics r^2 and RMSE for the PCA-SVR, PCA-RF, and CNN for case U3 with different training set sizes. It shows that the PCA-SVR, combined with the data-augmentation technique, again outperforms the other two models on these metrics. This consistent observation across all three slope cases indicates that the PCA-SVR is best in FoS prediction for a slope modeled with the min-max range of spatial correlation lengths as commonly found in practice, according to Phoon and Kulhawy [39].

Additionally, while the performance metrics for case U3 are comparable to those for case U2, they still fall short of the results for case U1. This pattern highlights a consistent challenge: the ML models struggle more with predictions for slopes that exhibit smaller spatial correlation lengths.

Figure 4.24 displays the predictions of p_f, calculated based on the ML-predicted FoS datasets, for different training sample sizes. It shows that the p_f predictions made by the CNN are again closest to the p_f obtained from a full RFEM analysis. This consistent observation across all three slope cases indicates that the CNN is best capable of predicting the lower tail of the FoS distribution, especially with smaller spatial correlation lengths. A possible explanation for the latter is that the CNN is capable of capturing the relative position of the weaker to stronger zones in the random field by the convolution operation. For smaller spatial correlation lengths, this becomes more and more important, as the failure surface tends to 'search' for the weakest travel path.

Furthermore, the figure shows that there are more realisations in the training set needed for the CNN model to have a relative error within 10% and a COV[p_f] lower than 0.1, compared to case U1. Specifically, 500 realisations are needed for the CNN model in the training set to reach this level of accuracy, whereas only 300 realisations were sufficient for case U1.

Additionally, the figure again displays the smallest range of predicted p_f values across the 20 training and predicting cycles for the PCA-SVR, highlighting the outperforming robustness of this model.

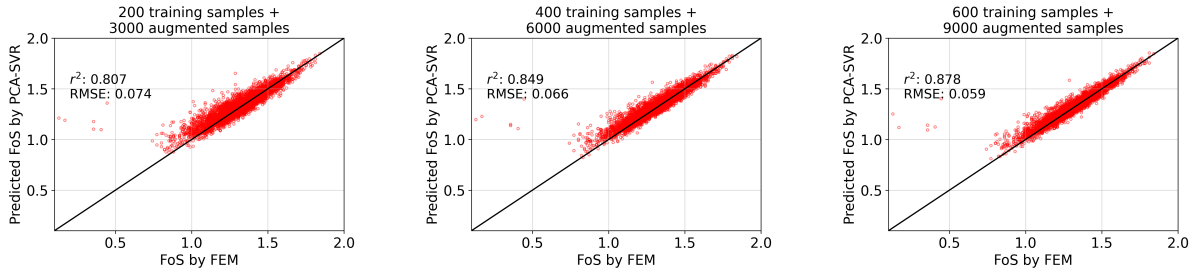


Figure 4.20: FEM vs. PCA-SVR predictions on the FoS for three different training set sizes.

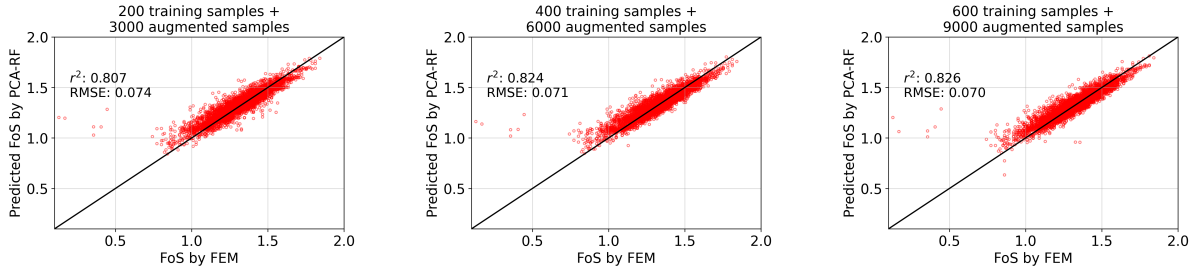


Figure 4.21: FEM vs. PCA-RF predictions on the FoS for three different training set sizes.

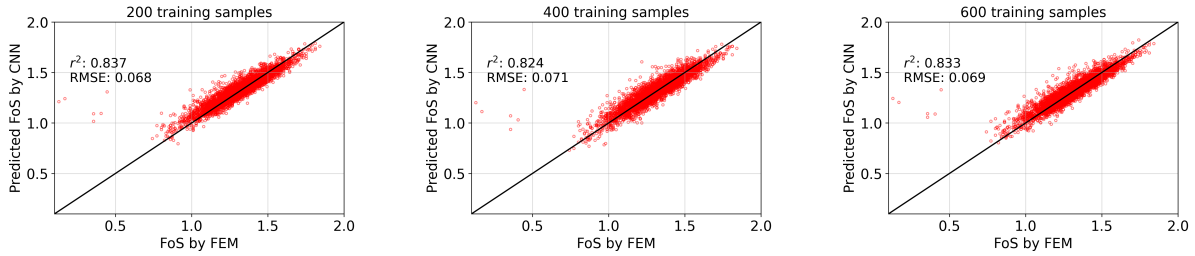
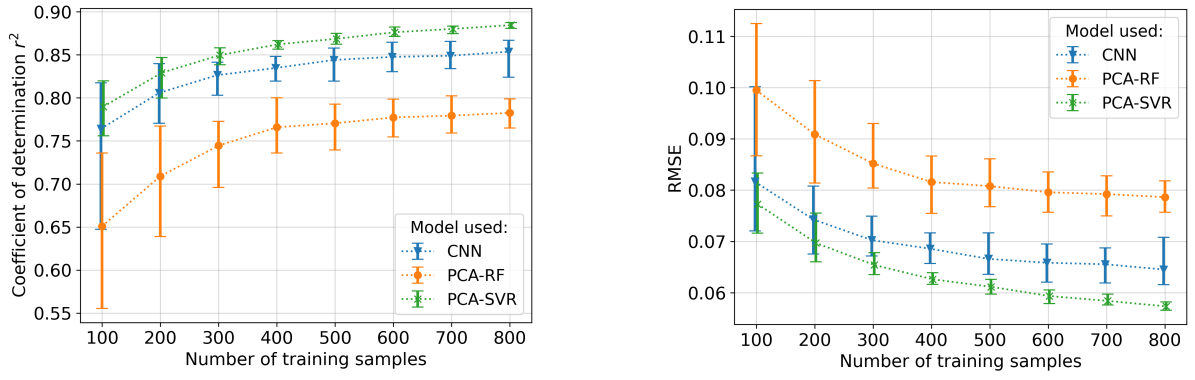


Figure 4.22: FEM vs. CNN predictions on the FoS for three different training set sizes.

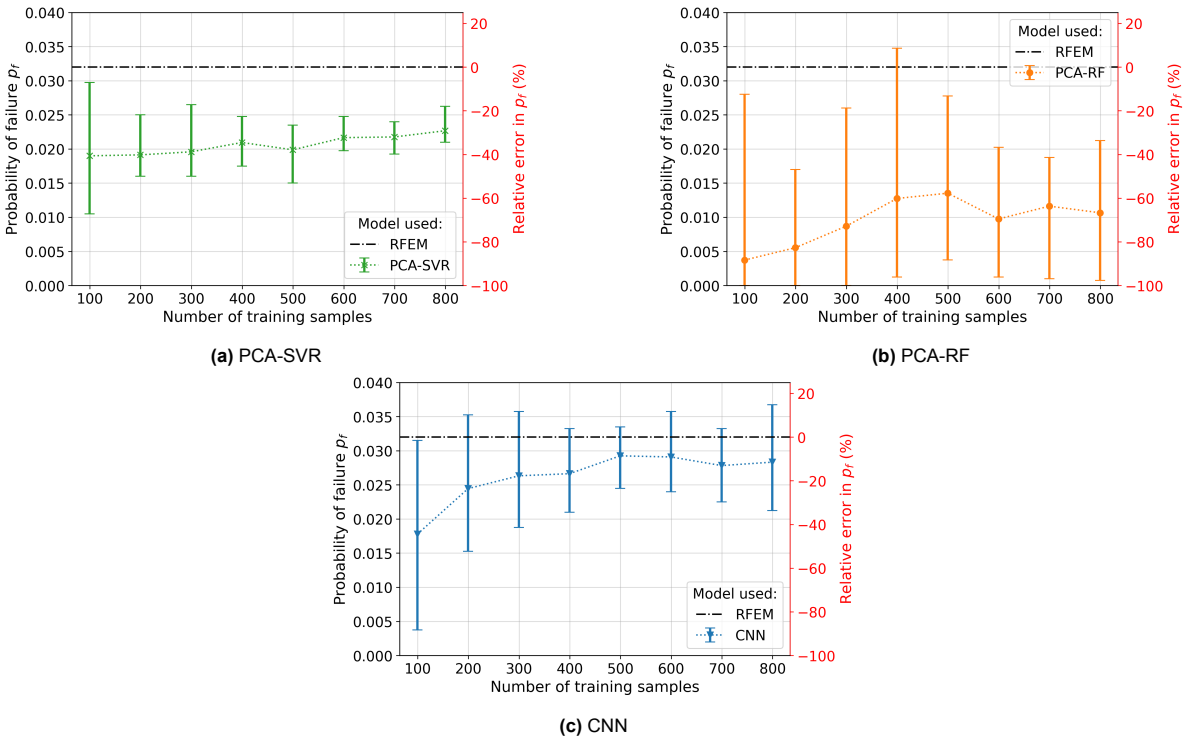


(a) r^2 **(b)** RMSE

Figure 4.23: Performance metrics r^2 (a) and RMSE (b) obtained by the three ML models on the test set of Case U3 using different training sample sizes. The dots indicate the mean value across 20 independent training and prediction cycles. The error bars show the range, indicating the minimum and maximum values across the 20 cycles. Note that the PCA-SVR and PCA-RF make use of the proposed data augmentation technique.

4.2.4. Full-surrogate model

So far, the training instances used by the ML models originated from a stochastically diverse dataset for a specific case with specified geometry, point statistics, and spatial variability. In practical terms, this means that if one wants to use an ML model to aid slope reliability analysis, there is a need to perform an *n* number of FE simulations of the identified case to serve as the training dataset. Although the total computational time can be reduced largely with this 'semi-surrogate' modeling technique, hundreds



(a) PCA-SVR **(b)** PCA-RF

(c) CNN

Figure 4.24: The computed probability of failure p_f on the test set of Case U3 for different training sample sizes by the PCA-SVR (a), PCA-RF (b), and CNN (c). The dots indicate the mean value across 20 independent training and prediction cycles. The error bars show the range, indicating the minimum and maximum values across the 20 cycles. The black dotted line indicates the p_f obtained from a full MCS, comprising 4000 FE simulations.

of FE simulations and multiple steps are still required to achieve reasonable results. Hence, a step further is to train a single surrogate ML model based on all available FE simulations such that no additional simulations will be needed when applying this 'full-surrogate' model to another different slope case. Specifically, the ML models are trained on the FE simulations of the 3 investigated cases with specified point statistics and scales of fluctuation as displayed in Table 4.2. Each case comprises 6400 simulations, adding up to a total of 19200 simulations. The trained models are then used to make predictions on 8 cases: the 3 investigated cases on which the models are trained (Cases U1, U2, and U3), and 5 cases that the models have not seen before (Cases N1, N2, N3, N4, and N5). The point statistics and scales of fluctuation for all test cases are listed in Table 4.12.

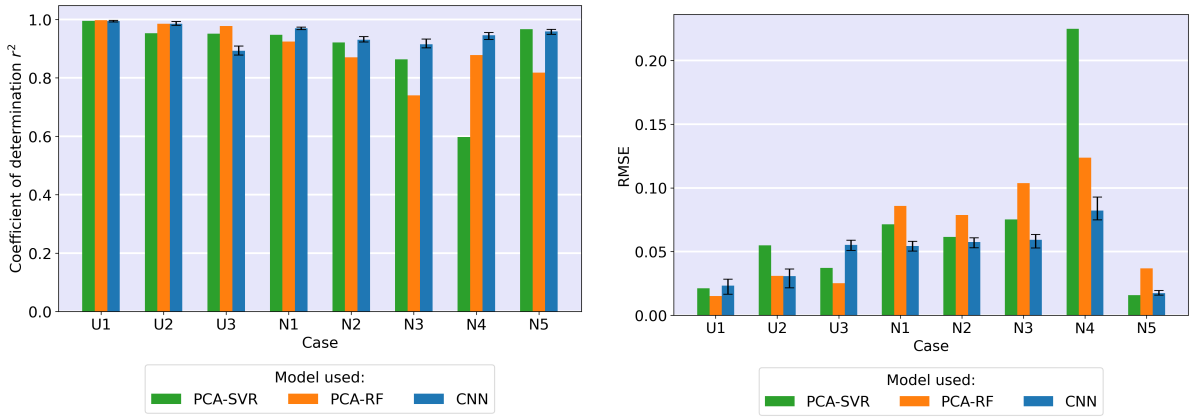
The scales of fluctuations for Cases N1 and N2 are arbitrarily chosen values that are within the range of values encountered in practice, according to Phoon and Kulhawy [39]. Specifically, Case N1 is assigned fluctuation scales between those of Case U1 and U2, while the values for Case N2 are between those of Case U2 and U3. Cases N3, N4, and N5 use Case U2 as a baseline, but each alters a statistical parameter of this case. For Case N3, the horizontal scale of fluctuation is decreased to an arbitrary value that is smaller than the width of the slope. For Case N4, the mean undrained shear strength is increased to an arbitrary value that is outside of the range on which the models have been trained, thereby increasing the point variability (standard deviation) of the random field cells. For Case N5, the COV of the undrained shear strength is lowered to 0.1, thereby reducing the point variability.

The mean of the undrained shear strength for cases N1 and N2 is adjusted until a p_f between 2.8% and 3.2% is obtained. This adjustment isn't made for Cases N3, N4, and N5, as the aim here is to explore the models' performances outside their training range. For each new case, 4000 FE simulations were performed and used for testing to ensure a reliable p_f (COV[p_f]<0.1) was obtained by the RFEM analysis for cases N1 and N2.

Table 4.12: The point statistics and spatial variability of the cases considered for the full-surrogate models.

Case	Undrained shear strength			Scale of fluctuation	
	Mean [kPa]	COV	Distribution	θ_v [m]	θ_h [m]
U1	27.76	0.3	Normal	6.1	60
U2	26.56	0.3	Normal	2.5	50
U3	24.34	0.3	Normal	0.8	46
N1	28.54	0.3	Normal	3.5	56
N2	25.71	0.3	Normal	1.5	48
N3	26.56	0.3	Normal	2.5	10
N4	33.39	0.3	Normal	2.5	50
N5	26.56	0.1	Normal	2.5	50

In the development of full-surrogate models, the hyperparameters and model architecture were largely kept consistent with those used in semi-surrogate modeling. A single modification was made for the PCA-RF: the number of trees (n_trees) in the forest was adjusted from 200 to 600. This was done to reduce the variance across multiple training and prediction cycles using the same training set. With this adjustment, the apparent variance in the r^2 metric was decreased to a level of 0.001. Ultimately, the variance is as low as possible for a reliable surrogate model.



(a) r^2 **(b)** RMSE

Figure 4.25: Performance metrics r^2 (a) and RMSE (b) obtained by the full-surrogate models on the test sets of 8 different cases.

Figure 4.25 shows the r^2 and RMSE of the predictions by the three full-surrogate models on the considered testing cases. For the CNN model, the bars in the figure illustrate the mean values of the metrics, averaged over 10 training and prediction cycles. Additionally, the error bars for the CNN indicate the minimum and maximum values observed across the 10 cycles. This repetition was done to account for the randomness of the training process of the CNN, stemming from random initialisations of the trainable parameters. Although the Random Forest model also incorporates randomness in its training via bagging, the effect on performance is minimal due to the large number of trees in the forest, which ensures low variance. Therefore, no repetitions were conducted for the Random Forest model. Similarly, repetitions were not done for the SVR due to the deterministic nature of the algorithm.

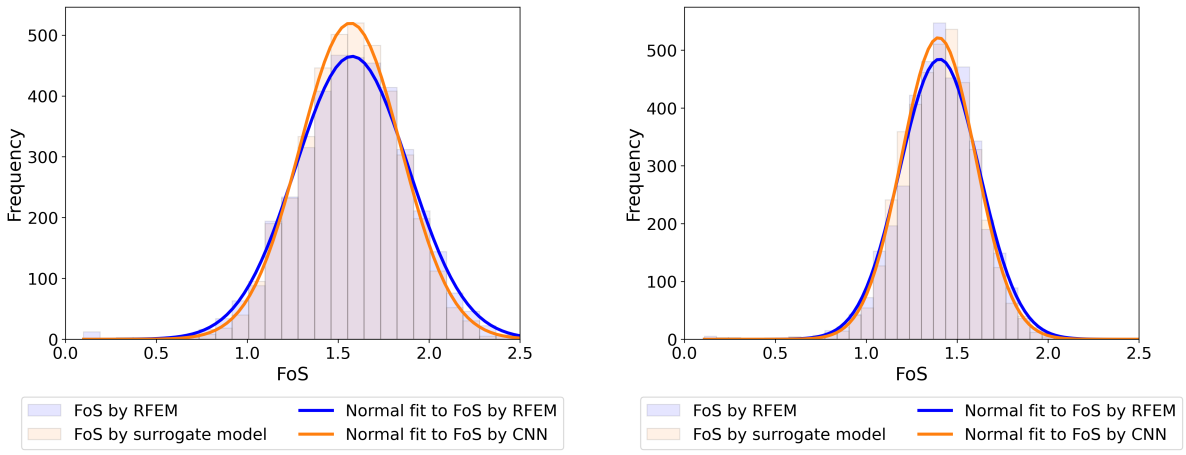
This figure is revealing in several aspects. First, all full-surrogate models demonstrate the ability to reasonably accurately predict the FoS across all cases (except for PCA-SVR for case N4), as evidenced by high r^2 values (>0.8) and low RMSE (<0.10) values, even when applied to cases that were not part of the training set.

