MSc. Thesis

Deep learning-based design model for suction caisson foundation in clay

Xilin Yin

Delft University of Technology

TU Delft

Delft University of Technology

MSc. Thesis

Deep learning-based design model for suction caisson foundation in clay

Student:
Xilin Yin
(5271649)

Committee:
Prof. Dr. K.G. Gavin
Dr. F. Pisanò
Dr. H. Wang
Dr. H. P. Zhou

September 23, 2022

Preface

This is the last part written at the end of my thesis. Past experiences flashed through my mind. From the first contact with the project last October to the completion in September 2022, I have gained knowledge, friendship and growth. The companionship and understanding of my parents, the guidance and encouragement of my mentors and the care and help of my friends.

My deepest and sincerest gratitude goes to my supervisor, Huan. He gave me the most encouragement and technical support, from guiding FE modelling step by step to organizing thesis structures and directions. And he is responsible for providing a lot of guidance on academic writing. The same gratitude to Hongpeng for bring unique perspectives and insights to my research. Also, the impressive PhD defense greatly encouraged me and made me firm in my future direction. I am very grateful to Federico for his crucial comments on improving my model and thesis. Many thanks to Ken, the chair of my thesis committee, whose unforgettable class on soil-structure interactions allowed me to define the direction of my research. Also many thanks to Paul and Xinyuan for their generous advice and supports. I would also say thanks to my friends who helped me along the way and gave me confidence when I was frustrated.

Finally I must thank my parents for giving me the opportunity to pursue my dream in the Netherlands during these two years. These two years of experience will be a valuable asset in my life, and I will bravely move forward.

Xilin Yin
Delft, The Netherlands
September 2022

Contents

List of Figures

List of Tables

Abstract

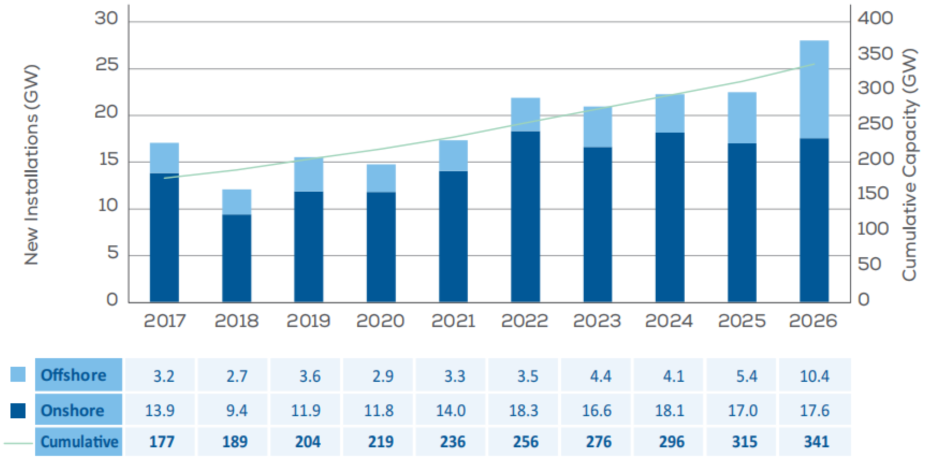
Suction caissons have been used extensively for anchoring and supporting the offshore installations like oil platforms and wind turbines. These foundations are normally subjected to complex combinations of the vertical, horizontal and moment loads (i.e. V, H, M) from the self-weight, wind, wave and currents. In the past decades, extensive studies have been conducted to investigate the combined V-H-M loading behaviour of suction caissons in clay. However, most existing studies are focused on the ultimate bearing capacity, while the deflection response is more critical in foundation design for recent infrastructures like offshore wind turbines. Due to the complex load conditions, predicting the three-dimensional (3D) deflection response of the foundation is still challenging. Machine learning (ML) appears on the research horizon due to its excellent capacity of solving nonlinear problems with desired speed and accuracy. However, conventional machine learning approaches were limited in their capacity to analyze raw natural data without artificial interventions. Meanwhile, the deep learning technique (DL), as a branch of machine learning, allows a machine to be fed with raw data, automatically extract the features, and discover intricate structures in high-dimensional data. The deep learning technique has been used in many fields like language translation, auto-pilot and image recognition. And Deep neural networks, including deep learning algorithms and architectures, are gradually being developed. In light of these backgrounds, this study proposed to develop a deep learning based surrogate model to predict the 3D deflection response of suction caissons under combined V-H-M loading. The advanced three-dimensional nonlinear finite element (FE) simulations under complex V-H-M loading paths were performed on suction caissons of different geometric configurations and in clay soils with different stiffness and strength properties. The 3D FE simulation data was then used to train the deep learning based design model. Three popular neural network structures, i.e., Feed forward Neural Network (FNN), Convolution Neural Network (CNN), Recurrent Neural Network (RNN) have been employed to develop the hybrid surrogate design model. In this study, two different training strategies were proposed for this geotechnical problem. In the first category, the 3D load-deflection behaviour of suction caisson is idealized as a point-to-point mapping problem, i.e. mapping between the deflections (i.e. displacement and rotation) with loads (i..e force and moment). This task was achieved by Fully-Connected Neural Network model (FC-NN) based on FNN, One Dimension Convolution Neural Network model (1D-CNN) based on CNN and Long Short Term Memory model (LSTM) based on RNN. In the second training strategy, the load-deflection response was idealized as a time series process, a line-to-line mapping problem, mapping between the past loading paths (i.e. 10 groups of forces and moments) with future loading paths (i.e. 90 groups of forces and moments). Besides the three neural network models mentioned before, another two complex and advanced models, LSTM Model combined with convolution neural network (1D-CNN+LSTM) and Temporal Convolutional Network model (TCN), are also applied for temporal prediction. The performance and training efficiency of these models were also systematically evaluated by interpolation and extrapolation experiments. Basically, all the models can well capture the 3D deflection response of the foundation with significantly high accuracy (i.e., root mean squared error is smaller than 0.05 and coefficient of determination is near 1.000) than the traditional design approach (such as macro-elements model), and with greater efficiency than the 3D FE simulations. Among all the models, the TCN model has the highest prediction accuracy and robustness. However, the FC-NN model has the simplest model structure and highest computational efficient in learning the non-linear relationship between deflection response and V-H-M load. Besides capturing the relationship between input and output, the deep learning model can also assist to identify the intrinsic failure mechanism. By observing the fluctuation of generalisation ability, the evolution of the failure mechanism of suction caisson with embedment depth was revealed.

Introduction 1

1.1 Research background

1.1.1 Offshore wind energy

Developing renewable energies has become the consensus around the world to solve the climate and energy problems. Wind energy is one of the most promising renewable energy sources with greater consistency and efficiency output. Compared with the on-shore wind, offshore winds are faster, unobstructed and more consistent, making them more accessible (Abdel-Rahman and Achmus, 2005). Therefore, many countries are putting their bats on offshore wind energy to combat global environmental problems. By the end of March 2021, the global offshore wind capacity in operation has reached nearly 35.3 GW, compared with 3.3 GW in 2011. According to projections from the Global Wind Energy Council (GWEC), 469 GW new capacity of global offshore wind will be added in the next five years.The trend in new capacity growth for offshore wind in Europe is shown in Figure 1.1. It should be noted that more than 75% of all commissioned wind farms are constructed in offshore areas with water depths of less than 30 m (Rystad, 2021). Driven by the successful leap from onshore to shallow offshore, the industries are making another significant move to build offshore wind farms in deeper water to maximize the potential global offshore wind resource.



	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026
Offshore	3.2	2.7	3.6	2.9	3.3	3.5	4.4	4.1	5.4	10.4
Onshore	13.9	9.4	11.9	11.8	14.0	18.3	16.6	18.1	17.0	17.6
Cumulative	177	189	204	219	236	256	276	296	315	341

Figure 1.1: Expected new installations from 2022 to 2026 (Source from WindEurope)

1.1.2 Foundation types for offshore wind turbines

For a typical offshore wind turbine, the foundation cost can take up more than 30%, as shown in Figure 1.2 (Wang, 2020), making it a very important role in the economic feasibility

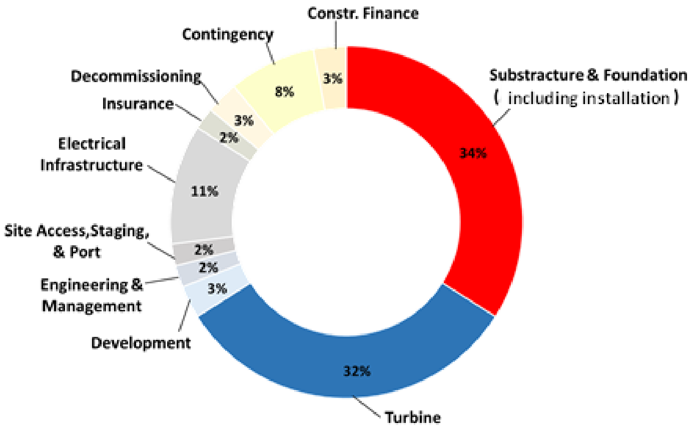


Figure 1.2: Cost share of a typical offshore wind turbine (Wang, 2020)

of the whole project. Up to date, many different foundation types have been used to support offshore wind turbines as shown in Figure 1.3, depending on the water depth and seabed conditions. In shallow waters with water depths ranging from 0 m to 25 m, gravity bases, suction caissons, and monopiles can be utilized. For the offshore wind turbine installations in shallow waters (depth less than 30 m), the monopile is the most widely used foundation type covering more than 81% of total installations in Europe (as shown in Figure 1.4). However, when the construction moves to deeper water, the difficulty in installation and the increase in the cost of the monopile make it less preferable. Instead, the multi-pod support structures such as tripods and jackets, are preferred (Lau, 2015). In particular, the suction caisson foundation has been considered as a promising alternative for its easy installation.

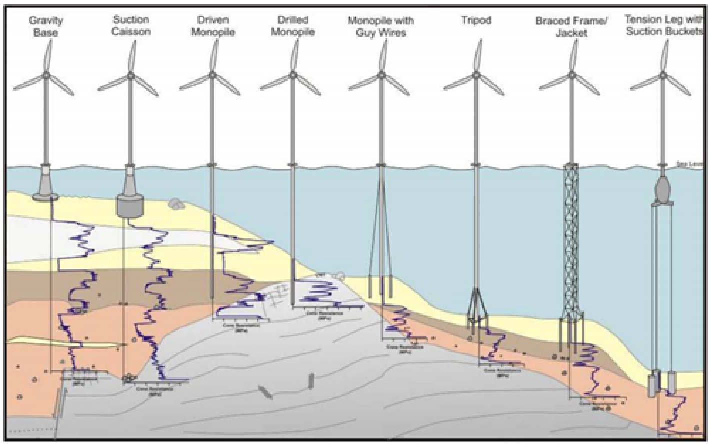


Figure 1.3: Wind Turbine Foundation Options(Lau, 2015).

The typical suction caisson foundation consists of a steel cylindrical shell with a top plate and various valves that allow water to be pumped into or out of the shell. It has an open bottom that allows soil to enter the internal volume of the caisson. In the field, the suction caisson will first penetrate the soil under its own weight, forming a sealed space. Then the caisson is embedded into the seabed to target depth by pumping out the water in the

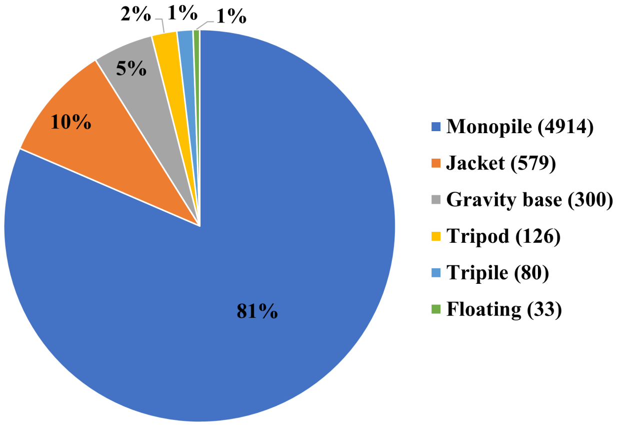


Figure 1.4: Offshore wind foundations in Europe.

shell, creating a negative (suction) pressure inside the caisson. As a result, the resultant pressure differential across the top plate will effectively push the caisson into the seabed. This installation procedure is illustrated in Figure 1.5. Compared with other traditional offshore foundations, the suction caisson has many advantages, including:

- Installed easily without noise pollution
- High overturning resistance
- Less steel costs
- Using reusable materials
- Located accurately

Therefore, the suction caisson foundation has a bright future for the application.

1.1.3 Design requirement of offshore wind turbine foundation

The offshore wind turbine is a long slender dynamic sensitive structure. For a typical 5 MW offshore wind turbine, the hub height can be up to 110 m with a rotor diameter of 126 m. Due to their high and flexible structural characteristics and the distribution of mass and stiffness, offshore wind turbines have very strict requirement for the system frequency and structure deflection.

For the safety, it is important to avoid the possible resonance from the overlap of the natural frequency of offshore wind turbines with those of the external loads, like the wave and the “shading effect” generated by blade passing (Bhattacharya et al., 2013). Figure 6 presents a summary of the typical loading frequency applied to the offshore wind turbine (Jonkman et al., 2009). The 1P and 3P denote the rotational frequency of the turbine and the blade passing frequency, respectively. To avoid any resonance, the natural frequency of the system should be designed lower than 1P (i.e., ‘soft–soft’), between 1P and 3P (i.e., ‘soft–stiff’) or

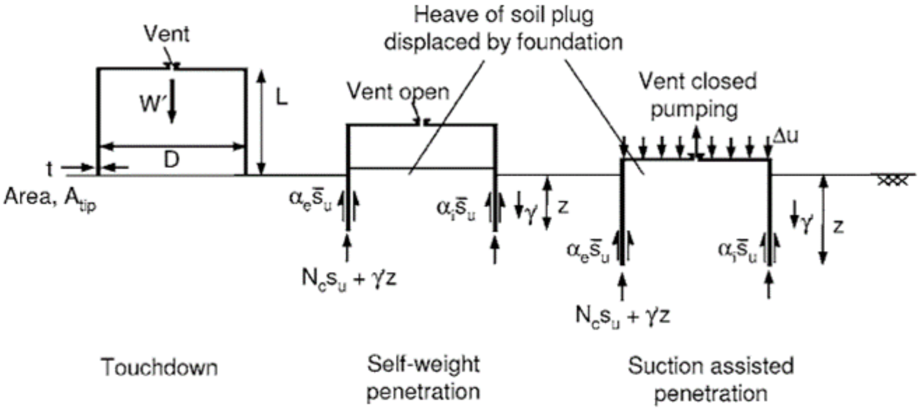


Figure 1.5: Steps and forces during installation of a suction caisson (Randolph and Gourvenec, 2017).

larger than 3P (i.e., 'stiff–stiff' regions). Typically, the initial natural frequency of most offshore wind turbines is designed to be 'soft–stiff', lying in the interval between turbine and blade passing frequencies, accounting for the cost and design feasibility (Bhattacharya et al., 2013; Yu et al., 2015). Therefore, the frequency is very strictly limited, typically between 0.22 Hz and 0.31 Hz.

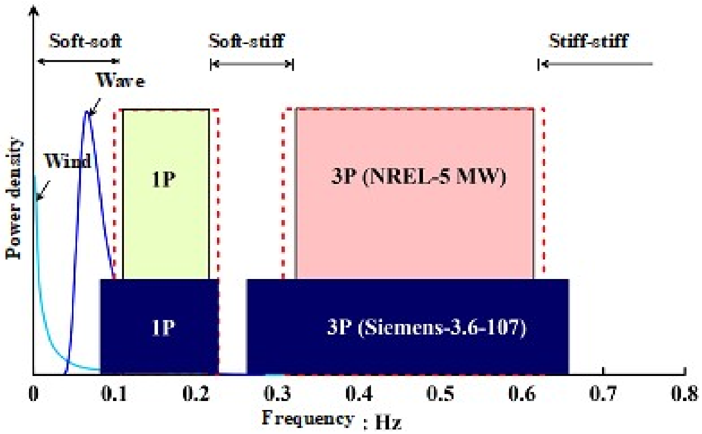


Figure 1.6: Typical loading frequencies and dynamically sensitive regions of an NREL 5 MW turbine structure(Jonkman et al., 2009).

According to the DNVGL design guideline, the maximum rotation of the offshore wind turbine should be limited to 0.25 degrees. However, the soil will exhibit nonlinear response under loading, with its stiffness degrading significantly at small strain. This will then affect the foundation deflection response under external loads. As illustrated in Figure 1.7, results from large-scale footing tests on 3 m diameter caisson foundations show the clear degradation of foundation stiffness with the rotation (Houlsby, 2016). Therefore, the design of the offshore wind turbine foundation requires an accurate prediction of the foundation stiffness more than the bearing capacity under small deflection.
Meanwhile, the offshore wind turbines are subjected to complex loads from self-weight,

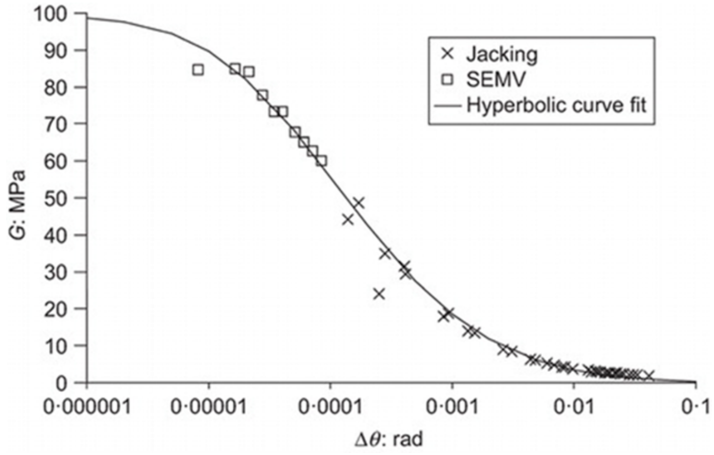


Figure 1.7: Change of rotational stiffness of a foundation as a function of rotation amplitude(Houlsby et al., 2006).

wind, wave and current. As shown in Figure 1.8, the dominant loads on the supporting substructures of offshore wind turbines are the lateral force and moment, rather than the huge vertical load on the offshore platform (Houlsby, 2016). Traditionally, the horizontal, vertical and moment load components are treated separately in the bearing capacity approach which ignore the interaction between them. This may lead to a non-conservative prediction. Therefore, to accurately predict the foundation response of offshore wind turbines, it is necessary to study the deflection response or stiffness response of the foundation under complex 3D loading condition.

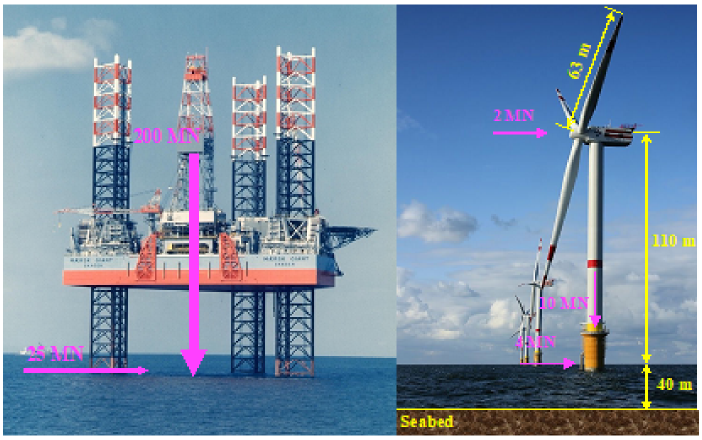


Figure 1.8: The representative loads on a typical offshore platform and a wind turbine(Houlsby, 2016).

1.1.4 Artificial intelligence applications in geotechnical engineering

In 1956, a group of computer scientists proposed that computers could be programmed to think and reason. Then they described this principle as “artificial intelligence.” Artificial intelligence (AI) is a field focused on automating intellectual tasks normally performed by humans, and machine learning (ML) and deep learning (DL) are specific methods of achieving this goal (Choi et al., 2020). Neural networks are a network structure that incorporates deep learning algorithms and have been used extensively in recent years. Deep learning, in turn, is a branch of machine learning that can actively extract features between data and is more intelligent than conventional machine learning. The relationship of these concepts are shown in Figure 1.9. The field of data science encompasses artificial intelligence (AI), which comprises classical programming and machine learning (ML). ML includes several models and techniques, such as deep learning algorithm (DL) with various neural networks.

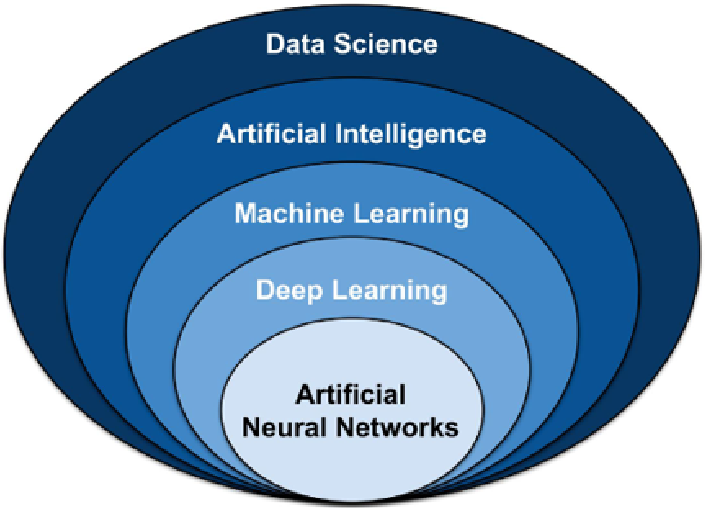


Figure 1.9: The relationship between Machine Learning, Deep Learning, and Neural Networks(Choi et al., 2020).

Since the last century, artificial intelligence (AI) provides several advantages over more traditional computing techniques. Numerous mathematical models are incapable to simulate the complex behavior of most geotechnical engineering problems. In contrast, AI methods are data-driven methodologies in which model development is determined by training input-output data pairs to define the model's structure and parameters. From the early 21st century, Artificial Neural Network (ANN), a kind of deep learning algorithm, emerged as the most widely used, with around half of studies relying on it. ANN and other machine learning methods have been successfully applied in the diverse field of geotechnical engineering (shown in Figure 1.10). Particularly, in the field of shallow and pile foundations, highly accurate predictions of bearing capacity, settlement estimation and load-settlement response have been achieved with deep learning neural networks (Shahin, 2016). This new method gives ideas for studying the deflection response or stiffness response of foundations under complex three-dimensional loading conditions.

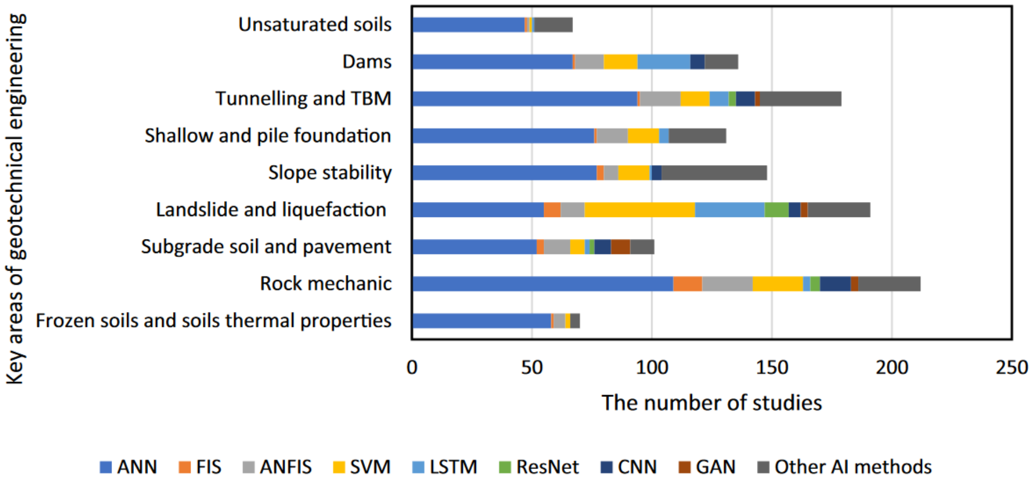


Figure 1.10: Distribution of the use of different AI techniques in the geotechnical engineering(Baghbani, 2022).

1.2 Research objectives and strategy

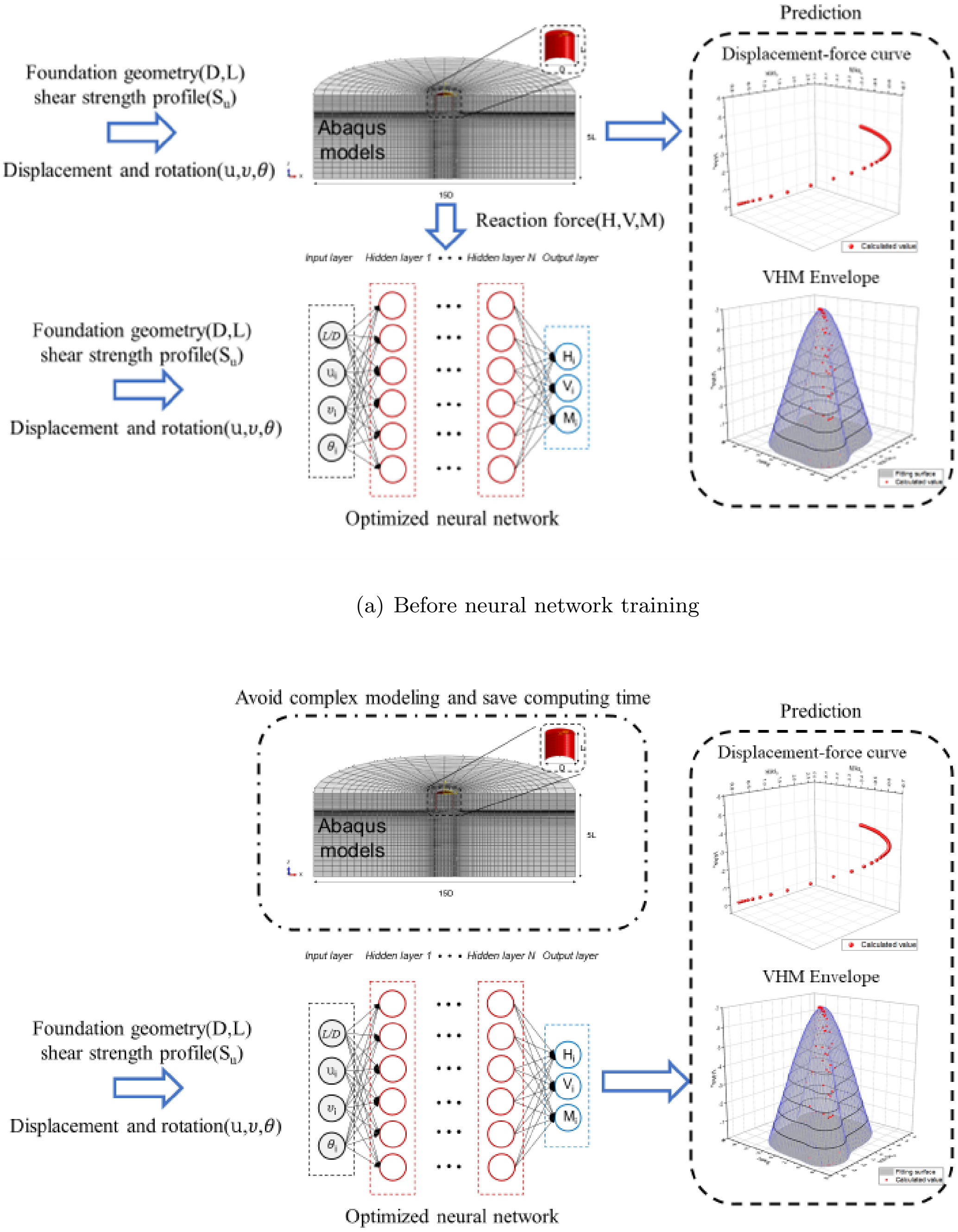
The previous background can be summarized as follows:

1. The offshore wind turbine foundation is a highly non-linear system under various load combinations with significant stiffness degradation at slight deflection.

2. The field data on suction caisson is scarce, and the deflection responses of various suction caissons under different load combinations and field conditions are impossible to obtain. And the FE simulation process is very time-consuming.

3. The deep learning technique is famous for describing the complex multi-factor non-linear system. The deep learning model can provide highly accurate predictions after trained, and this technique has been used in many geotechnical fields.

In these lights, the deep learning's excellent fitting capabilities are used to forecast the mechanical response of the foundation in an effort to replace conventional computational methods in the geotechnical domain. In this project, the response of the foundation under the combined loads were analyzed first using the advanced three-dimensional (3D) finite element modelling. Each simulation represents the mechanical response of the foundation under one load combination, and the mechanical response under different time steps form the loading path. After numerous simulations with different load combinations, the ultimate state of the loading paths constitutes the failure envelope of the suction caisson. These load-displacement response of the foundation were then studied. After finite element modelling, the load-displacement data of caissons with different geometric configurations and in different clay soils was fed to the deep learning model, which can directly learn the mapping relationship between the displacement and external loads of caisson foundations from the raw numerical data. The process from modelling to training the model is illustrated in Figure 1.11(a).



(a) Before neural network training

(b) After neural network training

Figure 1.11: Structure diagram of the whole research

After training, deep learning algorithms attempt to predict as precisely as possible mechanical response and failure envelope. When the prediction accuracy reaches the acceptable level, the trained deep learning model will be regarded as an alternative method of the complex FE modelling process and save computational cost (demonstrated in Figure 1.11(b)). In the end, an AI-based model for the suction caisson foundation in clay was developed. This trained AI-based model can implicitly incorporate the intrinsic control mechanism of the foundation under varied conditions without the limitations of the predefined assumptions in the traditional design model. This is of great benefit for the design companies, allowing them to predict the load response of the caisson within seconds.

When the deep learning model mimic the nonlinear relationship between force and displacement, the essential inputs to the training model are the horizontal displacement (u), the vertical displacement (v), the angle of rotation (θ), the embedment ratio (L/D), and the heterogeneity parameter (κ) of the soil. The horizontal reaction force (H), vertical response force (V), and moment (M) are considered as corresponding outputs. This selection of features follows the principle that the less input and output parameters decrease the difficulty of training a deep learning model. In addition, different deep learning models were studied to evaluate their accuracy and efficiency when using in foundation design. The deep learning models can predict the foundation response with both high accuracy and efficiency. The performance of each model was also systematically evaluated.

The main objectives of this project are summarized below:

- Investigate the three-dimensional response of the suction caisson under combined loads(V-H-M) by performing 3D FEM simulations under different loading paths (i.e. ratio between the u-v-θ), which will serve as the training database for deep learning;

- Study the influence of soil strength profiles (i.e. constant and linear s_u profiles against depth) and foundation geometric configurations (i.e. length to diameter ratio, L/D) on the 3D load-deflection response of caisson;

- Evaluate the performance of existing deep learning techniques when being used for foundation design;

- Develop the AI-based model for offshore foundations by selecting the most appropriate computational model and neural network structure;

1.3 Research contributions

- **The 3D response of the foundation researched by FE modeling**
  After FE modeling, the behaviour of suction caisson foundations under three-dimensional loading in different soil profile with different foundation geometries are investigated. The calculation results reveal the non-linearity of the load-displacement response of the foundation and the variation of the bearing capacity envelope with changes in burial depth and soil profile under different deformation conditions. And comparing the behaviour of the foundation under different reference point configurations, a unique failure mechanism is proposed when the reference point is set at the mudline.

- **Data driven model developed for 3D mechanical response of suction caisson prediction.**
  Up to date, existing studies on the three-dimensional (3D) response of suction caisson always tends to simplify their investigation into a two-dimensional (2D) problem and only focus on the ultimate state. This was mainly limited by the high computational expense and the complex of soil non-linearity induced deformation of the foundation. While the development of hybrid surrogate model will greatly facilitate reproducing the three-dimensional response of suction caisson under various load combinations. The 3D load-deflection behaviour of suction caisson is idealized as a point to point mapping problem. A fully connected (FC) neural network based model trained from raw data predict nonlinear mechanical responses by fitting the internal relationship within load and displacement. Moreover, this model can capture the 3D response of foundations with a wide range of embedment ratios and strength heterogeneity for the extensive applications. This advanced method will investigate the behaviour of suction caissons more comprehensively and efficiently. The modelling process and model's applicability will be demonstrated in detail in Chapter 4.

- **The unknown response of foundation predicted with limited data.**
  Deep learning algorithms are capable of accurately predicting future trends based on historical time series data, (i.e. stock prices, weather patterns, etc.), thanks to their superior computational power. Since 2017, time series prediction has also been increasingly applied for geotechnical problems such as tunnel construction and landslide displacement. After the point-to-point mapping problem was proven feasible, the variation of H-V-M load response with the caisson deflection will be treated as a time series of data, a line-to-line mapping problem. Therefore, the initial 10% data of FEM simulations were used to predict the whole range response (100% data) of the foundation. It means that future behaviour of suction caissons can be predicted based on existing inspection data. This kind of temporal prediction has promising applications for foundation monitoring and maintenance. The modelling process and predictions will also be demonstrated in detail in Chapter 4.

- **The deep learning models' performances compared in foundation behaviour prediction.**
  Although deep learning models have been widely used in the shallow and pile foundation field, there are insufficient guidelines for selecting deep learning models. Therefore, various state-of-the-art models were applied when performing non-linear regression prediction and time-series prediction. The accuracy, robustness, efficiency, and generalisation of the deep learning models were evaluated by interpolation and extrapolation predictability. All the models can well capture the 3D deflection response of the foundation with much higher accuracy than the traditional design approach. With the simplest model structure, the Fully Connected Neural Network (FC-NN) model was considered the most suitable for fitting the non-linear relationship between force and displacement. The Temporal Convolutional Network (TCN) model was discovered balancing accuracy, efficiency and robustness in the temporal prediction, challenging the dominating long short-term memory (LSTM) model in the temporal

domain. In Chapter 4, the prediction accuracy, computational efficiency and robustness of the individual models are demonstrated.