Secondly, the CNN outperforms the PCA-SVR and PCA-RF on the test cases that were not part of the training set. Furthermore, the CNN proves to be the most robust model, maintaining high performance on cases N3, N4, and N5, which indicates its adaptability to different spatial variability. To further illustrate the CNN's high performance, Figure 4.26 presents histograms of the FoS predictions for cases N1 and N2, made by both the CNN and the RFEM. Each histogram is fitted with a normal distribution curve, showing that the probability density functions of FoS generated by the CNN closely match those from the RFEM analysis.

Thirdly, the performance of the full-surrogate models on the test cases U1, U2, and U3, on which

they have been trained, is comparable to and sometimes even exceeds the performance of the semi-surrogate models applied to each case. This observation suggests that the performance of the full-surrogate models across a broader range of cases can be enhanced by enlarging and diversifying the training set with realisations having different input statistics.

Lastly, when comparing the performance of the full-surrogate models applied to test cases N1, N2, and N3, which have spatial correlation lengths not included in the training set, it appears that the performance improves with larger spatial correlation lengths. This observation aligns with previous findings in semi-surrogate modeling.



(a) Case N1 **(b)** Case N2

Figure 4.26: Histograms of the CNN-predicted and RFEM realised Factor of Safety for case N1 and N2.

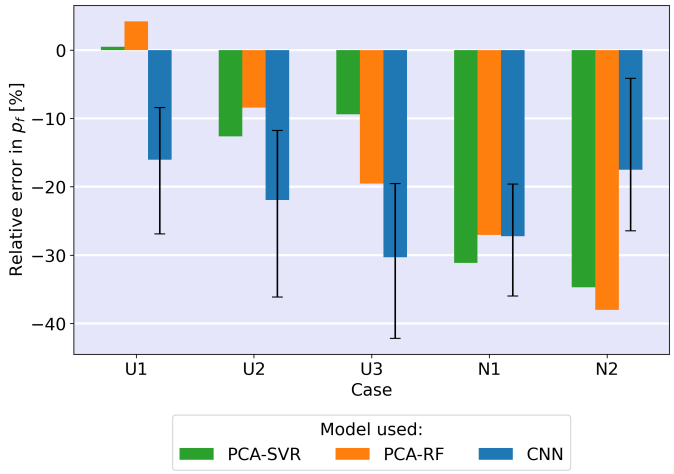


Figure 4.27: Relative error of the obtained p_f by the surrogate models to the p_f obtained by the FE-simulation for 5 different cases.

Figure 4.27 presents the relative error in ML-predicted p_f, calculated according to Eq. 4.6, for cases that have a RFEM derived p_f that is within the 2.8 - 3.2 % range. This condition is important because the predictability of the p_f is influenced by the number of failed instances in the training set, which is positively correlated with the p_f level. Consequently, the metric is not shown for cases N3, N4, and N5, for which the level of p_f is considerably different than this range.

This figure reveals some interesting points. First, the p_f can be predicted with reasonable accuracy (with a maximum relative error of 20%) by the PCA-SVR and PCA-RF for the slope cases that were part of the training set. This level of relative error in p_f is comparable or has decreased (for the PCA-RF) when compared to the results obtained from the semi-surrogate modeling for each individual case. Again, this suggests that the performance of the surrogate models can be improved on a broader range

of cases in terms of p_f prediction by diversifying the training set with realisations having different input statistics.

Secondly, for all cases (except case U1) the surrogate models are underpredicting the p_f. Underpredicting is a considerable drawback as engineers prefer a conservative prediction.

Lastly, the CNN outperforms the PCA-SVR and PCA-RF on p_f prediction when applied to cases with different input statistics than the realisations in the training set.

Regarding computational time, the full-surrogate models made predictions on 4000 random fields within seconds, whereas a full RFEM analysis consisting of the same number of simulations would take up to 67 hours on a local computer.

4.3. Main Findings

4.3.1. Semi-surrogate modeling

The main findings for semi-surrogate modeling for 2D RFEM slope reliability analysis are summarised as follows:

- All three machine learning semi-surrogate models predict the Factor of Safety with reasonable accuracy (RMSE between 0.05 and 0.10) using the random field of the undrained shear strength. This level of precision is reached when the models are trained on only 10% of the total number of realisations in RFEM slope reliability analysis. The accuracy of FoS predictions in this study is comparable with those obtained in previous research (e.g. [19, 50, 4]). This research found the PCA-SVR, when combined with the data augmentation technique, to perform best as a semi-surrogate model in FoS prediction. It achieved the highest performance in terms of r^2 and RMSE and demonstrated to be most robust to a varying training set.
- The performance of the considered ML models typically decreases for slope cases with smaller spatial correlation lengths of the undrained shear strength random field.
- This research found the CNN to perform best in predicting the probability of failure p_f. Using this model, the mean relative error in p_f prediction was as large as 10% for a slope case with relatively large spatial correlation lengths and as large as 17% for a slope case with relatively small spatial correlation lengths, when 400 realisations were used to train the model. However, the range of predictions across multiple training and prediction cycles using independent training sets can remain significant. As an illustrative example, a maximum COV[p_f] of 0.12 was noted across the three cases considered when 400 realisations were used for training.
- The data augmentation technique, as proposed by Jiang et al. [30], increases the PCA-SVR performance substantially, as evidenced by a consistently higher r^2, lower RMSE, and smaller relative error in p_f prediction.
- The total computational time needed to perform reliability analysis can be reduced largely with ML semi-surrogate modeling. To illustrate, for a slope with large spatial correlation lengths for the undrained shear strength random field [39], the computational time can be reduced from 67 hours for a full RFEM analysis to 4 hours (16-fold decrease) when the CNN is used as a semi-surrogate model while achieving a relative error in p_f of 8%. For a slope with smaller spatial correlation lengths for the undrained shear strength random field [39], more realisations are needed to achieve similar performance. Specifically, 500, instead of 300, realisations are needed to achieve the same relative error. Additionally, the range of predictions remain larger, even when more realisations are used for training. Using 500 realisations in the training set reduces the computational time to 8 hours, representing an 8-fold decrease.

4.3.2. Full-surrogate modeling

The main findings for full-surrogate modeling for 2D RFEM slope reliability analysis are summarised as follows:

- A full-surrogate CNN model is developed that makes reasonable FoS predictions (RMSE<0.075 and r^2>0.9) using the random fields of the undrained shear strength. This accuracy is achieved on eight testing cases, including slope cases with input statistics that were outside the range used in creating the training set. Predictions on thousands of random fields are made within seconds, compared to multiple days of computational time for a full MCS.

- The performance of the considered ML full-surrogate models typically decreases when applied to slope cases with smaller spatial correlation lengths of the undrained shear strength.
- This research found the PCA-SVR and PCA-RF full-surrogate models to predict the p_f most accurately when applied to random fields that were also part of the training set (max. relative p_f error of 20%). When applied to slope cases with different input statistics than the realisations on which the ML full-surrogate models were trained, the CNN demonstrated the best overall p_f predictions (max. relative p_f error of 27%).
- The performance of a full-surrogate ML model is expected to improve across various slope scenarios when the training dataset is enlarged and diversified in terms of the input statistics of the FE simulations.

Figure 4.28 presents an overview of the best performing ML (semi-)surrogate models for the 2D RFEM found in this research, identified for various goals.

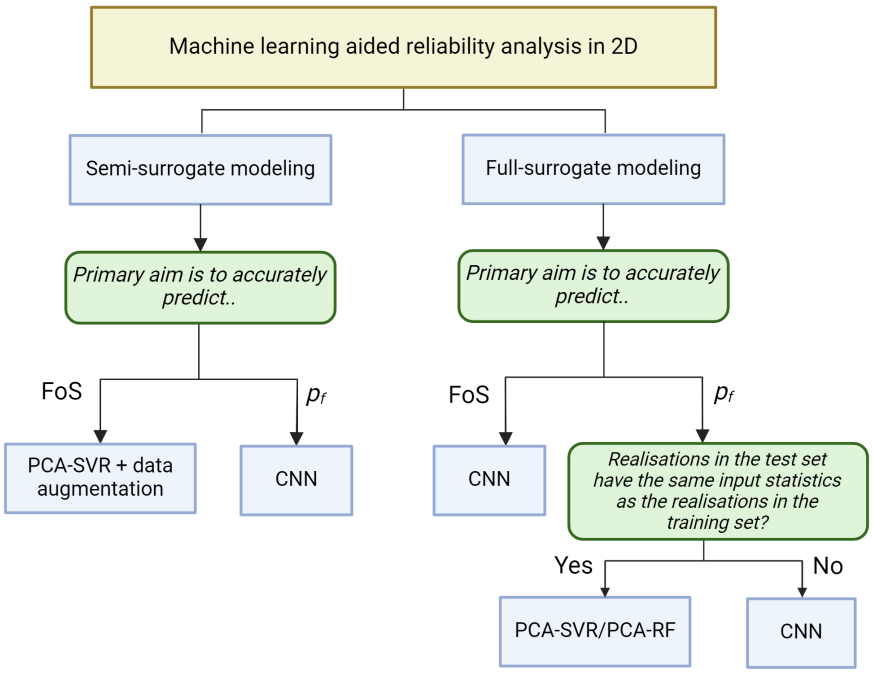


Figure 4.28: Overview of the best performing ML (semi-)surrogate models for the 2D RFEM found in this research.

5

3D Slope Reliability Analysis

In this chapter, the machine learning-aided slope reliability analysis is expanded to three dimensions (3D). This expansion is driven by the fact that no slope is truly two-dimensional. The presence of heterogeneous materials in the third dimension means that slope failures typically occur in 3D, which significantly influences the computed response and the predicted reliability [24].

This chapter starts by explaining the methodology, encompassing a description of setting up the RFEM model, two investigated slope cases with varying point statistics and spatial correlation lengths, and the utilisation and optimisation of three different ML models to make predictions of the FoS and probability of failure.

After going through the methodology, the results of the ML models as (semi-)surrogate models for the 3D RFEM are presented. This starts by assessing the performance of the ML models as semi-surrogate models for the 3D RFEM applied to the slope cases. Hereafter, full-surrogate models for the RFEM in 3D are evaluated, eliminating the need for additional numerical simulations when applied to a broader range of scenarios. Here, two strategies for creating such a surrogate model are compared.

To conclude the analysis in 3D, the main insights are presented.

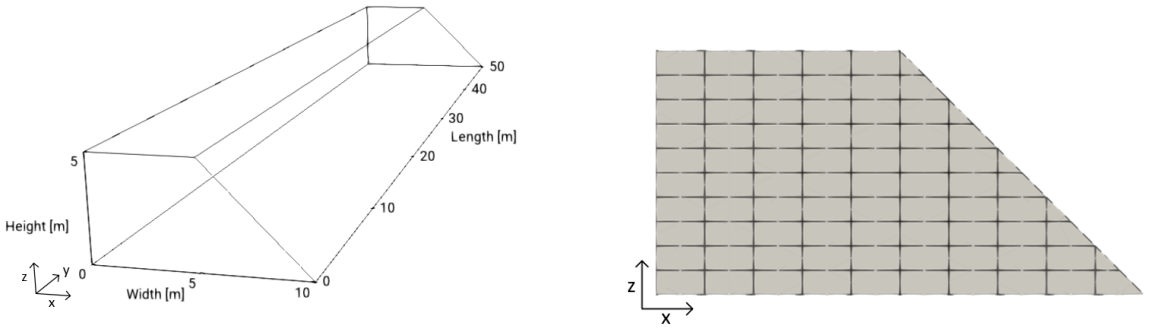
5.1. Methodology

5.1.1. Initialization of the RFEM

Figure 5.1b shows the idealised, fictive slope with respect to a Cartesian x, y, z coordinate system that is investigated in this research. The slope is characterised by an angle of 45°, a height of 5 meters, and a width of 10 meters, which are the same dimensions of the cross-sectional geometry used by Hicks and Spencer [24] and Varkey et al. [47]. In contrast to the slope geometry considered for the 2D RFEM analysis in this research, this slope doesn't have a foundation, which is done to reduce the overall number of elements and hence lower the computational cost. The length of the modeled slope is 50 meters, which proved to be long enough to cause failure through horizontal semi-continuous weaker zones [47]. This is important because such failure usually exhibits a lower FoS compared to a 2D solution.

The nodes on the base are fixed, preventing any movement in x-,y-, and z-direction. On the mesh ends, the boundary conditions are set as rollers, allowing movement only in the vertical direction. The nodes on the back side of the slope (vertical plane at $x=0$) are set as rollers, allowing movement in both the vertical as the out-of-plane direction. The validity of these adopted boundary conditions has been tested by Spencer [42].

The slope is discretized into 4000, 20-node, regular hexahedral elements, with each of size 1.0 × 1.0×0.5, except for those along the slope face, which have been distorted to fit the slope geometry. A cross-section of the slope mesh is shown in Figure 5.1b. Each element has 2×2×2 Gaussian integration points. For optimal utilisation, random field (cell) values are mapped onto these points instead of the elements themselves. In this configuration, eight random field cells cover the same volume as one finite element, leading to 32.000 random field cells.



(a) Isometric projection of the slope

(b) Element mesh through cross-section

Figure 5.1: Isometric projection of the slope geometry (a) and element mesh of cross-section (b).

The slope consists of clay modeled by a linear elastic, perfectly plastic stress-strain behaviour using the Mohr-Coulomb failure criterion and a non-associated flow rule (dilation angle = 0°). The elastic component of this model is defined by Young's modulus *E* = 100.000 kPa, and Poisson's ratio ν = 0.3. The unit weight of the clay γ = 20 kN/m^3. The undrained shear strength s_u is a stochastic parameter modeled by a random field, generated by the LAS technique with an underlying Gaussian distribution. The point statistics and the spatial correlation lengths are subject to change throughout several cases.

A maximum number of 500 iterations in the numerical calculations is used to indicate when failure has occurred, which was previously identified by Hicks and Spencer [24] to be sufficient to define failure. Using the SSRM, the FoS is computed with a resolution of 0.01.

5.1.2. Investigated cases

In this study, two slopes with varying point statistics and spatial correlation lengths are investigated, as detailed in Table 5.1.

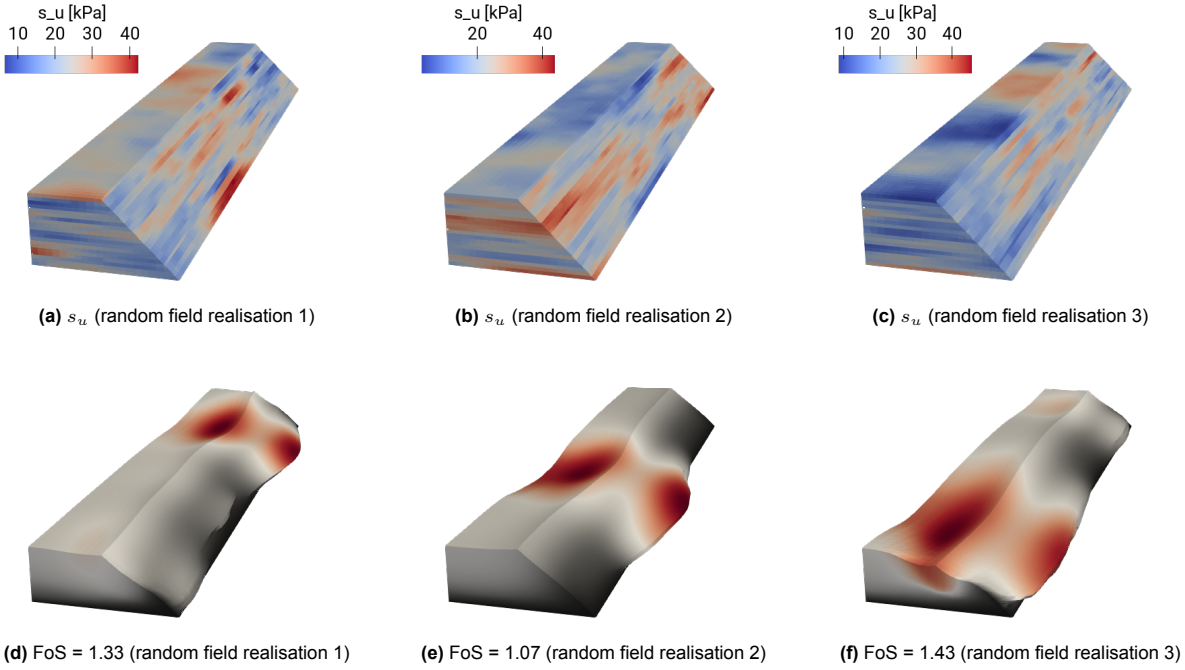
Table 5.1: The point statistics and scales of fluctuation for the cases considered.

Case	Undrained shear strength			Scale of fluctuation		
	Mean [kPa]	COV	Distribution	θ_z [m]	θ_x [m]	θ_y [m]
T1	21.96	0.25	Normal	1	20	20
T2	22.04	0.25	Normal	1	20	100

These slopes, labeled T1 and T2, have a consistent vertical scale of fluctuation θ_z of 1 meter, aligning with the value used by Varkey et al. [47] and Hicks and Spencer [24]. The scales of fluctuation in the horizontal plane, θ_x and θ_y, are set equal to each other for case T1, which is a common assumption made in 3D modeling. The value is set to 20 meters, which ensures 'mode 2' failure. This mode of failure is characterised by discrete three-dimensional failure that tends to pass through horizontal (semi-continuous) weaker zones [24]. According to Hicks and Spencer [24], such failure can be modeled if the horizontal scale of fluctuation θ_h meets the condition $H < \theta_h < \frac{L}{2}$, where *H* is the height of the slope and *L* is its length. Ensuring this mode of failure is occurring is important because it typically has a lower FoS compared to a 2D solution. Therefore, this failure mode is considered the most critical.

In case T2, the scale of fluctuation along the slope length, θ_y, is increased to 100 meters. This modification is based on findings from a study conducted on a Dutch dike, as reported by De Gast et al. [9]. Their research highlighted that the spatial correlation length along the dike's direction can vary significantly and often exceeds the correlation length across the dike.

The mean undrained shear strength s_u is assumed constant in each case and has been determined to ensure that the p_f derived from 4000 FE simulations is within the 2.8 - 3.2% range. The level of p_f is determined arbitrarily to strike a balance between the total number of simulations needed to achieve a consistent p_f and a practical value. In addition, considering the slopes with the same probability of failure ensures consistency and allows for a fair comparison between the errors in the ML-predicted p_f and the p_f obtained using 4000 FE simulations. The COV for the undrained shear strength is set



(a) s_u (random field realisation 1) **(b)** s_u (random field realisation 2) **(c)** s_u (random field realisation 3)

(d) FoS = 1.33 (random field realisation 1) **(e)** FoS = 1.07 (random field realisation 2) **(f)** FoS = 1.43 (random field realisation 3)

Figure 5.2: Three examples of random fields of the undrained shear strength (a, b, c) and the displacements (magnification = 500) at failure (d, e, f) computed using the FEM for slope case T1.

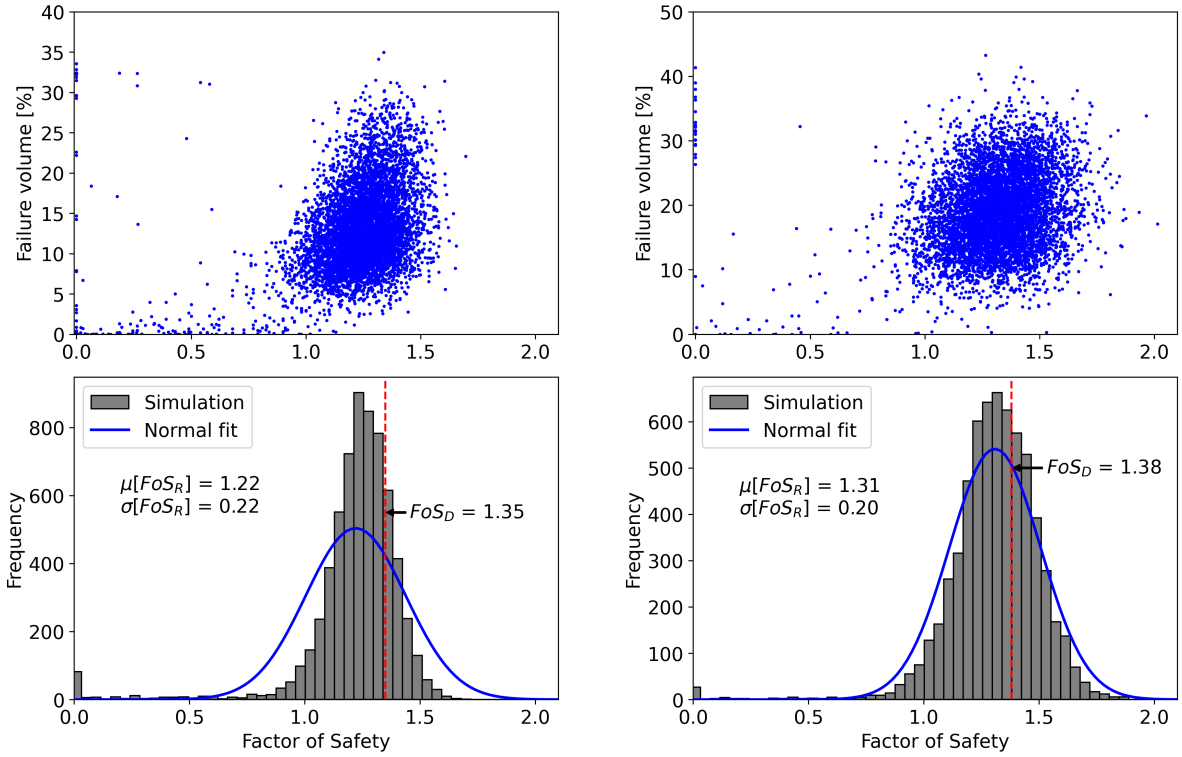
constant to 0.25 for both cases, which is within the typical range used in practice [39]. This represents a slight decrease from the COV value used in the previous 2D slope investigation. The adjustment aims to reduce the presence of extremely weak random field cells near the slope face, which can trigger local failures. The likelihood of encountering these weak random field cells near the slope face was expected to rise in 3D for two reasons: first, because of an increase in the portion of random field cells near the slope face, and second, because of the use of a lower mean undrained shear strength. Figure 5.2 presents three examples of generated random fields of s_u along with the resulting displacements at failure for case T1. It shows that discrete 3D failure occurs and that the failure surface passes through a weak zone (mode 2 failure) located in the lower portion of the slope. Figure B.3 in the appendix presents three examples of random fields generated for s_u and the resulting displacements at the point of failure for case T2. These examples show that mode 2 failure can still be the failure mode even if the spatial correlation length in *y* direction is much larger than the length of the slope. This finding is in agreement with the observations reported by Hicks et al. [22].

The choice to use 4000 realisations in the RFEM analysis is based on the fact that the COV of the p_f is well below 0.1 when the slope is at a p_f of 2.8%, according to Eq. 1.1.

In addition to calculating the FoS for each realisation, the slide volume is computed based on the number of elements having an average out-of-face displacement greater than a calibrated threshold value, following the methodology as described in Hicks et al. [22]. For case T1, the calibrated threshold was 23% of the maximum computed out-of-face displacement.

Histograms of the obtained factors of safety for each RFEM case using 6400 FE simulations are shown in the lower plots of Figure 5.3. A normal distribution is fitted on each histogram, and the mean and standard deviation are displayed in the top left corners, denoted as $\mu[FoS_R]$ and $\sigma[FoS_R]$, respectively. The FoS obtained using a deterministic analysis based on the mean undrained shear strength is denoted by the red dashed line and is denoted as FoS_D. The computed failure volume, expressed as a percentage of the total mesh volume, is plotted against the corresponding FoS for each realisation in the upper plots of Figure 5.3.

The figure points out that, despite efforts to reduce their occurrence by decreasing the COV of the undrained shear strength, a considerable number of realisations in both slope cases still exhibit an extremely low FoS. These realisations skew the fit of the normal distribution to the histogram. The corresponding failure consequence in terms of volume % for the majority of these outliers is very small, especially noticeable for case T1. Such small failure consequences are considered insignificant for



(a) Case T1 **(b)** Case T2

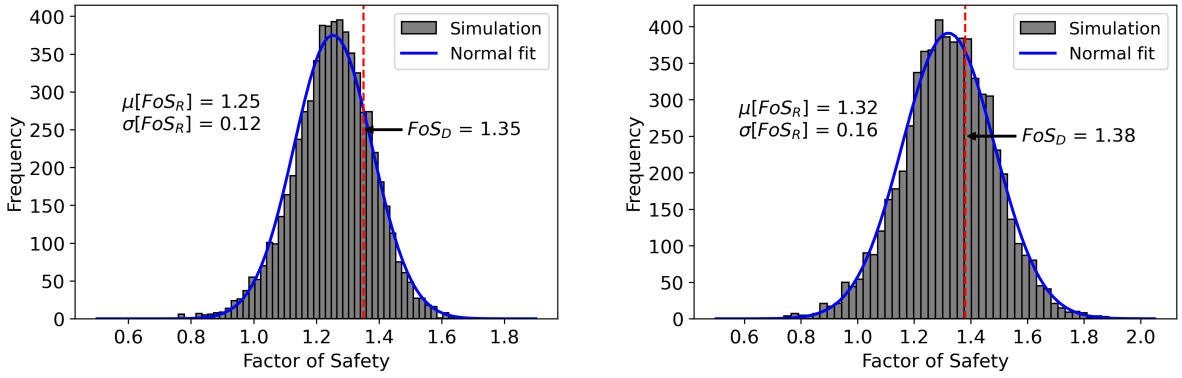
Figure 5.3: Histograms of the realised Factors of Safety using the RFEM analysis (FoS_R) for the two considered cases. The Factor of Safety based on the mean undrained shear strength is denoted by FoS_D.

reliability analysis, as it is (very) likely that they will not compromise the slope's overall functionality or the associated risk is very small. Consequently, instances where the slide volume was less than 2% of the total mesh volume were excluded from the datasets, resulting in the removal of 166 and 26 realisations for cases T1 and T2, respectively.

Additionally, Figure 5.3 points out that some realisations exhibit an extremely low FoS alongside a relatively large failure volume. Closer examination showed that these realisations also have a very locally failed zone but that the maximum displacement did not occur at the center of the failed zone but rather at a different location within the slope. An example of this is shown in Figure 5.5, where the displacements at the point of failure are depicted. This situation led to an automatic overestimation of the slide volume, presenting a consistent challenge to approximate the slide volume for such small local failures. As a workaround to exclude all very locally failed slopes, a simple strategy has been implemented: instances with a FoS below 0.75 are excluded. This led to an additional exclusion of 56 and 52 realisations for cases T1 and T2, respectively. Adopting this threshold value led to a closer match between the histogram and the fitted normal distribution, as depicted in Figure 5.4. Achieving this match is considered important, as normal distributions have previously been used to describe the FoS distribution effectively for 3D spatially varying slopes that are characterized by a Gaussian distribution of strength parameters (e.g., [20, 47]).

Consistent with prior studies (e.g., [24, 21]), the mean FoS obtained by the RFEM differs from the deterministic FoS, highlighting the importance of accounting for spatial variability of the soil. This discrepancy is relatively large for slope case T1, as strong and weak zones within each random field alternate frequently (relatively small θ_y), and consequently, the failure surface can go through weaker zones more easily. Conversely, for slope case T2, the random field instances appear to be composed of horizontally more uniform 'layers' (relatively large θ_y), and therefore it becomes more difficult for the failure surface to avoid crossing the strong zones, ultimately leading to an overall higher FoS.

Additionally, the histogram of case T1 shows a relatively small range of outcomes. This is because the failure surfaces pass through more diverse random field cells, leading to more strength averaging over the failure surface. Conversely, slope case T2 exhibits a relatively large range of outcomes as the



(a) Case T1 **(b)** Case T2

Figure 5.4: Histograms of the realised Factors of Safety using the RFEM analysis (FoS_R) for the two considered cases. The Factor of Safety based on the mean undrained shear strength is denoted by FoS_D.

failure surface passes through random field cells that are more alike, either being more consistently strong or weak.

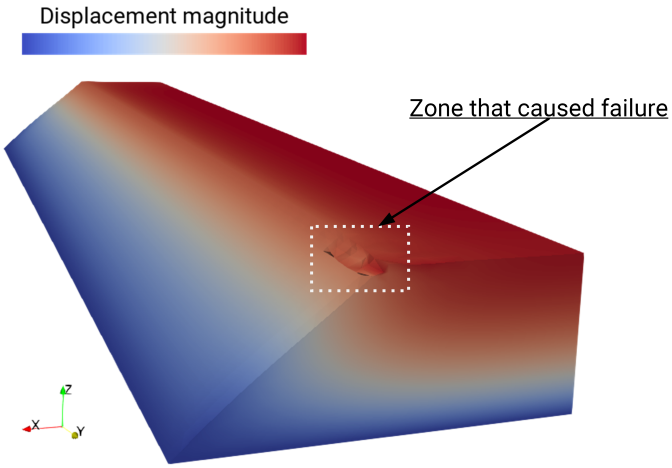


Figure 5.5: Example of a locally failed slope where the maximum displacement did not occur at the center of the slope. The calculated Factor of Safety for the slope is 0.25. The approximated failure volume, being 30% of the total volume, is likely to be inaccurate.

Note that a single realisation typically takes up to 30 minutes on average on the DelftBlue cluster, equipped with Intel XEON E5-6248R 24C 3.0GHz CPUs [10].

5.1.3. Data splitting

The same data splitting technique as used in 2D slope reliability analysis is employed (refer to Section 4.1.3). Thus, for each slope case, the initial dataset consists of 6400 random fields with associated FoSs. Note that slopes with very local failures have already been excluded from this dataset.

5.1.4. Machine learning models for FoS prediction

The same (combined) ML models are explored for 3D RFEM (semi-)surrogate modeling as are done for 2D RFEM (semi-)surrogate modeling. Specifically, the PCA-SVR, PCA-RF, and CNN models are investigated. This selection is based on their demonstrated effectiveness in 2D RFEM analysis. Additionally, there is a lack of previous research on 3D RFEM (semi-)surrogate modeling to serve as a benchmark, making this exploration novel.

The procedure for applying the ML models as a (semi-)surrogate model for 3D RFEM analysis is mostly the same as for 2D RFEM analysis, with a few adjustments to the inputs and the hyperparameters. This section lists these for every model considered.

5.1.4.1 CNN

Input

Following the same procedure for inputting 2D slopes into the CNN, the 3D random fields are first mapped onto 3D digital images. This mapping is done using the 'closest voxel' procedure, which is the 3D counterpart to the 2D 'closest pixel' procedure, as detailed in Section 4.1.5.3. In addition, zeros are added to the 3D digital images for parts that do not represent any part of the slope. As an illustrative example, Figure 5.6 shows the associated 3D digital image of the slope depicted in Figure 5.2a. For the 3D slope geometry considered in this study, the 3D digital images are structured with dimensions of 20×100×20 voxels.

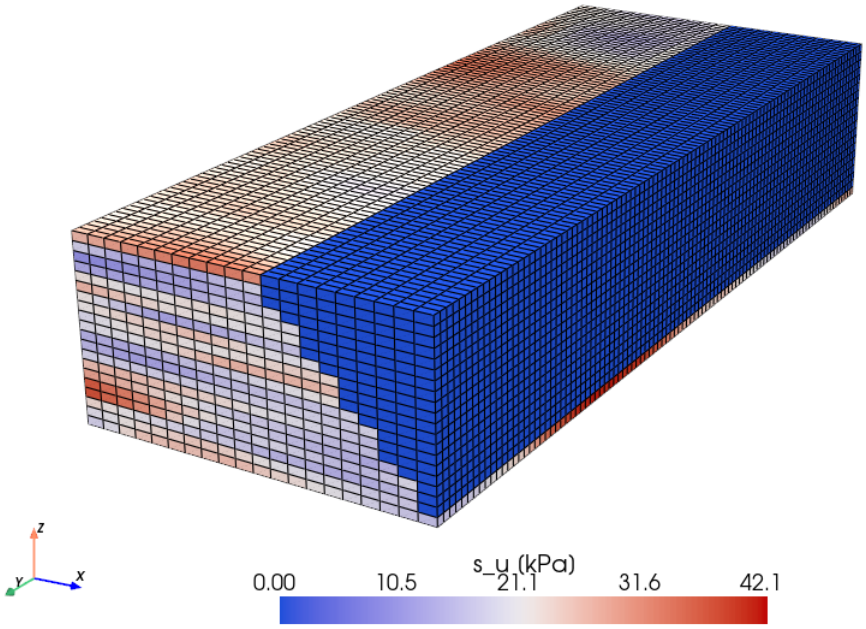


Figure 5.6: Example of a pre-processed 3D random field represented in a matrix format.

Architecture

The CNN architecture is subject to hyperparameter tuning by iteratively adjusting the hyperparameter until the model becomes complex enough to have good performance while maintaining a low number of trainable parameters to reduce training time. Experiments were done using either one or two stacked convolutional layers, and either one or two fully connected (FC) layers. Table 5.2 displays an overview of the hyperparameters tested.

Table 5.2: Hyperparameters tested for the CNN model.

Layer	Hyperparameters	Settings tested
Conv3D layer 1	Kernelsize	3x3, 5x5, 7x7
Conv3D layer 1	Filters	10, 20
Conv3D layer 2	Kernelsize	3x3, 5x5
Conv3D layer 2	Filter	10, 20
Pooling layer 1 & 2	Pooling size	2x2
FC layer 1	Neurons	128, 256, 512
FC layer 2	Neurons	128, 256, 512
Dropout layer	Dropout rate	0, 0.1, 0.2, 0.3, 0.4, 0.5
-	Batch size	256
-	Learning rate	0.0001
-	Optimizer	Adam
-	Loss function	Mean squared error

Two FC layers within the CNN architecture showed a competitive advantage over one for both slope cases, highlighting the increased complexity of the 3D slope problem compared to the 2D problem. Specifically, the network that showed the lowest loss on the validation set comprises two convolutional layers, two MaxPooling layers, and two FC layers. The two convolutional layers use a kernel size of 7×7 and 5×5, and the two stacked MaxPooling layers have a filter size 2×2. The pooled maps are flattened after the second MaxPooling layer, and the two FC layers are included hereafter. Finally, a dropout layer with a dropout rate of 0.4 is added. The output layer is a single neuron. All convolutional and FC layers use the ReLU activation function.