- **Providing a new direction of exploration in the geotechnical mechanism combined with AI.**
  Based on the model generalisation results in Chapter 5, an attempt was made to find the intrinsic mechanism for the poor generalisation ability of the partially embedded depths. Through the analysis of the anomalous data, the changes in the data distribution pattern were found to coincide with the changes in the failure mechanisms of the foundation under different embedment depths. The effect of the skirt geometry on the suction caisson failure mechanisms will be explained again from the perspective of deep learning. This way of argumentation forms a closed loop clarifying, to some extent, that deep neural networks can not only work as a black box to provide an accurate prediction of foundation response. But AI also has the potential to identify or reveal the intrinsic mechanism in the geotechnical field.

1.4 Thesis outline

The entire article is organized as shown below.

Chapter 1 concentrates on the research background, the methodology and objectives of the study.

Chapter 2 presents a comprehensive literature review related to the existing studies on the 3D combined response of suction caisson foundation and their limitations. Following the discussion on the suction caisson, this chapter provides a comprehensive description of the deep learning technique evolution and the AI technique applied in geotechnical engineering.

Chapter 3 explains how the database for this study was created. The FE simulations were performed first to provide the training data for the deep learning algorithm. The data set was generated by 3D FE modelling using Abaqus in both homogeneous and heterogeneous clay. The data validation was verified from the perspective of the ultimate bearing capacity and internal mechanisms. The FE simulation cases covered the foundations of a wide range of load combinations and embedded ratios in both homogeneous and heterogeneous soil.

Chapter 4 is the core of the research. Section 4.1 describes how the deep learning algorithm was combined to make predictions about the 3D mechanical response of foundation. Section 4.2 provides an overview of the deep learning models used for predictions. In Section 4.3, the 3D response force predictions at a given displacement were achieved by fitting a non-linear relationship between force and displacement. Section 4.4 explained how the unknown mechanical response of the foundation could be predicted from limited data. Moreover, the performance of the different models, such as prediction accuracy, computational efficiency and model robustness, was compared as well.

Chapter 5 further analyses the models prediction error and generalisation ability in different embedment depths. Several attempts have been made (i.e. model splitting, data augmentation and model complexity adjustment) to improve the generalisation of the model according to the regularisation theory. Finally, the generalisation ability of the improved model is tested by fast prediction of the failure envelope experiment.

Literature review 2

As highlighted in the preceding section, supporting substructures of offshore wind turbines are subjected to complex loads. It is therefore critical to understand the load-deflection response of a suction caisson foundation and develop an accurate and efficient design model for foundation design. The suction caisson foundations have been used in oil and gas as foundation or anchors since the last century. Extensive studies were conducted to understand its behaviour. In this section, a thorough review is provided on the existing studies about the suction caisson foundations, including the elastic model, bearing capacity envelope, macro-element model, FEM method and the new trend of deep learning surrogated model. The pros and cons of each theory/model have been discussed when being used in offshore wind turbine foundation design.

2.1 Geotechnical methods in foundation design

2.1.1 The elastic design theory of suction caisson

The most simple and classic design method for the suction caisson foundation is the elastic model which is based on a set of governing equations for the determination of elastic stress changes within a half-space due to a surface point load (Randolph and Gourvenec, 2017). The elastic response of a caisson, treated as a rigid cylindrical foundation, can be represented by the following matrix equation:

$$\begin{bmatrix} V/G_G D^2 \\ H/G_G D^2 \\ M/G_0 D^3 \end{bmatrix} = \begin{bmatrix} K_V & 0 & 0 \\ 0 & K_H & K_C \\ 0 & K_C & K_M \end{bmatrix} \begin{bmatrix} W/D \\ u/D \\ \theta \end{bmatrix} \tag{2.1}$$

Where:

$V, H, M.$ = vertical, horizontal and moment loading,
G_0 = small-strain shear modulus,
D = caisson outer diameter,
K_V, K_H, K_M, K_C = vertical, horizontal, moment and coupling stiffness coefficients, respectively.
w, u, θ = vertical, horizontal and rotational deformation, respectively

Although the elastic model is simple, it should be noted that the model does not reflect the non-linear response of the soil. As is widely recognized (Atkinson, 2000), the soil only exhibits linear elastic response at very small strains (less than 10^{-5}). Figure 2.1 shows the typical degradation of the soil modulus with shear strain and the most related strain range for a foundation (Atkinson, 2000). It can be seen there is a huge degradation of the stiffness from the elastic range to the strain level of a typical foundation, which means that using the elastic model will highly overestimate the foundation stiffness and also cannot capture the nonlinearity of foundation stiffness (Houlsby, 2016).

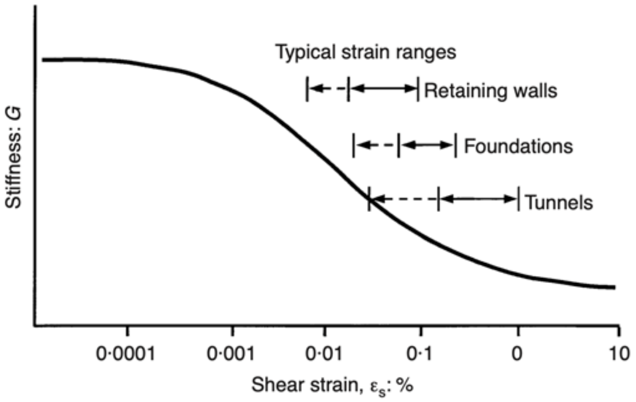


Figure 2.1: Characteristic stiffness-strain behaviour of soil with typical strain ranges for laboratory tests and structures(Atkinson, 2000).

2.1.2 The bearing capacity of suction caisson

Besides the studies on the elastic mode, a number of studies were performed to study the ultimate bearing capacity of the suction caisson foundation under combined loading. To capture the foundation capacity under complex vertical (V), horizontal (H) and moment (M) loads, the failure envelope approach is usually adopted, as shown in Figure 2.2. The failure envelope is a hypersurface that defines the multi-dimensional combination of bearing capacity loads (Suryasentana et al., 2020). The failure envelope approach was first introduced by Roscoe (1956) and has been widely adopted to describe the combined foundation bearing capacity, for a broad range of foundation types, such as mudmat (Feng et al., 2014), rigid pile (Graine et al., 2021), suction caisson (Gerolymos et al., 2015) and shallow foundation (Gourvenec, 2007).

Many studies have been conducted to establish the failure envelope of the suction caisson foundation (Bransby and Randolph, 1998; Bransby and Yun, 2009; Gourvenec and Barnett, 2011; Hung and Kim, 2014; Karapiperis and Gerolymos, 2014; Gerolymos et al., 2015; Vulpe, 2015; Mehravar et al., 2016). Numerous approximating expressions have been proposed to predict HVM failure envelopes, listed in Table 2.1 below. Taking Hung and Kim (2014)'s equation as an example in Figure 2.3, the computed and fitting outcomes are quite similar. It indicates that these expressions are efficient in obtaining the failure envelope with high accuracy. However, they only focused on the ultimate bearing capacity and due to the limitation of the function itself, it's impossible to capture the whole-range (from the initial to ultimate) response of the foundation with this simple equation. Moreover, it should be noted that the foundation deflection required to mobilize the bearing capacity is normally very large and exceeds the service limit conditions of a foundation. This is extremely important to the supporting foundations of offshore wind turbines. As explained in the preceding section, the

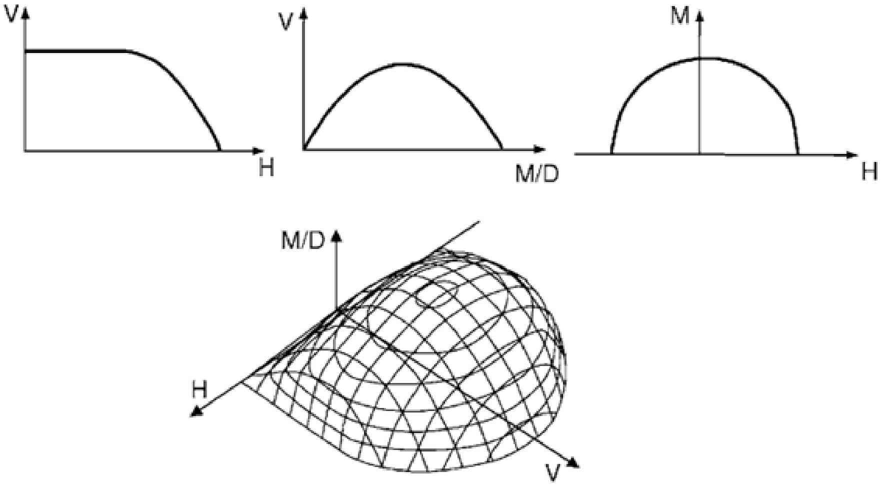


Figure 2.2: Representation of failure envelopes in two- and three-dimensional load space(Randolph and Gourvenec, 2017).

maximum allowed rotation of an offshore wind turbine is 0.25 degrees, which is significantly away from the deflection required for mobilizing the bearing capacity. Therefore, although the bearing capacity will be checked in the design of an offshore wind turbine foundation for the ultimate limit state, the foundation design is more controlled by the service limit state, i.e., the nonlinear stiffness at small deflection.

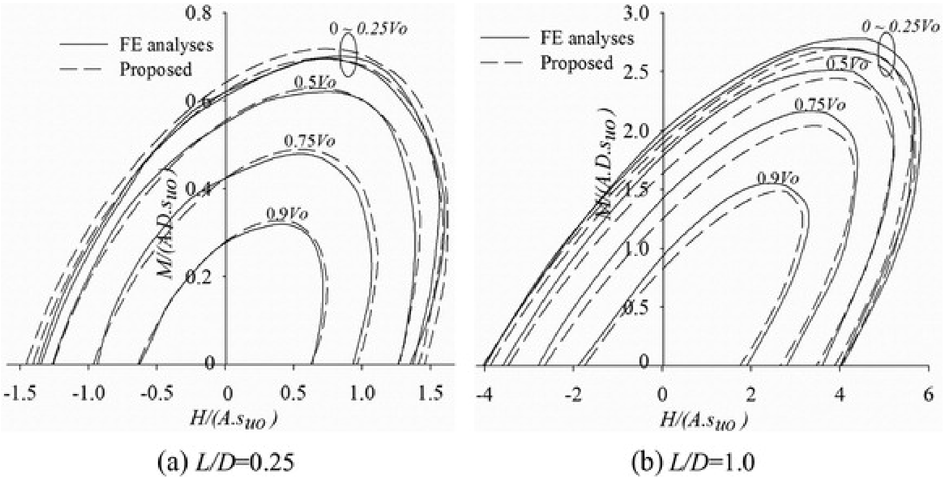


Figure 2.3: Comparison of VHM capacity envelopes between the proposed equation and FE analyses(Hung and Kim, 2014).

Table 2.1: Summary of published work on an approximating expression to predict HVM failure envelopes

Reference	Proposed Equation
Murff (1994)	$f=\left(\left\lvert\frac{V}{V_0}\right\rvert\right)^2-1+\left[\left(\left\lvert\frac{M^*}{M_0}\right\rvert\right)^m+\left(\left\lvert\frac{H}{H_0}\right\rvert\right)^n\right]^{1/p}$
Bransby and Randolph (1998)	$f=\left(\left\lvert\frac{V}{V_0}\right\rvert\right)^{2.5}-\left(1-\left\lvert\frac{H}{H_0}\right\rvert\right)^{1/3}\left(1-\left\lvert\frac{M^*}{M_0}\right\rvert\right)+\frac{1}{2}\left(\left\lvert\frac{M^*}{M_0}\right\rvert\right)\left(\left\lvert\frac{H}{H_0}\right\rvert\right)^5$
Taiebat and Carter (2000)	$f=\left(\frac{V}{V_{\mathrm{utt}}}\right)^2+\left[\left(\frac{M}{M_{\mathrm{ut}}}\right)\left(1-\alpha\frac{HM}{H_{\mathrm{utl}}\lvert M\rvert}\right)^2\right]+\left(\frac{H}{H_{\mathrm{ult}}}\right)^3\vert-1=0$
Gourvenec and Barnett (2011)	$\left(\frac{h}{h^*}\right)^\alpha+\left(\frac{m}{m^*}\right)^\alpha+2\beta\left(\frac{hm}{h^*m^*}\right)=1$
Hung and Kim (2014)	$f_{\mathrm{VHM}}=\left(\frac{M}{M_0}\right)^2-\alpha\beta\lambda\left(\frac{H}{H_0}\right)\left(\frac{M}{M_0}\right)+\left(\frac{H}{H_0}\right)^2+\left(\frac{V}{V_0}\right)^2-1=0$
Karapiperis and Gerolymos (2014)	$f=\left(\frac{Q}{Q_u}\right)^2+\left(\frac{M}{M_u}\right)^2+m\left(\frac{Q}{Q_u}\right)\left(\frac{M}{M_u}\right)-1=0$
Vulpe et al. (2014)	$\left(\left\lvert\frac{h}{h^*}\right\rvert\right)^\alpha+\left(\frac{m}{m^*}\right)^\alpha+2\beta\frac{hm}{h^*m^*}=1$
Gerolymos et al. (2015)	$f=\left(\frac{Q}{Q_u}\right)^{n_1}+\left(\frac{M}{M_u}\right)^{n_2}+n_3\left(\frac{Q}{Q_u}\right)\left(\frac{M}{M_u}\right)-1=0$

2.1.3 The macro-element model for suction caisson

To compensate for the deficiency of the elastic model and the failure envelope approach, the macro-element model was proposed to model the nonlinear load-deflection response of a foundation from the initial elastic state to the final ultimate capacity state. The macro-element is originated from the plasticity theory by mimicking the foundation response as a soil element, where the external loads (i.e.$\mathrm{V}, \mathrm{H}, \mathrm{M}$) and foundation deflection (i.e.u, v, θ) are equivalent to the stress and strain of soil, respectively. Same as a typical plasticity-based model, the macro-element models can account for nonlinear soil behaviour in the integrated soil-structure analyses. Many scholars have found that nonlinear analysis methods based on macro-element can accurately simulate the elastic-plastic behaviour of structures by comparing them with finite elements. The response computed by the macro-element agreed well with the response computed in FEA and has the advantage of fewer degrees of freedom and less computational effort than finite element software analysis. The macro-element concept has its roots back to the pioneering work of Roscoe (1956), but the development of macro-elements dedicated to integrated structural analysis accelerated in the 1990s. A typical macro-element model consists of four parts:

- The elastic stiffness for foundation response at very small deflection
- The yield surface for separating the elastic and plastic response

- The flow rule for determining the ratio between different deflection components

- The hardening law for determining the change of yield surface with plastic deflection

In the early years, the development focused on flat footings, but it became gradually more application oriented focusing in particular on spudcan behaviour for jack-ups. Along with the development of offshore wind over the last two decades, research has also focused on modelling of suction caisson foundation response for OWT applications (Ibsen et al., 2014; Villalobos Jara, 2006; Byrne, 2000; Pisanò et al., 2016; Skau et al., 2018; Yin et al., 2020). These models have mostly built on formulations developed for surface foundations and spud-cans, since the overall framework suits shallow foundation behaviour in general. However, it should be noted the shape of the yield surface, the flow rule (i.e. associated or non-associated) and the hardening law are strongly dependent on the geometric configuration of the foundations and also the geotechnical properties of the seabed (e.g. the nonlinearity of soil, the profile of undrained shear strength of clay, etc.). This shows the complexity of defining a complete macro model and the difficulty of changing the framework once it is defined (OWA, 2019). Up to date, most existing macro-element models are limited to a certain foundation configuration (e.g. the aspect ratio) in homogeneous soil.

2.1.4 The finite element modelling for suction caisson

The finite element (FE) modelling is another approach to understand the foundation response under complex loading. Benefiting from the advance on soil constitutive model and the computer computation power, it becomes more common to directly model the foundation and soil as continuum bodies in finite element model. The FE model is more flexible than the macro element and is a good predictor of the deflection of the foundation. However, finite element analysis requires experts with extensive engineering knowledge to design each model. Meanwhile, in an offshore wind project, the design of the foundation for a wind turbine usually requires thousands of integrated dynamic simulations under different load combinations. It is impossible to use advanced finite element modelling in all the simulations. It is, therefore, necessary to develop a model that inherits the accuracy and flexibility of the model but is simpler and more efficient.

2.2 Application of deep learning in foundation design

Continued from Section 1.1.4, machine learning (ML) is more focused on the "learning" aspect than AI and algorithm is developed based on the human learning process. Machine learning technique involves comparing the actual output of the ML algorithm with the desired output and calculating the error. Based on this error, the optimiser adjusts the parameters of model until a certain criterion is met and then stops adjusting. This process is known as 'learning' and 'training'. Without any previous assumptions, machine learning algorithms are capable of capturing the possible connections between information. However, Machine learning technique requires careful human selection of data features before training(Reimers and Requena-Mesa, 2020). While a specific kind of machine learning, known as deep neural learning, requires more data and achieves greater fitting power and flexibility. In deep learning, features are selected purely based on their usefulness for regression or classification steps, rather than their interpretability. This means that the deep learning technique does not focus on the physical meaning behind the parameters, but only on whether the combination of parameters will improve the prediction accuracy. As a result, this algorithm ends up producing complex features that are difficult to interpret. The deep learning becomes a black box in which the principles of feature extraction are difficult to explain.

The diversity, complexity, and engineering uncertainty of geology requires the engineers to address the complex features of the data. Many mathematical models fail to simulate the complex behaviour of most geotechnical engineering problems. In contrast, deep learning is based on the data alone in which the model can be trained on input-output data pairs to determine the structure and parameters of the model. In this case, there is no need to simplify the problem nor incorporate any assumptions (Shahin et al., 2001). The automated feature selection and good scalability of deep learning algorithms are therefore useful and make them a promising choice for analyzing geotechical engineering problems(Reimers and Requena-Mesa, 2020). Up to now, deep learning algorithms have been successfully applied to soil-structure interaction of geotechnical engineering with typical and novel neural networks (Cheng and Vanapalli, 2021; Zhang et al., 2020a; Nejad and Jaksa, 2017; Shahin, 2014; Momeni et al., 2014; Kuo et al., 2009; Padmini et al., 2008). These papers have demonstrated the promising potential of deep learning in geotechnical engineering, which is more accurate and efficient than traditional theoretical and modelling calculations. In particular, Zhang et al. (2020b) employed Long Short-Term Memory (LSTM) model to identify the mechanical response of caisson foundations in sand. The model was applied to predict the mechanical response of soil-structure interaction and the failure envelope of unknown caisson foundations under different specification tests. The results show that the LSTM-based model is more flexible than the macro element method, allowing the failure mechanism of caisson foundations to be understood directly from the raw data while providing higher computational efficiency and accuracy compared to physical and numerical simulations. However, this study is limited to the 2D problem (H-M) and does not consider the 3D response of the foundation, ignoring the influence of vertical load. And there is still potential for improvement in the model's applicability since it only considers single soil properties.

2.3 Summary

In this light, this study aims to develop an efficient surrogate model by combining the deep learning and finite element modelling. The advanced finite element modelling will be used to generate high-quality training data for deep learning. Once the model is trained properly, we can simply input some basic parameters such as the configuration of the foundation and the soil profile to obtain the response of the foundation directly. This method avoids the complicated and time-consuming process of finite element modelling but maintains the prediction accuracy of the 3D mechanical response.

3 Finite element model of suction caisson in clay

3.1 Introduction

Besides the many approximate expressions proposed for the failure envelope in Chapter 2, extensive studies have been carried out on the bearing capacity of the suction caisson foundation (in Table 3.1. It was found that the size and shape of the failure envelope are controlled by a wide range of factors(e.g. soil profile, drain condition).

Table 3.1: Summary of factors influencing the failure envelope

Influence factors	Reference
Embedment depth	Fu et al. (2017) Gerolymos et al. (2015) Gourvenec and Barnett (2011) Gourvenec (2008) Bransby and Randolph (1999)
Foundation geometry	Bransby and Yun (2009)
Shear strength profile	Fu et al. (2017) Vulpe et al. (2014) Gourvenec and Barnett (2011) Gourvenec and Randolph (2003)
Undrained and drained conditions	Gourvenec and Cassidy (2005)
Deformable and non-deformable soil plug	Vulpe (2015)
Limited interface shear	Gerolymos et al. (2015)
Tension strength	Taiebat and Carter (2000)

However, failure envelopes only represent the ultimate bearing capacity and the ultimate state of the foundation, which is less relevant to the foundation design of offshore wind turbines. Therefore, this research is still centered on the foundation's 3D response predictions under various load combinations, embedment depths and undrained shear strength. Failure envelopes are only used to assist in proving the validation of the simulated data. And this chapter presents a detailed description of the development of FE model, data validation in terms of bearing capacity and failure envelope, and the analysis of the failure mechanism of the foundations. Finally, the 192,000 sets of data incorporating homogeneous and heterogeneous soils were supplied as a database for deep learning algorithms.

3.2 Finite Element Modelling

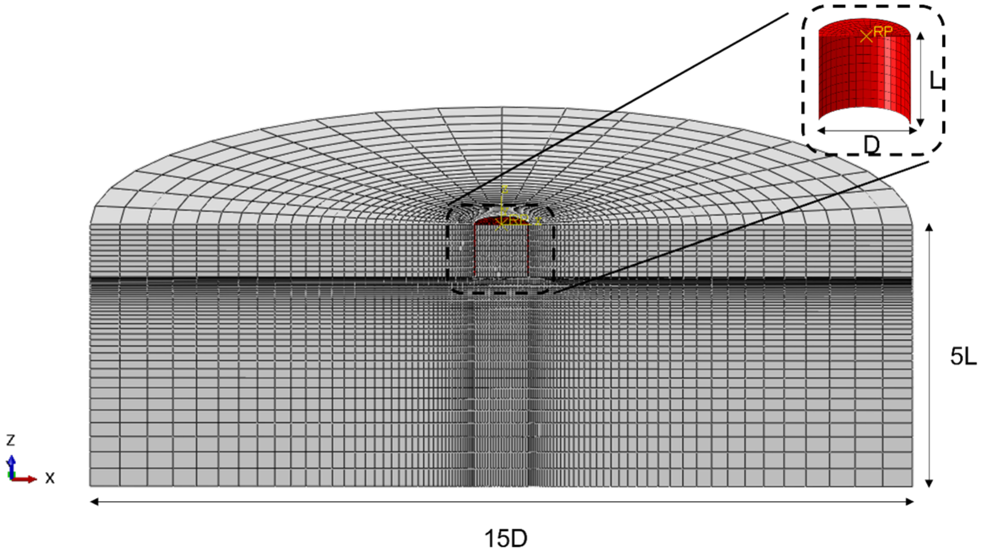


Figure 3.1: A typical FE model mesh of caisson with diameter of 10 m and aspect ratio of 1

In this project, the influence of foundation geometric configurations and the soil properties on the shape and size of the envelope will be systematically studied using the perfect elasto-plastic model with the Tresca yield criteria. Commercial finite element software Abaqus 6.14 was adopted to simulate the suction caisson foundation in clay seabed under combined loads (Manual, 2020). The aim of the simulation is to find the H-V-M response for foundations with different embedment ratios for two different soil conditions, i.e. over-consolidation clay with constant strength profile, and normally consolidated with linear shear strength profile. In this study, the installation process of the skirted foundations was not modelled; rather, the foundations were assumed to be ”wished-in-place”. Therefore, the modelling process in this case only focuses on the response of the foundation under loading.

Caissons are thin-walled large-diameter steel cylinders, open-ended at the bottom and closed at the top, typically less than 20m in foundation diameter (D) with an aspect ratio (L/D, L is the foundation embedment depth) typically less than 1 and thickness ratio (D/t) ranging between 80-300 (Gourvenec and Cassidy, 2005). Therefore, a typical suction caisson is modelling in Figure 3.3 with $L = 10m$. The model suction caisson has a diameter of 10 m, with embedment depth to foundation diameter ratios of $L/D = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$ under different foundation geometry. And the skirt thickness to foundation diameter ratio ($t/D = 0.01$) for all embedment ratios were considered. The suction caisson was represented as rigid bodies with a single load reference point (RP) located at mudline level along the centerline of the foundation. Meanwhile, in order to save computational resources and computational time, only a half model is built considering the symmetry of the problem. The detailed boundary conditions are shown in Figure 3.2. The mesh boundary of the model extends five times the foundation diameter vertically and fifteen times the foundation diameter horizontally from the centerline of the foundation to eliminate the boundary effects. The roller boundaries are applied around the mesh circumference and the base of the mesh was fixed in all three directions.

The skirted circular foundation was modelled as a rigid body and the soil foundation interface was deemed fully bonded to the foundation to simplify the problem. In this paper, the clay is simulated using three-dimensional eight-node hexahedral cells with hybrid, constant pressure (C3D8H). It is advised to represent the behaviour of almost incompressible materials using hybrid components which is appropriate for undrained soil conditions (Gourvenec, 2008). A fine mesh domain was constructed around the foundation skirts and a coarser mesh domain was established at the far field border, allowing the situation be correctly modeled with a significant reduction in computing expenditure. The overall soil domain consisted of 40000 (50*40*20) elements. The mechanical properties in FE modelling are shown in the Table 3.2 below:

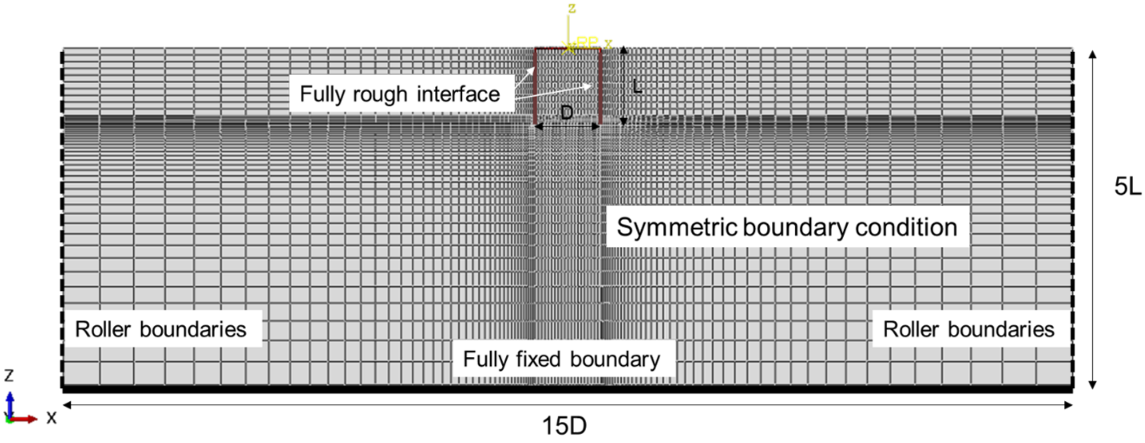


Figure 3.2: Boundary conditions of the FE model

Table 3.2: Mechanical properties in FE modelling

Suction caisson	Outer diameter (D)	10 m
	Length (L)	$1 - 10$ m
Clay	Angle of internal friction (ϕ')	0.01°
	Angle of dilation (ψ)	0.01°
	Shear modulus to undrained shear strength ratio (G/s_{u})	500
	Poisson's ratio (ν_s)	0.495
	Shear strength (s_u)	10kPa
Steel	Young's modulus (E_s)	$210 * 10^6$kPa
	Poisson's ratio (ν_s)	0.25

Sign convention and nomenclature

As recommended by Butterfield et al. (1997), the sign system for displacements and loads described in this study uses right-handed axes and clockwise positive signs. The marked loads and capacities represent the resistance contribution from both the foundation's base and skirts. The sign convention for loads and displacements are illustrated in detail in Figure 3.4.

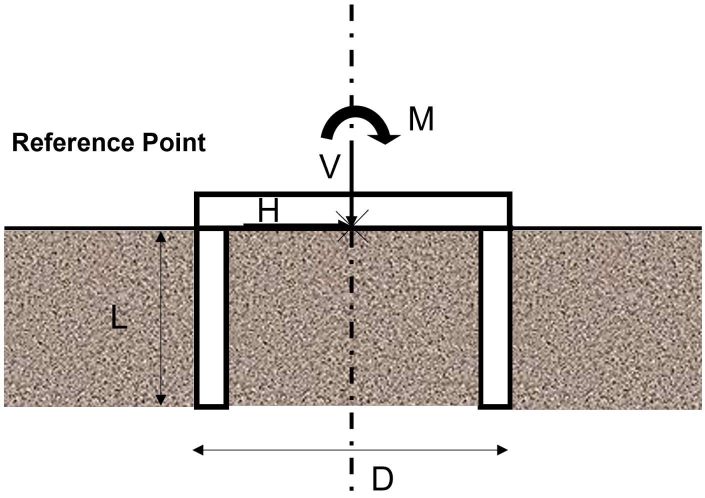


Figure 3.3: Example of skirted foundation geometry and loading

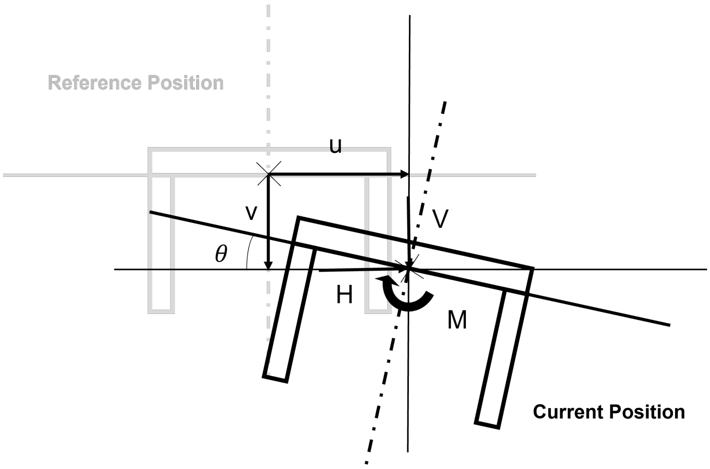


Figure 3.4: Sign convention for loads and displacements

3.3 Loading strategy

There are a range of loading methods such as the Load probe test, displacement probe test, side swipe and sequential swipe test according to the exist researches. The failure envelope of the foundation obtained by different methods are compared in Figure 3.5.

In the figure, the simple sweep method significantly underestimate the failure envelope. And there are no significant differences among the other methods. The sequential swipe

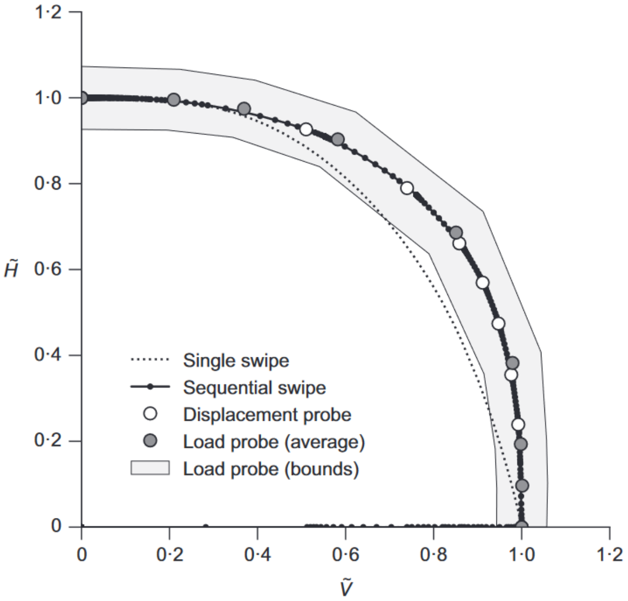


Figure 3.5: Dimensionless VH failure envelopes of caisson foundation(Suryasentana et al., 2020)

test shows better performance than the widely used displacement probe test in terms of accuracy and efficiency. But the sequential swipe method obtain the ultimate state of the foundation instead of the whole loading path. It is not conducive to analysing the behaviour of the foundation. And load probe test cannot guarantee the fast converge of model every time. Therefore, the displacement probe test was chosen as the determining method for the failure envelope(Bransby and Randolph, 1998; Suryasentana et al., 2020). Meanwhile, the combined response of suction caisson (in Figure 3.6) can also be obtained by means of displacement-controlled probe test, which is more accurate and controllable compared to force-controlled probe test. This sort of probing experiment employs displacement as the independent variable (input parameter) and response force as the dependent variable (output parameter). The purpose of using deep learning method in Chapter 4 is to mimic this one-to-one relationship between force and displacement and to identify the mechanisms involved. The foundation's complete behavior under load will be represented by a loading path generated during a probe test. In contrast to conventional two-dimensional probe test, this model concurrently probes horizontal displacement, vertical displacement, and rotation, enabling them to represent a variety of displacement combinations. Therefore, a spherical co-ordinate system is used to establish the displacement combinations in three dimensions, which provides a better representation of the ellipsoidal surface compared to the traditional Cartesian coordinate system. After determining the radius, each point on the sphere will be indicated with only two parameters θ and ϕ. θ is the positive angle to the x-axis when rotating counterclockwise and ϕ is the positive angle to the z-axis when rotating clockwise. For a better understanding, consider θ to be the earth's longitude and ϕ to be the earth's latitude (as shown in Figure 3.7). After weighing the computational time against the size of the data set, 96 probe test directions for each depth of embedment were established. This means that θ take 15°, 45°, 75°, 105°, 135°, 165°, 195°, 225°, 255°, 285°, 315°, 345° respectively (30 degree increments for a total of 12 angles). ϕ takes 15°, 30°, 60°, 75°, 80°, 85°, 88°, 90° respectively (a total of 8 angles). θ and ϕ should be equally spaced to cover the entire computation range. However, when spherical coordinate system is converted to Cartesian coordinate system, the z ($\rho * \cos\phi$) values are distributed unevenly. To better capture the failure pattern at smaller vertical displacements, ϕ distribution will shift from sparse to dense after 80°.

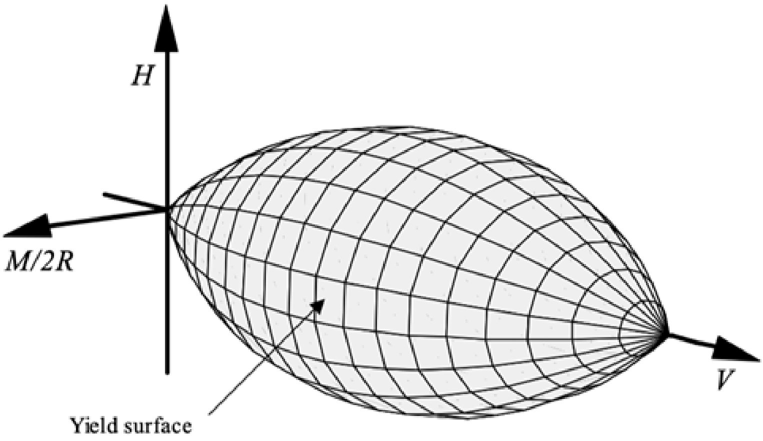


Figure 3.6: The typical shape of failure envelope under combined loads

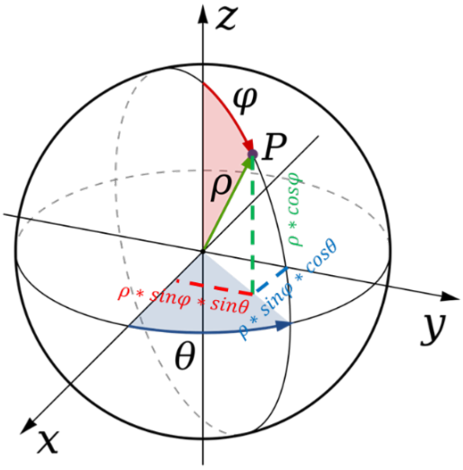
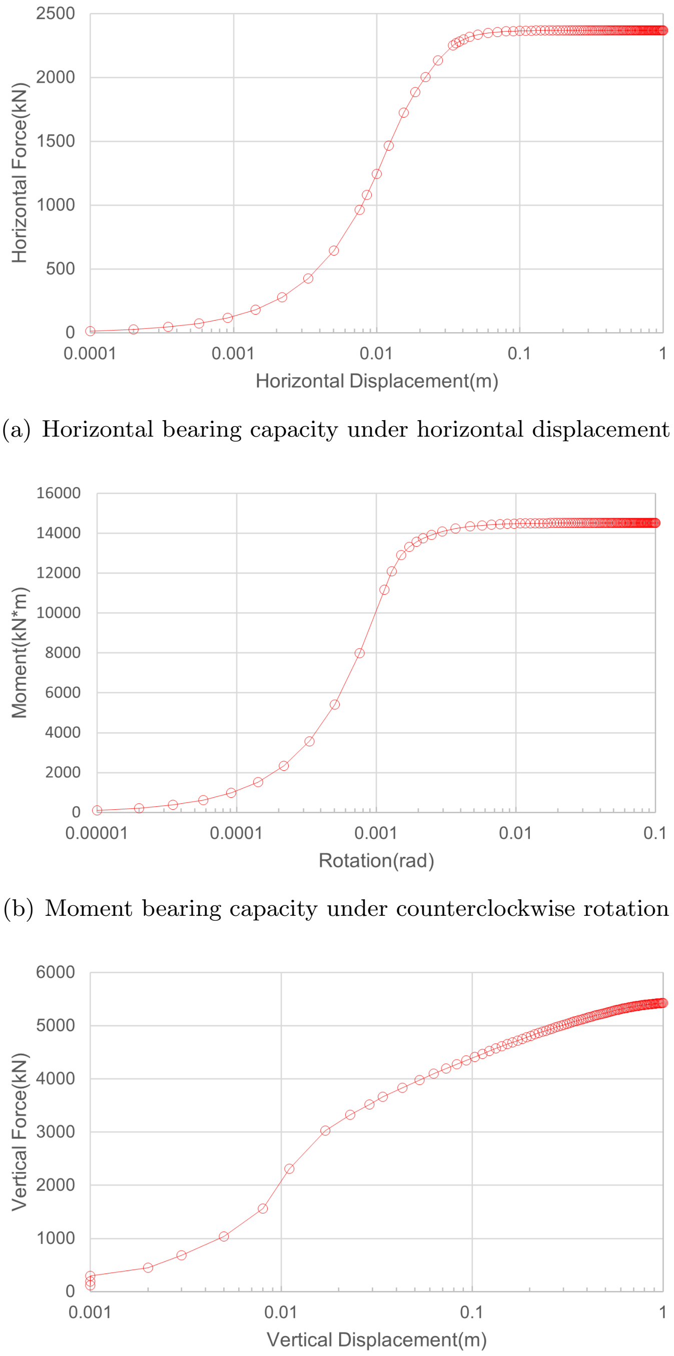


Figure 3.7: Spherical coordinate system

Determination of ultimate displacement

As the ultimate bearing capacity of H, V, M is different, the displacements required to mobilize the ultimate bearing capacity are also different. It is necessary to first study the required ultimate displacements for the H, V, M bearing capacity, separately, which will be used as reference values for the inputs of probe test simulations. In the case of the H (horizontal force) under $L/D = 1$ burial depth, just the horizontal displacement is set, while the displacements in the other two directions are equal to 0.



(a) Horizontal bearing capacity under horizontal displacement

(b) Moment bearing capacity under counterclockwise rotation

(c) Vertical bearing capacity under vertical displacement

Figure 3.8: The ultimate bearing capacity of 10m buried caisson

In Figure 3.8(a), the entire loading process is represented by a displacement-force curve. By presenting horizontal displacements in semi-log space, it is evident that the bearing capacity of the foundation is fully activated at 0.1 m. To guarantee that the horizontal ultimate bearing capacity of the foundation will be mobilized at varying burial depths, the horizontal displacement limit has been set to 0.2 m. Similar to the horizontal displacement-force curve, the rotation-moment curve for θ-M (shown in Figure 3.8(b)) demonstrates that the moment will not increase after 0.01°. Therefore, the ultimate rotation is set to be 0.02° to guarantee that the moment capacity will be completely mobilized in the subsequent numerical simulation. The third v-V curve(shown in the Figure 3.8(c)) demonstrates a steady increase across the range of the setup and a drop in the growth rate at 0.02 m. Therefore, in all the simulations of this study, the ultimate value for vertical displacement will be set to 1 m. The three ultimate bearing capacities of the caisson with an embedded ratio of 1 and in homogeneous clay with s_u of 10 kPa and the meanings of the force and displacement symbols are shown in Table 3.3.

Table 3.3: Summary of notation for loads and displacements

	Vertical	Horizontal	Rotational
Load	V	H	M
Ultimate load	$V_{\mathrm{ult}} = 5500kN$	$H_{\mathrm{ult}} = 2300kN$	$M_{\mathrm{ult}} = 14500kN * m$
Dimensionless load	V/As_{u0}	H/As_{u0}	M/ADs_{u0}
Bearing capacity factor	$N_{\mathrm{c}V} = V_{\mathrm{ult}}/As_{\mathrm{u}}$	$N_{\mathrm{c}H} = H_{\mathrm{ult}}/As_{\mathrm{u}}$	$N_{\mathrm{c}M} = M_{\mathrm{ult}}/ADs_{\mathrm{u}}$
Displacement	v	u	θ

Note: where $A = \pi D^2/4, s_{\mathrm{u0}}$ is undrained shear strength at mudline level.

To simplify the calculations, the set ultimate displacements in all directions and all embedded depths are consistent with the ultimate displacement for $L/D = 1$ which has been mentioned before. Figure 3.9 displays the ultimate 3D displacements under 96 directions, with the arrow indicating the loading direction of the suction caisson. These 96 directions are plotted as a hemispheric envelope. Because in this project only the compressive bearing capacity of the shallow foundation will be evaluated, not its pull-out resistance. Therefore, simulations indicating the pull-out resistance (negative direction of v) are not considered.

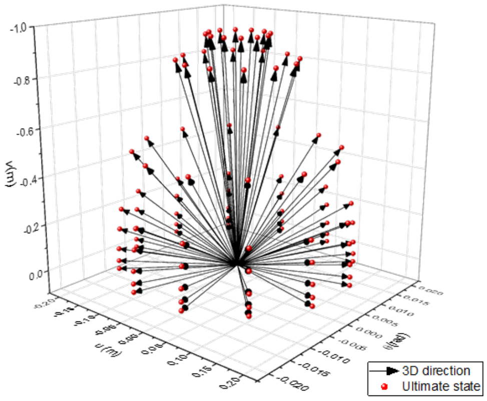


Figure 3.9: Ultimate displacement values in 96 directions (created by 8 latitudes and 12 longitudes combination)

3.4 Suction caisson response in homogeneous soil

In the simulation, the embedment ratio was considered instead of specific combinations of D (foundation diameter) and L (embedment depth) to accommodate as many feasible embedment depths combinations as possible while ignoring the impact of the foundation's dimensions. The embedment ratios are split into 10 equal parts from 0 to 1, while embedment ratios larger than 1 and no-skirt foundations (embedment ratio = 0) are not taken into consideration. The loading paths in each direction are represented by 100 data points under 96 loading directions for each embedment ratios. And each loading path experiences a significant shift after touching the failure envelope. A total of 96,000 (10*96*100) deep learning data points were generated and dimensionlessly processed, shown in the Figure 3.10 below and in Appendix A.0. Data points at different embedment ratio are distinguished by different colours. The response force is continuously mobilised with increasing displacement in each direction. The bearing capacity will be fully mobilised when the loading path touches the failure envelope.

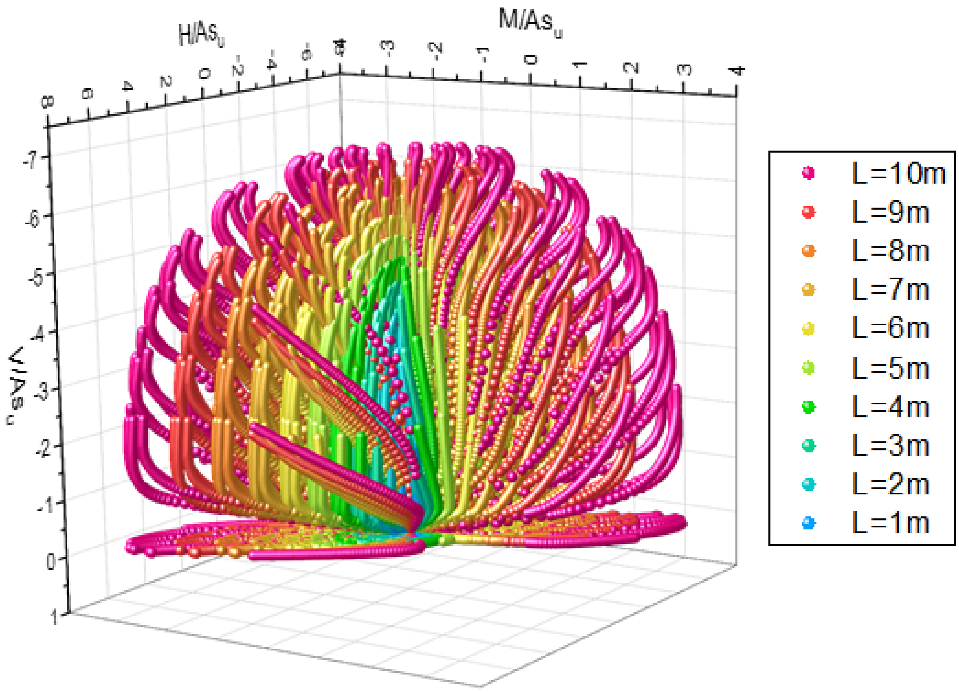
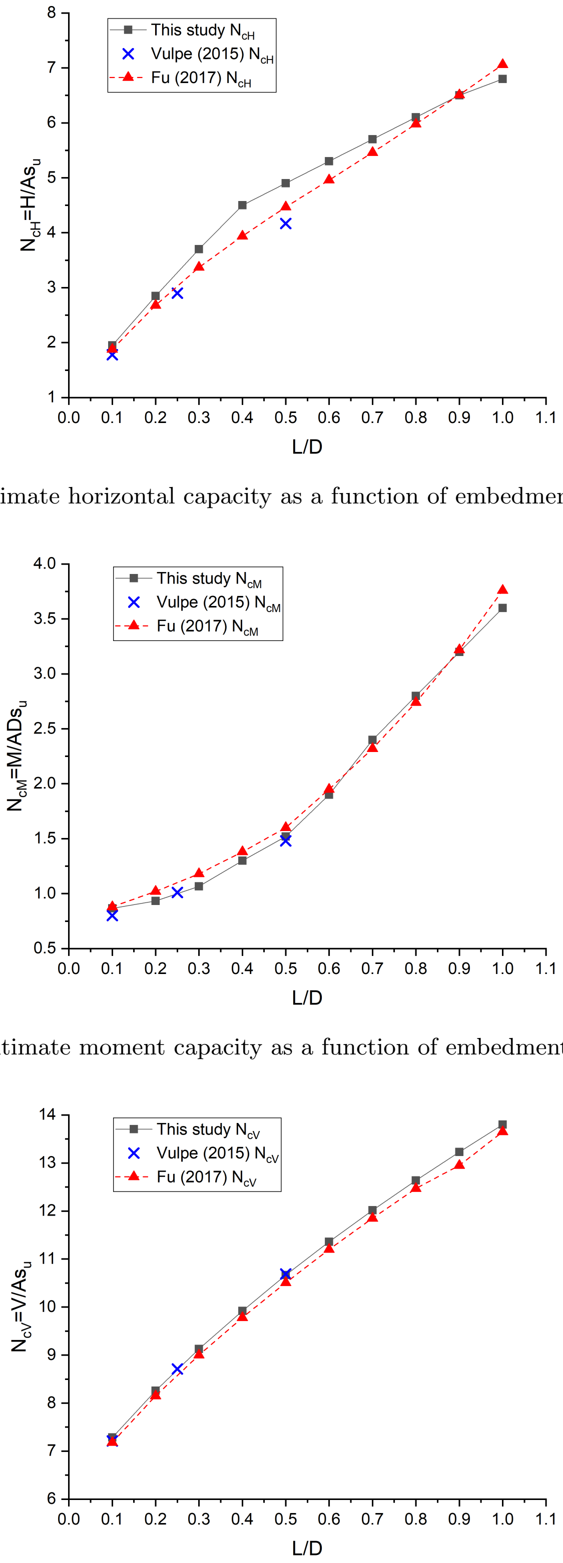


Figure 3.10: All data points in ten embedded depths

3.4.1 Bearing capacity

Figure 3.11 shows the variation of three bearing capacity factor N_{cV},N_{cH},N_{cM} (The exact formula is shown in the Table 3.3) as a function of embedment ratio L/D. Predicted bearing



(a) Ultimate horizontal capacity as a function of embedment ratio

(b) Ultimate moment capacity as a function of embedment ratio

(c) Vertical bearing capacity as a function of embedment ratio

Figure 3.11: Ultimate bearing capacity as a function of embedment ratio

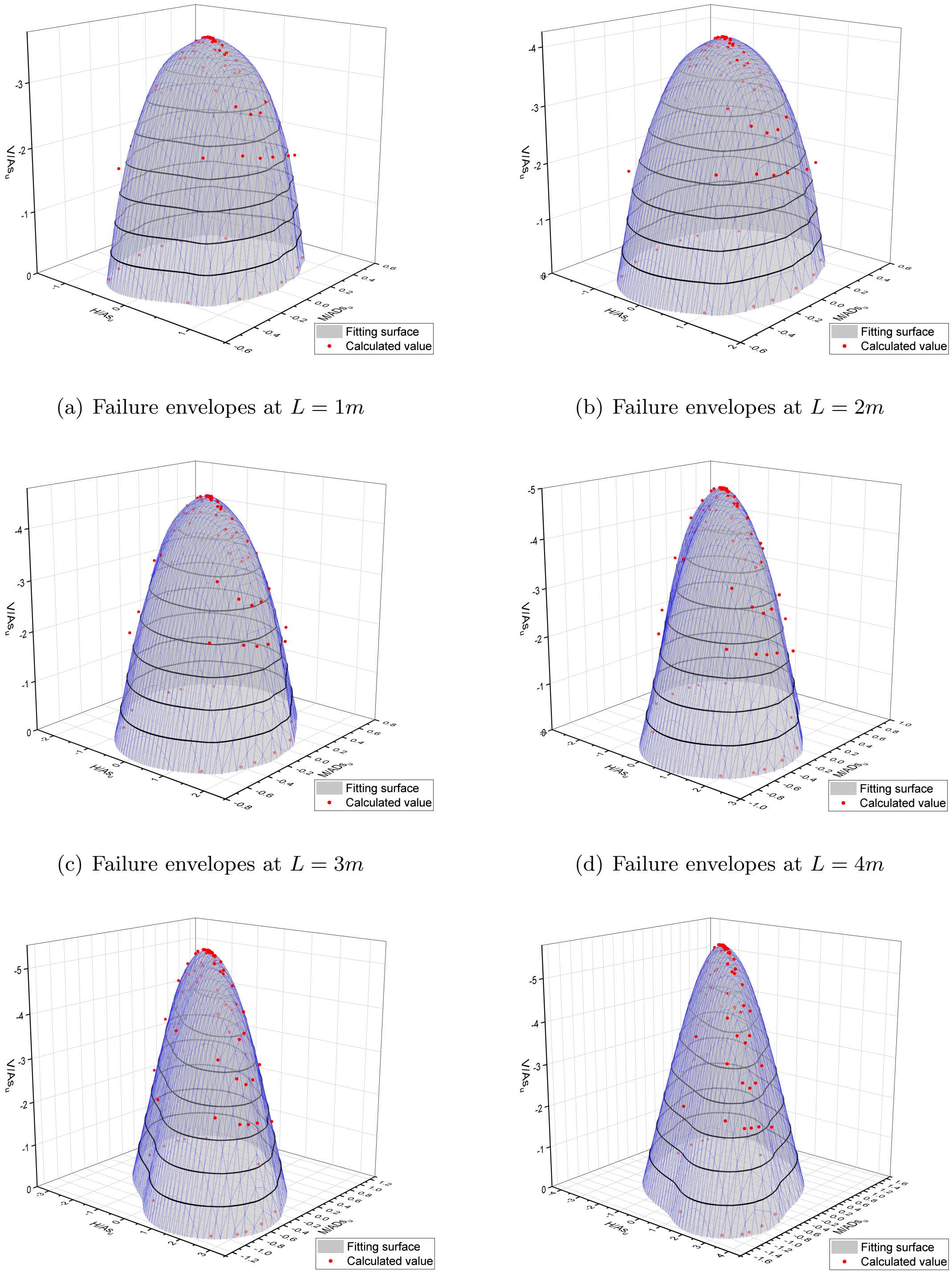
capacities in this study are compared with ultimate bearing capacity from analytical and numerical solutions supported by Vulpe (2015) and Fu et al. (2017).

As observed in Figure 3.11, the pattern of the variation in bearing capacity with embedment ratio produced from this simulation is consistent with the estimates of Vulpe (2015) and Fu et al. (2017). In homogeneous soil, the variances with vulpe's published numerical values for vertical, horizontal, and moment bearing capacities are within 0.6%, 15%, and 8%, respectively. And it is in good agreement with Fu's data, for vertical, horizontal, and moment capacities are within 1.3%, 4.4%, and 6%, respectively. The variance in bearing capacity is mostly attributable to the different positions chosen for the reference points during the modelling. Different reference points result in distinct failure mechanisms, which are primarily responsible for disparities in H and M. However, these slight differences is acceptable, demonstrating the model's validation.

3.4.2 Effect of embedment on failure mechanisms

Once the bearing capacity is determined, the size of the failure envelope is determined. Based on the ultimate data points at each burial depth, ten initial ellipsoidal envelope were sketched. The envelope is depicted using 96 data points in 96 directions extracted from 9600 data at each embedment ratio. All 96 data points lie on the envelope and can represent the ultimate state of the loading paths. Origin 2018 was used to plot the H-V-M envelope. The red points on the graph reflect the calculation points, whereas the grey surface with blue mesh represents the fitted envelope (plotted in Figure 3.11). All fitted surfaces are not ideal ellipsoids and have a few folds in the middle. Theoretically all envelopes should be convex. The concave surface appearing in the figure is the fitting error caused by the uneven distribution of points. But it is still approximated as an ellipsoidal surface. In all 10 figures, the same viewpoint is used. On successive views the failure envelope is seen to be 'rotating'. These trends will be analysed by splitting the H-V-M envelope into three 2-dimensional envelope (H-M, H-V, M-V) respectively. This is shown in Figure 3.12 below.

By modelling the response of the foundation at $\phi = 90°$ (which means V=0) for different embedment depths, the interrelationship between dimensionless H and M is shown in Figure 3.13. Each envelope represents result in an embedding depth and has twelve red hollow points representing twelve loading directions. The H-M envelope is asymmetrically elliptical at all embedment depths. It is eccentric to the negative side of the horizontal axis when $\mathrm{H/As_u} > 0$ and eccentric to the positive side of the horizontal axis when $\mathrm{H/As_u} < 0$. It has the opposite direction of eccentricity compared with the H-M failure envelope, in which the reference point is set at the tip level (Bransby and Yun, 2009). And when the reference point is set at skirt tip level, the effect of embedment depths is less significant for the shape of H-M envelope. These two reference point configurations have minimal impact on V and H. However, when the H applied at the tip level is equivalently translated to the mudline, a counterclockwise additional overturning moment is generated. The maximum moment is generated from rotation and horizontal forces on the reference point. The additional overturning moment balances the clockwise moment applied at the reference point. This results in the $M_{\max}$ (maximum moment bearing capacity) requiring the interaction of negative horizontal loads and $H_{\max}$ (maximum horizontal bearing capacity) also requiring



(a) Failure envelopes at $L = 1m$

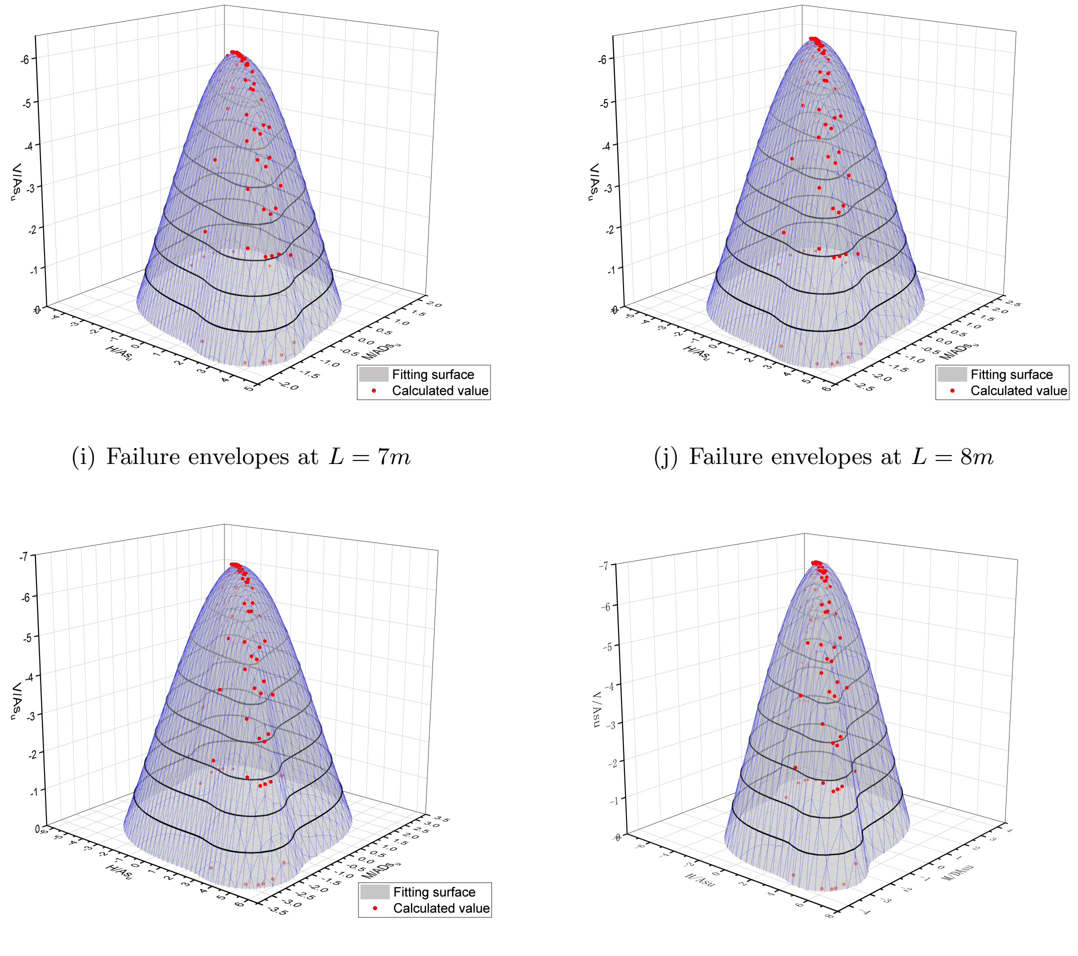
(b) Failure envelopes at $L = 2m$

(c) Failure envelopes at $L = 3m$

(d) Failure envelopes at $L = 4m$

(e) Failure envelopes at $L = 5m$

(f) Failure envelopes at $L = 6m$



(i) Failure envelopes at $L = 7m$

(j) Failure envelopes at $L = 8m$

(k) Failure envelopes at $L = 9m$

(l) Failure envelopes at $L = 10m$

Figure 3.11: 3D failure envelopes at $L = 1 - 10m$ calculated using FE analysis

the interaction of counterclockwise moments. Meanwhile, the additional overturning moment produced by H steadily grows as the depth of burial increases, resulting that the max moment develop more quickly than the max horizontal force. This interrelationship explains the gradual 'rotation' of the fitted surface as the depth of embedment changes in Figure 3.11. The failure envelope enlarges with increasing embedding depth. This phenomenon indicates that both the bearing capacity and the shape of the failure envelope are highly dependent on the embedment ratio. This is congruent with the results from Gourvenec (2008) and Bransby and Randolph (1999).

When comparing with the H-M Failure envelope (Figure 3.14) at $L/D = 0.1$ and 1, it is evident that the eccentricities of the two envelopes point in opposite directions. This difference is attributed to a change in the failure mechanism. When the embedded depth is altered, directions $\theta = 75°$ and $\theta = 255°$ experience a shift between positive and negative in horizontal force. The loading in $\theta = 75°$ and $\theta = 255°$ is rotationally symmetrical. Therefore, the vector diagrams for $\theta = 75°$ direction was selected for analysis.

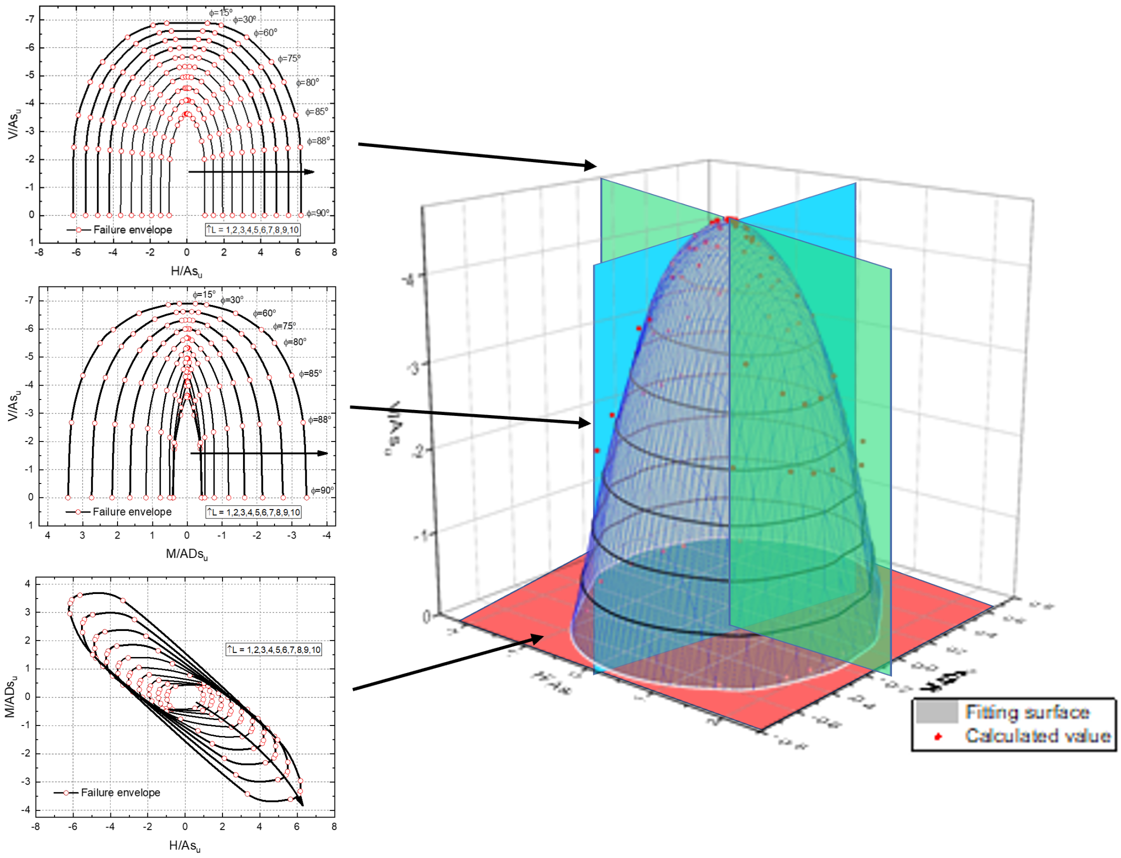


Figure 3.12: 3D failure envelopes at $L = 1 - 10m$ calculated using FE analysis with slice (H-M slice in red, H-V slice in green and M-V slice in blue)

The vector diagram for $L/D = 0.1$ (Figure 3.15(a)) shows that the failure mechanism can be regarded as a scoop-slide mechanism (shown in Figure 3.16), with the failure concentrated in the sheared wedge at mudline. And a primary scoop mechanism can be identified for the $L/D = 1$ foundation, with no wedge generate at the mudline in Figure 3.15(b). The tip of the skirt is particularly vulnerable to shear force and the effect of scoop mechanism expands. The rotational center is considerably lower than the position of the reference point. This indicates that the rotational center of the scoop mechanism also shifts downwards as the depth of skirt increases. Therefore, the horizontal forces around the foundation at mudline under the two embedded depths are in opposite directions, resulting in positive and negative alternated horizontal forces in $\theta = 75°$ and $\theta = 255°$ directions.

When comparing with the Bransby and Yun (2009)'s study, an interesting phenomenon can be observed. The vector diagram for $L/D = 1$ in this experiment is similar to the vector diagram for the solid foundation of $L/D = 1$ in Bransby's experiment, where the scoop failure mechanism is prevalent and the internal processes of the skirt are not readily apparent. And the vector diagram for $L/D = 0.1$ is similar to the vector diagram for the solid foundation of $L/D = 0.1$ in Bransby's experiment. Both of them consist of the

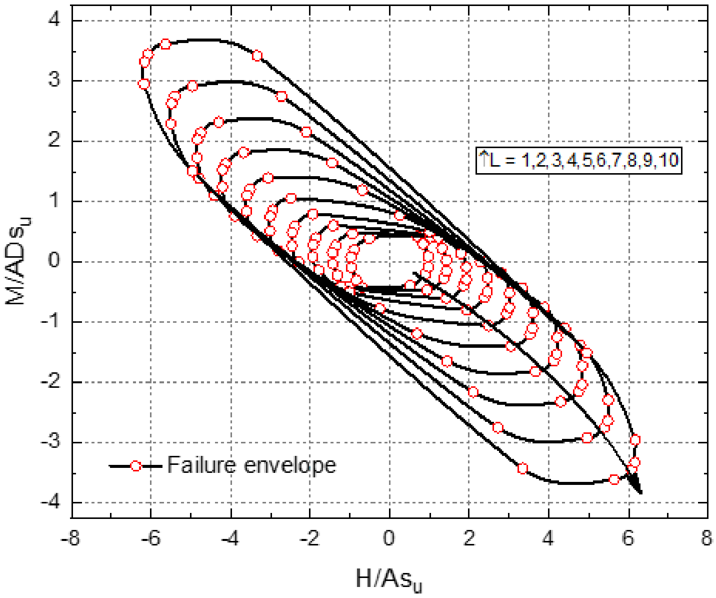
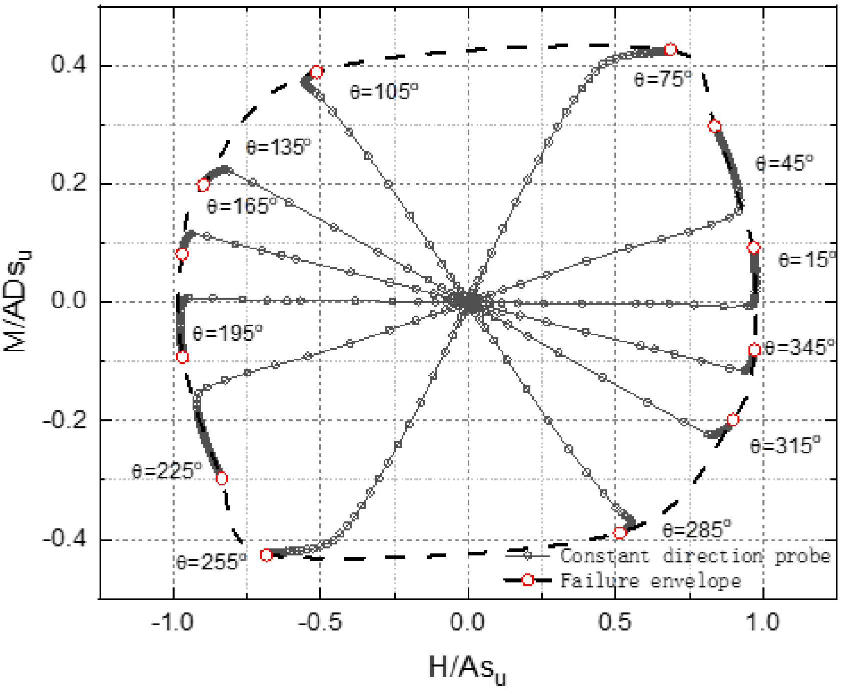


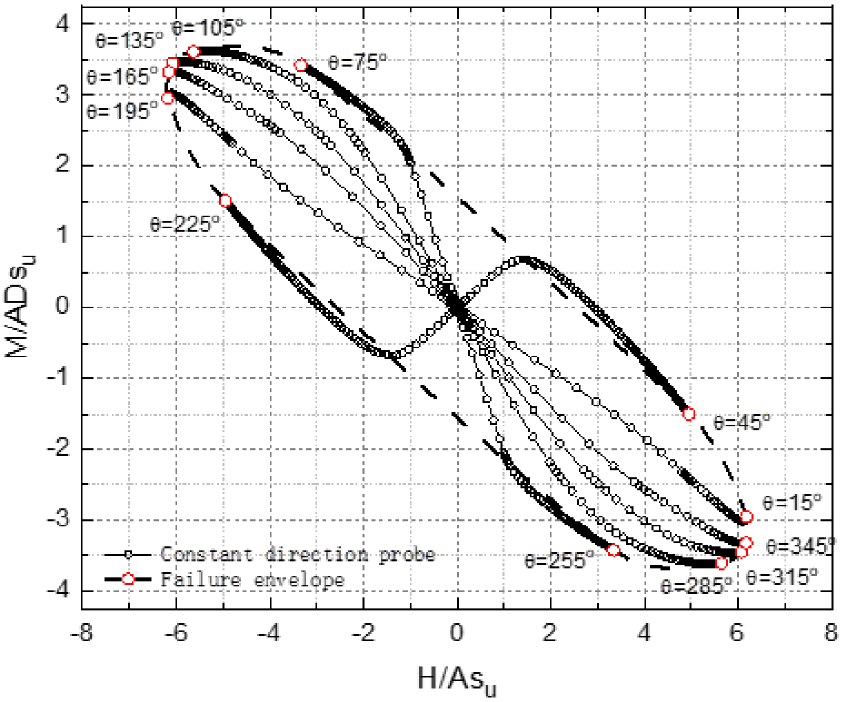
Figure 3.13: Comparison of failure envelopes of skirted circular foundations in different embedment predicted by FE analyses (H-M)

scoop-slide mechanism without the internal Hansen's failure mechanism. This phenomenon indicates that the setting of reference point at the mudline leads soil within the skirt to move in tandem with the caisson and skirt's unique internal failure mechanisms are not clearly evident. The suction caisson can be considered as a solid foundation under this assumption.