Training

During training, 30% of the training data is allocated for validation. A maximum of 5000 epochs is used for training, and an early stopping mechanism is employed to halt training when there is no improvement in the validation loss after a specified number of epochs—in this case, 500 epochs. The learning rate is set to 10^{-4}, and the Adam optimizer is used. Training is performed on batches of 256 samples. An overview of the training hyperparameters used, as well as the hyperparameters for the CNN architecture, is shown in Table 5.3.

Figure 5.7 displays an example of the evolution of the loss function (MSE) during training of the CNN. It shows that both the training and validation losses decrease rapidly, with the validation loss stabilizing after approximately 400 epochs. The training loss remains higher than the validation loss over time, which can be attributed to the use of dropout during training; this involves randomly omitting some neurons, whereas, during validation, all neurons are utilised.

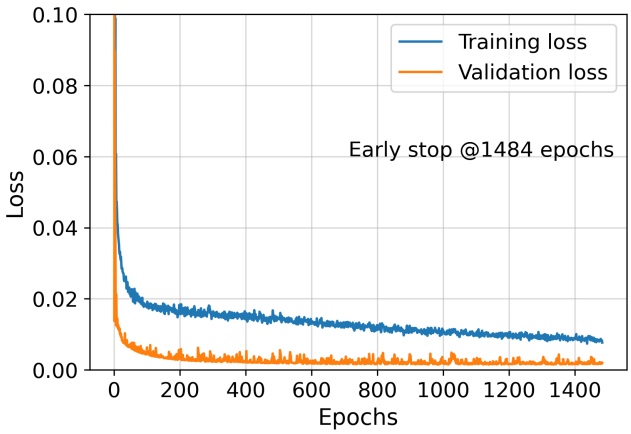


Figure 5.7: Evolution of the losses (MSE of the FoS prediction) during training of the CNN model.

Table 5.3: Hyperparameters used for the CNN model.

Layer	Hyperparameters	Setting used
Conv3D layer 1	Kernelsize	7x7
Conv3D layer 1	Filters	20
Conv3D layer 2	Kernelsize	5x5
Conv3D layer 2	Filter	20
Pooling layer 1 & 2	Pooling size	2x2
FC layer 1	Neurons	512
FC layer 2	Neurons	256
Dropout layer	Dropout rate	0.4
-	Batch size	256
-	Learning rate	0.0001
-	Optimizer	Adam
-	Loss function	Mean squared error

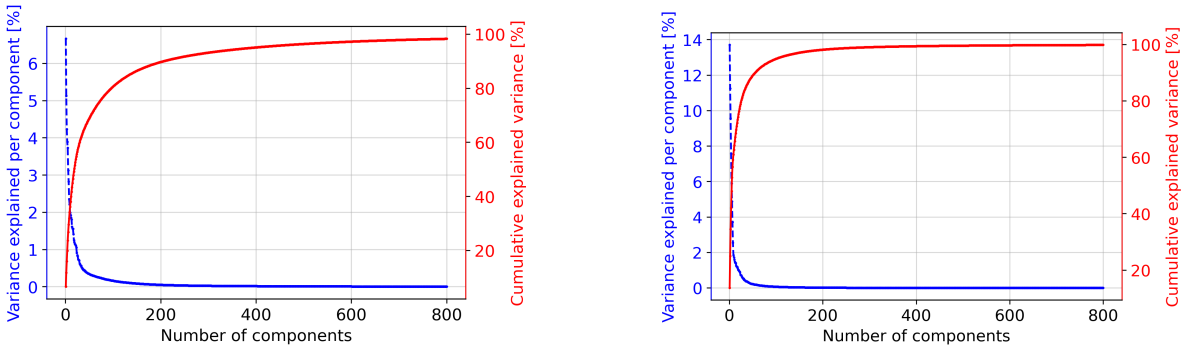
Due to the high computational demands of training the CNN on 3D matrices compared to 2D, the process is carried out on the Delft University of Technology's Geoscience & Engineering (GSE) server, which is equipped with an Nvidia A100 80GB GPU.

5.1.4.2 PCA-SVR

Input

The input for the PCA-SVR in 3D RFEM analysis is similar to that used in 2D RFEM analysis, in that both utilise 1D arrays. However, there's a difference in how these arrays are generated. Because of storage limits on the DelftBlue cluster, either the 3D matrices or 1D arrays could be stored remotely. Due to the memory limits of the personal laptop (8GB of RAM), the matrices could not be created out of the 1D arrays locally. This way, the choice was made to create the 3D matrices remotely and backfigure the 1D arrays from the 3D matrices locally by flattening the 3D matrices and removing the zeros. As the nearest voxel procedure used for creating the 3D matrices overwrites some random field cells close to the slope face, the dimensionality of the 1D arrays was reduced to 30.500, compared to the original 32.000 random field cell values. However, this information being lost is not considered crucial, as the undrained shear strength of neighboring cells is very similar to that of these cells with the specified spatial correlation lengths.

Figure 5.8 presents the scree plots resulting from the PCA conducted on the training set for the cases under study. The plots display the percentage of total variance each principal component explains (shown on the left axis) and the cumulative explained variance as a function of the number of components (shown on the right axis). Similar to the 2D PCA results, a small number of components capture most of the variance. However, due to the additional dimension affecting the strength parameter, explaining an equivalent portion of the total variance requires a significantly larger number of components. Specifically, for case T1, which features a shorter spatial correlation length in the *y* direction, achieving the same level of cumulative variance as in case T2 necessitates a greater number of principal components, aligning with expectations.



(a) Case T1

(b) Case T2

Figure 5.8: Scree plots for the cases under study.

Hyperparameter Optimisation

For hyperparameter optimisation, a grid search in combination with 5-fold cross-validation is employed using the full training set (2400 realisation) for each investigated case.

The hyperparameter grid that is tested for the PCA-SVR is detailed in Table 5.4. The metric that decided the best hyperparameter combination is the mean squared error (MSE).

Table 5.4: Hyperparameter range for optimization of the Support Vector Regression.

Parameters	Kernel	Min	Max	Type	Steps	Scale
C	linear, RBF, polynomial	0.001	10	Real	10	logarithmic
$\mathcal{E}$	linear, RBF, polynomial	0.001	1	Real	4	logarithmic
degree	polynomial	1	5	Integer	1	linear
PCA components	linear, RBF, polynomial	100	300	Integer	10	linear

The hyperparameter combination that showed the lowest MSE appeared to be the same for both cases considered: the radial basis function (rbf) kernel, $C = 0.3$, $\mathcal{E} = 0$, and *PCA components* = *130*. With 130 principal components used, between 85% (for Case T1) and 95% (for Case T2) of the total variance in the data is captured (Figure 5.8).

Note that during semi-surrogate modeling, the number of PCA components must be equal or smaller than the number of training simulations, for the PCA algorithm to calculate a full rank covariance matrix of the input features. Consequently, the number of principal components is set to match the number of available training instances but does not exceed 130.

5.1.4.3 PCA-RF

Input

The input for the PCA-RF model is the same as the 1D arrays containing the random field cell values, as for the PCA-SVR.

Hyperparameter optimisation

For hyperparameter optimisation, a random search is deployed in combination with 5-fold cross-validation on the full training set for both investigated cases. Using this method, 300 random combinations of hyperparameters in the parameter space are explored. The ranges of the random-search hyperparameters are detailed in Table 5.5. For the description of each hyperparameter, the reader is referred to Section A.2. The metric that decided the best hyperparameter combination is the mean squared error (MSE).

Table 5.5: Hyperparameter range for optimization of the PCA-RF.

Parameters	Min	Max	Type
Min_samples_split	2	21	Integer
Min_samples_leaf	1	21	Integer
Max_depth	20	50	Integer
Max_features	-	-	None, sqrt, log2
PCA components	3	200	Integer

For both cases considered, the PCA-RF demonstrated low MSE performance with the following hyperparameters: *PCA components* = 6, *Min_samples_split* = 2, *Min_samples_leaf* = 4, and *Max_features* = *None*. Setting *Max_features* to *None* means that all six PCA components are utilized in each decision tree within the forest. This configuration essentially drops the 'feature randomness' aspect of random forests, as every tree has access to the full set of features (PCA components) rather than a random subset for making splits. The number of trees in the forest that showed consistent performance was found to be 200.

With six principal components used, between 30% (for Case T1) and 57% (for Case T2) of total variance in the data is captured (Figure 5.8).

5.2. Results and Discussion

5.2.1. Case T1

For reference, the input statistics of Case T1 are repeated in Table 5.6.

Table 5.6: The point statistics and scales of fluctuation for Case T1.

Case	Undrained shear strength			Scale of fluctuation		
	Mean [kPa]	COV	Distribution	θ_z [m]	θ_x [m]	θ_y [m]
T1	21.96	0.25	Normal	1	20	20

Figures 5.9, 5.10, and 5.11 present the ML-predicted FoS values using the PCA-SVR, PCA-RF, and CNN, respectively, against the FEM-predicted FoS values on the test set. The predictions are made by training the ML models on 100, 300, and 600 random FE-simulations. Here, the PCA-SVR and PCA-RF used the data augmentation technique, as they showed significantly improved performance compared to without it, as shown in Figures C.10 and C.11. Note that the predictions displayed are made using a single training set (out of 20 sets). The r^2 and RMSE are denoted in the top left corner of every plot. A general observation is that the predictions made by all models are reasonable, as indicated by the reasonable fit to the diagonal line in each subplot. However, a strong relationship between ML and FEM FoS prediction is only observed when the PCA-SVR and CNN models are used with 600 samples in the training set, as indicated by an r^2 above 0.8.

All three models exhibit improved performance with increased training size, with the PCA-SVR and CNN especially benefiting largely from a larger training set. It should be pointed out that for every training set size shown here, the models underpredict the FoSs in the upper tail of the distribution and overpredict the FoSs in the lower tail of the distribution.

Figure 5.12 further displays the evolution of the performance metrics r^2 and RMSE for the PCA-SVR, PCA-RF, and CNN for case T1 for different training set sizes. In addition, they help assess the consistency of the ML models by displaying the mean, minimum, and maximum values of the performance metric across the 20 random training samples that are used. The mean metric values are shown with dots, and the error bars indicate the ranges, each spanning the lowest and highest values from the 20 training runs. Note that the PCA-SVR and PCA-RF make use of the data augmentation technique, as discussed before. It is observed that the r^2 increases and the RMSE decreases with increasing training sample size. These observations suggest that all three models are effectively learning the relationship between random fields of s_u and associated FoS. In addition, the shrinking range of the error bars suggests that the models become more robust with increasing training sample sizes. It can be seen that the average performance of the PCA-SVR model, when used in combination with the data augmentation technique, is comparable to that of the CNN and surpasses the PCA-RF model. Notably, the PCA-SVR model produces a narrower range of predictions compared to those of the CNN, indicating a more robust model for varying training sets. This comparison of ML model performance aligns with findings from ML semi-surrogate modeling in 2D.

Figure 5.13 displays the predictions of p_f, calculated based on the ML-predicted FoS datasets, for different training sample sizes. The right axis shows the relative percentage error of the ML p_f predictions to the p_f obtained by the RFEM, which is itself depicted by the black dotted line. Again, the dots in the plot represent the mean computed values, and the error bars indicate the range of the computed p_f.

In line with observations from Figure 5.12, the predictions generally improve with increasing training set size. The relative error in p_f remains high (ranging from 40 to 100%) for all training set sizes. Such high errors make the ML semi-surrogate models unsuitable for reliability analysis.

Among the models analysed, the CNN exhibits the smallest relative error. Furthermore, the CNN appears to be the most effective in learning from more data, as indicated by the steepest gradient in p_f predictions with increasing training set size. This gradient remains prevalent even after increasing the training set size from 700 to 800 realisations, suggesting that the performance of the CNN could be enhanced significantly when more realisations are added to the training set.

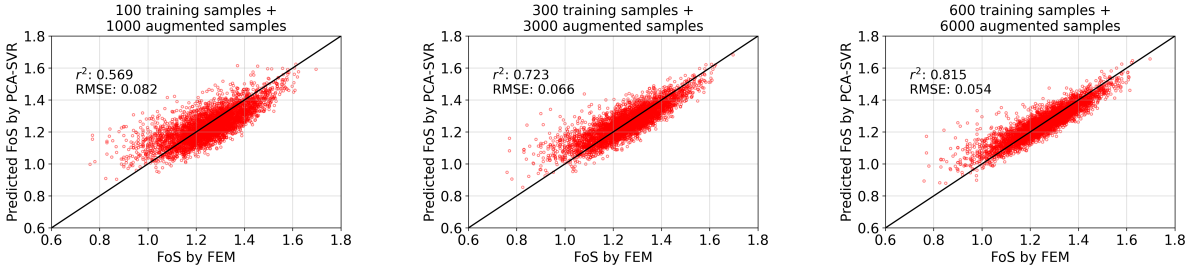


Figure 5.9: FEM vs. PCA-SVR predictions on the FoS for three different training set sizes.

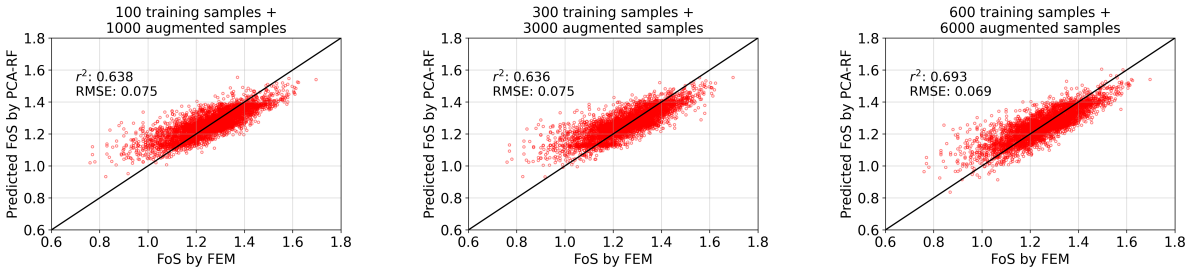


Figure 5.10: FEM vs. PCA-RF predictions on the FoS for three different training set sizes.

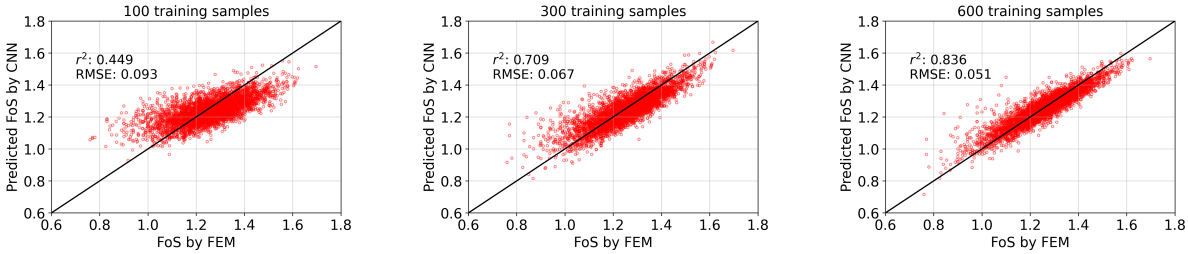
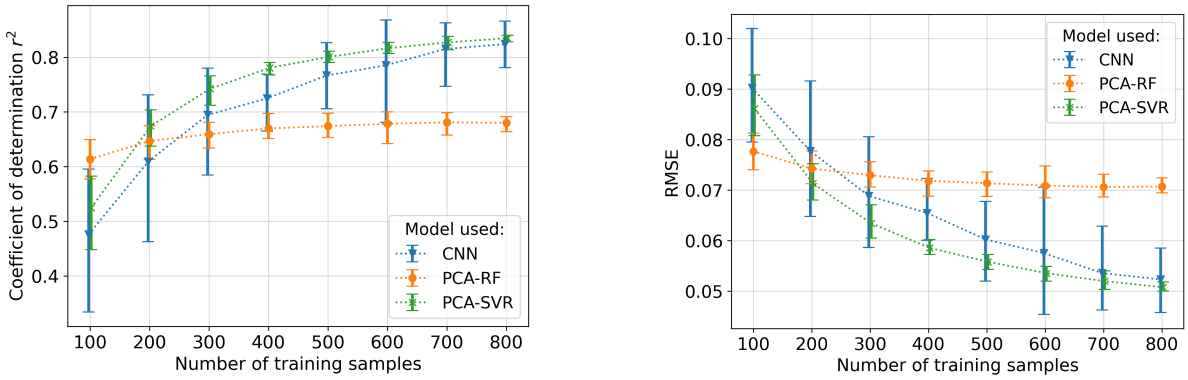


Figure 5.11: FEM vs. CNN predictions on the FoS for three different training set sizes.



(a) r^2 **(b)** RMSE

Figure 5.12: Performance metrics r^2 (a) and RMSE (b) obtained by the three ML models on the test set of case T2 using different training set sizes. The dots indicate the mean metric value across 20 independent training and prediction cycles. The error bars show the range, indicating the minimum and maximum values across the 20 cycles. Note that the PCA-SVR and PCA-RF make use of the data augmentation technique.