In Figure 3.17 and Figure 3.18, According to the formula for vertical force ($V = \rho * \cos\phi$) ϕ reflects the value of V with an inverse correlation. When $\phi = 90°$, it means that $V = 0$, and the H-M relationship will be illustrated. When $\phi = 0°$, the envelope's maximum vertical bearing capacity will be demonstrated. Furthermore, by increasing embedment ratios the failure envelopes expand, which confirms the effect of the embedment depth on increasing the load bearing capacity again. In Figure 3.17, the H-V envelope increases linearly with increasing embedment depth. While in the M-V envelope (in Figure 3.18), an accelerated increasing trend of the ultimate vertical bearing capacity was observed with the increasing embedment ratio in every direction. However, the patterns indicate that such an increase is less pronounced for L/D $\geq$ 0.7, which is caused by the change in the failure mechanism, consistent with Mehravar et al. (2016) conclusion. It is also apparent that the shape of the H-M envelope is almost independent of V when combine Figure 3.17 and Figure 3.18. While envelope size becomes progressively larger as V decreases, eventually forming an ellipsoid. This pattern is effectively depicted in Figure. 3.11.



(a) Failure envelope (H-M) at $L/D = 0.1$



(b) Failure envelope (H-M) at $L/D = 1$

Figure 3.14: Failure envelope (H-M) at $L/D = 0.1$ and 1 using probe tests from FE analyses

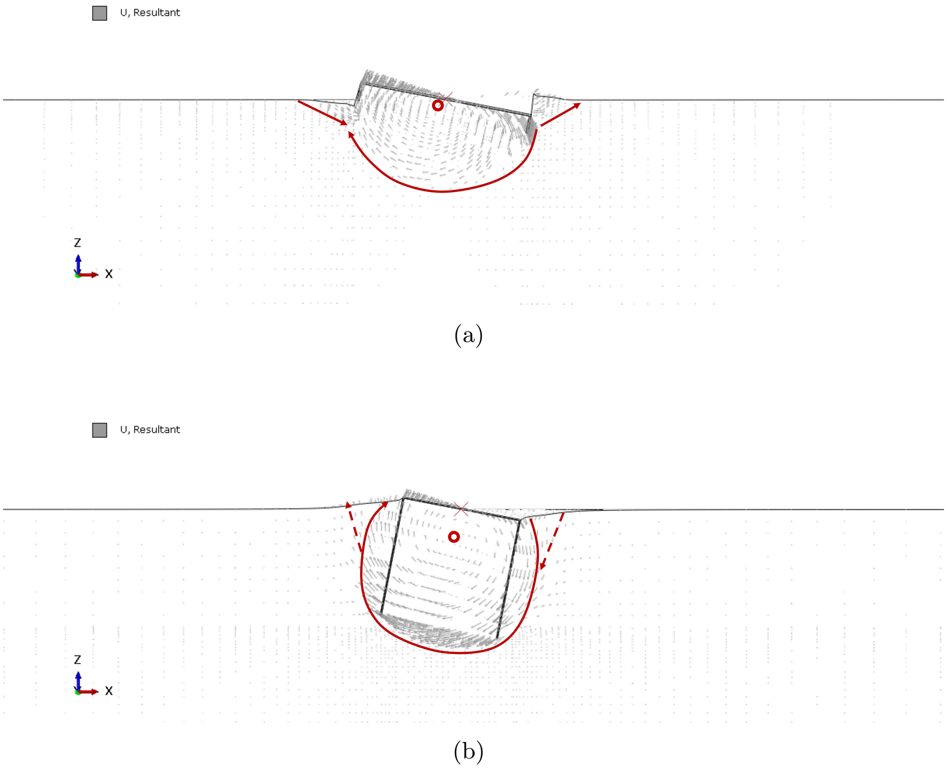


Figure 3.15: Vector diagram of $\theta = 75°$ and $\phi = 90°$ (a) $L/D = 0.1$,(b) $L/D = 1$

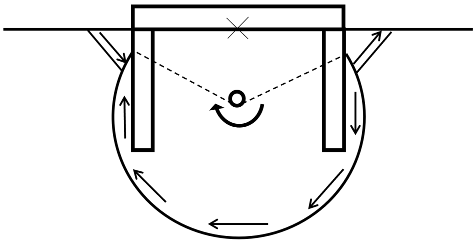


Figure 3.16: Scoop-slide failure mechanism

3.4.3 Effect of load combinations on failure mechanisms

The change in the failure mechanism is not only connected to the depth of embedment, but also to the direction of loading. By comparing the failure mechanism of foundation in this project with the conclusions of the references, the validation of the data can also be demonstrated. The affect of loading direction will be demonstrated in this part by three separate perspectives (H-M, H-V, M-V). The following vector diagrams are all plotted under $L/D = 1$.

Figure 3.19 depicts the vector diagram of the H-M envelope for each load combination, with the failure mechanism changing as the direction varies. After generalisation, the failure mechanism will be decomposed into two halves (up-scoop and down-scoop). The up-scoop is formed by the clockwise rotation generated by the positive moment, and the

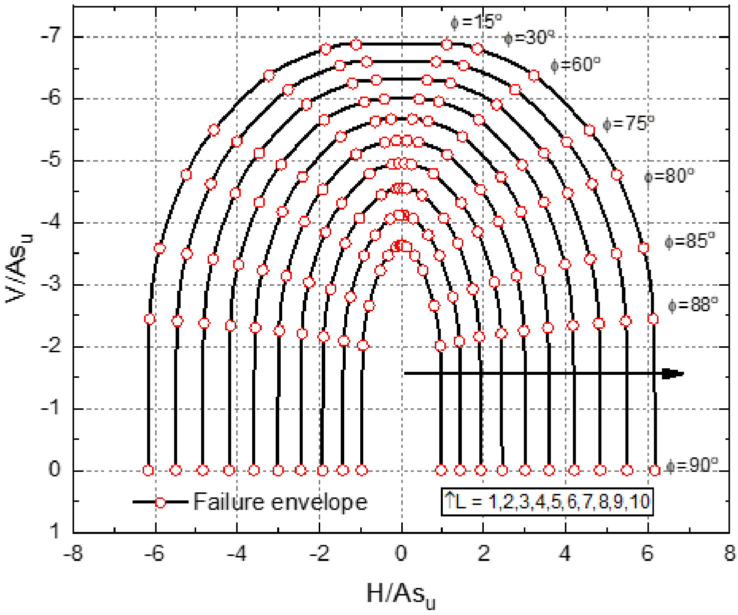


Figure 3.17: Comparison of failure envelopes of skirted circular foundations in different embedment predicted by FE analyses (H-V)

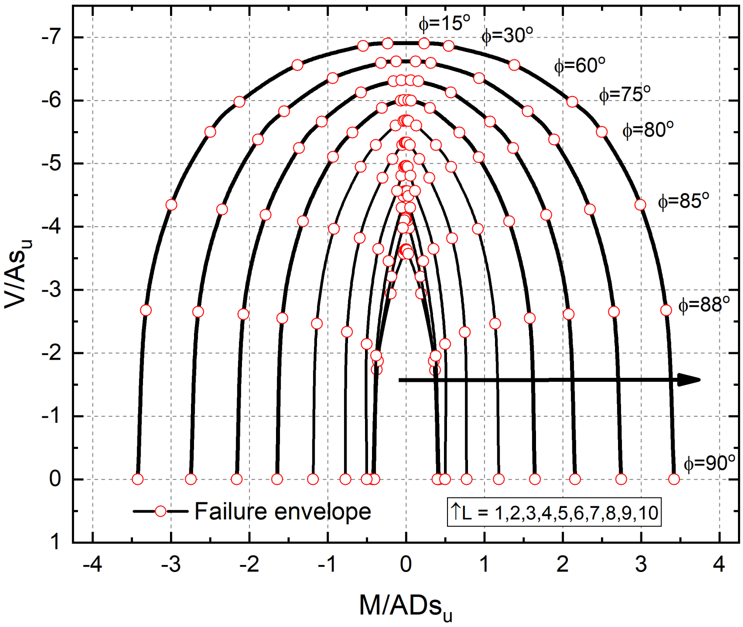


Figure 3.18: Comparison of failure envelopes of skirted circular foundations in different embedment predicted by FE analyses (M-V)

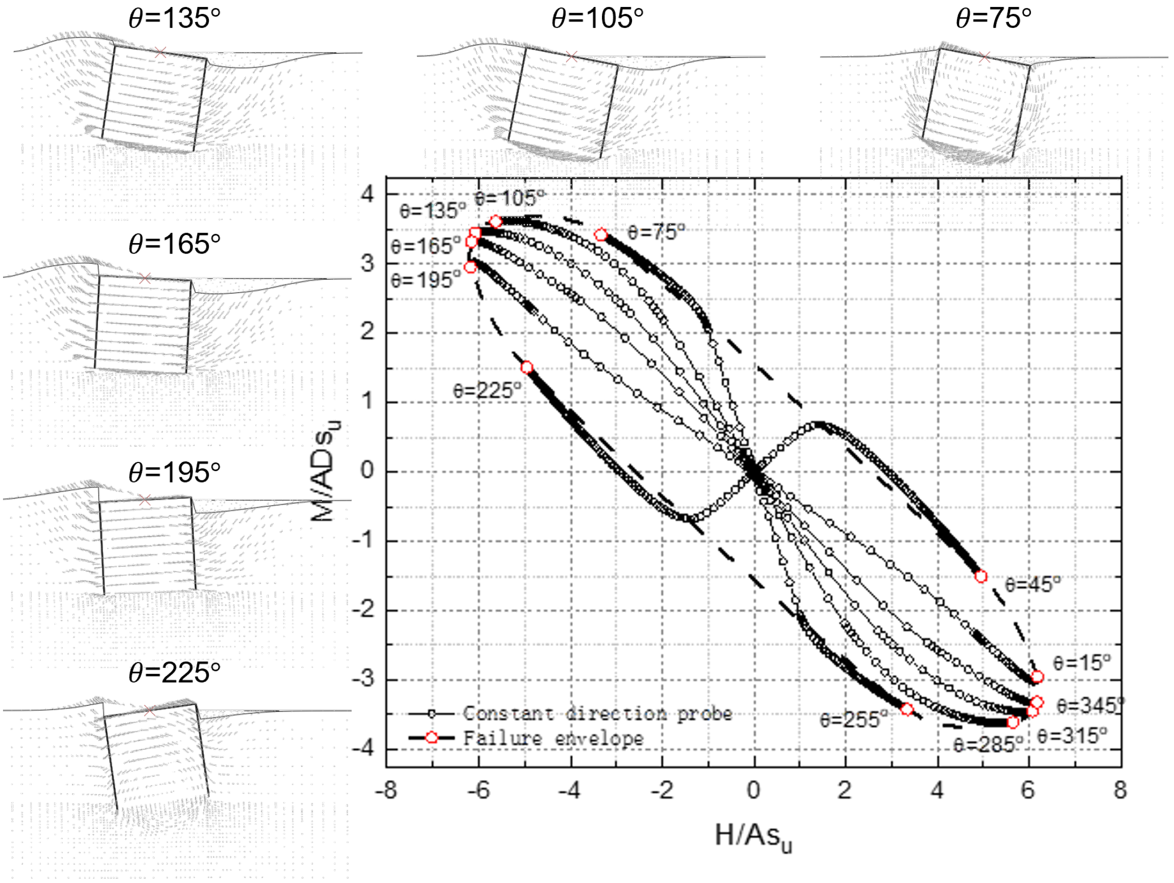


Figure 3.19: Effect of of load combinations on failure mechanisms under H-M loading ($\phi = 90°$; $L/D = 1$)

down-scoop is formed by the counterclockwise rotation generated by the negative horizontal force. These two scoops are in opposition to one another, but the up-scoop produces a much larger rotation than the down-scoop (shown in Figure 3.20). H_{max} occurs at $\theta = 165°$ when the vector is completely toward left and sliding is dominant, as seen in Figure 3.19. The process of foundation failure is dominated by shear failure at the base of the caisson, without soil sheared within the skirts. $\theta = 165°$ is also an equilibrium point, where the maximum displacement is concentrated in the middle of the burial depth. As θ increases, the maximum displacement rises to the mudline and the maximum displacement falls to the tips when θ decreases. Meanwhile, M_{max} lies between $\theta = 105°$ and 75°, at this point the double scoop gradually degrades to one single scoop. The counterclockwise rotation of H is balanced by the predominant clockwise rotation of M, which can be ignored. The vector diagram at M (Figure 3.21(a)) illustrates that the centre of rotation drops to the minimum position (around $L/2$). When θ is larger than 165°, the foundation tilts to the left and the down-scoop begins to dominate the failure mechanism, constantly balancing the positive clockwise rotation generated by M. At $\theta = 225°$, the center of rotation of the down-scoop climb up as the value of θ increases. The centre of rotation rises to the tips in Figure 3.21(b), showing the opposite "vortex" to the Figure 3.21(a).

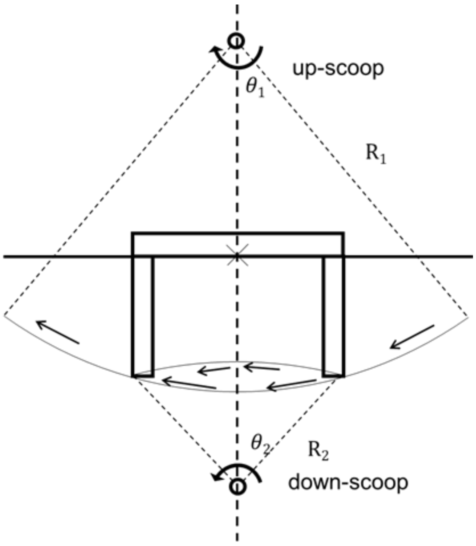
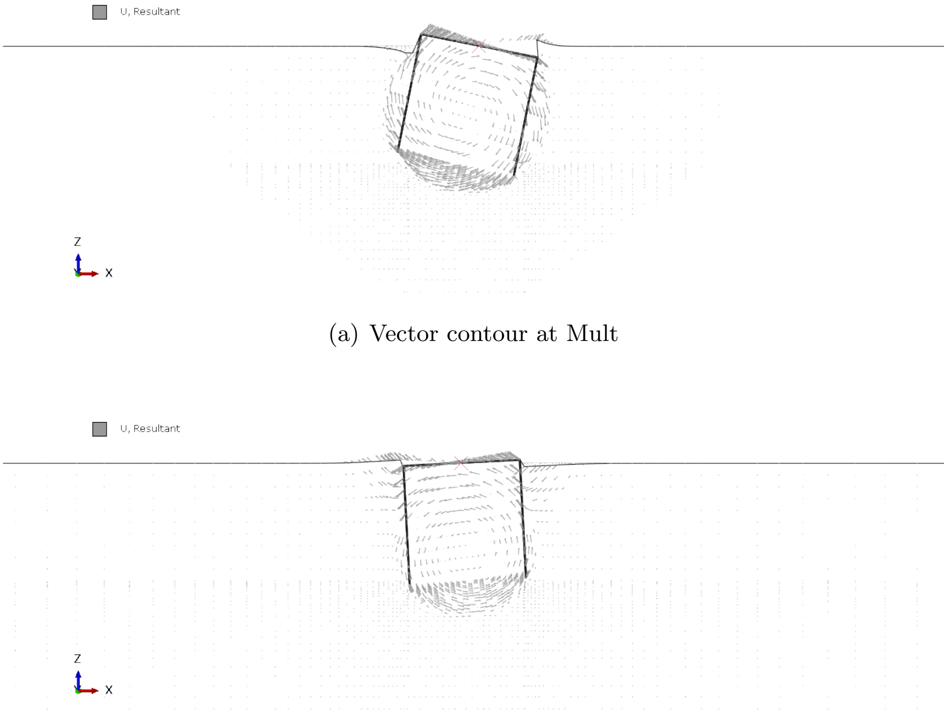


Figure 3.20: Double scoop failure mechanism



(a) Vector contour at Mult

(b) Vector contour at Hult

Figure 3.21: Vector contour in ultimate state

Figure 3.22 depicts the soil deformation mechanisms at several positions along the envelope, demonstrating that there are distinctly different mechanisms at different directions. At $\phi = 30°$, a double wedge mechanism is visible, with a small horizontal force to the right that does not conceal the fact that the soil is being pushed out in two Planter-type wedges. As ϕ grows, the value of V decreases and the value of H increases. In addition, the failure mechanism shifts progressively from piercing failure to slide failure. The symmetrical nature of the vector diagram transforms into an asymmetrical one. When $\phi = 88°$, the vector points downhill to the right as a result of the dominating horizontal force and the residual vertical force.

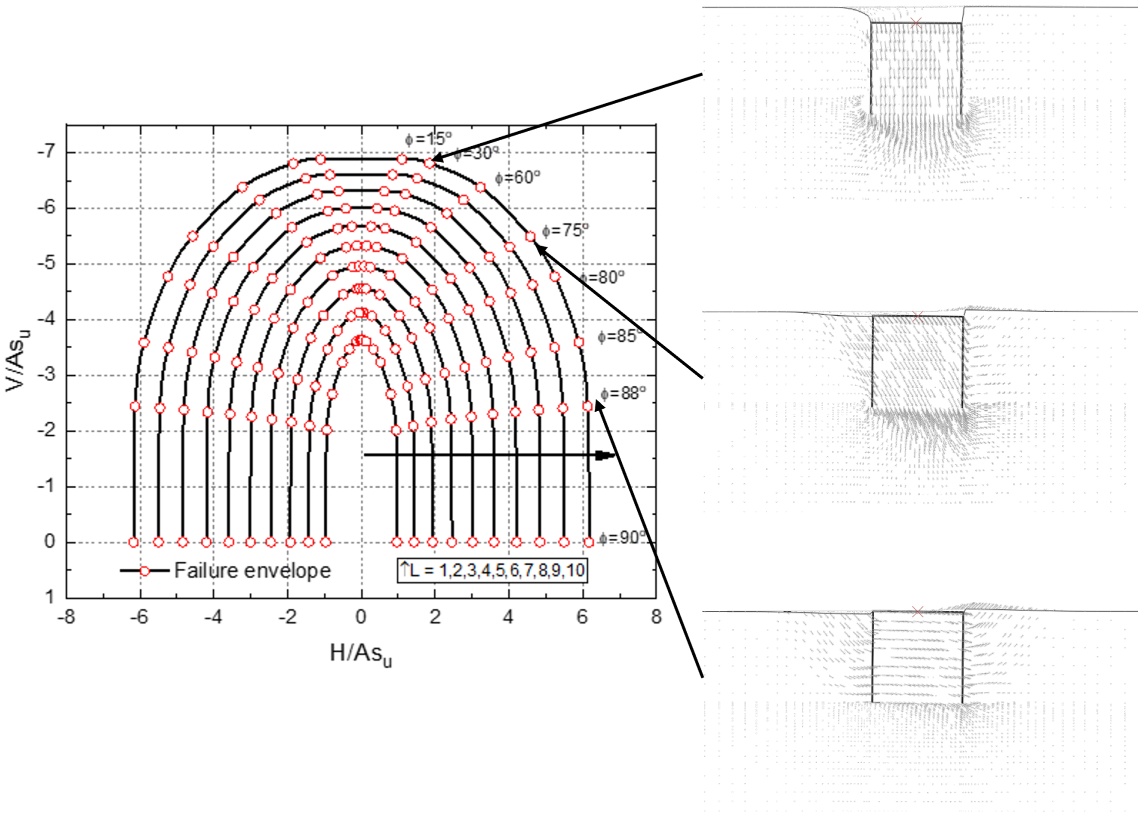


Figure 3.22: Effect of of load combinations on failure mechanisms under H-V loading ($\theta = 15°$ and 195°; $L/D = 1$)

When $\phi = 30°$, a basic symmetrical double-wedge mechanism occurred with small rotation, showing a similar pattern in Figure 3.23. The scoop-type mechanism dominated when $\phi = 88°$. Due to the existence of positive vertical load at this location, the centre of rotation of the scoop is slightly eccentric. Whereas in other load combinations, a combined scoop and slide mechanism predominated at intermediate positions.

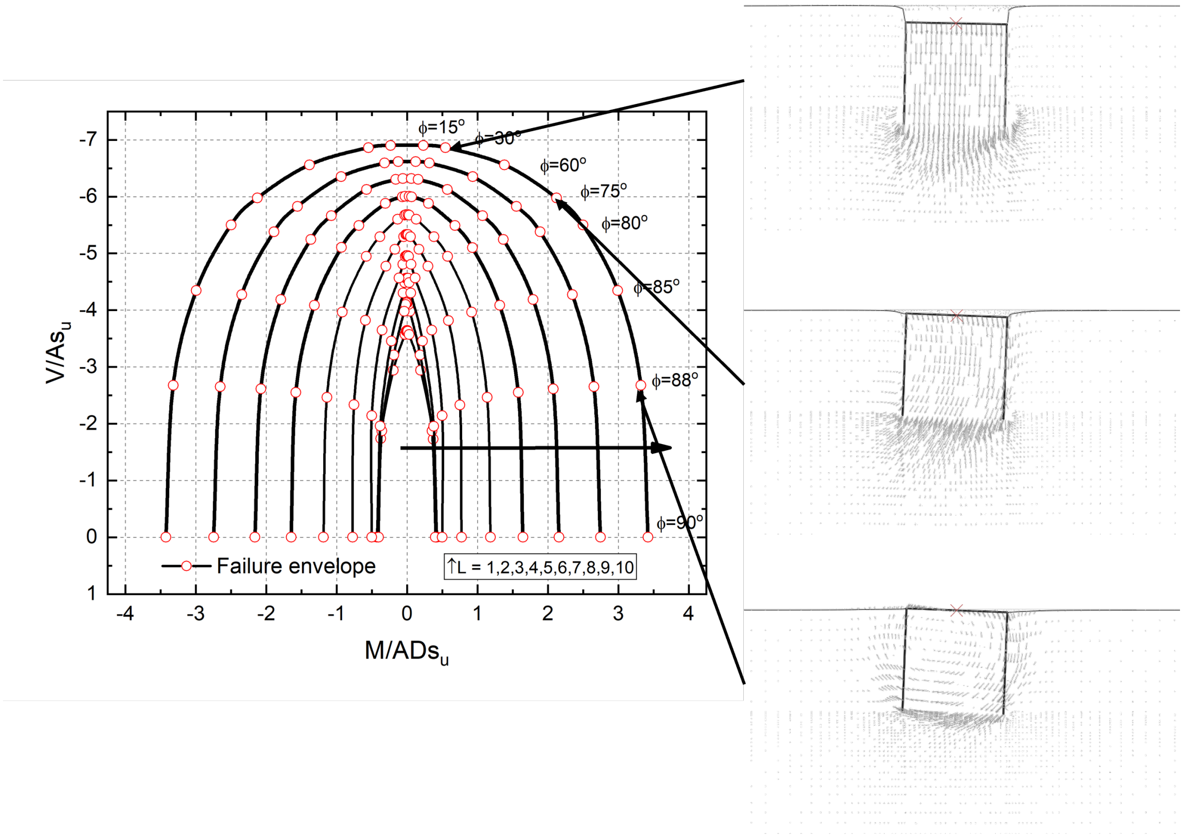


Figure 3.23: Effect of of load combinations on failure mechanisms under M-V loading ($\theta = 75°$ and $255°$; $L/D = 1$)

3.5 Suction caisson response in heterogenous soil

In heterogenous soil, the undrained shear strength of soil profile increased linearly with depth, rather than the uniform strength of homogeneous soils. And the undrained shear strengths are defined by the equation below:

$$s_{\mathrm{u}} = s_{\mathrm{um}} + kz \tag{3.1}$$

where s_{um} is the shear strength at the mudline and k is the shear strength gradient with depth z (Figure 3.24). The degree of soil strength heterogeneity can be expressed in terms of a dimensionless index as:

$$\kappa = \frac{kD}{s_{\mathrm{um}}} \tag{3.2}$$

Assume the shear strength gradient(k) to be 1 and the s_{um} is equal to 0.01. This assures that the shear strength of the heterogeneous soil at caisson tip is identical to in the homogeneous soil. The soil strength heterogeneity index (κ) is calculated to be 1000. In the heterogeneous soil simulation, the mechanical properties are the same as those for homogeneous soils, shown in Table 3.2 except soil shear strength. In the project, the deep learning algorithm was designed to identify the soil shear strength change or not, rather than identify the specific changes in soil strength. Therefore, the heterogeneous soil experiment data is used as supplementary, 96 directions were calculated at the every depth of embedment, which contains 12 θ values and 8 ϕ values (same as the experimental setup in the homogeneous soil).

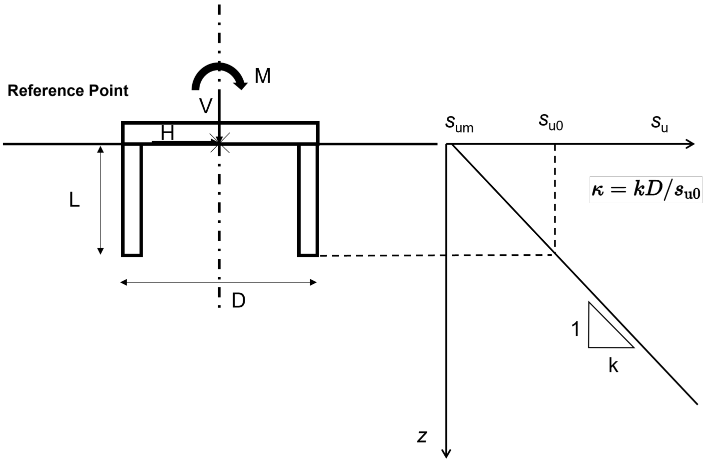
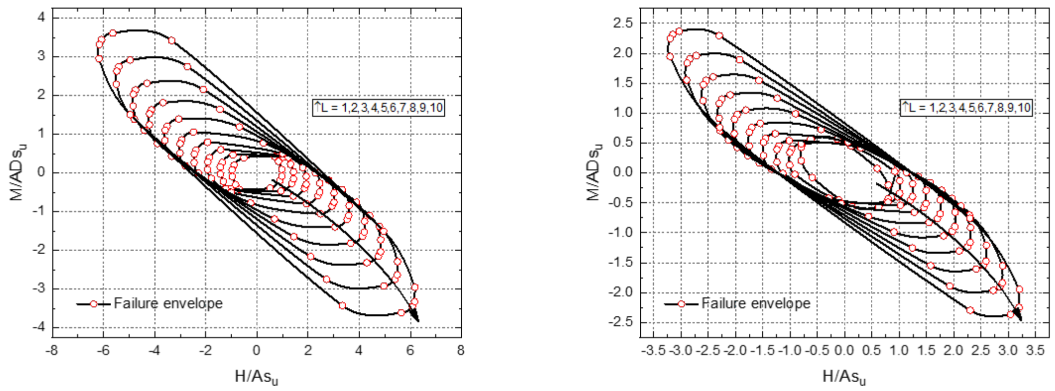


Figure 3.24: Soil shear strength profile

The following figures compare homogeneous and heterogeneous soils from the prospective of H-M plane(Figure 3.25), H-V plane (Figure 3.26) and the M-V plane (Figure 3.27) respectively. Homogeneous and heterogeneous soil plots show the same size change trend that the size grows when the embedment ratio increase. Furthermore, all envelopes demonstrate the influence of embedment depths and load combinations on the shape and size of the envelopes. In the H-V envelope, the size increases linearly with increasing embedment depth in both soil profiles. Bearing capacity in a specific ϕ in homogeneous soil increases with the embedment ratio, while an opposite trend shows in heterogeneous soil. This inverse phenomenon can also be observed in the M-V envelope. This phenomenon is attributed to the constantly changing heterogeneous soil shear strength at the foundation tip. As the depth increases, the moment bearing capacity increases exponentially in the M-V envelope. However, the shear strength of the heterogeneous soil is less than that of the homogeneous soil at all embedment depth, except at embedment depth $L/D = 1$, where the shear strength of both is the same. The variation in shear strength affects the horizontal and moment bearing capacity more significantly than the vertical bearing capacity. When vertical displacement is applied, the higher shear strength soil under the base of the caisson is mobilised. Therefore, the size of the envelope of the heterogeneous soils is smaller than that in the homogeneous soils.

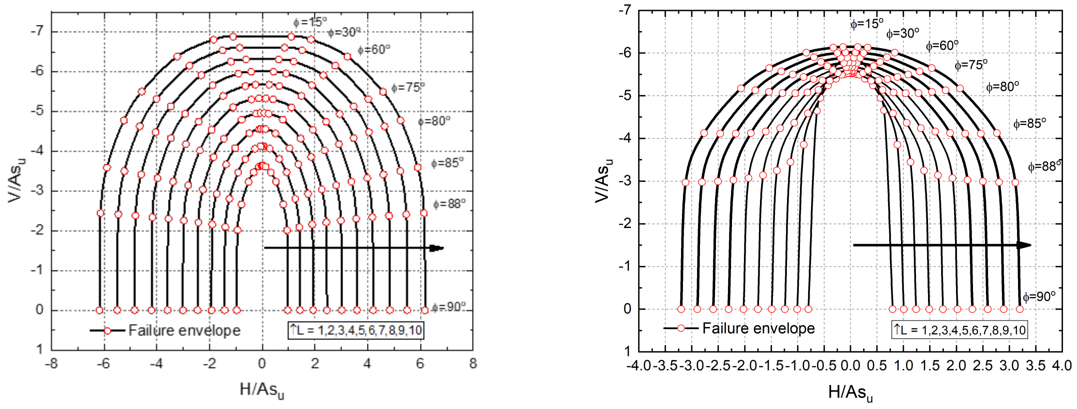
Moreover, it has been demonstrated that when the embedment depth decreases, heterogeneous soils exhibit the opposite eccentricity of the envelope as compared to homogeneous soils (Bransby and Yun, 2009). However, this phenomenon was not observed in this simulation. A reasonable explanation is the change in the failure mechanism originated from the difference in reference points. Although the skirted foundation was modelled, the actual failure mechanism is similar to that of the solid foundation. The internal double scoop mechanism reduce the foundation moment capacity significantly, as it involve shearing of shallower and weaker soil beneath the top cap for the normally consolidated soil conditions. For the solid foundation, internal Hansen mechanism and internal double scoop mechanisms are not admissible, and so the foundation capacity is much greater. This results that the phenomenon of negative eccentricity is not observed. This phenomenon further illustrates the insignificant effect of shear strength on the shape of the envelope when the reference point set at the mudline.



(a) Homogeneous soil

(b) Heterogenous soil

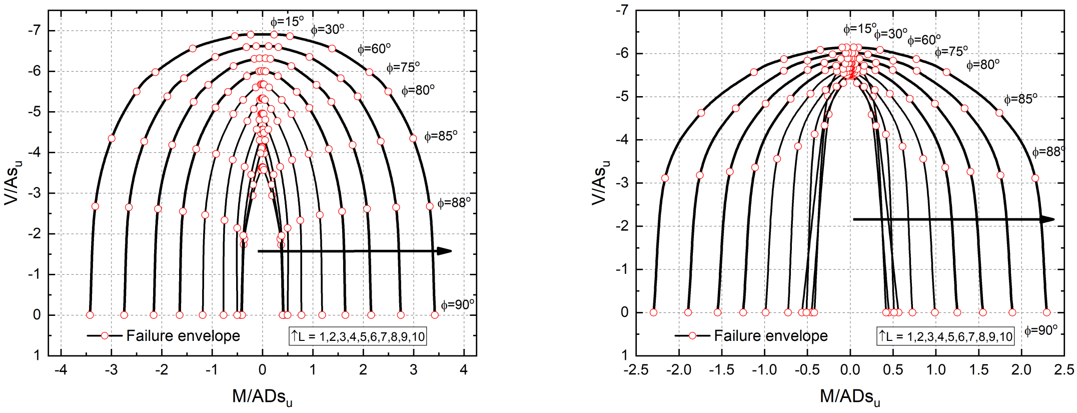
Figure 3.25: Comparison of H-M failure envelopes of skirted circular foundations in soil strength profile predicted by FE analyses



(a) Homogeneous soil

(b) Heterogenous soil

Figure 3.26: Comparison of H-V failure envelopes of skirted circular foundations in soil strength profile predicted by FE analyses



(a) Homogeneous soil

(b) Heterogenous soil

Figure 3.27: Comparison of M-V failure envelopes of skirted circular foundations in soil strength profile predicted by FE analyses

3.6 Section conclusion

This chapter explained how finite elements were used to simulate the response of foundations at various burial depths and load combinations in homogeneous and heterogeneous soils, respectively. To better simulate the mechanical response of suction caissons under various soil profiles and foundation configurations, 192,000 sets of computational data were established. The validity of the modelling data was then confirmed by analysing the bearing capacity and failure envelope and comparing them with existing researches. Additionally, the failure mechanisms of the foundations were further investigated. The transition of the failure mechanism with burial depth and load combination was analysed. A novel "double scoop" failure mechanism was presented when the reference point was set at the mudline. The 3D FE simulation data in this chapter would be used to train and test the deep learning based design model in Chapter 4.

4 Deep learning-based loading response prediction

4.1 Introduction

Training process
Considering its outstanding mapping ability, the deep learning is adopted in this study to predict the three-dimensional (3D) nonlinear load response of the caisson. All the models are implemented using Keras, a high-level deep learning toolkit based on Python. The entire training process is illustrated in Figure 4.1 below. The displacement values of horizontal displacement (u), vertical displacement (v) and rotation (θ), combined with the foundation embedment ratio (L/D) are input features of the model. These input values are transformed by randomly initialising the weight metrics (W) and biases (b) through each layer of the neural network to obtain the predicted values $\hat{H}$, $\hat{V}$, $\hat{M}$. The H, V, M obtained through FE simulations are used as labels for comparison with the predicted values. In each training, error value is calculated by mean square error (MSE) loss function or mean absolute error (MAE) loss function. The optimiser then revises the initialised weight metrics and biases in each neuron of the model reversely based on the error value, until a certain criterion is met and then stops adjusting. This process is referred to as 'training'. The updating of parameters can also be regarded as the 'learning' process of the neural network. The trained network can predict the load response of the caisson directly after inputting the new displacement and foundation configuration, thus skipping the cumbersome process of FE simulations. The main hyperparameters to be adjusted during the training process are marked as blue in Figure 4.1. The definition and selection of these hyperparameters will be explained in detail in Section 4.3.2.

Training design
In finite element simulations, the 3D response of suction caisson is obtained by multiple displacement probes from different directions. In deep learning, the unknown mechanical response can be predicted from the displacements at each moment or the limited known mechanical response in the past. The nonlinear regression predictions in this study are focused on the relationship between force and displacement. The 3D displacements are the input, and 3D forces are the prediction goal. In this approach, neural networks are employed to reproduce the finite element calculation process. The loading paths under each direction do not intersect before the ultimate state. It means that only one mechanical response can be mobilized for every displacement combination. This one-to-one relationship greatly facilitates the training of the neural network. However, this approach of training only focuses on the relationship between displacements and forces at one moment, ignoring the relationship between mechanical responses over different periods (past and future). The mechanical response at each moment forms an entire loading path, which can also be considered as a time series. The load response at the future moment heavily depends on the load response in the past. Therefore, in this study, two different training strategies are proposed: i) the load and deflection response at each moment is treated as an independent

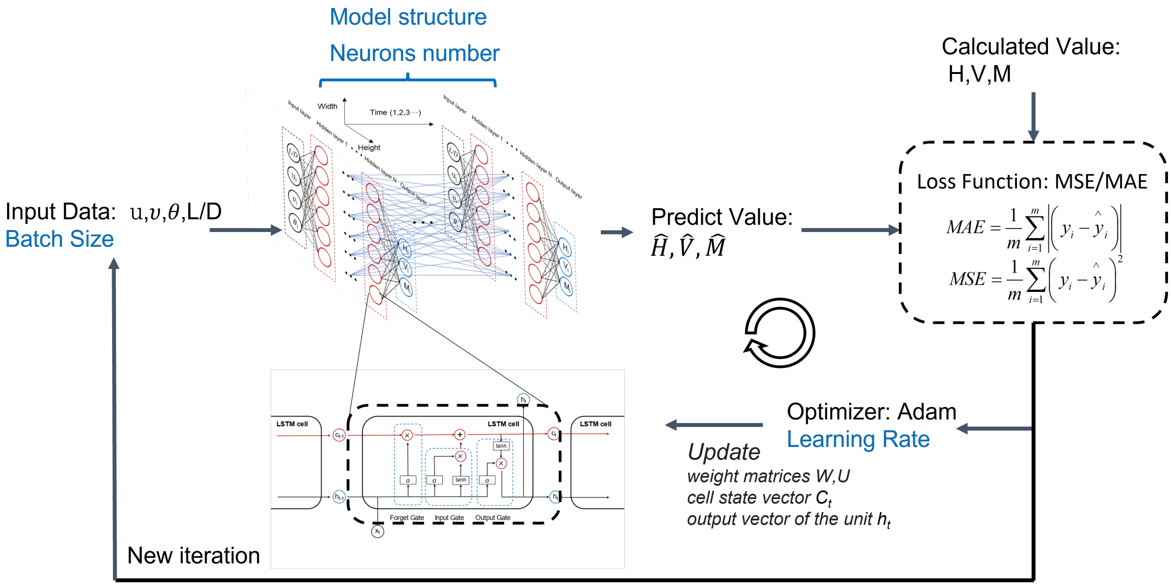


Figure 4.1: Constitutive modelling procedure using the LSTM deep-learning method (Hyper-parameters are shown in blue)

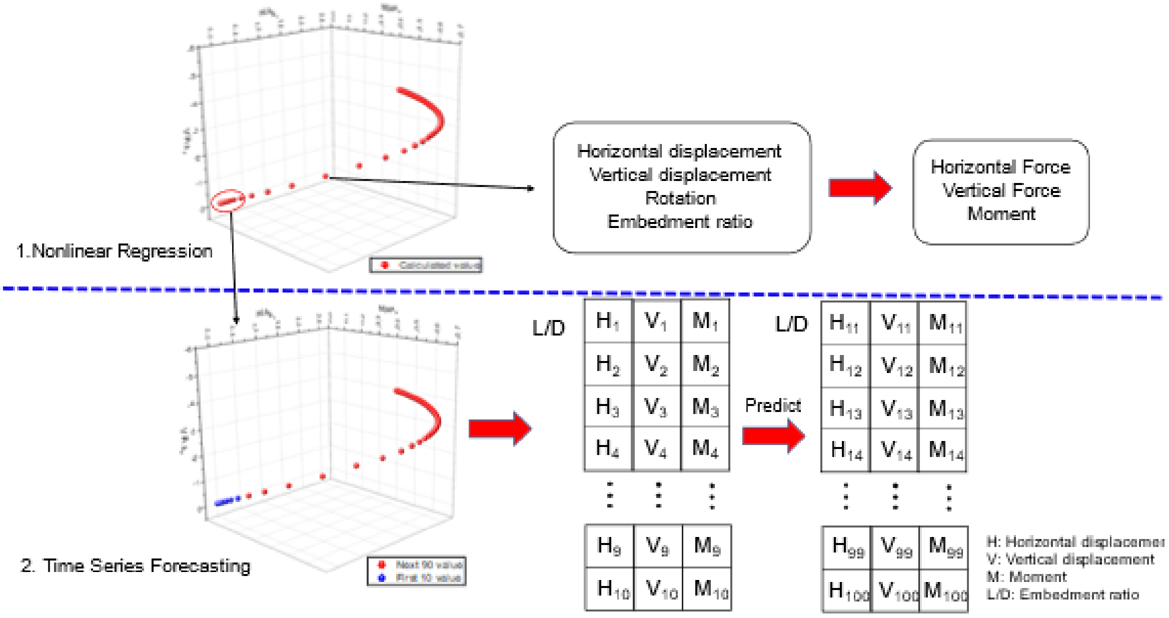
point, i.e. one-to-one prediction; ii) the load and deflection response at different moments of a fixed direction is treated as a time series and are dependent on each other, i.e. the temporal prediction. Therefore, these two primary training approaches are utilized to comprehensively study 3D response prediction for the suction caisson foundation in clay. A combination of three displacements at a particular state of the foundation configuration is used to predict the associated mechanical response at that moment (in Section 4.3). While the time series forecast for the subsequent ninety data points is achieved based on the initial ten data points along the loading path (in Section 4.4). The input and output of two primary training approaches are shown in Figure 4.2.

4.2 Methodology

In this project, all the data simulated by Abaqus will be imported into the neural network for training, and the neural network can transform the input data into a suitable internal representation, which can be used for the regression research objective. This section describes the neural networks and model structures that will be utilized in subsequent deep learning applications. In a deep learning model, the two most important parts are the neurons and the network structure.

Neurons

Biologists discovered the structure of biological neurons in the early 20th century. A biological neuron usually has several dendrites and one axon. Dendrites are used to receive information and axons are used to send messages. An artificial neuron, or Neuron, is the basic unit of a neural network, which mimics the structure and properties of a biological neuron, receives a set of input signals and produces an output. Figure 4.3 depicts a schematic of a single neuron (or



H_1	V_1	M_1
H_2	V_2	M_2
H_3	V_3	M_3
H_4	V_4	M_4
⋮	⋮	⋮
H_9	V_9	M_9
H_{10}	V_{10}	M_{10}

H_{11}	V_{11}	M_{11}
H_{12}	V_{12}	M_{12}
H_{13}	V_{13}	M_{13}
H_{14}	V_{14}	M_{14}
⋮	⋮	⋮
H_{99}	V_{99}	M_{99}
H_{100}	V_{100}	M_{100}

Figure 4.2: Two different training strategies to catch foundation response

node) with four inputs, initially introduced as a perceptron. These four inputs, corresponding to the inputs in the point-to-point predictions in Section 4.3, should be multiplied by a weight matrix w_i respectively, and their aggregate is added to the bias b. Next, an activation function f is applied to the net input value z and calculates the neuron's output y. The neuron's output y will be sent to the next neuron, completing a loop. This process can also be expressed by the following formula:

$$z = \sum_{i=1}^{n} w_i x_i + b = \boldsymbol{w}^\top \boldsymbol{x} + b \tag{4.1}$$

$$y = f(z) \tag{4.2}$$

with z is the net input value to the activation function $f(\cdot)$, x_i is the input and main component of z.

Network structure

A biological nerve cell has a relatively simple function, while an artificial neuron is just an idealized and simple implementation of a biological nerve cell with an even simpler function. A single neuron is insufficient to imitate the capabilities of the human brain; several neurons must collaborate to conduct complicated operations. Neurons that collaborate in this manner, through specific connections or information transfer, can be regarded as a network, a neural network. Researchers have created great quantities of neural network architectures to far. In the present study, the feed forward networks and memory networks are adopted

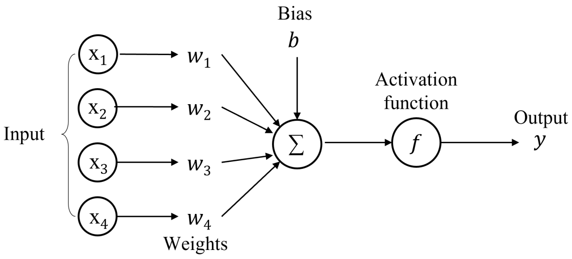


Figure 4.3: Schematic of single neuron with inputs

(Bishop and Nasrabadi, 2006).

Feedforward networks

Neurons in a feedforward network are divided into distinct groups based on the sequence in which it receives data. Each group can be regarded as a neural layer(Schmidhuber, 2015). The neurons in each layer receive the output of the neurons in the previous layer and provide output to the neurons in the next layer. The whole network propagates information in a single direction, and there is no reverse information propagation, as represented by a directed, loop-free graph (Figure 4.4). Feedforward networks consist of, among others, fully-connected feedforward networks (Section 4.2.1) and convolutional neural networks (Section 4.2.2) (Nebauer, 1998). A feedforward network can be viewed as a function that provides a sophisticated mapping from input space to output space by combining numerous basic nonlinear functions. These networks have a straightforward design and are simple to construct.

Memory networks(Recurrent networks)

In contrast, neurons in memory networks, also known as feedback networks, will receive information not just from other neurons but also from their past. Unlike neurons in feedforward networks, neurons in memory networks have a memory function which changes at different moments (Mau et al., 2020). Therefore, a directed recurrent graph can represent unidirectional or bidirectional information transmission in a memory network (Figure 4.9). Recurrent neural networks are included in memory networks (Section 4.2.3). A memory network is also regarded as a software with enhanced computational and memory capabilities (Cheng et al., 2016).

4.2.1 Feedforward Neural Network(FNN)

Given a collection of neurons, it is possible to create a network using the neurons as nodes. The topologies of network connections vary between neural network models. The feedforward network has a relatively straightforward topology. Feedforward Neural Network (FNN), also known as Multi-Layer Perceptron (MLP), was the first simple artificial neural network to be developed. However, feedforward neural networks are composed of multiple layers of logistic regression models (continuous non-linear functions) as opposed to multiple layers of perceptrons. This terminology is being gradually replaced (Bishop and Nasrabadi, 2006). In this study, the neurons in two adjacent layers are fully interconnected, forming

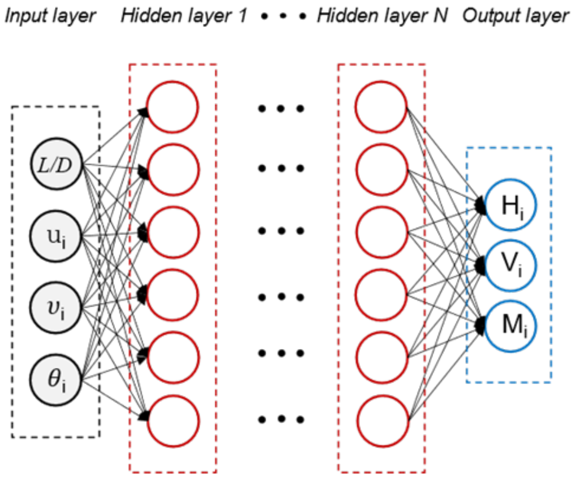


Figure 4.4: Multilayer feedforward neural network

a fully-connected (FC) neural network. The neurons in each layer can both receive signals from the neurons in the preceding layer and send signals to the neurons in the following layer. Layer 0 is referred to as the input layer, the final layer as the output layer, and the intermediate layers as the hidden layers. As seen in Figure 4.4, there is no feedback in the whole network; the signal propagates in a single path from the input layer to the output layer.

In accordance with the Universal Approximation Theorem (Haykin, 2009), a neural network can be treated as a "universal" function to some extent. Common continuous nonlinear functions can be modeled by feedforward neural networks, which have a powerful capability for fitting data. Theoretically, FC neural networks are ideal for the nonlinear regression problem under investigation. Nonetheless, learning the parameters of a neural network is more challenging than learning the parameters of a linear model, primarily due to the non-convex optimisation problem and the gradient vanishing problem, as detailed in Section 4.3.2.

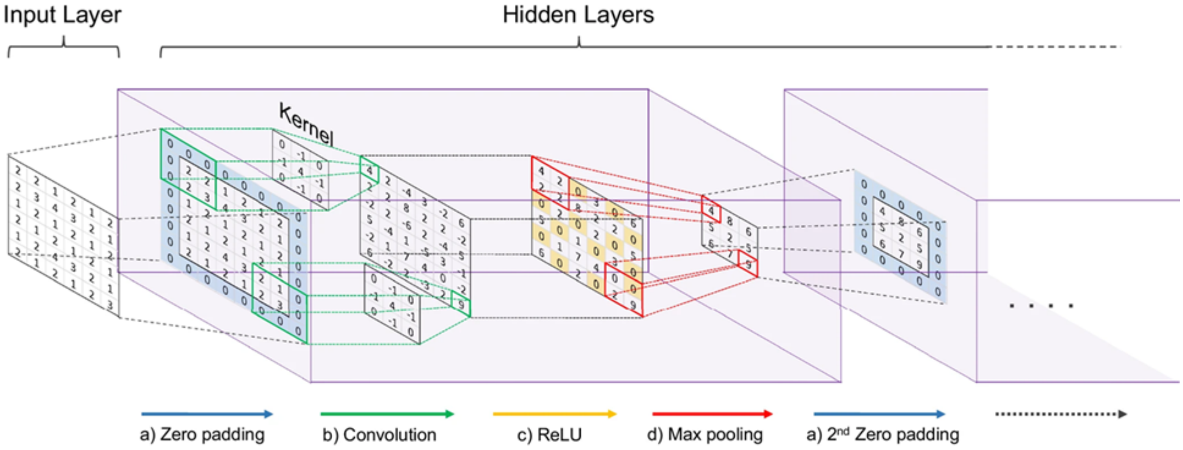


Figure 4.5: Schematic illustration of convolution and pooling processes in a CNN (Kunz et al., 2020)

4.2.2 Convolution Neural Network(CNN)

Convolutional Neural Network (CNN) is a deep feed-forward neural network with local connectivity and weight sharing. Local connectivity decreases the excessive number of parameters in FC neural networks, and weight sharing allows convolutional neural networks to extract local invariant features. Therefore, the convolutional neural networks were first used to process image information.

Each neuron in the convolutional layer (assumed to be layer l is connected only to a neuron in a local window in the next layer (layer $l-1$), forming a locally connected network. The number of connections between each layer is also greatly reduced. While weight sharing means that the weights are the same for each layer. A convolutional kernel captures only one specific local feature of the input data. Therefore, to extract multiple features requires the use of multiple different convolutional kernels. These properties speed up the computational efficiency of the model and make convolutional neural networks invariant to a certain extent in terms of translation, scaling and rotation. And compared to feedforward neural networks, convolutional neural networks have fewer parameters. Current convolutional neural networks are generally feedforward neural networks consisting of a cross-stack of convolutional, pooling and fully connected layers (shown in Figure 4.5).

The role of the convolution layer is to extract features from a local region, with different convolution kernels corresponding to different feature extractors. And the pooling layer is responsible for feature selection, reducing the number of features, and consequently the number of parameters. After the convolution layer, a pooling layer is added to minimize the number of features and prevent overfitting. Currently, the use of pooling layers is typically decreased by employing various strides and zero padding. This is because pooling layers minimize the number of features and lose some data information.

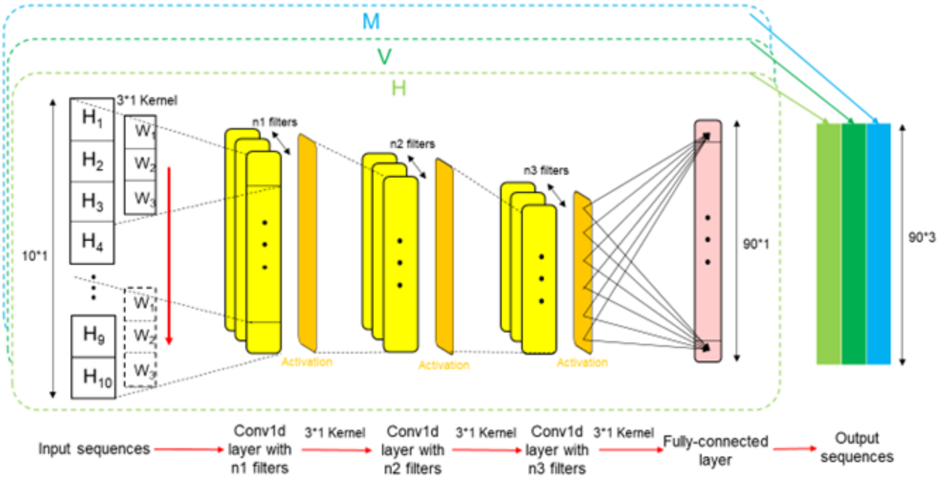


Figure 4.6: Schematic illustration of convolution processes and data transfer by using Conv1d

4.2.2.1 1 Dimension Convolution Neural Network(1D-CNN)

In time series prediction, the only types of data examined are time series data rather than pictures. In light of this, 2D-CNNs are no longer appropriate, and 1D-CNNs are believed to be the best option for predicting the data. Compared to 2D-CNNs, 1D-CNNs are significantly more computationally efficient with the same complexity. And shallow architecture 1D-CNNs can be trained and implemented more efficiently. Although the neural network's structure is similar to that of 2D-CNNs, the computational procedure has been substantially simplified. A representation of the training flow of 1D-CNNs based on temporal prediction for foundation behaviour is illustrated in Figure 4.6.

4.2.2.2 Temporal Convolutional Networks (TCN)

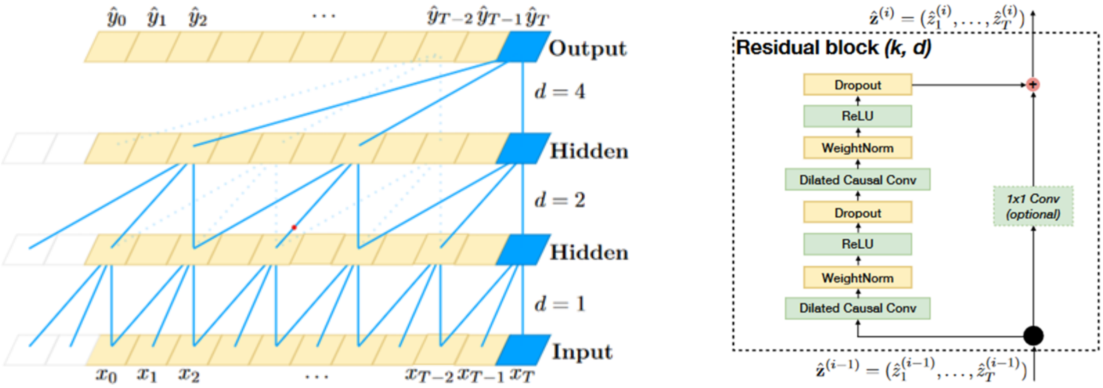
The TCN is a new network based on convolutional neural networks, specifically for the prediction of time-series data. This network's structure incorporates the most advanced scripted algorithms and model structures available at the time. The characteristics are as follows:

- Similar to an RNN, the architecture can transfer any length input sequence to an output sequence of the same length. The TCN employs a 1D fully-convolutional network (FCN) architecture (Long et al., 2015), in which each hidden layer has the same length as the input layer and zero padding of length is added to guarantee the subsequent layers have the same length as the previous layers. This feature is represented in Figure 4.7(a) by the same sequence length at each layer.

- The architecture's convolutions are causal, hence there is no information ”leakage” from the future to the past. This signifies that the current prediction is based only on data from the past and is consistent with reality. In Figure 4.7(a), the blue squares indicate the present moment and the yellow squares represent the past moment. The predictions in the blue squares depend entirely on the historical information in the yellow squares.

- By applying dilation convolution (Oord et al., 2016), fewer convolution layers can be configured to provide a broader field of vision, resulting in a longer-lasting memory. As shown in the Figure 4.7(a), despite the size of the convolution kernel of 3, the referenced historical input data have far exceeded the decision data of ordinary convolution. Altering the dilation factor d also affects the quantity of information accessible for decision making. A large dilation factor means that the information extracted by the next layer of neurons refers to more previous layer neurons. The more information is input into the next layer of neurons (i.e. the broader field of decision and feature extraction).

- A number of dilation convolutions, activation functions and batch normalization layers make up the residual block, which is shown in Figure 4.7(b). Residual blocks have a unique 'shortcut' mechanism which gives shortcuts to the neural network, while avoiding network degradation due to the complexity of the network (He et al., 2016). Utilizing residual blocks enables deeper network architectures to circumvent issues such as gradient vanishing.



$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ ... $\hat{y}_{T-2}$ $\hat{y}_{T-1}$ $\hat{y}_T$
Output
$d=4$
Hidden
$d=2$
Hidden
$d=1$
Input
x_0 x_1 x_2 ... x_{T-2} x_{T-1} x_T
$\hat{\mathbf{z}}^{(i)} = (\hat{z}_1^{(i)}, \ldots, \hat{z}_T^{(i)})$
Residual block (k, d)
Dropout
ReLU
WeightNorm
Dilated Causal Conv
Dropout
ReLU
WeightNorm
Dilated Causal Conv
1x1 Conv (optional)
$\hat{\mathbf{z}}^{(i-1)} = (\hat{z}_1^{(i-1)}, \ldots, \hat{z}_T^{(i-1)})$

(a) Visualization of a stack of dilated causal convolutional layers

(b) TCN residual block

Figure 4.7: Unique structure and algorithm of TCN (Bai et al., 2018)

The flow of TCN-based model training is shown in the Figure 4.8. A dilation convolution with an activation function (ReLU) plus a shortcut from a one-dimensional convolution form a residual block. The residual block, the flatten layer and the fully connected layer are cross-stacked to form a feed-forward neural network. Firstly, input data are processed by three layers of residual blocks. Then, the data are stretched by flattening and completely connected layers. The training phase is finished after data is reshaped back to the initial format.

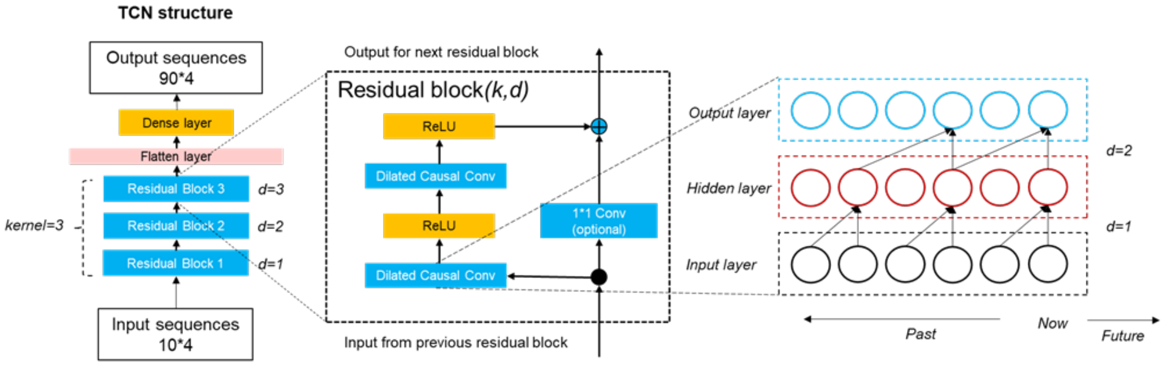


Figure 4.8: Overall architecture of the TCN-based model

4.2.3 Recurrent Neural Network(RNN)

In feed-forward neural networks, there is a unidirectional flow of information. While this limitation makes the network easier to learn, it diminishes the neural network model's performance to some extent. Recurrent Neural Network (RNN) is a type of neural network which has the capacity for short-term memory. In a recurrent neural network, neurons can receive input from other neurons in the past, similar to how the dimension of time is added to a fully connected neural network, as depicted in Figure 4.9.

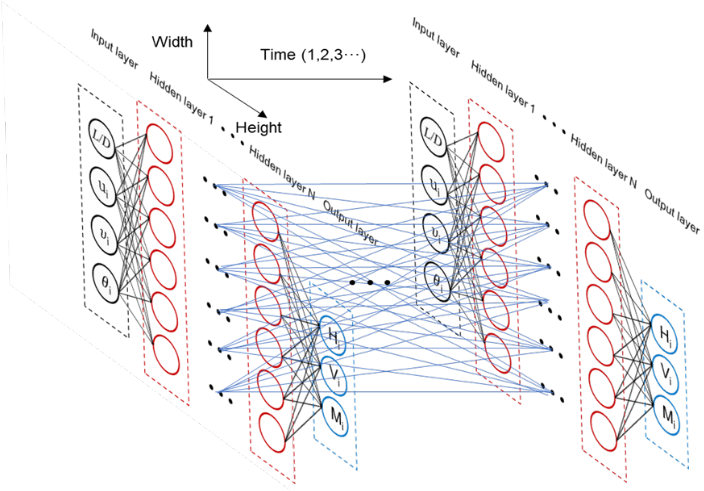


Figure 4.9: Three-dimensional architecture of the recurrent deep-learning network (Hochreiter and Schmidhuber, 1997b)

Let the vector $\boldsymbol{x}_t$ denote the input to the network at moment t and $\boldsymbol{h}_t$ denote the hidden layer state (i.e. the hidden layer neuron activity value), then $\boldsymbol{h}_t$ is related not only to the input at the current moment $\boldsymbol{x}_t$, but also to the hidden layer state at the previous moment $\boldsymbol{h}_{t-1}$ related. The equation for the recurrent network at moment t is:

$$\boldsymbol{h}_t = f\left(\boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{b}\right) \tag{4.3}$$

where $\boldsymbol{U}$ is the state-state weight matrix, $\boldsymbol{W}$ is the state-input weight matrix, $\boldsymbol{b}$ is the bias vector, and $f(\cdot)$ is the nonlinear activation function.

4.2.3.1 Long Short Term Memory(LSTM)

Gradient explosion and disappearing, commonly known as the long-range dependence problem, occurs when the input sequence is long (Bengio et al., 1994; Hochreiter and Schmidhuber, 1997b). Numerous enhancements have been made for recurrent neural networks to address this issue. Among them, the most influential one is the gating mechanism. The network structure that contains this gating mechanism is called Long Short-Term Memory Network (LSTM). LSTM network is a variant of recurrent neural networks that can effectively solve the gradient explosion or disappearing problem of the traditional recurrent neural networks. Hidden states in recurrent neural networks can hold past data and can be treated as a kind of Memory. In a basic recurrent network, the hidden state is rebuilt at each moment and is hence comparable to a Short-Term Memory. Instead, the Long-Term Memory is a network parameter that represents the training data-learned experience, and its update time is much slower than that of Short-Term Memory. The memory unit c of an LSTM network is capable of capturing a critical piece of information at a particular moment and storing it for a certain time period. The lifetime of the information stored in memory unit c is greater than that of short-term memory h, but much less than that of long-term memory. The LSTM network introduces the internal state $\boldsymbol{c}_t$, i.e. the memory unit c at moment t, specialises in linear recurrent information transfer. The LSTM outputs the hidden layer's information simultaneously to the external state $\boldsymbol{h}_t$. The internal state $\boldsymbol{c}_t$ is calculated by the following equation:

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \tag{4.4}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{c}_t\right) \tag{4.5}$$

where $\boldsymbol{f}_t$, $\boldsymbol{i}_t$, $\boldsymbol{o}_t$ are the three gates to control the path of information transfer; $\odot$ is the vector element product; $\boldsymbol{c}_{t-1}$ is the memory cell at the previous moment; $\tilde{\boldsymbol{c}}_t$ is the candidate state obtained by the nonlinear function, at each moment t, the internal state $\boldsymbol{c}_t$ of the LSTM network records the historical information up to the current moment.

The forgetting gate $\boldsymbol{f}_t$ determines how much information should be discarded regarding the internal state $\boldsymbol{c}_{t-1}$ at the last moment. The input gate $\boldsymbol{i}_t$ determines how much data will be stored for the present moment of the candidate state $\tilde{\boldsymbol{c}}_t$. The output gate $\boldsymbol{o}_t$ determines how much information from the internal state $\boldsymbol{c}_t$ need to be transmitted to the external state $\boldsymbol{h}_t$ at this time. The formula for the three gates is as follows.

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) \tag{4.6}$$

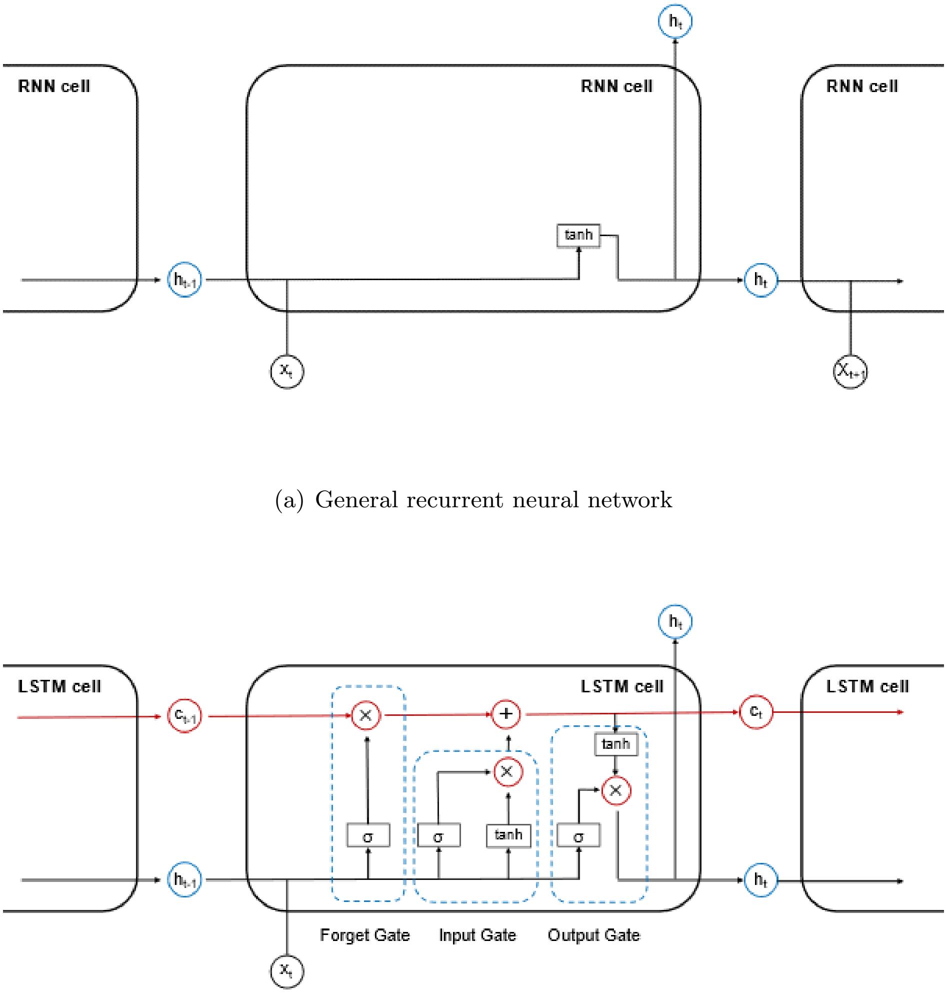
$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \tag{4.7}$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) \tag{4.8}$$

where $\sigma(\cdot)$ is the logistic function. The cell structure of the LSTM network and the hidden layer information transfer process of LSTM are shown in Figure 4.10

4.2.3.2 Long Short Term Memory Model Combined With Convolution Neural Network(1D-CNN+LSTM)

Convolutional layers are distinguished by their capacity to extract meaningful knowledge and learn the internal mechanism of time-series data, whereas LSTM networks stands out at



(a) General recurrent neural network

(b) LSTM's unique gate control mechanism

Figure 4.10: The hidden layer information transfer process of general recurrent neural network and LSTM

recognizing both short-term and long-term relationships. The purpose of the 1D-CNN+LSTM model (Livieris et al., 2020) is to integrate the benefits of these deep learning techniques in an effective manner. This model has two primary components: The first component includes of convolutional and pooling layers in which complex mathematical processes are done to produce input data features, while the second component utilizes the obtained features by LSTM and dense layers. The structure of the model is shown in Figure 4.11. It shows better prediction results compared to the CNN and the LSTM alone. Therefore, it was adopted as a new model for temporal prediction.