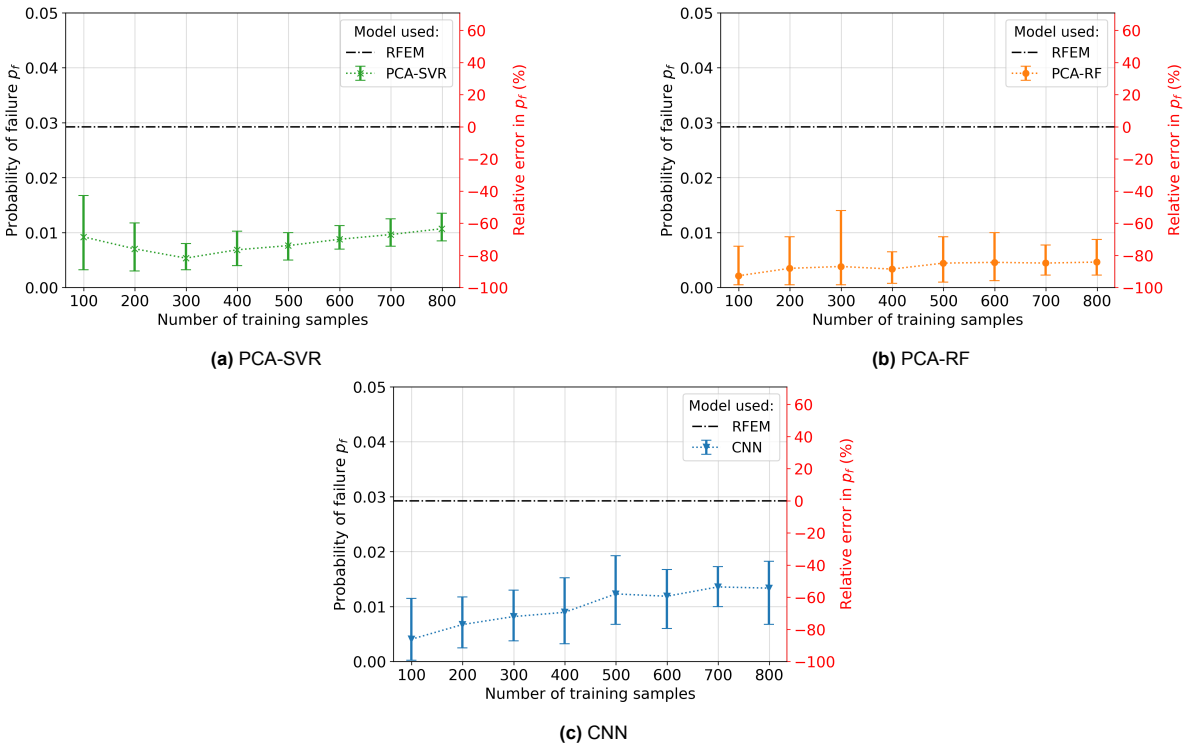


Figure 5.13: The predicted probability of failure p_f on the test set of case T1 using different training set sizes by the PCA-SVR (a), PCA-RF (b), and CNN (c). The dots indicate the mean metric value across 20 independent training and prediction cycles. The error bars show the range, indicating the minimum and maximum values across the 20 cycles. The black dotted line indicates the p_f obtained from a full MCS, comprising 4000 FE simulations. Note that the PCA-SVR and PCA-RF make use of the data augmentation technique.

5.2.2. Case T2

For reference, the input statistics of case T2 are repeated in Table 5.7.

Table 5.7: The point statistics and scales of fluctuation for case T2.

Case	Undrained shear strength			Scale of fluctuation		
	Mean [kPa]	COV	Distribution	θ_z [m]	θ_x [m]	θ_y [m]
T2	22.04	0.25	Normal	1	20	100

Figures 5.14, 5.15, and 5.11 present the ML-predicted FoS values using the PCA-SVR, PCA-RF, and CNN, respectively, against the FEM-predicted FoS values on the test set. The predictions are made by training the ML models on 100, 200, and 400 random FE-simulations. Notably, the PCA-SVR and PCA-RF used the data augmentation technique, as it showed significantly improved performance to as without it, as shown in Figures C.12 and C.13.

Similar findings to those observed for case T1 are noted again.

Figure 5.17 further displays the evolution of the performance metrics r^2 and RMSE for the PCA-SVR, PCA-RF, and CNN for case T2 for different training set sizes. Similar to case T1, PCA-SVR, when combined with data augmentation, outperforms the other models based on these metrics. However, compared to case T1, an improvement in performance is noted, as evidenced by a higher r^2 and a lower RMSE. This highlights the positive correlation between the performance of ML semi-surrogate models and the scale of fluctuation, a trend also observed in 2D analysis.

Figure 5.18 displays the predicted p_f for different training sample sizes. Similar observations as for the p_f predictions of case T1 are noted. A notable difference is the decrease in the relative error of p_f. This improvement suggests that the models are better at predicting the p_f for slope cases with larger spatial correlation lengths, a finding that aligns with results from 2D analysis. However, the relative

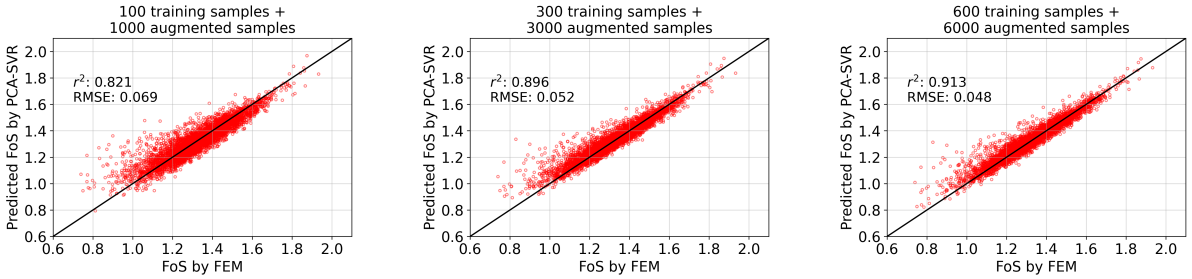


Figure 5.14: FEM vs. PCA-SVR predictions on the FoS for three different training set sizes.

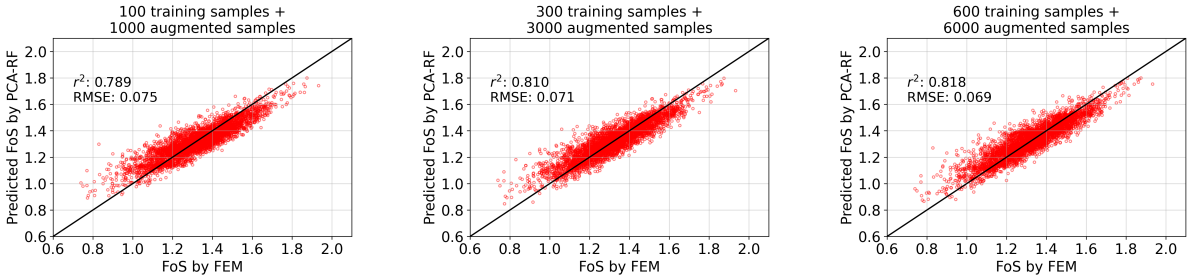


Figure 5.15: FEM vs. PCA-RF predictions on the FoS for three different training set sizes.

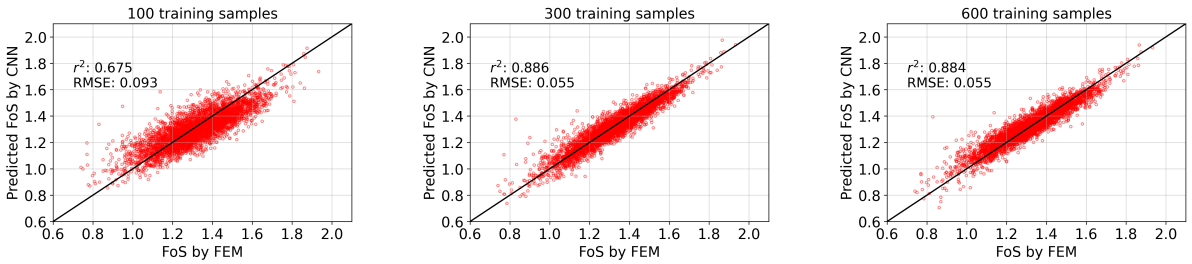
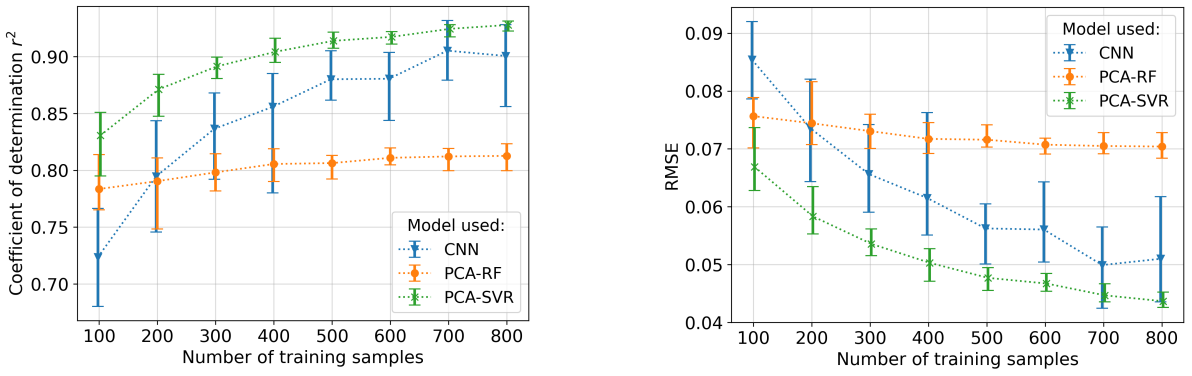


Figure 5.16: FEM vs. CNN predictions on the FoS for three different training set sizes.



(a) r^2 **(b)** RMSE

Figure 5.17: Performance metrics r^2 (a) and RMSE (b) obtained by the three ML models on the test set of case T2 using different training set sizes. The dots indicate the mean metric value across 20 independent training and prediction cycles. The error bars show the range, indicating the minimum and maximum values across the 20 cycles. Note that the PCA-SVR and PCA-RF make use of the data augmentation technique.

error in p_f, even if the large number of 800 realisations are used for training, is considered too large for effective reliability analysis.

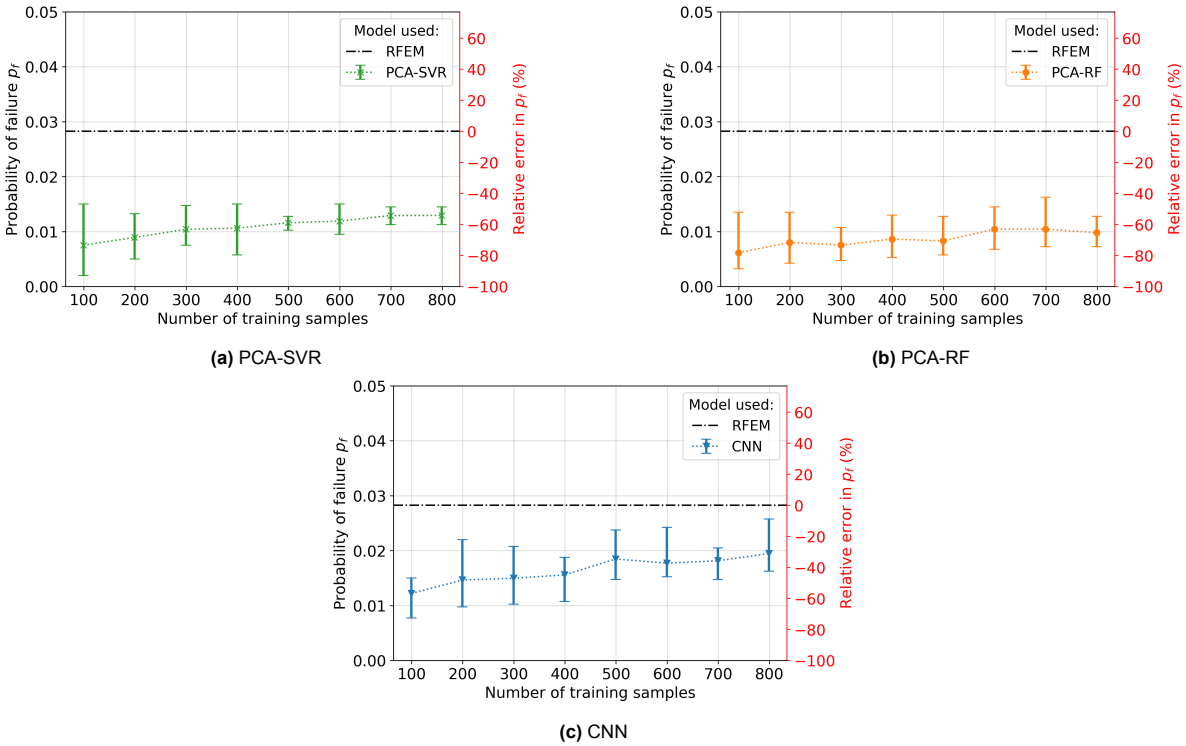


Figure 5.18: The predicted probability of failure p_f on the test set of case T2 using different training sample sizes by the PCA-SVR (a), PCA-RF (b), and CNN (c). The dots indicate the mean metric value across 20 independent training and prediction cycles. The error bars show the range, indicating the minimum and maximum values across the 20 cycles. The black dotted line indicates the p_f obtained from a full MCS, comprising 4000 FE simulations. Note that the PCA-SVR and PCA-RF make use of the data augmentation technique.

5.2.3. Full-surrogate model

Similar to the full-surrogate model in 2D analysis, a single full-surrogate model is here developed, such that no additional simulations are needed when applying this full-surrogate model to different slope cases.

Two sets of full-surrogate models are developed, each using a distinct training set:

1. **Case-Based Training Set**: This training set consists of realisations from cases T1 and T2, with each case contributing an equal number of realisations.
2. **Distributed Sampling Training Set**: This training set is made up of realisations, each generated with input parameters drawn from a specified distribution.

The performance of the two sets of surrogate models, trained on their respective datasets, is evaluated and compared to determine the effectiveness of the proposed strategy for developing a full-surrogate model, as detailed in the section on 2D full-surrogate modeling (Section 4.2.4).

To allow for a correct performance comparison of the two sets of surrogate models, only the scale of fluctuation in *y* direction (θ_y) is sampled from a uniform distribution in the *Distributed Sampling Training Set*. This approach is taken because θ_y is the sole changing input parameter in the realisations within the *Case-Based Training Set*. Additionally, to ensure a correct comparison, a total of 4000 realisations are included in both training sets. The choice for this number is to strike a balance between the computational costs and marginal ML performance improvements with a larger training set size. An overview of the input statistics of the two considered training sets is depicted in Table 5.8.

Table 5.8: The point statistics and scales of fluctuation of the training sets used to develop the full-surrogate models.

Training set reference	Case reference	Number of simulations	Undrained shear strength			Scale of fluctuation		
			Mean [kPa]	COV	Distribution	θ_z [m]	θ_x [m]	θ_y [m]
Case-Based Training Set	T1	2000	21.96	0.25	Normal	1	20	20
	T2	2000	22.04	0.25	Normal	1	20	100
Distributed Sampling Training Set	-	4000	22.00	0.25	Normal	1	20	∼U(20, 100)

The trained ML models are then used to make predictions on 5 slope cases: the two investigated cases on which the Case-Based Training approach was trained (cases T1 and T2), and three new slope cases (cases V1, V2, and V3). The new slope cases use case T1 as a baseline, but each alters a single aspect of this case. For case V2, θ_y is chosen to be the theoretical mean value of θ_y of the realisations in the training sets. For cases V1 and V3, the value of θ_y is set to be 10 and 120 meters respectively, outside the range of θ_v values used for the realisations in the training set.

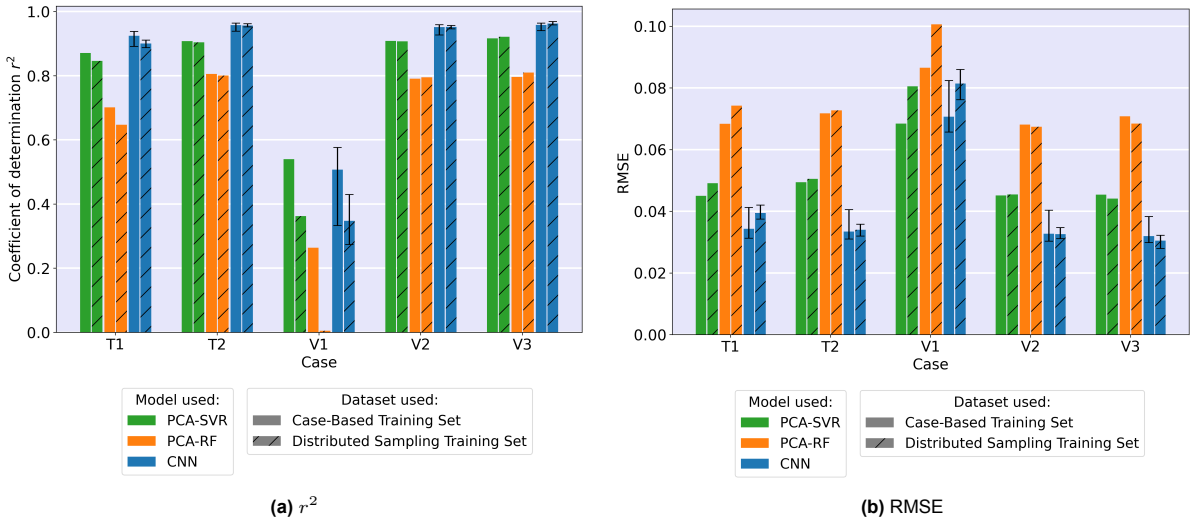
The mean undrained shear strength for cases V1, V2, and V3 is adjusted until a p_f between 2.8% and 3.2% was obtained. For each case considered, 4000 FE simulations are performed and used for testing, which ensures a reliable p_f (COV[p_f]<0.1). The COV of the undrained shear strength is set constant to a value of 0.25. An overview of the point statistics and scales of fluctuation for all test cases are detailed in Table 5.9.

Table 5.9: The point statistics and scales of fluctuation of the test cases considered for the full-surrogate models.

Case	Undrained shear strength			Scale of fluctuation		
	Mean [kPa]	COV	Distribution	θ_z [m]	θ_x [m]	θ_y [m]
T1	21.96	0.25	Normal	1	20	20
T2	22.04	0.25	Normal	1	20	100
V1	21.19	0.25	Normal	1	20	10
V2	22.00	0.25	Normal	1	20	60
V3	22.00	0.25	Normal	1	20	120

Figure 5.19 shows the r^2 and RMSE of the predictions by the two sets of three surrogate models on the testing sets of the considered cases. For the CNN model, the height of the bars in the figure illustrates the mean values of the metrics, averaged over 10 training and prediction cycles. Additionally, the error bars for the CNN indicate the minimum and maximum values observed across the 10 cycles. This repetition was done to account for the model-induced variance of the trained CNN, stemming from

factors such as random initialisation of the trainable parameters and numerical errors. In contrast, such repetition was not done for the PCA-RF model, as the large number of trees in the forest ensures minimal variance in performance. Similarly, repetitions were not done for the SVR due to the deterministic nature of the algorithm.



(a) r^2 **(b)** RMSE

Figure 5.19: Performance metrics r^2 (a) and RMSE (b) obtained by two sets of full-surrogate models on the test sets of 4 different cases.

A few things can be pointed out from the figure. First, both sets of PCA-SVR and CNN surrogate models demonstrate a strong ability to accurately predict the FoS for slope cases with a θ_y within the range of values they have been trained on, as evidenced by high r^2 (>0.83) and low RMSE (<0.05) values for cases T1, T2, and V2. The CNN outperforms the other two models overall on both metrics. Additionally, both sets of ML models show high performance when applied to case V3, even though its θ_y, specifically being larger than 100 meters, falls outside the range used for training. This supports previous findings that ML (semi-)surrogate models perform better with larger spatial correlation lengths. Upon examining the slope realisations for this case, it was found that the failure mode of many realisations was similar to what is called failure mode 3, as described by Hicks and Spencer [24]. This failure mode is characterized by the failure extending along the length of the slope, which mirrors slope failure in 2D. Consequently, slope analysis in 2D would be sufficient for such a case.

Secondly, both sets of models show the weakest performance when applied to case V1, characterised by the smallest longitudinal correlation length (θ_y) of 10 meters among all the tested cases. While the RMSE remains within an acceptable range, the r^2 is significantly low. This observation can be attributed to the models almost consistently overpredicting FoS for this case, which is illustrated in Figure 5.20a. The reason behind this overestimation lies in the smaller spatial correlation lengths, which cause more frequent alternations in the random field. This condition allows the failure surface to pass through fewer strong zones, consistently resulting in a lower FoS. These realisations were not included in the training dataset, leading to the observed overestimation. Note that although case V3 also features a θ_y not represented in the training set, the ML surrogate performance on this case is high. This is due to the relatively minor change in θ_y compared to an edge case in the training set and the fact that model performance generally improves with larger spatial correlation lengths.

Lastly, when comparing the performance of ML models trained on the *Distributed Sampling Training Set* with those trained on the *Case-Based Training Set*, there isn't a consistent trend of under- or over-performance between them. Consequently, the full-surrogate model development strategy deemed promising based on 2D analysis might not be as effective as anticipated.

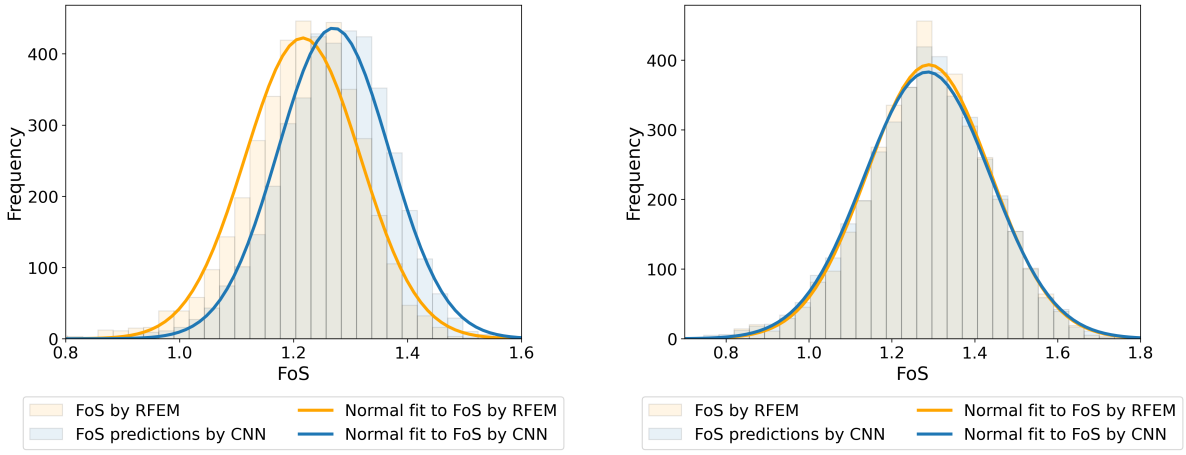
Figure 5.21 presents the relative error in ML-predicted p_f, calculated based on the ML-predicted FoS datasets, and the p_f derived from 4000 FE simulations for all cases under study.

The figure reveals some interesting things. First, both sets of CNN models demonstrate superior performance over both sets of PCA-RF and PCA-SVR models applied to all cases.

Secondly, the models make better predictions for cases with larger spatial correlation lengths. Notably, the relative error in p_f prediction by the CNN is as low as 6.6% for the slope case where the

θ_y is 60 meters. To further demonstrate the accuracy of the predictions by the CNN model applied to this slope case, Figure 5.20b displays the CNN-predicted and RFEM-computed Factors of Safety, presenting both sets of results in histograms with a fitted normal distribution curve.

Lastly, the set of ML models trained on the *Case-Based Training Set* consistently outperforms those trained on the *Distributed Sampling Training Set* across all considered cases. This implies that the full-surrogate model development strategy deemed promising based on 2D RFEM analysis is unsuitable when the primary aim is to accurately predict the p_f. A user would be better off including realisations in the training set with input statistics of edge cases, characterised by minimum and maximum spatial correlation lengths.



(a) Case V1 **(b)** Case V2

Figure 5.20: Histograms of the CNN-predicted and RFEM realised Factors of Safety (FoS) for cases V1 and V2.

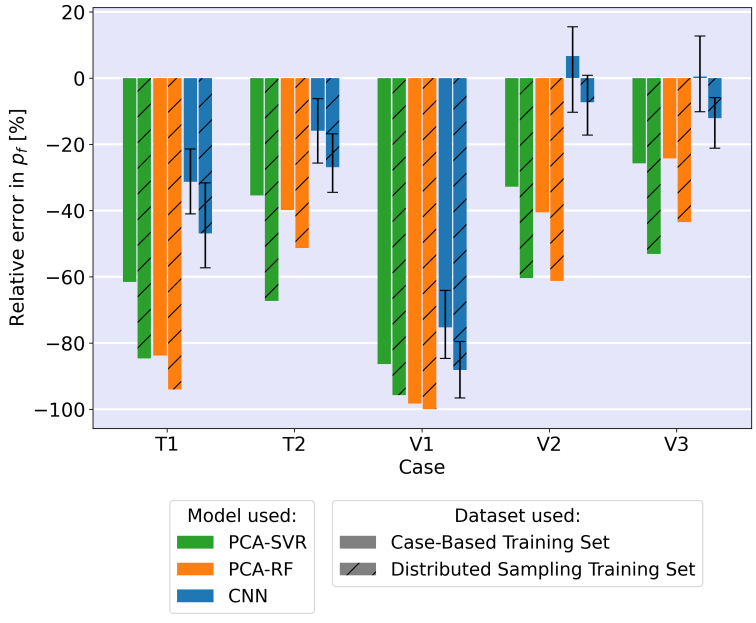


Figure 5.21: Relative error of the obtained p_f by the surrogate models to the p_f obtained by RFEM analysis for 5 different cases.

5.3. Main Findings

5.3.1. Semi-surrogate modeling

The main findings for semi-surrogate modeling for 3D RFEM slope reliability analysis are summarised as follows:

- The CNN and PCA-SVR, when combined with the data augmentation technique, both demonstrated to predict the FoS with reasonable accuracy (RMSE<0.06) using the random field of the undrained shear strength. Only 15% of the total number of realisations within an RFEM analysis is needed to achieve such results. The PCA-SVR, when combined with the data augmentation technique, exhibited superior performance (highest r^2 and lowest RMSE) compared to both the CNN and PCA-RF models.
- The performance of the considered ML models typically increases for slope cases with larger spatial correlation lengths of the undrained shear strength.
- All three considered ML models demonstrated difficulty in predicting the probability of failure p_f accurately, as evidenced by large relative errors (>50%). This indicates a need for further exploration of ML semi-surrogate modeling for 3D reliability analysis.
- The data augmentation technique introduced by Jiang et al. [30] significantly enhances the performance of both PCA-SVR and PCA-RF models. Employing this technique led to consistent improvements in r^2, RMSE, and p_f predictions for both models.

5.3.2. Full-surrogate modeling

The main findings for full-surrogate modeling for 3D RFEM slope reliability analysis are summarised as follows:

- A full-surrogate ML model is developed that can make good FoS predictions (max. RMSE of 0.04) using the random fields of the undrained shear strength. It is demonstrated that the spatial correlation length along the length of the slope (θ_y) in a new case should be within or exceed the range of values used for the realisations in the training set to have accurate results. A large FoS prediction bias is observed when θ_y is taken smaller due to the lack of extrapolation ability.
- The performance of the considered ML surrogate models typically decreases when applied to slope cases with smaller spatial correlation lengths of the undrained shear strength random field.
- This research found the CNN surrogate model to predict the p_f most accurately. This model predicted the p_f with a 30% to 0.47% relative error, given that the θ_y of the new slope case was within or exceeded the range of θ_y used for the realisations in the training set.
- Improved p_f predictions result when the training set of the full-surrogate model includes realisations featuring edge cases of θ_y, rather than sampling θ_y for each realisation from a distribution.

Figure 5.22 presents an overview of the best performing ML (semi-)surrogate models for the 3D RFEM found in this research, identified for various goals.

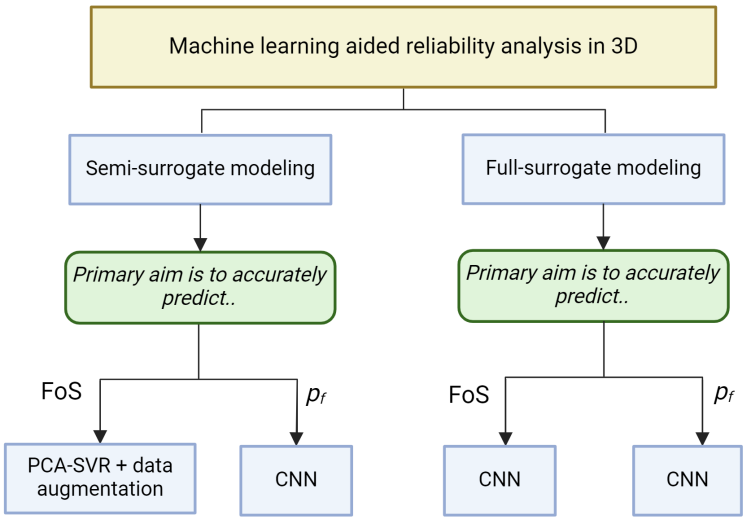


Figure 5.22: Overview of the most effective ML (semi-)surrogate models for the 3D RFEM found in this research.

6

Conclusions

This thesis investigated the performance of machine learning (ML) models as surrogate models for RFEM slope reliability analysis, aiming to reduce the extensive computational time associated with direct Monte Carlo (MC) simulation. Three distinct (combined) ML models– PCA-SVR, PCA-RF, and CNN– were used for predicting the Factor of Safety (FoS) of a slope with spatial variability. The random field variable was the undrained shear strength. Semi-surrogate and full-surrogate modeling techniques were investigated. For semi-surrogate ML model development, a small portion of the total number of realisations in direct Monte Carlo simulation was used for training. The models were then used for prediction on random fields that had the same input statistics as the random fields in the training set. For full-surrogate ML model development, a large dataset of realisations covering a range of spatial correlation lengths was used for training. The trained full-surrogate model was then ready to make predictions for another different slope case without additional numerical simulations and training. This study conducted both 2D and 3D analyses, applying these ML models to a variety of slope cases, characterised by varying spatial variability. Additionally, it utilised a data augmentation technique with the aim of improving ML performance. The conclusions of this thesis are split up into two main sections: 2D analysis and 3D analysis. Within each of these sections, the conclusions are further split for semi-surrogate modeling and full-surrogate modeling.

6.1. 2D Slope Reliability Analysis

6.1.1. Semi-surrogate modeling

This thesis further confirms previous work (e.g., [4, 19]) that ML models can be effectively used as semi-surrogate models for slope reliability analysis in 2D. Predictions for thousands of realisations are made with reasonable accuracy (RMSE between 0.05 and 0.10) at a fraction of the computational cost of an RFEM analysis. Notably, by training with just 10% of the total number of RFEM realisations, the ML models achieved high accuracy levels, comparable to those reported in prior studies [19, 50, 4]. However, the thesis also reveals that the performance of these semi-surrogate models varies with the scale of fluctuation, with reduced performance observed at smaller scales.

Among the ML models investigated, the PCA-SVR, when combined with the data augmentation technique, demonstrated the best performance in predicting the FoS. It achieved the highest performance in terms of r^2 and RMSE and demonstrated to be most robust to a varying training set. For p_f prediction, the CNN showed the best performance. Using this model, the mean relative error in p_f prediction is within 10% when the training set includes up to 300 realisations for slopes with large spatial correlation lengths of the strength property, and up to 500 realisations for slopes with small spatial correlation lengths. As a result, the total computational time required for reliability analysis of the considered slope cases can be reduced from 67 hours for a full RFEM analysis to between 4 and 8 hours using the CNN as a semi-surrogate model.

The variation in optimal models for different objectives emphasizes the importance of careful ML model selection, reinforcing the idea that there is *no such thing as free lunch*.

6.1.2. Full-surrogate modeling

The potential of developing a ready-to-use surrogate model for 2D RFEM slope reliability analysis, as demonstrated by Xu et al. [51] and He et al. [18], is further underscored in this work. The developed full-surrogate models make FoS predictions that are highly consistent with the results from RFEM simulations. Similar to semi-surrogate models, this thesis finds that the performance of full-surrogate models is positively correlated to the scales of fluctuation of the strength property, based on the range of scales of fluctuation considered in this project.

The CNN emerged as the most robust model, consistently performing well across eight test cases. Notably, the CNN maintained high performance for cases with spatial variability levels different from those in the training set. Predictions on thousands of random fields are made within seconds, compared to multiple days of computational time for MC simulation.

6.2. 3D Slope Reliability Analysis

6.2.1. Semi-surrogate modeling

This thesis explored the potential of semi-surrogate modeling for RFEM slope reliability analysis in 3D. It showed that the FoS can be predicted most accurately by the PCA-SVR, when combined with the data augmentation technique. Reasonable FoS prediction accuracy is obtained when 600 realisations are used for training. However, the study revealed that semi-surrogate modeling is less effective in 3D than in 2D, reflecting the greater complexity of 3D system responses. Additionally, all three tested models struggled to accurately predict the p_f, with relative errors exceeding 50%. This significant inaccuracy underscores the current limitations of semi-surrogate modeling in 3D and points to the necessity for further research in this area.

6.2.2. Full-surrogate modeling

This thesis demonstrated the potential of developing a full-surrogate model for the RFEM used in 3D slope reliability analysis. Five testing cases, characterised by a varying spatial correlation length along the length of the slope (θ_y), provided the basis for the investigation. Two sets of full-surrogate models were created, using distinct training sets. The results indicate that the full-surrogate models can make FoS predictions that are highly consistent with values from RFEM simulations, given that θ_y of the testing case is within or exceeds the range of values used for the realisations in the training set.

Among the models tested, the CNN demonstrated superior performance in terms of both FoS and p_f prediction across all cases. Furthermore, the analysis concluded that better p_f predictions are obtained when the training set of the surrogate model includes realisations featuring edge cases of θ_y rather than sampling θ_y from a distribution for each realisation. The full-surrogate model performs a stochastic analysis of 4000 realisations within seconds, compared to 83 days of computational time required for RFEM reliability analysis.

6.3. Recommendations

Building on this thesis, the following recommendations for future research are given:

- This research investigated the effect of varying spatial variability levels on ML (semi-)surrogate modeling performance by using various slope cases. Future research should expand the range of spatial variability levels examined in this study. Specifically, it is recommended to cover a wider range of spatial correlation lengths, extending from very small to very large. This way, the ML (semi-)surrogate performance is also assessed for 2D cases where the horizontal scale of fluctuation is smaller than the slope domain itself. Additionally, it is recommended to investigate how changes in point statistics affect the performance of ML (semi-)surrogate models.
- In this research, the ML (semi-)surrogate models demonstrated relative weak performance in the tails of the FoS distribution, likely due to an insufficient number of realisations in these regions for training. Consequently, the ML models relied on their extrapolation capabilities, which are generally not very effective. Since engineers are primarily concerned with the probability of failure, which typically pertains to the lower tail, it is advisable to enhance the training set with more

realisations from this region by using subset simulation (e.g. [44]). This approach could lead to better ML (semi-)surrogate performance in the lower tail and consequently better p_f predictions.