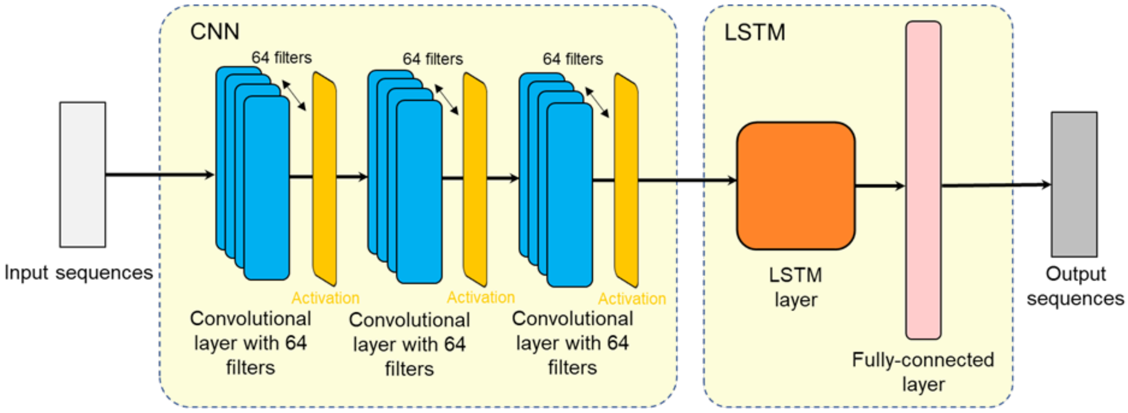


Figure 4.11: 1D-CNN+LSTM model architecture

4.3 Prediction of loading response by fully connected neural network

4.3.1 Data pre-processing

From Chapter 3, 10 embedment depths were simulated in the homogeneous soil. In each case, 96 directions and 100 data points are intercepted in each direction, generating a grand total of 96,000 data points. Similarly, 10 embedment depths were simulated in the heterogeneous soil. In each case, 48 directions and 100 data points are intercepted along each loading path, generating a total of 48,000 data points. The reason why 100 data points are captured is to assist the temporal training in Section 4.4. And in the Abaqus calculation, the first 100 data points already adequately represent the entire loading path. The remaining data points exhibit a negligible variance relative to the first 100 and can be disregarded. Each data point includes displacement information in three directions, corresponding to three forces, a foundation configuration and a soil shear strength profile. The force information of skirted foundation is represented by the bearing capacity factor (N_{cH}, N_{cV}, N_{cM}) . The configuration information of the foundation is represented by the embedment ratio (L/D) and the shear strength profile is represented by the soil strength heterogeneity index (κ). Due to the small displacement values are not dimensionless, the different scales of input and output parameters also affect model performance. The following normalising equation ultimately eliminates the size effect.

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} (\bar{x}_{\max} - \bar{x}_{\min}) + \bar{x}_{\min} \tag{4.9}$$

where $x_{\max}$ and $x_{\min}$ measured maximum and minimum of the parameter x; $\bar{x}_{\max}$ is the threshold to be scaled and $\bar{x}_{\min}$ is the upper boundary to be scaled. The upper boundary is typically set to 1 and the lower boundary is set to -1 in this project.

After normalization, the complete data is separated into three blocks, consisting of 64% training set, 16% validation set, and 20% test set. The test set consists of a random selection of 20% of the total data points. This portion of the data will not be trained. It will only be utilized for the purpose of evaluating the trained model. The 20% training data is consisted by the validation set, and the model is tested on it after each training epoch. The remaining 64% of data is actively engaged in the model's training. These two partitioning processes are controlled directly in the Keras environment with the parameters train_test_split, validation_split. All datasets are saved in comma-separated values (CSV) files for quick importation into Python.

4.3.2 Evaluation metrics

The RMSE (Root Mean Squared Error) and R^2 (Coefficient of determination) are used as the primary evaluators of the regression results, while the MAE (Mean Absolute Error) serves as the supplementary evaluator. RMSE is the root of MSE (Mean Squared Error) and has the same theoretical meaning as MSE. However, RMSE characterises the data better since MSE values do not match the unit scale of the error values, while RMSE better represents the error with the same scale.

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i-\hat{y}_i\right)^2} \tag{4.10}$$

Goodness of fit measures the degree to which a regression line matches the observed values. The coefficient of determination (R^2) is the statistical indicator of the goodness of fit. The greater the R^2, the better the fit, which is optimal at a value of 1. Negative R^2 values are possible if the model anticipates random values. The theoretical range of R^2 is $(-\infty, 1]$, with a typical value between [0, 1]. In practice, the better-fitting curve is typically selected to calculate R^2, so the value of $-\infty$ occurs infrequently. The closer R^2 is to 1, the more effectively the variables in the equation explain y, and the better the model fits the data. In contrast, the closer R^2 is to 0, the less well a model fits the data.

$$R^2 = 1-\frac{\sum_i\left(\hat{y}_i-y_i\right)^2}{\sum_i\left(\bar{y}_i-y_i\right)^2} \tag{4.11}$$

Considering that certain data points are 0, the Mean Absolute Percentage Error (MAPE) calculation would be biased. Consequently, the MAE is utilised as an alternative in the actual evaluation.

$$\frac{1}{m}\sum_{i=1}^{m}\left|\left(y_i-\hat{y}_i\right)\right| \tag{4.12}$$

4.3.3 Selection of deep learning model for nonlinear regression

The non-linear regression prediction for load-displacement relationship experiments was conducted using FC-NN, 1D-CNN and LSTM, respectively. Both 1D-CNN and LSTM models are more adept at analysing time-series data. Therefore, the time length of both 1D-CNN and LSTM models was set to 1, and no temporal relationship between data was introduced. All three models have only one hidden layer and the same number of neurons, and the network structure is in Table 4.1-4.3. The training results of the three neural networks were compared after selecting the most suitable model.

Table 4.1: Main hyperparameter during training FC-NN for nonlinear regression prediction test

FC-NN		
Hyperparameter	Description	Value
Nh	Number of hidden layers	1
Nn	Number of nodes in the hidden layer	128
η	Learning rate in the optimizer	0.001
Batch size	Number of training samples	200
Epoch	Number of iterations during training	50
validation split	Proportion of validation set in total training	0.2

Table 4.2: Main Hyperparameter during training LSTM for nonlinear regression prediction test

LSTM		
Hyper-parameter	Description	Value
Nh	Number of hidden layers	1
Nn	Number of nodes in the hidden layers	128
η	Learning rate in the optimizer	0.001
Batch_size	Number of training samples	200
Epoch	Number of iterations during training	50
validation_split	Proportion of validation set in total training set	0.2
input_length	Time length	1

Table 4.3: Main Hyperparameter during training 1D-CNN for nonlinear regression prediction test

1D-CNN		
Hyper-parameter	Description	Value
Nh	Number of hidden layers	1
filters	Number of filters in the hidden layers	128
kernel_size	Length of the convolution kernel	1
strides	Step size of convolution kernel shift	1
padding	padding mode	valid
dilations	Convolutional kernel dilation	[1]
η	Learning rate in the optimizer	0.001
Batch_size	Number of training samples	200
Epoch	Number of iterations during training	50
validation_split	Proportion of validation set in total training	0.2

Out of 96,000 sets of data from homogeneous soils, 76,800 sets were randomly selected to train thees models, leaving 19,200 sets for testing. The prediction results of the test are presented in the following Table 4.4 through three evaluation matrices.

Table 4.4: Prediction results of the three models

	FC-NN	LSTM	1D-CNN
RMSE	0.006	0.003	0.006
R^2	0.998	0.999	0.998
MAE	0.003	0.002	0.003

In the table, all three of the models show excellent model predictions, and the predictions are already very close to the actual values. However, the FC-NN model has the simplest structure with the most computationally efficient. Therefore, FC-NN was chosen for the following non-linear predictions.

4.3.4 Hyperparameter tuning

A deep neural network is a highly nonlinear model, and its hyperparameters must be fine-tuned for optimal performance. Hence, this section explains the hyperparameters of the neural network that were used for load response predictions, some of the hyperparameters have been marked in blue in Figure 4.1.

Activation function

The activation function is very important in neurons, as it enhances the representation and learning ability of the network. Two common activation functions are compared below.

ReLU (Rectified Linear Unit) (Nair and Hinton, 2010) is the activation function often used in deep neural networks nowadays. Graphically ReLU is actually a ramp function, defined as a neuron that only needs to perform addition, multiplication and comparison operations, making it more computationally efficient. The ReLU function is also considered to have biological plausibility. While Tanh is a Sigmoid-type function, which refers to a category of S-curve functions that are saturated at both ends. The Tanh function's output is zero-centered, but the ReLU function's output is always higher than or equal to 0. The non-zero-centered output induces a bias shift in the input of the adjacent layer's neurons. Due to the Sigmoid activation function's saturation, the derivative in the saturation zone is much closer to 0. Consequently, the transmission of the error through each neuronal layer is progressively diminishing. When a network is extremely dense, the gradient will continue to degrade or perhaps vanish, making it difficult to train the entire network. This is known as the problem of disappearing gradients. The ReLU, Tanh, Sigmoid function are all shown in Figure 4.12.

In terms of optimization, the ReLU function is a left saturated function and has a derivative of 1 when $x > 0$, whereas the Sigmoid function is saturated at both ends. This somewhat mitigates the gradient vanishing problem and accelerates the gradient descent convergence. The Sigmoid-type activation function produces a dense neural network, whereas the ReLU

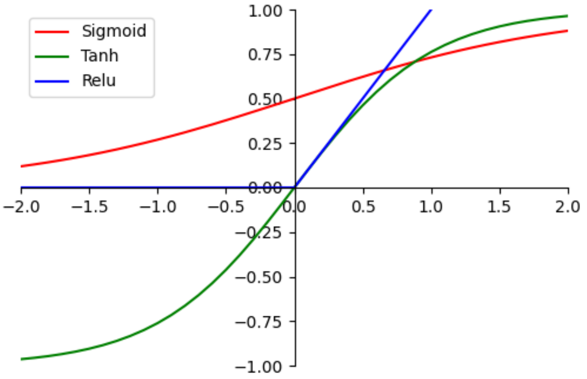


Figure 4.12: Hidden layer activation functions

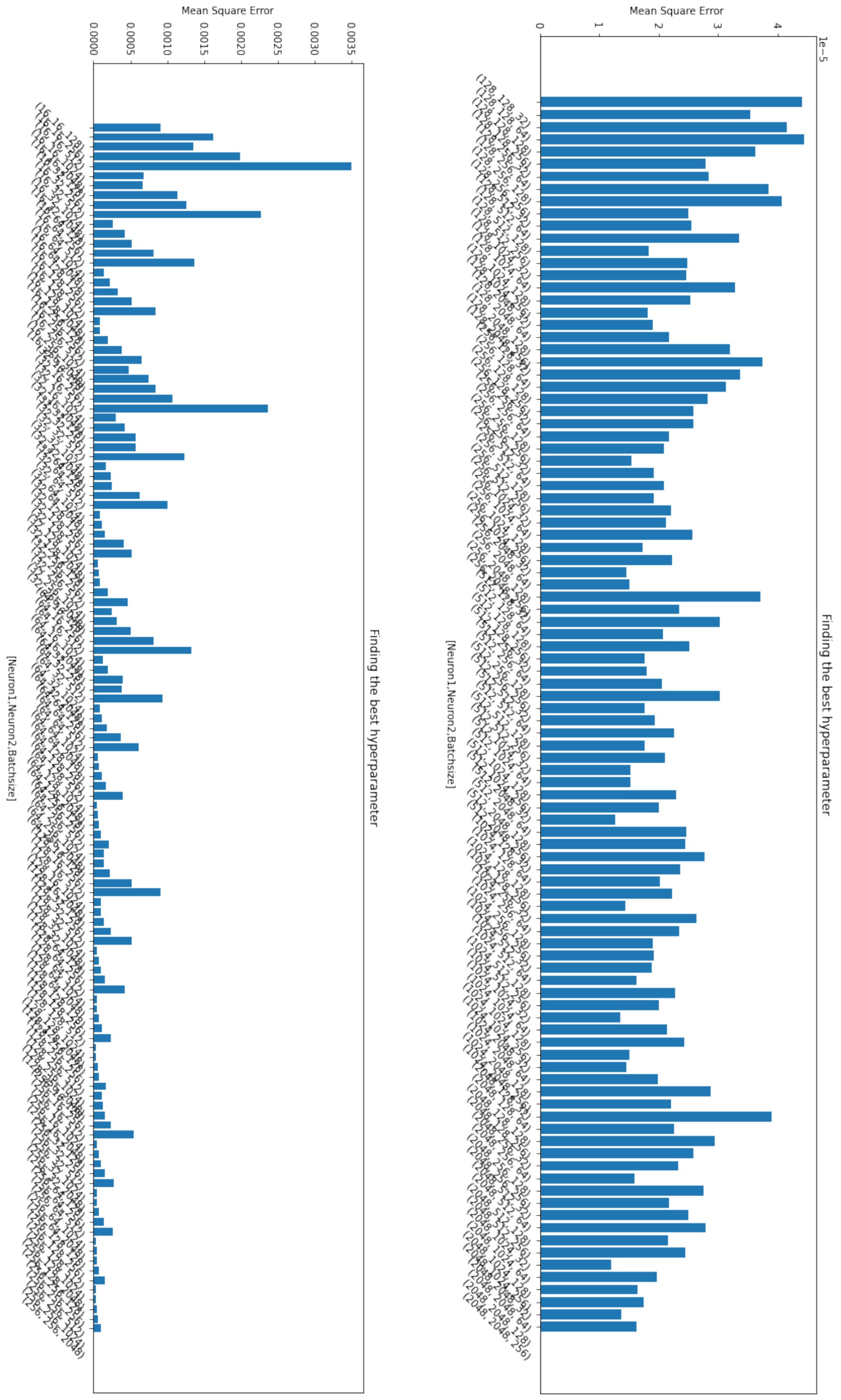
activation function produces a sparse neural network with around 50 percent of the neurons activated (Glorot et al., 2011). Therefore, ReLU is selected as the model's activation function.

Batch size and neurons

Using a grid search strategy, this part analyzes the number of neurons and batchsize of each layer. After trial computations, the FC neural network with a single hidden layer was able to predict the data. However, the stability and robustness of the prediction were inadequate. The model's performance can be enhanced by increasing the number of hidden layers. Theoretically, the more the accuracy should be, the greater the solution space that can be covered, and the deeper the network layers. However, increasing the number of layers does not directly improve convergence and precision. If three layers are configured, the network will face the challenge of overfitting, and much effort will be necessary to avoid it. Therefore, a fully-connected model structure with two hidden layers is desired. In order to maximize the performance of parallel computing within the GPU, the batchsize is often required to be a multiple of 8 (e.g., 32, 128). This project investigates the performance of FC neural network based models with batch size of 128, 256, 512, 1024 and 2048. The number of neurons each layer is 16,32,64,128,256, respectively, for a total of 125 (5*5*5) sets of trials, and the minimum mean squared error in each experiment generate the Figure 4.13(a) below.

From the Figure 4.13(a), it can be deduced that the optimal batch size is 128 and number of neurons in the first and second layers are 256, respectively. An obvious rule is that the higher the number of neurons and the smaller the batch size, the better the performance of the model. Theoretically, the greater the number of neurons and the number of layers, the better the model will perform. As batch size increases, the variance of the random gradient decreases, less noise is introduced, and training becomes more stable. While appropriately small batch sizes will lead to faster convergence. Therefore, the batch size must therefore take into account training stability and convergence speed.

A new expanded experiment was designed based on 256 neurons per layer neurons and



(a) Origin experiment

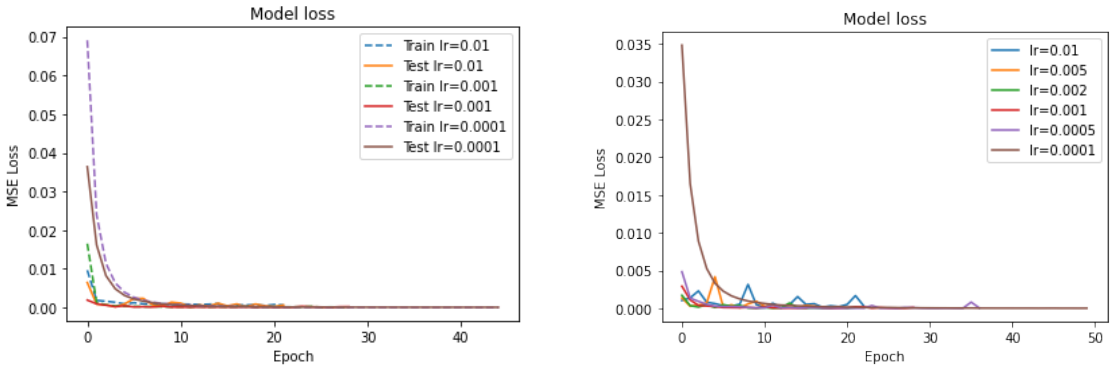
(b) Addition experiment

Figure 4.13: Find the optimal neurons and batchsize by grid search method

128 samples each batch. This time, the batchsize was set 32, 64, 128 and 256 neurons respectively, and the number of neurons per layer was set 128, 256, 512, 1024, and 2048 respectively. The new results are shown in Figure 4.13(b). 1024 neurons per layers and 128 samples in a batch are the optimal solutions. The optimal number of neurons at 1024 instead of 2048 indicates that the model's performance has begun to degrade and the optimal batch size doesn't change. After evaluating computational efficiency and precision, the number of neurons per layer is determined to be 256, while the batchsize is 128.

Optimiser and learning rate

The Adam algorithm (Adaptive Moment Estimation Algorithm) is selected as the optimiser in this study (Kingma and Ba, 2014), which is a combination of the momentum method and the RMSprop algorithm. In addition to using momentum as the direction of parameter update, it can also alter the learning rate adaptively. Nonetheless, the initial learning rate must be determined. When the batch size is small, the learning rate must be set to a low value in order for the model to converge. Typically, the learning rate accelerates as batch size grows. Figure 4.14(a) shows that both the training and test values have the same trend during training at different learning rates. Figure 4.14(b) then shows the training process at different learning rates. The model converges rapidly for all learning rates except for the learning rate of 0.0001. While at learning rates of 0.01 and 0.005, the training process shows fluctuations in loss value. This implies that at high learning rates, overfitting is prone to occur. The loss curve of training remained stable at learning rates of 0.001 and 0.002. Therefore, a learning rate of 0.001 was used as the initial value for model training.



(a) Comparison between train and test processes under same learning rate

(b) Comparison of train processes under different learning rates

Figure 4.14: Loss of train and test set at different learning rates

Loss function

Under the regression task, the loss function's are divided into two main categories, MSE (Mean squared error) and MAE (Mean absolute error). MAE loss is more robust to outliers, but its derivative discontinuity makes the process of finding the optimal solution inefficient; MSE loss is sensitive to outliers, but is more stable and accurate in the optimisation process. Since there are no incorrect data points or noise in this experiment, MSE is employed to produce a more precise result. It computes the mean of the squared discrepancies between the prediction and the target value. By square rooting the error, larger errors are punished more severely than smaller ones. Its ideal value is zero. The specific formulae has been shown in Section 4.3.2

Overfitting

The loss value learning curves for the training and testing sets are normally plotted and compared. When a large loss value on the training set and a low loss value on the testing set is observed, it means that the trained model has a problem with overfitting. Overfitting can be solved in various approaches, including employing Dropout, introducing Early Stopping, adding Batch Normalization, and changing the batch size.

The Dropout mechanism allows random disconnection between input or hidden layers. By introducing this random disconnection mechanism, the robustness of the model can be enhanced by avoiding over-reliance on a few points. Layer-wise Normalisation not only improves the efficiency of the optimisation but also serves as an invisible regularisation method. During training, the neural network's prediction of a sample is related to the other samples in the same batch. Because of the random nature of the batch selection, the neural network does not "overfit" to a particular sample, thus improving the network's generalisation ability (Luo et al., 2018). Early stopping refers to ending the training of a model while the loss is minimal or before the training and validation error begin to diverge, i.e., before overfitting begins.

After preliminary testing, the model has the potential to jump out of a local minimum after using the optimal batch size. With early stopping, the probability of loss fluctuation during training is significantly reduced. Therefore, the model's inherent stability is strong, and there is no need to include the dropout layer and Batch Normalization, which reduce computational efficiency. The final hyperparameters utilized for FC-NN training are displayed in Table 4.5.

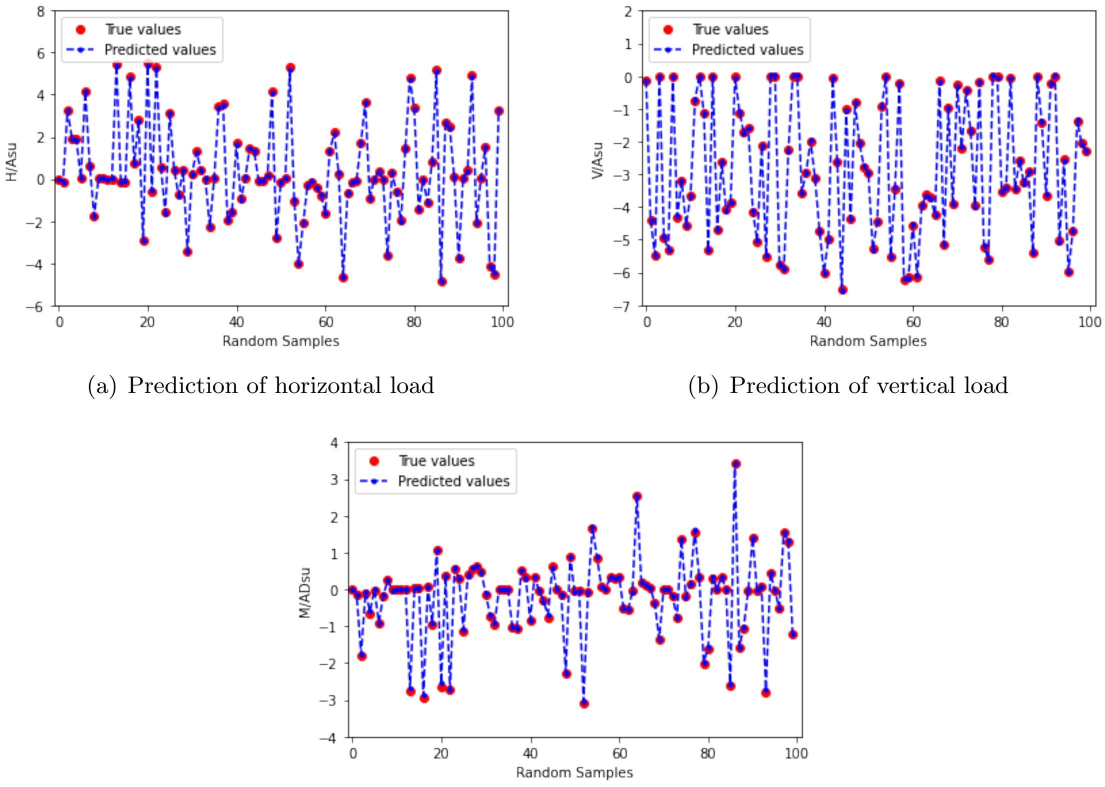
Table 4.5: Main hyper-parameters during training FC-NN for nonlinear regression prediction

FC-NN		
Hyperparameter	Description	Value
Nh	Number of hidden layers	2
Nn	Number of nodes in the hidden layer	256,256
η	Learning rate in the optimizer	0.001
Batch size	Number of training samples	128
Epoch	Number of iterations during training	50
validation split	Proportion of validation set in total training	0.1

4.3.5 Experiment results

The experimental results of the AI-based design model are divided into two parts, the training results for homogeneous soils only and the training results incorporating heterogeneous soils. The first part of the results, i.e. 96,000 sets of data for homogeneous soils only, consists of three inputs of deformation (u,v,θ) and burial depth ratio L/D, and three outputs of load response (H, V, M).

The predicted results are shown below, with the regression evaluation metrics calculated from the test dataset, with $RMSE = 0.035$, $R^2 = 1.000$ and $MAE = 0.023$. The red dot is the actual value and the blue dotted line is the predicted value. In Figure 4.15, the red dots represent the actual values, while the blue dashed line represents the anticipated value. The fact that the two significantly overlap indicates that the forecasts are accurate. $R^2 = 1.000$ indicates that the model fit has been optimal.



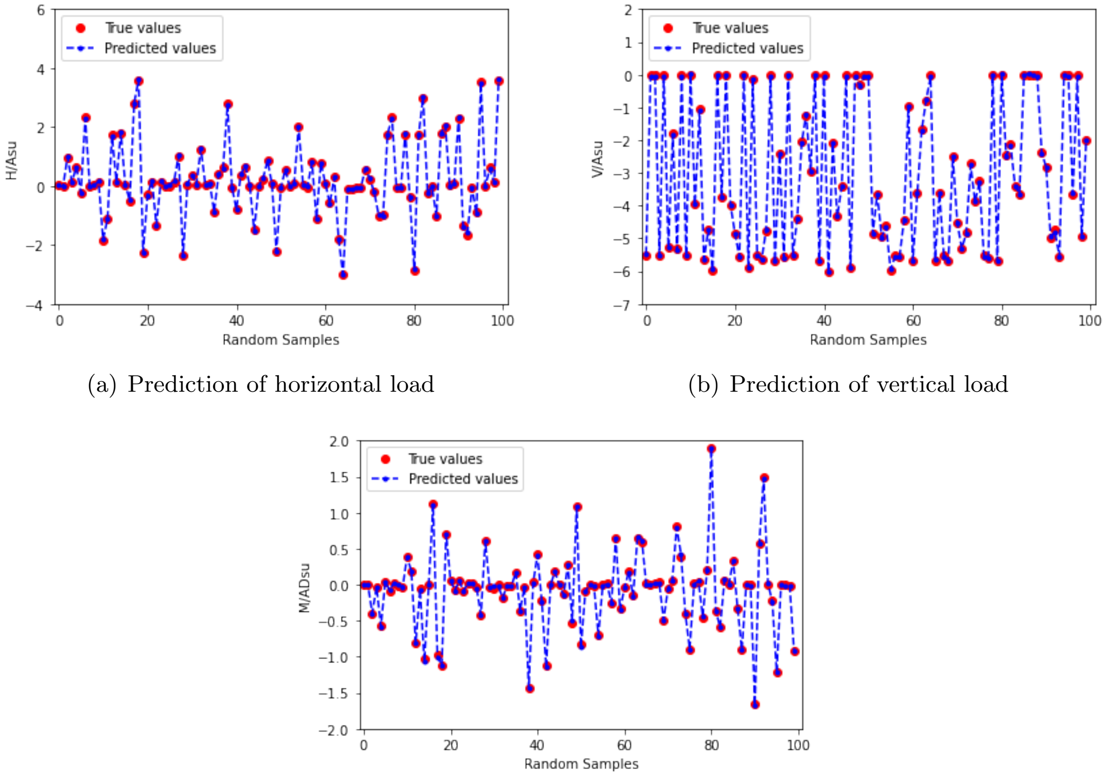
(a) Prediction of horizontal load

(b) Prediction of vertical load

(c) Prediction of moment

Figure 4.15: Prediction results for 100 randomly samples on H, V and M respectively in homogeneous (Test RMSE: 0.035Test R^2: 1.000 Test MAE: 0.023)

The second experiment is based on the first experiment with the addition of data generated when the soil profiles differ. The result of second experiment determines whether the model can accommodate predictions of the underlying behaviour under multiple soil shear strengths and failure mechanisms. In this phase, 192,000 data sets of homogeneous and heterogeneous soils were trained. The soil strength heterogeneity index was introduced to the model input in order to differentiate between the two types of soils. Three response forces were predicted from five input values, showing excellent accuracy (shown in Figure 4.16). It was found that even very “shallow” FC neural network model can learn the intrinsic failure mechanisms of the caissons from raw data and predict their nonlinear mechanical responses under complex loading effectively.



(a) Prediction of horizontal load

(b) Prediction of vertical load

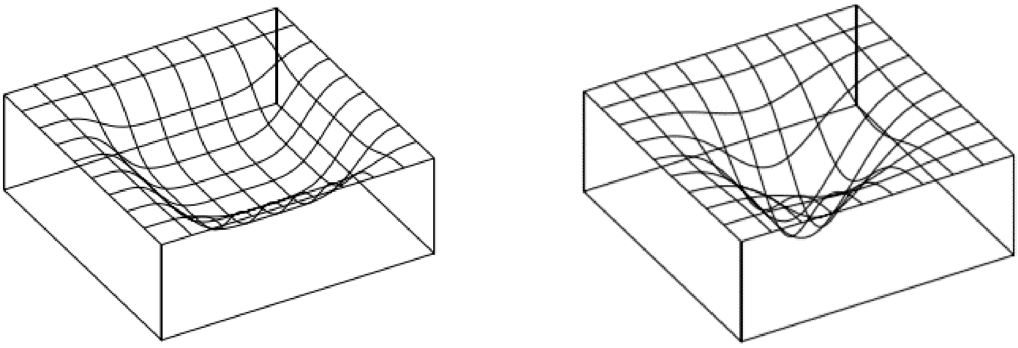
(c) Prediction of moment

Figure 4.16: Prediction results for 100 randomly samples on H, V and M respectively in homogeneous and heterogenous soil(Test RMSE: 0.024 Test R^2: 1.000 Test MAE: 0.016)

4.3.6 Evaluation of surrogate model performance

The flat minimum (Hochreiter and Schmidhuber, 1997a) (shown in Figure 4.17(a)) is often considered to have a relationship with generalisation ability. In general, a model is more robust when it converges to a local flat minimum, which means small parameter changes do not drastically affect the ability of the model, and less robust when it converges to a local sharp minimum (shown in Figure 4.17(b)). Models with good generalisation ability should generally be robust, and the ideal local minimum should be flat. The pursuit of global minima will lead to overfitting. The following experiments on the model's reproducibility are used to evaluate the robustness of the model.

The outstanding performance of the model can come from the fortuitous selection of weights and biases, which considerably helps model training and makes convergence very simple. Or possibly the data were generously divided by chance, and the challenging data were placed in the training set while the left data in test set is easily predicted. The random



(a) Flat minimum

(b) Sharp minimum

Figure 4.17: Example of a flat minimum and sharp minimum (Hochreiter and Schmidhuber, 1997a)

number seed (which symbolizes the randomness of the model) influences not just the model's initialisation weights and bias parameters, but also the partitioning of the test and validation sets. To prevent a model that performs exceptionally well with a particular amount of random number seeds. The robustness of the model was evaluated by repeating experiments with various random seeds. After testing 50 random number seeds, the MSE loss values for model training are shown below (Figure 4.18). The blue line represents the mean of the fifty training runs, while the red line represents the standard deviation of each epoch. The training process stabilizes after the fifth epoch, and the model rapidly converges. The error line reaches its highest value at 22th epoch, and the loss value swings by 0.0007, which is within an acceptable range. Experiments reveal that the model is very repeatable and robust.

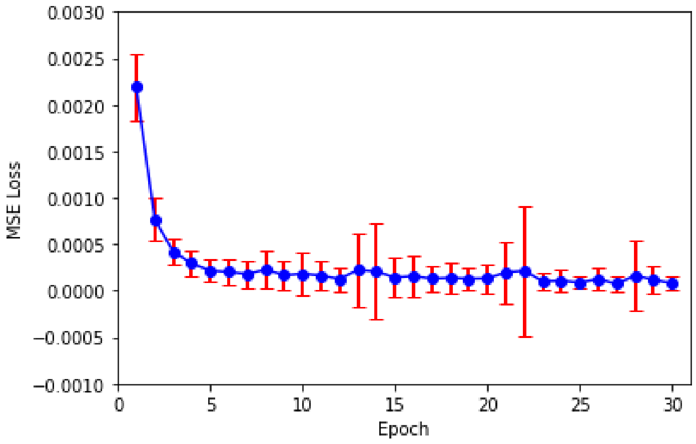
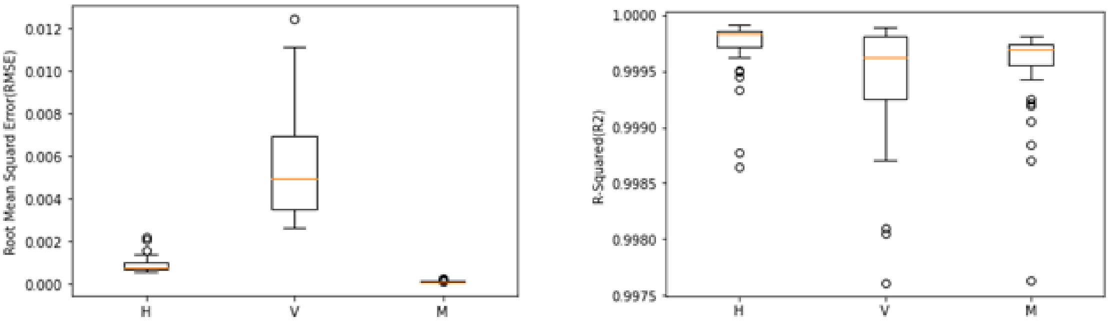


Figure 4.18: The mean MSE Loss under 50 random seeds

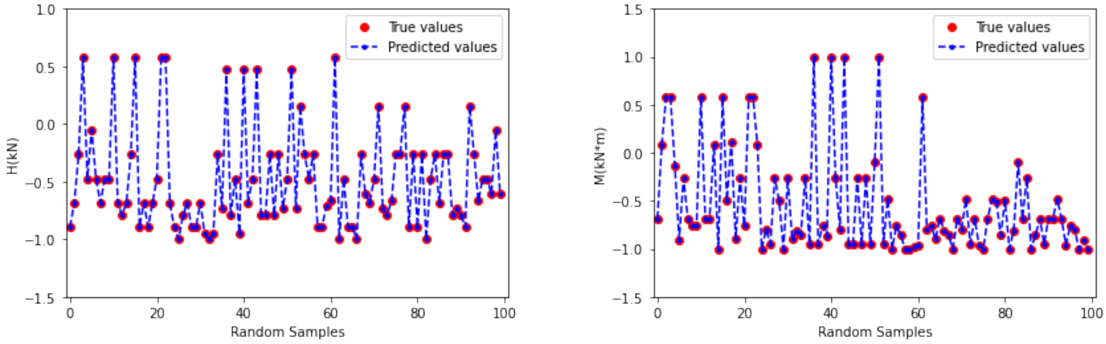
The predictions of H, V, and M from these fifty experiments are represented by RMSE and R^2 (shown in Figure 4.19), with the RMSE of prediction for the three forces being less than 0.012 and R^2 remaining above 0.9975. The overturning moment prediction has the smallest error and the horizontal force prediction converge fastest. The results of the 50 replicate experiments demonstrate the high reproducibility and robustness of the model.