- This investigation is limited to three ML models, of which two (PCA-SVR and CNN) were based on their demonstrated effectiveness in previous research for 2D slope reliability analysis [4, 50, 19]. The PCA-RF model was added to the investigation to include a new ML model in the comparison. For (semi-)surrogate modeling of the RFEM analysis in 3D, no previous research existed to compare to, making this selection of models novel. However, many ML models remain unexplored in both 2D and 3D contexts, suggesting the need for further research. When more complex geometries are dealt with in the future, the Graph Neural Network (GNN) shows promise. Like the CNN, the GNN is complex and highly adaptable, but unlike the CNN, it does not require a structured mesh.

  Additionally, potential ways to improve the current investigated model types are to investigate different CNN architectures and to replace PCA with different feature compressing tools in PCA-SVR and PCA-RF. While PCA is effective for compressing the high-dimensional random field in this research, exploring other tools is recommended when more complex problems involving random fields for multiple properties are dealt with. In such cases, more advanced methods, like autoencoders, may be required.
- The current investigation was limited to slopes with basic geometry without a phreatic surface, soil layering, or the use of advanced soil models. However, from a practical perspective, these streamlined conditions are rarely applicable. Therefore, it is recommended to increase the complexity of the RFEM model by changing these conditions one by one and investigating the performance of ML (semi-)surrogate models for such modifications. This way, a better understanding of how ML models perform as (semi-)surrogate models applied to such cases can be obtained.
- The PCA, used in combination with SVR and RF, reduced the high-dimensional random field to a small set of principal components. This reduction improved the computational efficiency of the combined ML models and reduced data noise. Future research should investigate the extent to which PCA captures the physical features of the system that are key for accurate prediction. When such features are identified, a better understanding of the system response is obtained, potentially improving prediction performance by selectively incorporating these features in the training set.
- The full-surrogate models developed for both 2D and 3D RFEM analysis were tested on slope cases with the same geometry as those in their training set. Future research should aim to develop and investigate the performance of a full-surrogate model capable of accurately handling slopes of various dimensions and angles, similar to the approach by Xu et al. [51]. The CNN model is particularly promising due to its flexibility: it only requires that the input matrix dimensions remain consistent between training and testing phases. Specifically, it does not require the number of random field cells within the slope to be identical, as areas not displaying any part of the slope are simply filled with zeros.
- In this research, a challenge was encountered with the occurrence of very locally failed realisations in the RFEM analysis. These failures had minimal consequences and were thus considered insignificant for reliability analysis. The question arises if these realisations just appeared to have such small consequences or if the FEM defined failure at a too-low Strength Reduction Factor.

  To address this issue, a method was adopted that initially excludes these minor failures based on the automatically approximated sliding volume and, subsequently, eliminates instances that fall below a certain FoS threshold. However, more sophisticated approaches for handling such localised failures are recommended for future research. This includes refining the sliding approximation method and modifying the failure criterion used in the FEM program. For the former, a potential modification is to compute a spatial gradient for displacements and set a minimum of this value for the elements to be considered as failure within the elements having a larger out-of-face displacement than the calibrated threshold. For the latter, potential modifications include increasing the number of iterations to better identify non-convergence, reducing the plastic tolerance, or adopting a different or combined criterion for defining failure. One such alternative criterion could be the bulging of a slope [41].

References

[1] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, S CG, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[2] Abdolrasol MG, Hussain SS, Ustun TS, Sarker MR, Hannan MA, Mohamed R, Ali JA, Mekhilef S, Milad A. "Artificial neural networks based optimization techniques: A review". Electronics. 2021; 10(21), 2689.

[3] Aminpour M, Alaie R, Kardani N, Moridpour S, Nazem M. "Highly efficient reliability analysis of anisotropic heterogeneous slopes: machine learning-aided Monte Carlo method". Acta Geotechnica. 2023; 18(6), 3367–3389.

[4] Aminpour M, Alaie R, Khosravi S, Kardani N, Moridpour S, Nazem M. "Slope stability machine learning predictions on spatially variable random fields with and without factor of safety calculations". Computers and Geotechnics. 2023; 153, 105094.

[5] Au SK, Beck JL. "Estimation of small failure probabilities in high dimensions by subset simulation". Probabilistic engineering mechanics. 2001; 16(4), 263–277.

[6] Au SK, Ching J, Beck J. "Application of subset simulation methods to reliability benchmark problems". Structural safety. 2007; 29(3), 183–193.

[7] Boser BE, Guyon IM, Vapnik VN. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, 144–152.

[8] Cherubini C. "Reliability evaluation of shallow foundation bearing capacity on c'φ'soils". Canadian Geotechnical Journal. 2000; 37(1), 264–269.

[9] De Gast T, Vardon PJ, Hicks MA. "Assessment of soil spatial variability for linear infrastructure using cone penetration tests". Géotechnique. 2021; 71(11), 999–1013.

[10] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 1)*. `https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1`. 2022.

[11] Di Matteo L, Valigi D, Ricco R. "Laboratory shear strength parameters of cohesive soils: variability and potential effects on slope stability". Bulletin of Engineering Geology and the Environment. 2013; 72, 101–106.

[12] Fenton G, Griffiths D. *Risk assessment in geotechnical engineering*. John Wiley & Sons New York, 2008.

[13] Fenton GA, Vanmarcke EH. "Simulation of random fields via local average subdivision". Journal of Engineering Mechanics. 1990; 116(8), 1733–1749.

[14] Ghanem RG, Spanos PD. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[15] Griffiths DV, Fenton GA. "Influence of soil strength spatial variability on the stability of an undrained clay slope by finite elements". In: *Slope stability 2000*. 2000, 184–193.

[16] Griffiths DV, Fenton GA. "Probabilistic slope stability analysis by finite elements". Journal of geotechnical and geoenvironmental engineering. 2004; 130(5), 507–518.

[17] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, et al. "Recent advances in convolutional neural networks". Pattern recognition. 2018; 77, 354–377.

[18] He X, Wang F, Li W, Sheng D. "Deep learning for efficient stochastic analysis with spatial variability". Acta Geotechnica. 2022; 17(4), 1031–1051.

[19] He X, Xu H, Sabetamal H, Sheng D. “Machine learning aided stochastic reliability analysis of spatially variable slopes”. Computers and Geotechnics. 2020; 126, 103711.

[20] Hicks MA. “Application of the random finite element method”. ALERT Doctoral School. 2014, 181.

[21] Hicks MA, Li Y. “Influence of length effect on embankment slope reliability in 3D”. International Journal for Numerical and Analytical Methods in Geomechanics. 2018; 42(7), 891–915.

[22] Hicks MA, Nuttall JD, Chen J. “Influence of heterogeneity on 3D slope reliability and failure consequence”. Computers and Geotechnics. 2014; 61, 198–208.

[23] Hicks MA, Samy K. “Reliability-based characteristic values: a stochastic approach to Eurocode 7”. Ground Engineering. 2002; 35(12).

[24] Hicks MA, Spencer WA. “Influence of heterogeneity on the reliability and failure of a long 3D slope”. Computers and Geotechnics. 2010; 37(7-8), 948–955.

[25] Huang F, Xiong H, Chen S, Lv Z, Huang J, Chang Z, Catani F. “Slope stability prediction based on a long short-term memory neural network: Comparisons with convolutional neural networks, support vector machines and random forest models”. International Journal of Coal Science & Technology. 2023; 10(1), 18.

[26] Huang J, Fenton GA, Griffiths DV, Li D, Zhou C. “On the efficient estimation of small failure probability in slopes”. Landslides. 2017; 14, 491–498.

[27] Huang YH. “Slope stability analysis by the limit equilibrium method: Fundamentals and methods”. In: American Society of Civil Engineers. 2014.

[28] Jiang S.-H, Huang J.-S. “Efficient slope reliability analysis at low-probability levels in spatially variable soils”. Computers and Geotechnics. 2016; 75, 18–27.

[29] Jiang S.-H, Li D.-Q, Zhang L.-M, Zhou C.-B. “Slope reliability analysis considering spatially variable shear strength parameters using a non-intrusive stochastic finite element method”. Engineering geology. 2014; 168, 120–128.

[30] Jiang S.-H, Zhu G.-Y, Wang ZZ, Huang Z.-T, Huang J. “Data augmentation for CNN-based probabilistic slope stability analysis in spatially variable soils”. Computers and Geotechnics. 2023; 160, 105501.

[31] Kingma DP, Ba J. “Adam: A method for stochastic optimization”. arXiv preprint. 2014, arXiv:1412.6980.

[32] Kumar S, Choudhary SS, Burman A. “Recent advances in 3D slope stability analysis: A detailed review”. Modeling Earth Systems and Environment. 2023; 9(2), 1445–1462.

[33] Li D.-Q, Xiao T, Cao Z.-J, Zhou C.-B, Zhang L.-M. “Enhancement of random finite element method in reliability analysis and risk assessment of soil slopes using Subset Simulation”. Landslides. 2016; 13, 293–303.

[34] Li D.-Q, Zheng D, Cao Z.-J, Tang X.-S, Phoon KK. “Response surface methods for slope reliability analysis: review and comparison”. Engineering Geology. 2016; 203, 3–14.

[35] Liu L.-L, Cheng Y.-M. “System reliability analysis of soil slopes using an advanced kriging metamodel and quasi–Monte Carlo simulation”. International Journal of Geomechanics. 2018; 18(8), 06018019.

[36] Liu W.-S, Cheung SH, Cao W.-J. “An efficient surrogate-aided importance sampling framework for reliability analysis”. Advances in Engineering Software. 2019; 135, 102687.

[37] Matthies HG, Brenner CE, Bucher CG, Soares CG. “Uncertainties in probabilistic numerical analysis of structures and solids-stochastic finite elements”. Structural safety. 1997; 19(3), 283–336.

[38] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. “Scikit-learn: Machine Learning in Python”. Journal of Machine Learning Research. 2011; 12, 2825–2830.

[39] Phoon K.-K, Kulhawy FH. “Characterization of geotechnical variability”. Canadian geotechnical journal. 1999; 36(4), 612–624.

[40] Reddy JN. *Introduction to the finite element method*. McGraw-Hill Education, 2019.

[41] Snitbhan N, Chen W.-F. “Elastic-plastic large deformation analysis of soil slopes”. Computers & structures. 1978; 9(6), 567–577.

[42] Spencer WA. *Parallel stochastic and finite element modelling of clay slope stability in 3D*. The University of Manchester (United Kingdom), 2007.

[43] Sudret B, Der Kiureghian A. *Stochastic finite element methods and reliability: a state-of-the-art report*. Department of Civil and Environmental Engineering, University of California, 2000.

[44] Van Den Eijnden A, Hicks M. “Efficient subset simulation for evaluating the modes of improbable slope failure”. Computers and Geotechnics. 2017; 88, 267–280.

[45] Vanmarcke EH. “Probabilistic modeling of soil profiles”. Journal of the geotechnical engineering division. 1977; 103(11), 1227–1246.

[46] Vanmarcke EH. *Random fields: Analysis and synthesis*. MIT Press, 1983.

[47] Varkey D, Hicks MA, Vardon PJ. “An improved semi-analytical method for 3D slope reliability assessments”. Computers and Geotechnics. 2019; 111, 181–190.

[48] Wang ZZ, Goh SH. “A maximum entropy method using fractional moments and deep learning for geotechnical reliability analysis”. Acta Geotechnica. 2022; 17(4), 1147–1166.

[49] Wang ZZ, Xiao C, Goh SH, Deng M.-X. “Metamodel-based reliability analysis in spatially variable soils using convolutional neural networks”. Journal of Geotechnical and Geoenvironmental Engineering. 2021; 147(3), 04021003.

[50] Wang Z.-Z, Goh SH. “Novel approach to efficient slope reliability analysis in spatially variable soils”. Engineering Geology. 2021; 281, 105989.

[51] Xu H, He X, Pradhan B, Sheng D. “A pre-trained deep-learning surrogate model for slope stability analysis with spatial variability”. Soils and Foundations. 2023; 63(3), 101321.

[52] Zhang W, Gu X, Hong L, Han L, Wang L. “Comprehensive review of machine learning in geotechnical reliability analysis: Algorithms, applications and further challenges”. Applied Soft Computing. 2023, 110066.

[53] Zhao J, Duan X, Ma L, Zhang J, Huang H. “Importance sampling for system reliability analysis of soil slopes based on shear strength reduction”. Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards. 2021; 15(4), 287–298.

[54] Zhu D, Lee C, Jiang H. “Generalised framework of limit equilibrium methods for slope stability analysis”. Géotechnique. 2003; 53(4), 377–395.

A

Machine Learning Concepts

This chapter provides an overview of two machine learning practices employed in this thesis. Section A.1 discusses the cross-validation technique, while Section A.2 details the hyperparameters used in the Random Forest regressor.

A.1. Cross-validation Technique

Cross-validation is a widely used technique in machine learning that aims to maximize the usage of the available data for both training and validation. Although this study does not have a limit on the amount of available data (extra data can be generated using the FEM program), it is still a useful technique during hyperparameter optimization by reducing the risk of overfitting and ensuring that the model is not overly biased toward a specific subset of the training data.

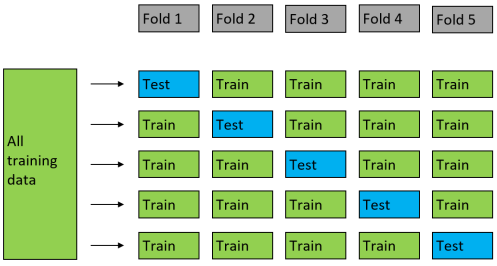


Figure A.1: 5-fold cross-validation technique.

Cross-validation starts by dividing the full training dataset into n distinct folds. In an n-fold cross-validation process, the model is trained and validated n times, each time using a different fold for validation and the remaining $n-1$ folds for training. Figure A.1 illustrates this process for a 5-fold cross-validation. It is important to note that each fold serves as the validation set exactly once throughout the process. After the model has been trained and validated across all folds, the performance metrics from each iteration are averaged to provide a comprehensive assessment of the model's performance.

A.2. Random Forest Hyperparameters

This section outlines the tuned hyperparameters in the Random Forest regressor.

1. **n_estimators**: Number of trees in the forest. Increasing the number of trees can improve the model's performance but also increases computational cost. More trees help in reducing overfitting by averaging predictions.
2. **max_depth**: Maximum depth of each tree. Unset by default, which means that nodes expand until all leaves are pure or contain less than min_samples_split samples. Controlling the depth can prevent the model from becoming overly complex and overfitting.

3. **min_samples_split**: Minimum number of samples required to split an internal node. Higher values can prevent creating over-specific models, thus avoiding overfitting. Too high values might lead to underfitting.
4. **min_samples_leaf**: Minimum number of samples a leaf node must have. Aids in smoothing the model, especially in regression, by avoiding overly complex branches in the tree.
5. **max_features**: Determines the number of features to consider when looking for the best split. Options include:
    - 'sqrt': Takes the square root of the total number of features, adding randomness and increasing tree diversity.
    - 'log2': Uses logarithm to base 2 of the number of features, similar to 'sqrt'.

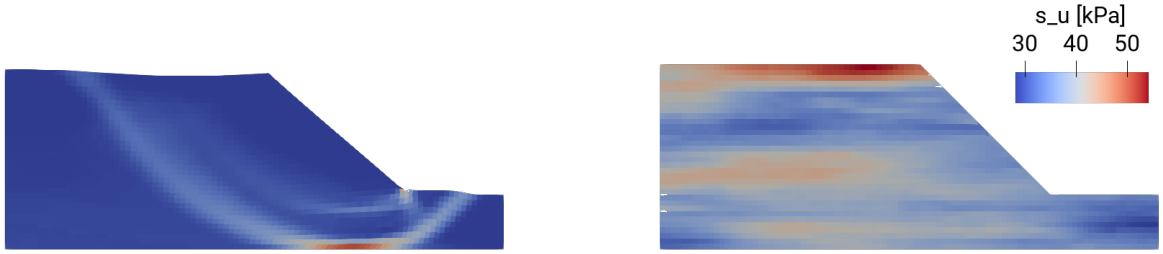
B

RFEM realisations

This chapter illustrates the appearance of the modeled slopes. Section B.1 is dedicated to realisations in 2D, while Section B.2 is dedicated to realisations in 3D.

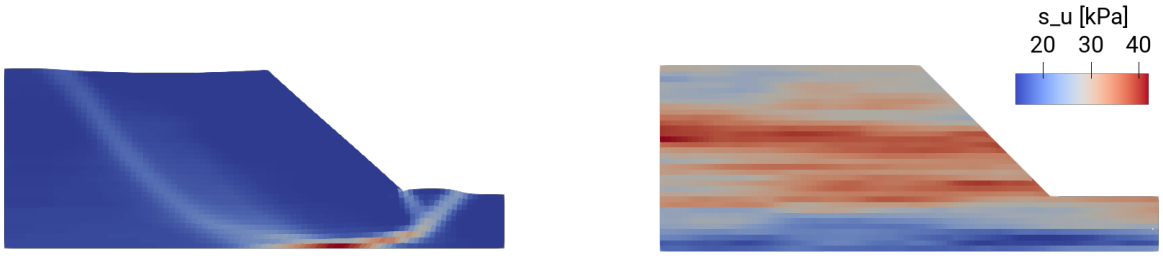
B.1. 2D Random Field Examples

Figures B.1 and B.2 present an example of a generated random field for s_u (a) along with the associated failure mechanisms deduced from the computed shear strains at failure (b) for cases U1 and U2, respectively. Together with Figure 4.2, these figures demonstrate that the frequency of alternating weak and strong zones increases in the order of case name U1, U2, and U3.