(a) RMSE for H, V, M predictions

(b) R^2 for H, V, M predictions

Figure 4.19: Distributions of H, V, M prediction error under 50 random seeds

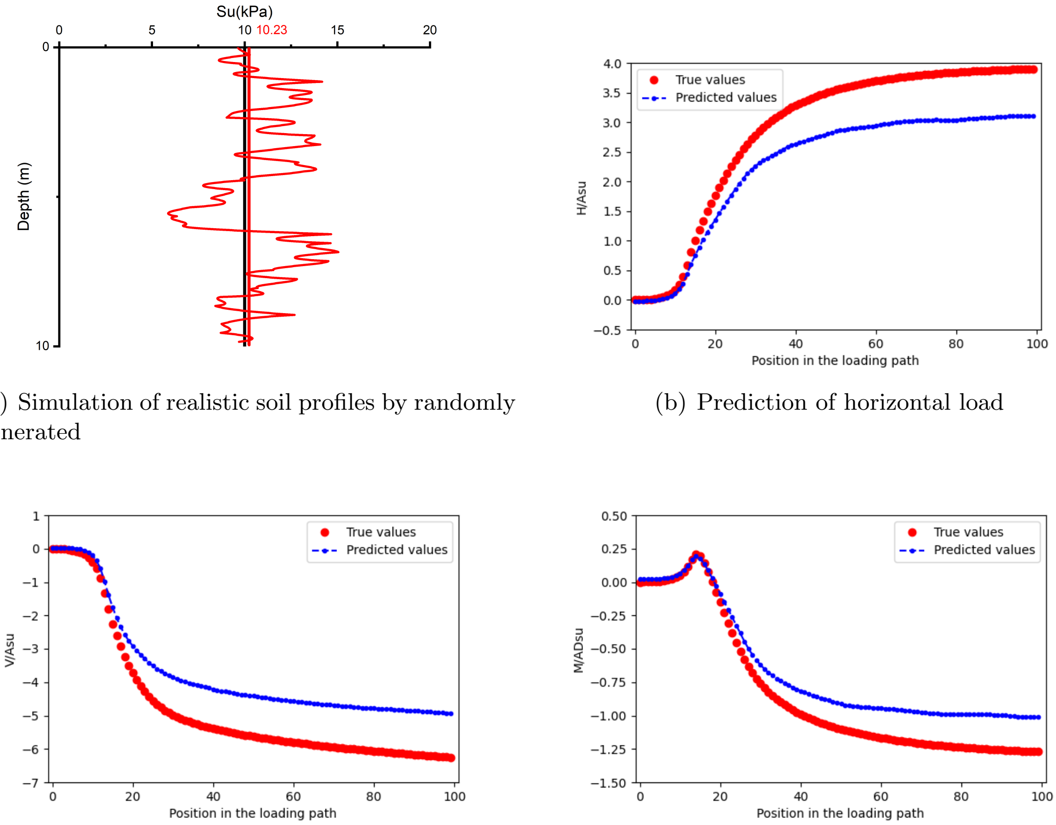


(a) Prediction of horizontal load

(b) Prediction of moment

Figure 4.20: Prediction results for 100 randomly samples on H and M respectively in sand (Test RMSE: 0.002 Test R^2: 0.999 Test MAE: 0.001)

The model is also tested against Zhang et al. (2020b)'s database which is generated from an advanced numerical modelling combining the smoothed particle hydrodynamics with the SIMSAND model. This database is validated by laboratory tests, physical model tests, and a caisson foundation field test. The hyperparameters setting of this model are not adjusted and the two columns of zhang's data are reduced, which is a non-essential input when in the FC-NN model. The model was trained and the experimental results are shown in Figure 4.20. Although the R^2 decreases slightly compared to the experimental results of this study, the fit is still very good. The SIMSAND model producing a noisy dataset is a reasonable explanation for this phenomenon. The MSE loss function would be affected by the noise, leading to the predicted bias. The good prediction results are attributable to the FC-NN model's structure, which is easy to train. The LSTM model used in Zhang's study did not perform as well as the FC neural network in the point-to-point problem. This is due to the high complexity of the LSTM model, which will be very effective in dealing with complex temporal problems, but its unique gating mechanism will not perform well in simple regression problems. However, when it comes to time series issues, the benefits of the LSTM will appear, and a comparison of these two models will be presented in section 4.4. The successful application of this FC-NN based model structure in the sandy dataset illustrates once again that the model's robustness is outstanding. This FC model structure can be adapted to different training data (i.e. clay and sand). The model structure also has transfer learning capabilities and can be applied to similar issues.



(a) Simulation of realistic soil profiles by randomly generated

(b) Prediction of horizontal load

(c) Prediction of vertical load

(d) Prediction of moment

Figure 4.21: Realistic soil profile and the prediction. (a) Simulation of realistic soil profiles by randomly generated, (b-d) Prediction results for $\theta = 45°$, $\phi = 80°$ loading path on H, V and M respectively in non-linear soil profile (Test RMSE: 0.760 Test R^2: 0.927 Test MAE: 0.583)

However, the above experiments were performed with data from the ideal situation. In the field, the soil profile cannot present a perfect uniform or linear. For this purpose, the inhomogeneity and randomness of the soils on site were simulated in Figure 4.21(a). The figure simulates a realistic soil profile by randomly generating soil strength. For better comparison with homogeneous soils, randomly generated soils has the similar average strength value of 10.23 kPa. A specific foundation at $L = 8m$ was built for comparison experiments, and 96 directions of loading were performed. 192,000 data points created with homogeneous and heterogeneous soils are used to train the deep learning-based model and then perform the prediction of non-homogeneous soils. The predicted results are shown in Figure 4.21. The prediction errors RMSE, R^2, MAE are 0.760, 0.927, 0.583 respectively. In the figure, the prediction error in H, V, and M are all large. The poor prediction results in nonlinear soils can be equated to the calculated difference between the FEM model in calculating linear and nonlinear soils. It is unrealistic to predict nonlinear soils with a model that has only learned the mechanism of uniform and linear soils. At the same time, there are also errors between the foundation response in nonlinear soil calculated by the FEM model and the actual field. As a result, more work has to be done before this technology can be applied directly in reality prediction. To improve the prediction accuracy of field data, it is not enough to learn heterogeneity parameters alone. A very feasible idea is to better simulate the strength distribution of the soil profile by introducing more input parameters, e.g. coefficients of variation, CPT data, etc. But how to characterize the soil strength parameters by field data will be another issue.

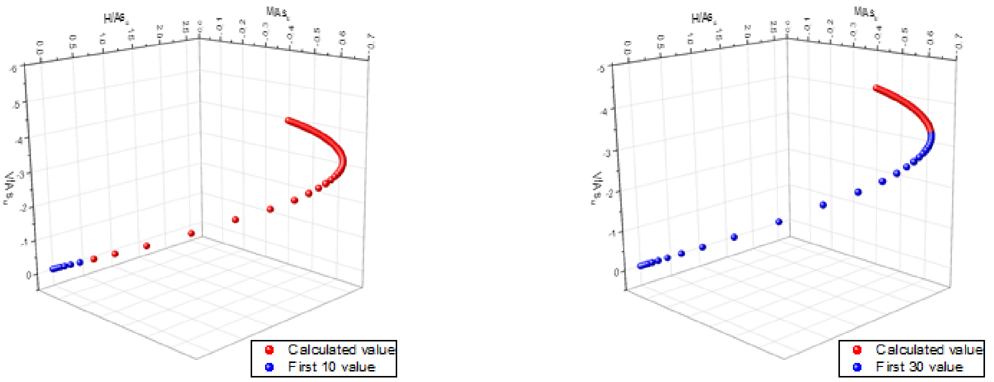
4.4 Prediction of loading paths by temporal neural networks

4.4.1 Introduction

The loading process of a foundation can be considered as a time series, whose loading response at each moment depends to a large extent on the loading state at previous times. On the one hand this time series prediction makes it possible to predict how the loading path will change at future moments. On the other hand after predicting the final state of the load, the failure envelope of the caisson can be plotted. Throughout the training, the loads at the first 10 points of each path will be used as input to predict the loads at the next 90 points. The initial experimental setup was to predict the data for the next 70 points from the first 30 points, as shown in Figure 4.22. As the step size is adaptive in the Abaqus calculation, the increase in displacement is not constant each time. For the model to converge, the displacement increment is decreased as the foundation approaches a state of failure. This results in an output loading path in which the data points are not evenly distributed and a significant number of data points are concentrated near the failure. During loading, the first ten data points already encompass the elastic deformation section and a portion of the plastic deformation section, with the initial trend of the curve also indicating a shift. This non-uniform distribution is also a prerequisite for building this experiment. If the future forecast is made using 30 data points, which already contain the majority of the loading path. Then only the post-failure trend of the loading paths needs to be modelled, which is relatively unchallenged. Therefore, this section uses three main neural networks, FNN, CNN and RNN for prediction, containing five specific models respectively FC-NN, LSTM, 1D-CNN, 1D-CNN+LSTM and TCN, which have been described in Section 4.2.

4.4.2 Experiment setup

The hyperparameter tuning in this section follows the same process as in Section 4.3.2, considering in turn the number of layers, neurons, batchsize, and then the learning rate and loss function settings. The activation function, optimizer, overfitting, and other parameters are configured similarly to Section 4.3. For the sake of objectivity, the number of layers,



(a)

(b)

Figure 4.22: Comparison of loading path predictions with 10 points and with 30 points

learning rate, batch size, and epoch number were fixed in the comparison validation trials. The procedure's details are not repeated here.

FC-NN

When FC-NN was applied in Section 4.3, only a point-to-point mapping was performed. In that case, 4 parameters (1, 4) in a data were input and 3 predictions (1, 3) were output at a time. This section focus on the prediction of whole loading paths. At each moment, 4 parameters of 10 data are input as the shape of (10, 4). And 4 parameters of 90 data are predicted for the future in the shape of (90, 4). The FC-NN model cannot accept multi-dimensional data as input, so the 10 × 4 data are ordered into the form of 1 × 40 data for input, and the output data are then reshaped from 1 × 360 dimensions to 90 × 4 dimensions. In this training method, the model needs to establish the relationship between 40 input data and 360 output data, so the model needs to be deeper and the number of nodes needs to be higher than the single-point regression prediction. The model learns not only the relationship between the parameters in each data point but also the relationship between the data points. The final hyperparameter settings are shown in Table 4.6. Figure A.1 depicts the data transit and network structure of the model (where None indicates that this dimension is variable, which can also be considered as batchsize from a data transfer perspective). The two matrices on the right represent the shape of the input data and the shape of the output data, with the shape of the output data in each layer corresponding to the number of hidden nodes in that layer. The first three dense layers are configured as hidden layers, whereas the final dense layer is just configured to alter the output data's shape. In the subsequent model, a dense layer is added at the end, which corresponds to the integration of information from all hidden layers and the specification of the output shape, and does not include it in the table of hyperparameters.

LSTM

Due to its unique gating mechanism, the LSTM handles temporal data exceptionally effectively. Changes can also be made to adjust the time step to 1 for standard regression

Table 4.6: Main hyper-parameters during training FC-NN for temporal prediction

FC-NN		
Hyper-parameter	Description	Value
Nh	Number of hidden layers	3
Nn	Number of nodes in the hidden layers	$64, 512, 1024$
η	Learning rate in the optimizer	0.001
Batch_size	Number of training samples	50
Epoch	Number of iterations during training	300
validation_split	Proportion of validation set in total training set	0.2

prediction. However, as shown in Section 4.3.6, the LSTM's ability degrades to an ordinary FNN after losing its unique long-time memory, and the prediction is, in turn, less effective than FC-NN. The time memory can be introduced to the current time-series problem by setting the time step parameter to 10. These 10 data points are read 10 times, and the data information of each (1, 4) is gradually retained and passed on to the next moment, which ultimately becomes the algorithm's input (10, 4). The two-layer LSTM was deemed suitable for this task, and the training loss curve was already fluctuating, indicating a high likelihood of overfitting. Therefore, the LSTM model was configured with two layers and 64 and 512 neurons in each layer, respectively. After adjusting the settings, Table 4.7 was obtained, and the training procedure is depicted in Figure A.2.

Table 4.7: Main hyper-parameters during training LSTM for temporal prediction

LSTM		
Hyper-parameter	Description	Value
Nh	Number of hidden layers	2
Nn	Number of nodes in the hidden layers	64,512
η	Learning rate in the optimizer	0.001
Batch_size	Number of training samples	50
Epoch	Number of iterations during training	300
validation_split	Proportion of validation set in total training set	0.2

1D-CNN

1D-CNN, as a type of convolutional neural network, has convolutional computation at its core, so the size of the kernel determines the area of the field of view. Since there are only 10 data, the size of each kernel layer is set to 3 while ensuring that no data information is lost. The computation of the first hidden layer in 1D-CNN receives only three adjacent data from the input layer, unlike the LSTM model, which combines 10 historical data for computation, so the memory is short term. The number of hidden layers was set to two to facilitate a more accurate comparison with the LSTM. The filters in 1D-CNN were the same amount of neurons in the FC-NN. No zero padding is set, as equal sequence size per layer is not considered. Also, no dilation convolution is employed so that the comparison with the following TCN model is more evident. The remaining hyperparameters are specified in Table 4.8.

Table 4.8: Main hyper-parameters during training 1D-CNN for temporal prediction

	1D-CNN	
Hyper-parameter	Description	Value
Nh	Number of hidden layers	2
filters	Number of filters in the hidden layers	128,512
kernel_size	Length of the convolution kernel	3,3
strides	Step size of convolution kernel shift	1
padding	padding mode	valid
dilations	Convolutional kernel dilation	$[1, 1]$
η	Learning rate in the optimizer	0.001
Batch_size	Number of training samples	50
Epoch	Number of iterations during training	300
validation_split	Proportion of validation set in total training	0.2

The pre-processing of the data is consistent with that of the LSTM. Theoretically, the model could be optimised for over-fitting by adding a batch normalisation (BN) layer to the model. However, trial calculations revealed that the model's MSE loss curve did not fluctuate considerably throughout training; hence, the BN layer was not included in the model. A flatten layer also appears in the model. The role of this layer is to reduce the dimensionality, stretching the two-dimensional data into one dimension before entering the dense layer for information integration and specification of the output shape. The LSTM model also has two-dimensional data, but the LSTM at the last layer only returns the final value of the sequence, thus completing the dimensionality reduction. This procedure is implemented directly in the Keras environment with the parameter return_sequences. The training procedure is depicted in Figure A.3.

1D-CNN+LSTM

The model is constructed by adding an LSTM computation layer to the 1D-CNN while preserving the original 1D-hyperparameter CNN's settings. Due to its two-layer 1D-CNN, one-layer LSTM, and one-layer dense structure, the model's total number of parameters exceeds 2M. This is twice the total number of parameters of the preceding three models, which maintain the total number of parameters at approximately 1M. This model was created to maximise the utility of both 1D-CNN and LSTM. The specific hyperparameter values are presented in Table 4.9, and the training flow is depicted in Figure A.4.

TCN

Theoretically, a network with a residual layer configuration can effectively tackle the problem of network degradation; hence it is quite feasible to configure 30 or more residual blocks for the most precise prediction. However, the final setup of TCN takes three residual blocks with 64, 128 and 256 neurons per layer. This is to control the model's size and facilitate comparison with other models. Each residual block consists of two hidden layers coupled to the activation function ReLU, and a shortcut for convolutional operations is configured. To take full advantage of TCN, the model also uses zero padding and dilations. The best prediction results were experimentally obtained with kernel sizes of 1, 3 and 3 for each layer,

Table 4.9: Main hyper-parameters during training 1D-CNN+LSTM for temporal prediction

1D-CNN+LSTM		
Hyper-parameter	Description	Value
Nh	Number of hidden layers	3
filters	Number of filters in the hidden layers	32,128
Nn	Number of nodes in the hidden layers	512
kernel_size	Length of the convolution kernel	3,3
strides	Step size of convolution kernel shift	1
padding	padding mode	valid
dilations	Convolutional kernel dilation	$[1, 1]$
η	Learning rate in the optimizer	0.001
Batch_size	Number of training samples	50
Epoch	Number of iterations during training	300
validation_split	Proportion of validation set in total training set	0.2

respectively. The settings of the hyperparameters are shown in Table 4.10 and the training flow is shown in Figure A.5.

Table 4.10: Main hyper-parameters during training TCN for temporal prediction

TCN		
Hyper-parameter	Description	Value
Nh	Number of hidden layers	2
Nn	Number of residual blocks	3
filters	Number of filters in the hidden layer	$64, 128, 256$
kernel_size	Size of the convolution kernel	$1, 3, 3$
strides	Step size of convolution kernel shift	1
padding	padding mode	same
dilations	Convolutional kernel dilation	$[1, 2, 4]$
η	Learning rate in the optimizer	0.001
Batch_size	Number of training samples	50
Epoch	Number of iterations during training	300
validation_split	Proportion of validation set in total training	0.2

4.4.3 Experimental results and model performance comparisons

Comparison of the model prediction accuracy

In Table 4.11 below, the performance of the models in forecasting the time series of the loading routes is shown. Overall, all models can accurately predict the whole loading path, and it has proven feasible to forecast the complete loading path using just 10 percent of the available data. This gives a very effective way for forecasting the foundation's three-dimensional response.

In Figure 4.23, all of the predictions are comparable to the actual values produced by Abaqus

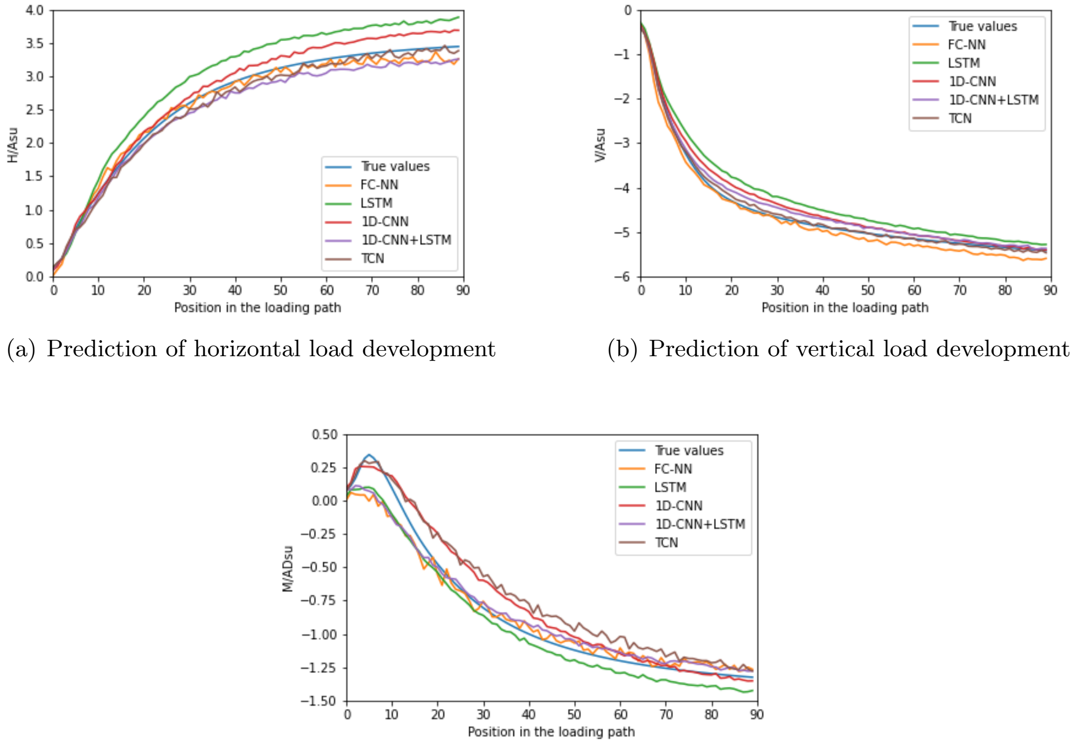
Table 4.11: Prediction results of the five models

	FC-NN	LSTM	1D-CNN	1D-CNN+LSTM	TCN
RMSE	0.159	0.103	0.140	0.085	0.071
R^2	0.993	0.997	0.995	0.998	0.999
MAE	0.086	0.073	0.100	0.052	0.046

and exhibit the same load response pattern. FC-NN and TCN provide the most accurate forecasts for H. From a numerical perspective, more than half models underestimate the horizontal bearing capacity of the foundation slightly, while LSTM and 1D-CNN overestimates it. For the prediction of V, most models overestimated slightly the load response before ultimate vertical bearing capacity reached, except for FC-NN which underestimates the response in the second half of the prediction. And only TCN learned the vertical response under load perfectly, with the projected curve almost covering the actual curve. In the forecasts for M, the 1D-CNN and TCN are the closest to the actual values in the first 15 points, with the slightly overestimating the ultimate load capacity. The rest models are closer to the true value in the back part and the slightly underestimate the loading development process. The M loading path has a point of steering, which can be explained by the fact that for small rotation the negative moment is balanced by the greater positive horizontal force, thus detecting the opposite moment at the reference point. As the negative moment is progressively applied, the curve undergoes a direction shift. From the prediction curves of V, it can be seen that the beginning and end of the model forecasts are pretty accurate, however the middle process prediction exhibits more variances. The accuracy of the prediction at the beginning is due to the proximity to the input data range, which has more reference. Accurate prediction at the end is owing to the unequal distribution of the data points, as shown in Figure 4.22, where the data points are spread more densely at the latter points. The growth and yield prediction of each loading curve are concentrated between the 1st and the 40th data. The remain of the curve is more intensive and deemed to have received "reinforced training." The predicted error is thus underestimated and the predictions from the 1st to the 40th points in each path should be focused more. From the data, the TCN demonstrates the best predictions for all loading paths, but in the actual curves, the other models' prediction errors for the behind of the loading path are more minor than in the figure. The model overall performance is TCN>1D-CNN+LSTM>LSTM>1D-CNN>FC-NN. This result is consistent with the expectation that TCN and 1D-CNN+LSTM have a more complex model structure and should have better prediction results. The LSTM, which is an 'expert' in dealing with the temporal domain, can also provide good computational results.

Comparison of the model training process

Because this is a controlled experiment, the structure and parameters of each model are kept as close as possible. Hence, the hyperparameter settings of some models are not optimal, it can be verified by the training process of the models. Figure 4.24 below depicts the trajectory of MSE Loss during training for the five models. Because the number of training epochs had to be regulated, the early stop parameter was not set. Several large fluctuations can be seen in the training process of the LSTM and 1D-CNN+LSTM, indicating that the learning rate has undergone adaptive changes at these epochs. This also implies that the stability and repeatability of the model will be relatively poor. Because there is a high probability that



(a) Prediction of horizontal load development

(b) Prediction of vertical load development

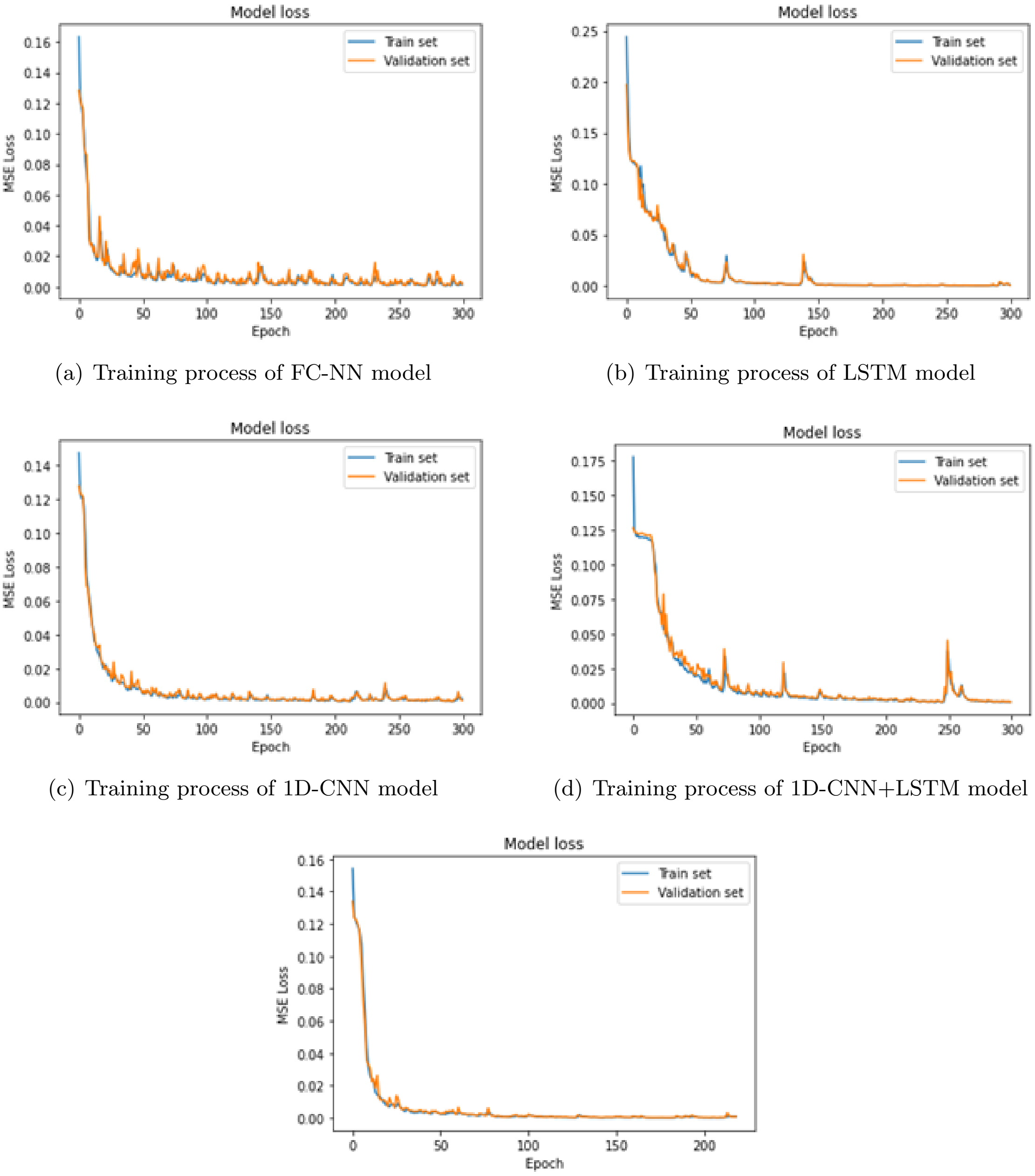
(c) Prediction of moment development

Figure 4.23: Prediction results (90 points) on H, V and M respectively in a random path

the training will stop in fluctuations. The FC-NN and 1D-CNN models, on the other hand, do not show particularly pronounced fluctuations, but rather small continuous fluctuations instead, indicating that the batchsize value is set too small and that there is also partial overfitting. On the contrary, the TCN is devoid of substantial fluctuations due to the usage of residual blocks, which significantly minimizes the likelihood of its overfitting.

Comparison of the model computational efficiency

The number of parameters and expense of training for each model are shown in Table 4.12 below. Theoretically, the more parameters a model has, the more complicated and difficult the training becomes. Due to the LSTM model's gating mechanism, its memory footprint is huge and its training time is significantly lengthened. The LSTM contains half the number of neurons of a 1D-CNN model, but more internal parameters, which slows down training. 1D-CNN+LSTM is a 1D-CNN model with an LSTM layer superimposed on top. The total number of parameters is twice that of 1D-CNN, and the training expense is double. This shows that combining with CNN can be an excellent solution to improve the shortcomings of LSTM's low computational efficiency. The FC-NN has the fewest parameters and is the simplest model with the highest computational efficiency, whereas the TCN has the most complex model structure, but the number of parameters does not change significantly due to the application of dilation convolution, indicating that the model is well optimised. In



(a) Training process of FC-NN model

(b) Training process of LSTM model

(c) Training process of 1D-CNN model

(d) Training process of 1D-CNN+LSTM model

(e) Training process of TCN model

Figure 4.24: Training process of five models

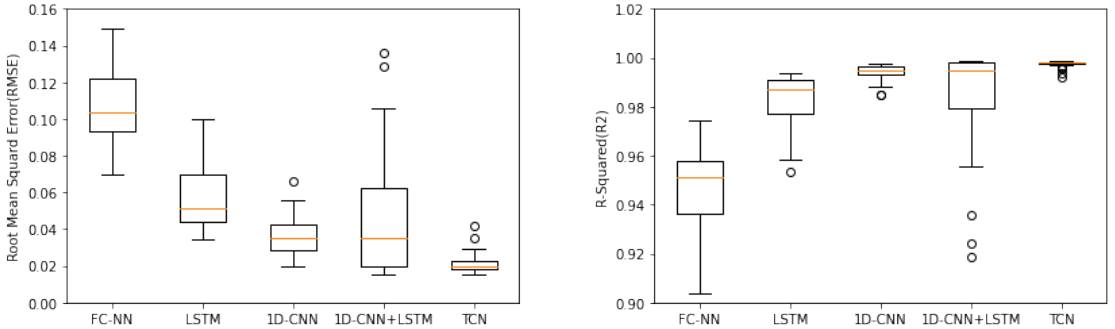
summary, FC-NN has the smallest number of parameters and is the most computationally efficient algorithm. Models using gating mechanisms are much less efficient and have more parameters. The model based on convolutional layers has an intermediate amount of parameters and processing efficiency compared to the previous two networks.

Comparison of Reproducibility and Robustness of models

Table 4.12: The number of parameters and training time in each model

	FC-NN	LSTM	1D-CNN	1D-CNN+LSTM	TCN
Number of parameters	930,216	1,384,040	1,305,064	2,431,880	1,419,432
Training time	27s	387s	41s	72s	43s

In order to remove the influence of contingency on the model, an additional 30 repeat experiments were performed. Early stop (minimum change of 0.00001, patience of 50 epoch) was used to enhance computing performance and prevent overfitting. The training is ended early if the value change of MSE Loss is smaller than 0.00001 within 50 epochs. The results of the experiments are shown below (in Figure 4.25). The red line reflects the median error over 30 repetitions, while the size of the box indicates the distribution of model predictions. LSTM training is contingent and its results from repeated tests are inferior to those of one-dimensional convolution model. The inclusion of LSTM did not increase the accuracy of 1D-CNN+LSTM model, but rather made the computational outputs less robust. This is due to the fact that LSTM models contain superfluous parameters, which are not conducive to training. The models based on 1D convolution have higher computational efficiency and stability. In particular, the TCN is very accurate and robust in its distribution. The FC-NN model can be used in place of the LSTM to solve this problem, as the LSTM has no discernible advantage. This is owing to the fact that the experiment's time series has just 10 historical data points. Compared to long short-term memory, which LSTM excels at, this experiment demands short-term memory, which hinders LSTM's effectiveness. On the other hand, the 10 historical data are ideal for a 1D-CNN with a field of view of 3. In addition, the capacity of TCN to employ dilation convolution to compensate for 1D-field CNN's of view deficiencies was not utilized. The highest computational stability of TCN is due to the use of residual blocks, and the highest computational accuracy due to the fact that each residual block is equivalent to two layers of convolutional computation, for a total of six layers of convolutional computation. This is supposed to be more accurate than the 1D-CNN model with only two hidden layers. In this scenario, the LSTM loses to the CNN represented by the TCN.



(a) RMSE for the five model predictions

(b) R^2 for the five model predictions

Figure 4.25: Prediction error distribution for multiple models under 30 random seeds

4.5 Section conclusion

This chapter presents a comprehensive study on developing AI-based design model for the suction caisson foundation in clay. To address this problem, two different training strategies were proposed: 1. point-to-point mapping, wherein the deep learning model was utilised to discover the nonlinear relationship between displacement and force. 2. line-to-line mapping, in which a deep learning model was employed to predict the relationship between response forces at various moments. Correspondingly, three main neural networks, including five deep learning models were employed. The performance of the different models was compared in terms of prediction accuracy, computational efficiency and robustness. It was found that both training strategies work well in predicting the behaviour of foundations. Among all the deep learning models, the FC-NN model is most efficient in capturing the relationship between displacement and load. By applying the same hyperparameter setting to the database of the suction caisson in sand from Zhang et al. (2020b), it was found that the FC-NN model works not only for foundation in clay of this study but also that in the sand. When it comes to the temporal prediction of the whole loading path, the TCN model shows the highest accuracy and robustness.

5 Model generalisation

5.1 Introduction

The exceptional mapping ability of neural networks has been demonstrated in the preceding section on the training data. However, it should be noted that the ”perfect” prediction on the training data can also result in over-fitting, making the model inaccurate when applying to the data outside the training one. An idea model should be stable with the perturbation. In general, in addition to the accuracy of the prediction results, robustness and generalisation are crucial measures of a model's quality. Basically the robustness is the redundancy of the model in terms of the sensitivity of prediction to the slight change of model parameter, while generalisation is the accuracy of the model when being applied to predict the new range and data. High robustness means that the system remains stable and effective when some parameters deviates slightly from the optimal value. And generalisation ability implies that the network model derived from a limited sample has strong prediction potential for additional variable domains. Good generalisation has the capacity to passively accept many external inputs without actively altering or damaging the model, and achieve the intended outcomes. In Chapter 4, the model performance has been validated against the data in test sets. The excellent robustness of the FC-NN based AI model is demonstrated by the repetitive experiments. Moreover, the good results when applying the model on Zhang's experimental data proves its generalisation ability. However, it should be noted that the validation in preceding section is an interpolation calculation. For example, the model was trained by the data from cases with L/D ratios from 0 to 1 and tested against the case with L/D ratios from 0 to 1 with new input. To further test the generalisation capability of the deep learning model, extrapolation calculations were performed. Furthermore, it was found the variation of model generalisation capability can be explained through the change of intrinsic failure mechanism of suction caissons.