(a) Generated random field of s_u **(b)** FoS = 2.17

Figure B.1: Example of a generated random field of the undrained shear strength s_u (a) and the deviatoric shear strains at failure (b) computed using the FEM for the slope case U1.

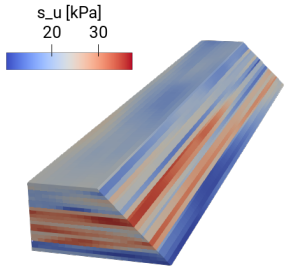


(a) Generated random field of s_u **(b)** FoS = 1.67

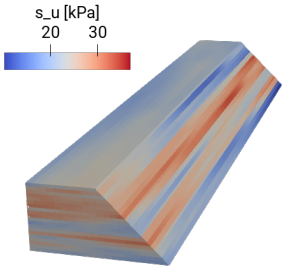
Figure B.2: Example of a generated random field of the undrained shear strength s_u (a) and the deviatoric shear strains at failure (b) computed using the FEM for the slope case U2.

B.2. 3D Random Field Examples

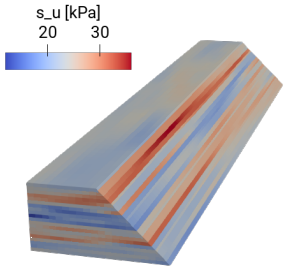
Figure B.3 displays three examples of generated random fields of s_u along with the resulting displacements at failure for case T2. It shows that discrete 3D failure occurs and that the failure surface passes through a weak zone (mode 2 failure) located in the lower portion of the slope.



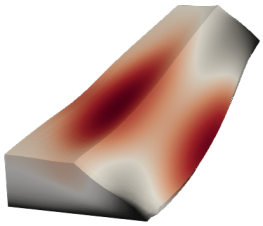
(a) s_u (random field realisation 1)



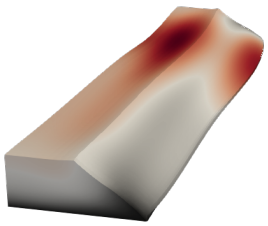
(b) s_u (random field realisation 2)



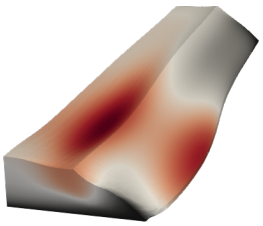
(c) s_u (random field realisation 3)



(d) FoS = 1.28 (random field realisation 1)



(e) FoS = 1.39 (random field realisation 2)



(f) FoS = 1.48 (random field realisation 3)

Figure B.3: Tree examples of random fields of the undrained shear strength s_u (a, b, c) and the displacements (magnification = 500) at failure (d, e, f) computed using the FEM for slope case T2.

C

Data Augmentation Effectiveness

This chapter displays the performance of the selected ML models as semi-surrogate models for the RFEM, particularly when combined with the proposed data augmentation technique. Section C.1 displays the performance of the ML models when applied in 2D context, while section C.2 displays the performance in 3D context.

C.1. 2D Analysis

Figures C.1, C.2, and C.3 illustrate the performance of the PCA-SVR, PCA-RF, and CNN, respectively, for case U1, as a standalone model and when used in combination of the proposed data augmentation technique. Similarly, Figures C.4, C.5, C.6 and C.7, C.8, C.9 show the performance outcomes for cases U2 and U3, respectively, under the same conditions. These figures demonstrate a consistent improvement in the performance of the PCA-SVR model when combined with the data augmentation technique across all evaluated cases.

C.2. 3D Analysis

Figures C.10 and C.11 illustrate the performance of the PCA-SVR and PCA-RF, respectively, for case T1, as a standalone model and when used in combination of the proposed data augmentation technique. Similarly, Figures C.12 and C.13 show the performance outcomes for case T2 under the same conditions. These figures demonstrate a consistent improvement in the performance of the PCA-SVR and PCA-RF models across all evaluated cases.

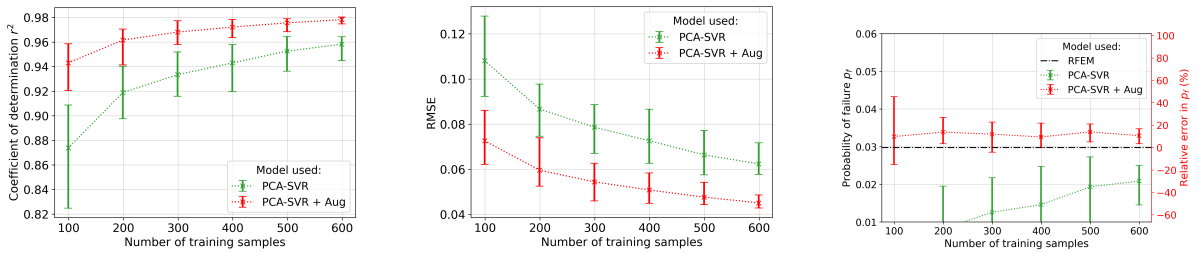


Figure C.1: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U1 obtained by the PCA-SVR using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

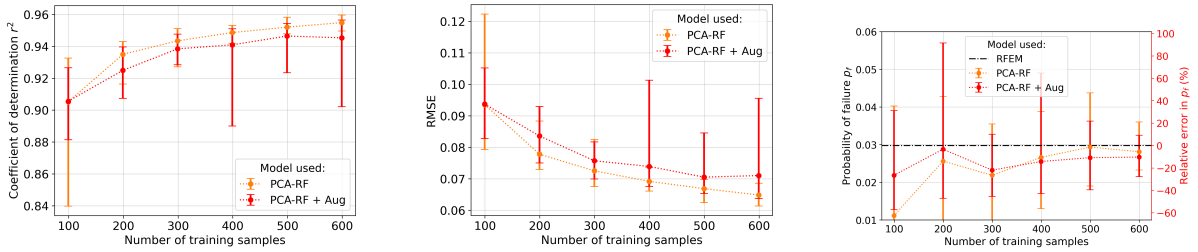


Figure C.2: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U1 obtained by the PCA-RF using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

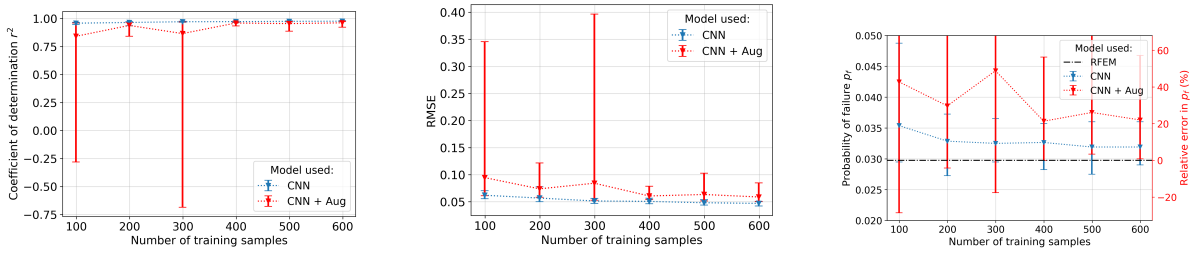


Figure C.3: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U1 obtained by the CNN using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

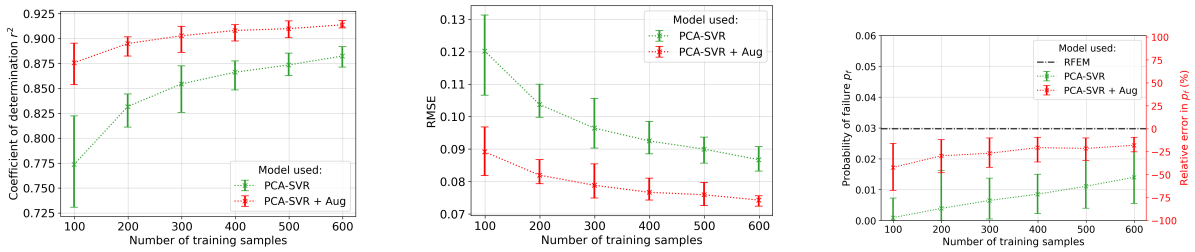


Figure C.4: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U2 obtained by the PCA-SVR using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

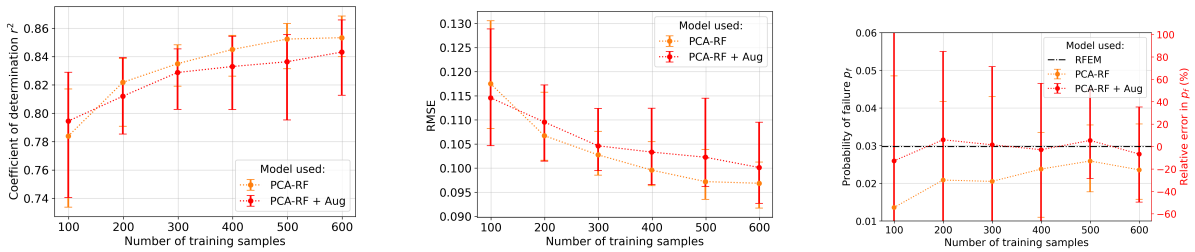


Figure C.5: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U2 obtained by the PCA-RF using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

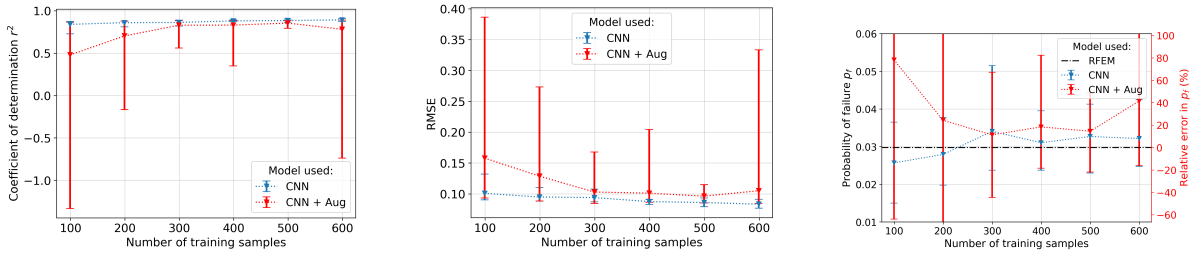


Figure C.6: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U2 obtained by the CNN using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

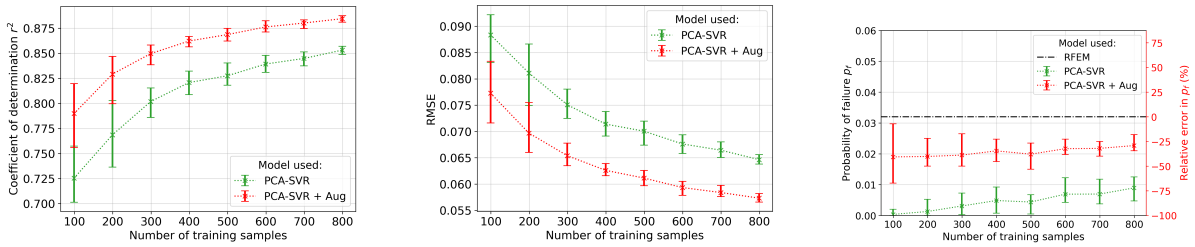


Figure C.7: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U3 obtained by the PCA-SVR using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

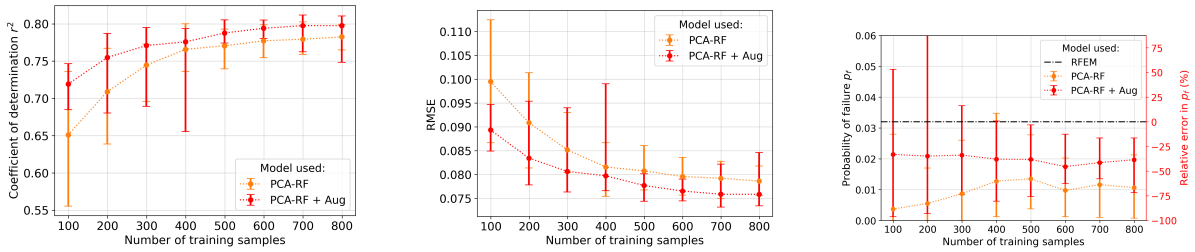


Figure C.8: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U3 obtained by the PCA-RF using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

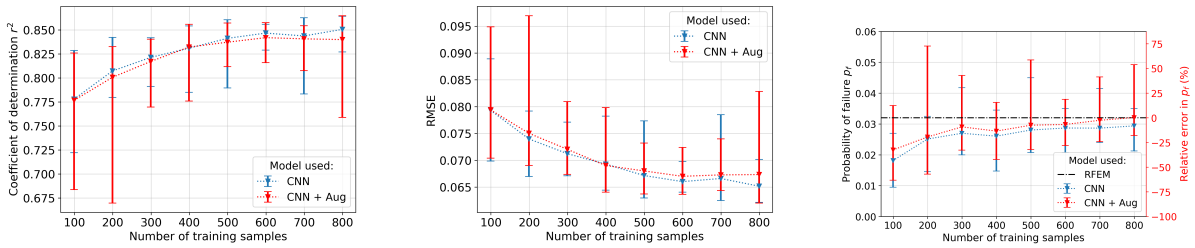


Figure C.9: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case U3 obtained by the CNN using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

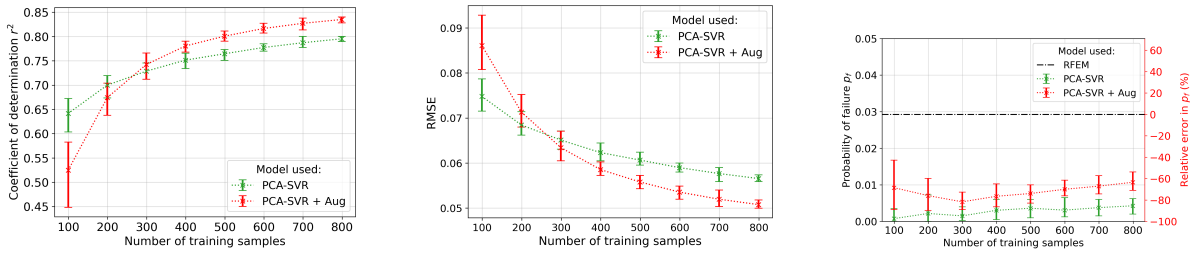


Figure C.10: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case T1 obtained by the PCA-SVR using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

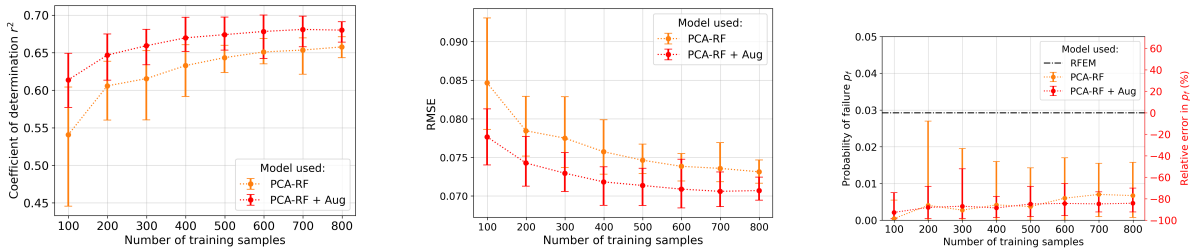


Figure C.11: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case T1 obtained by the PCA-RF using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

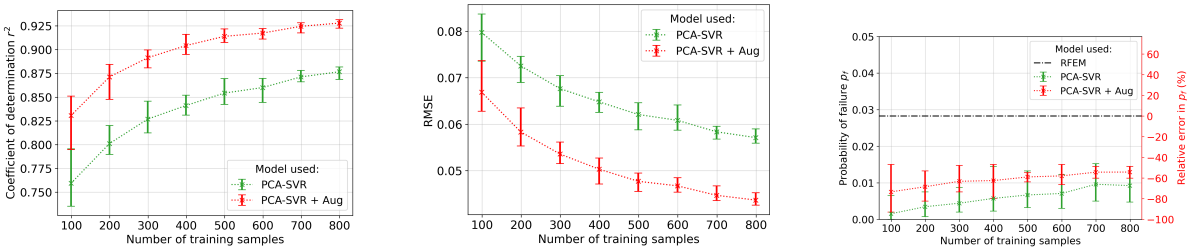


Figure C.12: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case T2 obtained by the PCA-SVR using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').

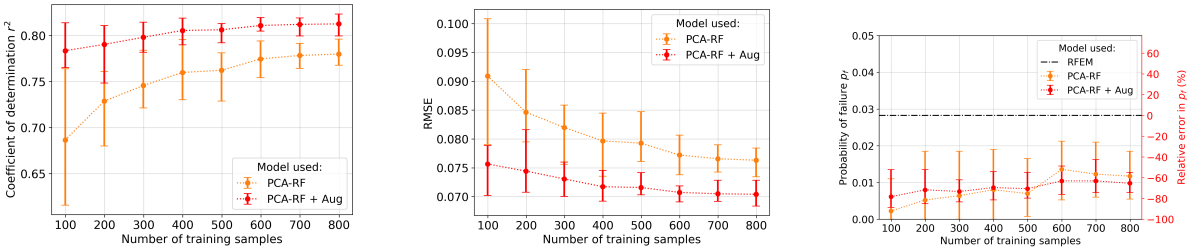


Figure C.13: Performance metrics r^2 (a), RMSE (b), and relative p_f error (c) for case T2 obtained by the PCA-RF using different training sample sizes, combined with and without the data augmentation technique (abbreviated with '+aug').