5.2 Interpolation experiments to analyse errors at different embedment depths

Although many studies have been conducted to apply the machine learning technique in the geotechnical domain, only 10.64% of them tested both the interpolation and extrapolation predictability of the models (Zhang et al., 2021). So what are interpolation and extrapolation experiments? Basically, if the trained deep learning model is a student, interpolation and extrapolation are two kinds of exams as an analogy. The interpolation experiment tests the deep learning model on the questions the model has already learnt. The exam questions are probably the same type of questions as trained but with different values. On the other hand, the extrapolation experiment requires the model to figure out the questions it has never been exposed to before, which is expected to be solved using the knowledge (algorithm) that model has learned. These two types of tests are critical for a comprehensive evaluation of the model's applicability.

This chapter continues the work in Chapter 4 by examining the model's non-linear regression predictions of the relationship between force and displacement at each embedded depths in homogeneous soil. By taking 80% of the data as the training set and the remaining 20% as the test set. After averaged, the test data in each depth of burial accounts for 2% of the total data (i.e. around 1920 data points). Since the training set already contains data from each burial depth, this is an interpolation test. The data was subjected to an inverse normalisation calculation to obtain the true error as shown in Figure 5.1.

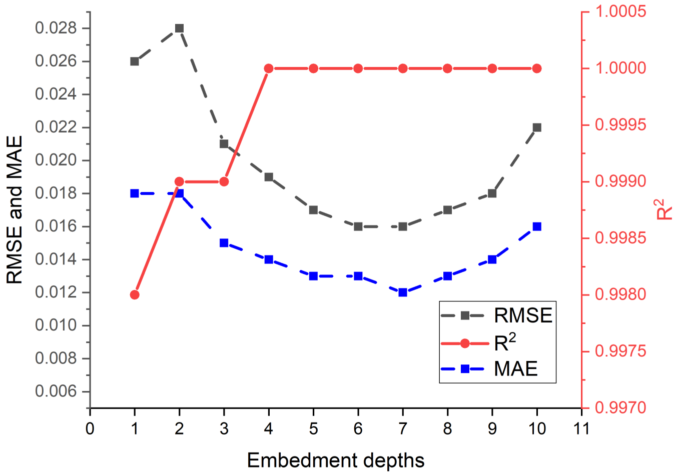


Figure 5.1: RMSE, R^2 and MAE of test prediction at different embedment depths

In the figure, RMSE and MAE show the same trend. RMSE and MAE can both determine the calculated error between the predicted and true values of the model, where RMSE is more sensitive to outliers in the model prediction and magnify the proportion of large errors in the total error. So the calculated error of RMSE is generally larger than that of MAE. Coefficient of determination (R^2) is used to indicate the fit of the model. When R^2 close to 1, it indicates the deep learning model has been fitted perfectly, i.e. the predicted trend is highly interpretable and the mechanism is same as the intrinsic mechanism of foundation. From the definition, the RMSE and MAE represent the straightforward errors generated in the predictions (i.e. error between predicted and true values). While the R^2 does not provide a direct error between predicted and true values, but a good indication of the fit of the model intrinsic mechanism.

Based on the analysis, the error in the RMSE and MAE calculations can be decomposed into three components. The most significant error is the error in the model's prediction of the failure mechanism. The prediction error of the failure mechanism can be further divided into the basic prediction error and the mechanism change error. The former error is the prediction error due to the structure of the neural network, which is constant for all depths of burial and unavoidable. The latter error is the prediction error due to changes in the intrinsic mechanism, so the mechanism prediction error is expressed mainly through the mechanism change error. Thanks to the coefficient of determination (R^2), which correlates well with the degree of model failure mechanism fit. Consequently, this portion of the error is analyzed primarily by R^2. The value of R^2 shows that the deep learning algorithm has learnt the failure mechanism well at large burial depths (when $L >= 4m$, $R^2 = 1.000$). While in the range of $L <= 4m$, R^2 increases continuously and RMSE and MAE decrease with increasing burial depth. When $L >= 4m$, R^2 is almost constant, indicating that the failure mechanism of foundation does not vary much. However when $L <= 4m$, the failure mechanism of foundation is not the same failure mechanism as when $L >= 4m$ and the intrinsic mechanism at small burial depths is constantly changing. The details of this mechanism change are researched in the next section. Based on R^2 in Figure 5.1, the value of which stays at 1 after $L > 4m$, it would be a misconception that the failure mechanism of the foundation is constant after $L > 4m$ and no additional mechanism change error trigger. In reality, however, the predicted failure mechanism error for the foundation still varies slightly, as only three decimal places are shown so R^2 is rounded to 1. This deduction will be demonstrated later. R^2 reaches its maximum value at $L = 10m$ and has the smallest mechanism prediction error, and error increases as the depth of burial decreases.

The second part of the error is mainly due to data boundary effects. When predict $L >= 4m$, the fit of the model is already reaching the optimum and the coefficient of determination is equal to 1. At large burial depths, the comparison is more about the variability of the RMSE and MAE data. The RMSE, for example, drops to the minimum value of 0.016 at $L = 7m$ and then rebounds, with RMSE gradually increasing to 0.022 at $L = 10m$. The phenomenon can't prove the error induced by mechanism change, as R^2 is still 1.000. This is because when the test set approaches the training set's data boundary, less data is available for the deep learning algorithm to train on, leading to an increase in the inaccuracy of the prediction results. The most obvious manifestation of this is the appearance of larger errors in the MAE and RMSE for large and small burial depths than for intermediate burial depths. The overall trend of RMSE and MAE exhibit a shape of the arc. This boundary effect is mainly observed in tests with $L = 1, 2, 9, 10m$, and the effect sharply vanish as the training set data increases.

Size effects account for the last portion of the prediction error. This is due to the fact that as the depth of burial grows, so does the bearing capacity of the foundation. The RMSE and MAE obtained in the figure reflect the actual errors after the inverse normalisation adjustment. Hence, the larger the burial depth, the more significant the error generated from the size effect. From $L = 5m$ to $L = 6m$, the MAE is almost constant, but the prediction error from the change in mechanism is actually decreasing and the boundary effect error has not yet developed. This is because the increasing size effect error balances the decreasing mechanism prediction error. Also, there is no significant change in MAE from $L = 1m$ to $L = 2m$. However, in reality, both the boundary effect and the mechanism prediction error are decreasing, which is also balanced by the size effect error.

Finally, MAE is plotted as an example to represent the prediction error's in three parts (in Figure 5.2). The three errors are plotted qualitatively with indeterminate quantities. The red line represents the prediction error for the mechanism change, the green line represents the data boundary effect error, and the blue line represents the error arising from the size effect.

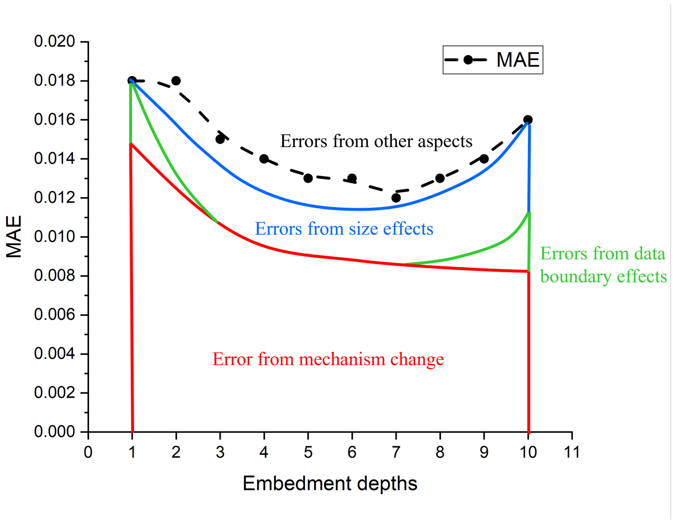


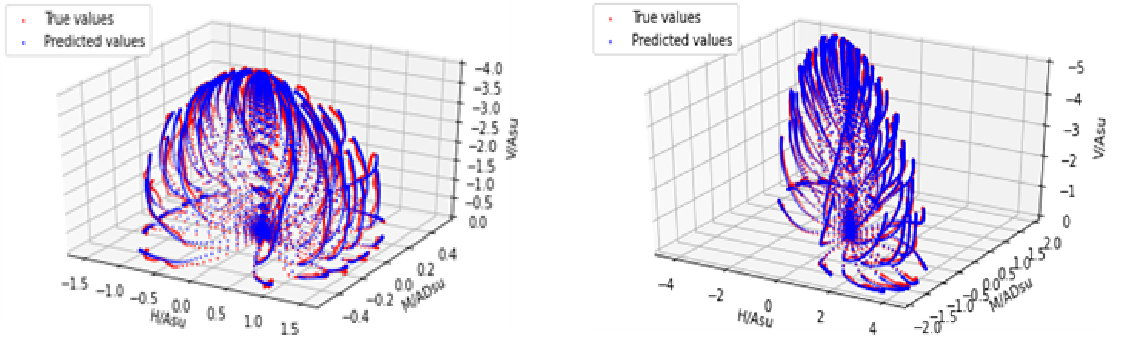
Figure 5.2: Three main components of MAE at different embedment depths

5.3 Extrapolation experiments to detect intrinsic mechanism changes at different embedment depths

In the interpolation experiment, R^2 captures the change in foundation failure mechanism very well. However, the constant R^2 value of 1 after $L > 4m$ does not establish that the failure mechanism of the foundation changes significantly from $L = 4m$. This led to the design of extrapolation experiments in an attempt to find the transition point in the intrinsic mechanism of the foundation. The extrapolation experiment was first used to verify the conclusion reached in the previous section that the deep learning model can capture the intrinsic mechanism of the foundation well for large burial depths, and is relatively poor at predicting mechanism for small burial depths.

In prior interpolation experiment, the 80% training set and the 20% test set were randomly split according to a preset ratio. While in extrapolation experiment, a specific buried depth was artificially divided as the test set. The deep learning model, now trained on the rest of the embedment depths data, will be evaluated to determine whether it can accurately predict the entire behaviour of foundation at a particular depth of embedment. This corresponds to utilizing 90% of the data for the training set and 10% of the data for the test set. Although there are more trained data, the model still faces a significant challenge because it lacks data for a particular depth. Only by identifying the underlying failure mechanism correctly from existing data, network can make precise predictions. The data from $L = 2m$ representing small burial depths and $L = 7m$ representing large burial depths were independently evaluated in Figure 5.3.

In Figure 5.3, it is evident that the prediction results for the data buried at a depth of $7m$



(a) Prediction by other embedded depths data trained model at $L = 2m$ RMSE: 0.033 R^2: 0.995 MAE: 0.024

(b) Prediction by other embedded depths data trained model at $L = 7m$ RMSE: 0.038 R^2: 0.999 MAE: 0.025

Figure 5.3: Prediction by other embedded depths data trained model at $L = 2m$ and $L = 7m$

are superior than those embedded at a depth of 2m. R^2 remaining at 1 at a 7m burial depth implies that the predictions are pretty precise, verifying the conclusions from the last section. But at $L = 7m$, the value of RMSE increases from 0.016 in the interpolation experiment to 0.038 and MAE increases from 0.012 to 0.025. Meanwhile, R^2 at $L = 2m$ decreases from 0.999 to 0.995, and both RMSE and MAE increase. The predicted value of the extrapolation experiment is obtained by the generalisation ability of the model, and the accuracy of the prediction also decreases. This phenomenon indicates that the extrapolation experiments are more challenging for the model than the interpolation experiments.

Based on prior knowledge in Chapter 3, it is known that the failure mechanism varies with the embedment ratio. The experiment result indicates that the model only learns the particular failure mechanism at large burial depths and is insensitive to variation in mechanisms. Since the foundation's failure mechanism is continuous over a range of burial depths, i.e. the failure mechanism at a specific burial depth is close to the failure mechanisms at the two adjacent embedded depths. Hence, the deep learning model is trained by utilizing data from two neighboring embedded depths to predict the response forces for a particular depth (e.g. training the neural network using data from $L = 1m$ and $L = 3m$ to predict the response force at $L = 2m$). Both homogeneous and heterogeneous soils were tested and the results are shown in Figure 5.4.

In the figure, the heterogeneous and homogeneous soils demonstrate the same pattern of variation, with one poorly fitted positions appearing about $5 - 6m$ of burial depth, respectively. The outstanding fit at burial depth of $4m$ suggests that burial depths of $3m$, $4m$ and $5m$ share the same failure mechanism. Similarly the excellent predictions for burial depths of $7m$ indicate the same failure mechanism for burial depths of $6m$, $7m$ and $8m$. However, the inferior prediction between $L = 5m$ and $L = 6m$ indicates a transition point (i.e. a switch of the failure mechanism) between these two burial depths. According to Section 3.4, the foundation has two main failure mechanisms (scoop and scoop-slide), which are swift with depth change. This mechanism transformation can be estimated reversely from the neural network's generalisation performance if the vector contours of $L = 5m$ and

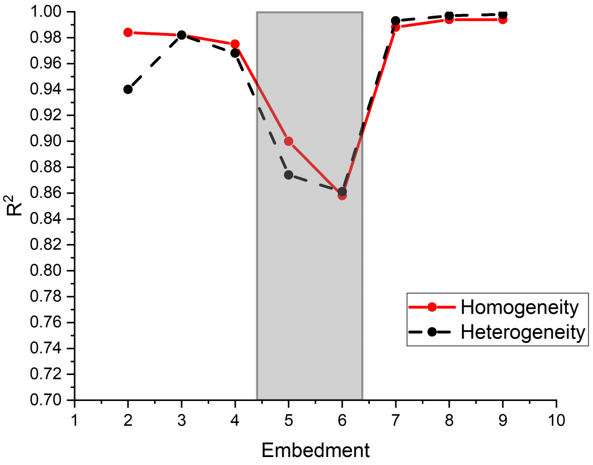
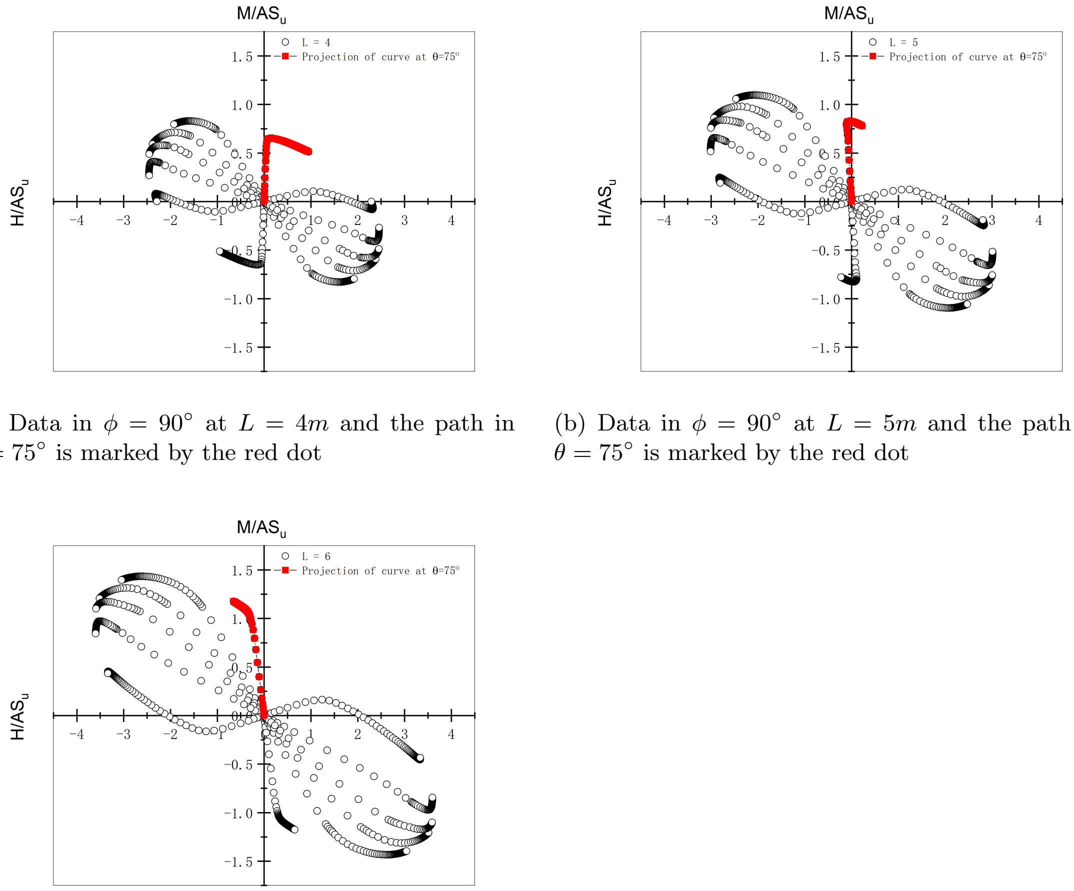


Figure 5.4: Variation of R^2 at different embedment depths in homogeneous and heterogeneous soils

$L = 6m$ prove this mechanism change.

According to the conclusions of Section 3.4, there is a change in the failure mechanism from 1 to $10m$ resulting in a positive and negative alternation in the horizontal forces. Figure 5.5 indicates that this transition should take place between $L = 5m$ and $L = 6m$. At $L = 4m$, the $\theta = 75°$ loading path for develops in a positive direction along the horizontal axis. Moreover, in this direction, the shape of the loading path changes significantly, and the scoop dominates the failure mechanism. When failure happens, the ultimate moment-bearing capacity is first reached. Then the path gradually develops towards the ultimate value of the horizontal force along the positive axis. At $L = 5m$, the loading path is almost perpendicular to the transverse axis, and after reaching the ultimate state of the moment, the path develops slightly along the positive horizontal axis. At this stage, the state of the loading path is quite close to the condition of the uniaxial moment bearing capacity experiment. And when $L = 6m$, the loading path shifts along the negative direction of the transverse axis, and keeps developing towards the negative ultimate horizontal bearing capacity after contacting the failure envelope. As the embedment depth steadily rises, the wedge caused by sliding vanishes at the mudline, and the displacement vector shifts from right (at $L = 4m$ in Figure 5.6(a)) to the left (at $L = 6m$ in Figure 5.6(c)). Therefore, the shift in the general direction of the loading paths make neural network more difficult to predict.

As two main failure mechanisms of foundation exist within the ten embedment depths, the deep learning model can be split into two models to learn these two failure mechanisms separately. It was observed whether the models would improve the foundation intrinsic mechanism predictions. Therefore, 96,000 dataset was divided into 48,000 sets of data when $L <= 5m$ and 48,000 sets of data when $L >= 6m$ to train two deep learning models (i.e. small burial depth model and large burial depth model) respectively. Take training the small buried depth deep learning model as an example, 80% of the 48,000 datasets when L¡5m



(a) Data in $\phi = 90°$ at $L = 4m$ and the path in $\theta = 75°$ is marked by the red dot

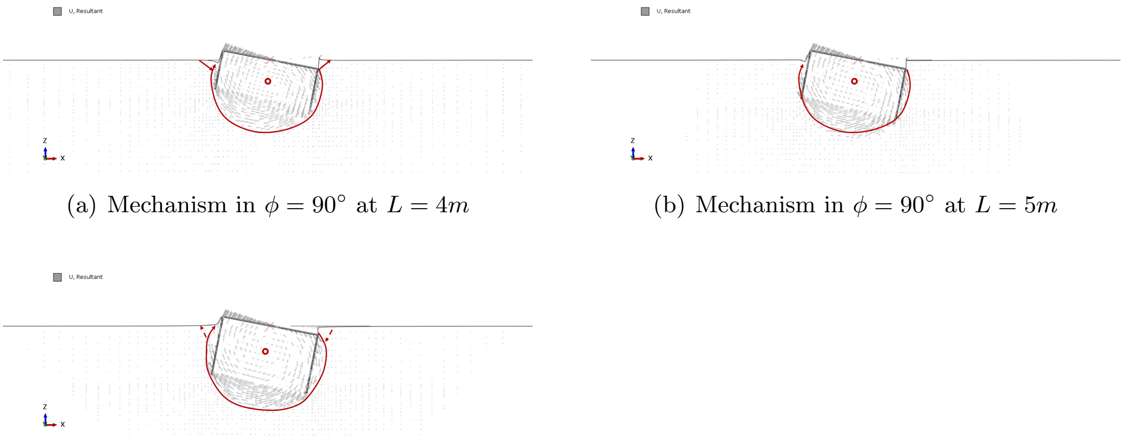
(b) Data in $\phi = 90°$ at $L = 5m$ and the path in $\theta = 75°$ is marked by the red dot

(c) Data in $\phi = 90°$ at $L = 6m$ and the path in $\theta = 75°$ is marked by the red dot

Figure 5.5: Comparison of data at three embedment depths

were randomly selected as the training set, and the remaining 20% of the data were used for interpolation test. The small burial depth model was also used for extrapolation experiments to predict the failure mechanism of the foundation with large burial depths (6-10m). The prediction is illustrated in Figure 5.7.

In the figure, the computational interpolation experiments result for both models are the same as the predictions in Figure 5.1, indicating that splitting one model into two smaller models to predict the foundation mechanism at different burial depths separately does not improve. Meanwhile, the prediction accuracy of both models gradually decreased as the test set moved away from the training set range when doing the extrapolation experiments. This is plausible given that both models have only learnt a single failure mechanism and Lacks the capacity to predict the other mechanism accurately. It is also worth pointing out that the small burial depth model's extrapolation predictions are superior to those of the large burial depth model. The large burial depth model exhibits a negative R^2 when predicting the foundation behavior at $L = 1m$, indicating that the model's predictions are extremely weak. This phenomenon



(a) Mechanism in $\phi = 90°$ at $L = 4m$

(b) Mechanism in $\phi = 90°$ at $L = 5m$

(c) Mechanism in $\phi = 90°$ at $L = 6m$

Figure 5.6: Comparison of mechanism at three embedment depths

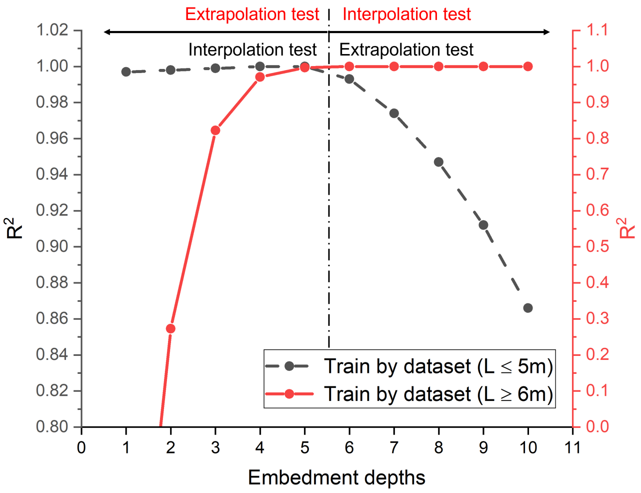


Figure 5.7: Mechanism prediction under different embedment depths by two deep learning models

also indicates that the mechanism changes faster in small burial depths than in large burial depths, and the rate of mechanism change decreases with increasing burial depth. Also the interpolation experiments of the small burial depth model are still lower than the predicted results at $L = 1m$ than at $L >= 2m$, demonstrating the rapid change of the mechanism here.

5.4 Extrapolation experiments to evaluate model generalisation ability

To further test deductions in Section 5.3 and the generalisation ability of the model. With burial depths of $L = 2.30m$, $L = 4.50m$, and$L = 7.80m$, three new data sets were generated.

At each burial depth, twelve completely new directions with $\theta = 37°, 127°, 217°$ and $307°$ respectively and $\phi = 40°, 70°$ and $83°$ were chosen at random. This represents a new burial depth and loading direction for the trained model. There are 100 data points in each load combination and 1200 data points for each embedment depth.

The predictions at $L = 2.30m$, $L = 4.50m$, and $L = 7.80m$ are shown in Table 5.1. The deeper the burial depth, the better the prediction outcomes, demonstrating that the AI model learns the failure mechanism in large burial depths more again. By comparing the generalisation in different directions at the same burial depth. It can be found that there is no obvious pattern between different θ predictions. And when $\phi = 40°$, the accuracy of the prediction is significantly lower than the other two ϕ values, which is caused by the uneven distribution of the data, resulting in a smaller amount of training data when ϕ is smaller.

Table 5.1: Generalisation prediction error at three new test sets

Data	Direction	θ=37°	θ=127°	θ=217°	θ=307°
$L = 2.3m$	ϕ=40°	RMSE: 0.012 R^2: 0.470	RMSE: 0.015 R^2: 0.883	RMSE: 0.015 R^2: 0.765	RMSE: 0.019 R^2: 0.644
	ϕ=70°	RMSE: 0.016 R^2: 0.959	RMSE: 0.025 R^2: 0.985	RMSE: 0.028 R^2: 0.962	RMSE: 0.015 R^2: 0.979
	ϕ=83°	RMSE: 0.026 R^2: 0.443	RMSE: 0.023 R^2: 0.996	RMSE: 0.030 R^2: 0.853	RMSE: 0.023 R^2: 0.977
$L = 4.5m$	ϕ=40°	RMSE: 0.022 R^2: 0.938	RMSE: 0.020 R^2: 0.947	RMSE: 0.018 R^2: 0.946	RMSE: 0.019 R^2: 0.969
	ϕ=70°	RMSE: 0.019 R^2: 0.996	RMSE: 0.019 R^2: 0.997	RMSE: 0.020 R^2: 0.994	RMSE: 0.015 R^2: 0.997
	ϕ=83°	RMSE: 0.022 R^2: 0.992	RMSE: 0.019 R^2: 0.999	RMSE: 0.051 R^2: 0.978	RMSE: 0.023 R^2: 0.998
$L = 7.9m$	ϕ=40°	RMSE: 0.027 R^2: 0.997	RMSE: 0.029 R^2: 0.996	RMSE: 0.024 R^2: 0.992	RMSE: 0.046 R^2: 0.979
	ϕ=70°	RMSE: 0.032 R^2: 0.997	RMSE: 0.038 R^2: 0.998	RMSE: 0.041 R^2: 0.995	RMSE: 0.038 R^2: 0.997
	ϕ=83°	RMSE: 0.052 R^2: 0.995	RMSE: 0.028 R^2: 0.999	RMSE: 0.064 R^2: 0.995	RMSE: 0.029 R^2: 0.999

The small burial depths model and large burial depths model were trained artificially by splitting the data into small burial depths (L <= $5m$) and large burial depths (L >= $6m$) according to the mechanistic variation between burial depths of 5m and 6m. Theoretically, better predictions should be obtained. In Figure 5.7, this "targeted training" of splitting a model into two smaller models to learn the two failure mechanisms separately did not improve in the interpolation test. However, this experiment did improve the generalisation ability of the model at partial burial depths. In particular, the following table 5.2 was obtained when the small burial depth model was used to evaluate generalisation ability at $L = 2.3m$. The coefficient of determination (R^2) of the prediction on $L = 2.3m$ data improves from 0.993 to 0.996, and the RMSE and MAE both decrease. However, the improvement is still concentrated at larger ϕ rather than $\phi = 40°$. In contrast, the generalised prediction performance of test sets at $L = 4.5m$ and $L = 7.9m$ decreased to some degrees. This phenomenon shows that the generalisation ability at$L = 4.5m$ is influenced by the data at large burial depths and that the generalisation ability of $L = 7.9m$ is influenced by the data at small burial depths. Therefore, the mechanism of foundation cannot be simply divided into two distinct failure mechanisms. In addition, there is a connection between these two mechanisms, which shift progressively with depth.

Table 5.2: Generalisation abilities at $L = 2.3m$ after training by divided data (Better performance is viewed in green and same performance is viewed in light green)

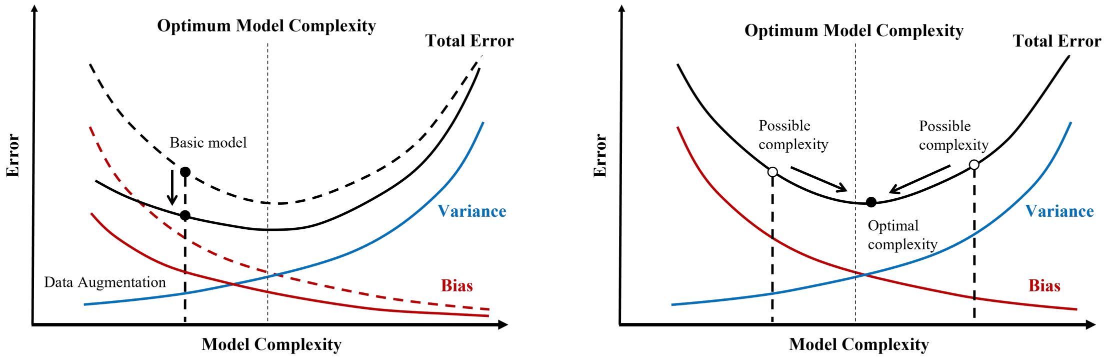
Data	Direction	θ=37°	θ=127°	θ=217°	θ=307°
Normal	ϕ=40°	RMSE: 0.012 R^2: 0.470	RMSE: 0.015 R^2: 0.883	RMSE: 0.015 R^2: 0.765	RMSE: 0.019 R^2: 0.644
	ϕ=70°	RMSE: 0.016 R^2: 0.959	RMSE: 0.025 R^2: 0.985	RMSE: 0.028 R^2: 0.962	RMSE: 0.015 R^2: 0.979
	ϕ=83°	RMSE: 0.026 R^2: 0.443	RMSE: 0.023 R^2: 0.996	RMSE: 0.030 R^2: 0.853	RMSE: 0.023 R^2: 0.977
Divided	ϕ=40°	RMSE: 0.013 R^2: 0.365	RMSE: 0.020 R^2: 0.858	RMSE: 0.017 R^2: 0.191	RMSE: 0.010 R^2: 0.963
	ϕ=70°	RMSE: 0.010 R^2: 0.978	RMSE: 0.024 R^2: 0.978	RMSE: 0.017 R^2: 0.972	RMSE: 0.026 R^2: 0.982
	ϕ=83°	RMSE: 0.033 R^2: 0.848	RMSE: 0.023 R^2: 0.996	RMSE: 0.024 R^2: 0.914	RMSE: 0.026 R^2: 0.987

5.5 Improvement of generalisation

The previous section has demonstrated that the failure mechanism of foundation is inseparable and can not improve generalisation results at different depths by splitting the deep learning model into two models. This section will explore more approaches to improve the generalisation of the model.

According to the theory, regularization is one of the primary ways to enhance neural network generalisation. Regularization improves generalisation, e.g. by utilizing L1 Norm and L2 Norm regularization, by restricting the complexity of the model and preventing overfitting. The effect of L1 Norm and L2 Norm regularisation is often less pronounced in deep learning models than in shallow machine learning models, especially in the case of over-parameterisation, where the number of model parameters is much larger than the number of training data. Therefore, when training deep learning models, other regularisation methods are often used, such as data augmentation, early stopping, dropout, integrated methods, etc. Since preventing model overfitting improves the model's capacity for generalisation in a sense. Therefore, the methods, such as early stop, change of batch size, and the Adam optimiser, are employed in Section 4.3.2 to prevent overfitting, indicating that the model's capacity to generalise has already been enhanced. In addition to the regularization methods already used, there are two more feasible methods for improving generalisation (i.e. data augmentation and model complexity change). Therefore, this section will further utilise data augmentation and model complexity change to enhance the model's generalisation ability.

Variance and bias are the crucial components of error, which evaluate fitness and precision of model respectively (Fortmann-Roe, 2012). Contrary to variance, bias diminishes as model complexity grows. The bias and variance of the two improvement methods are illustrated in Figure 5.8.



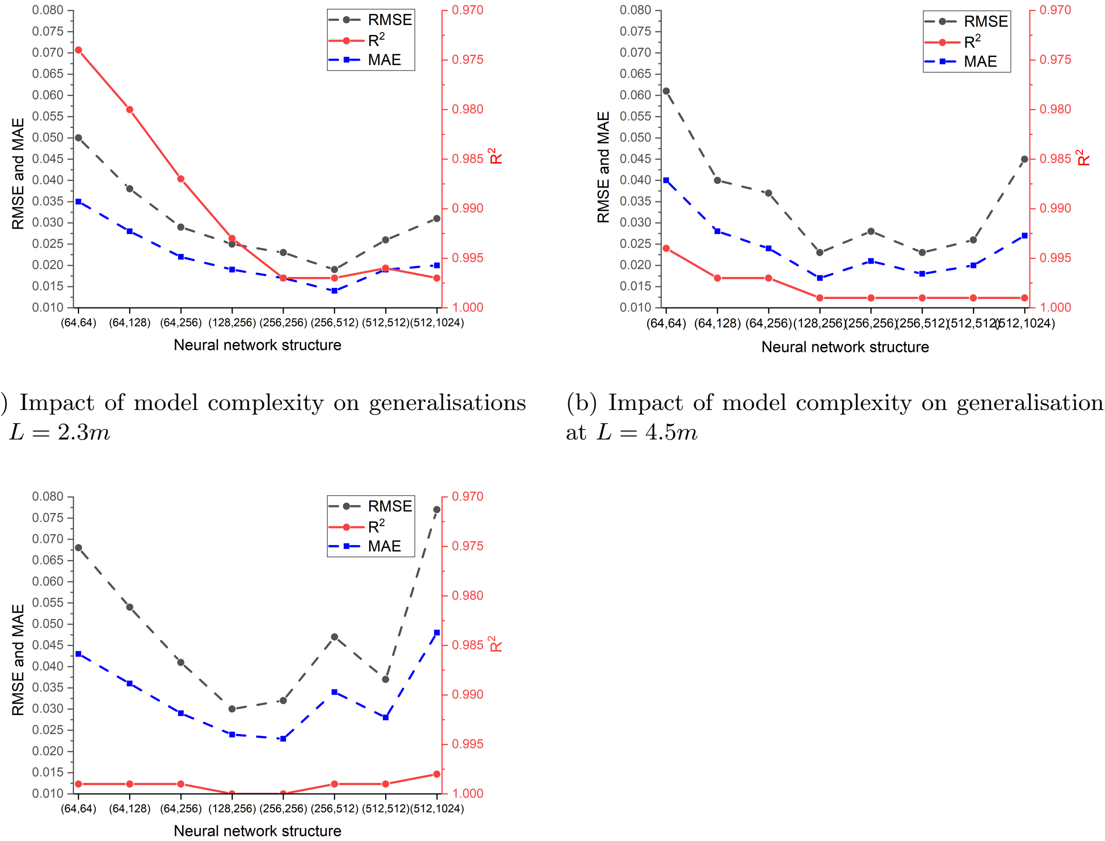
(a) Change in error after data augmentation

(b) Change in error after model complexity change

Figure 5.8: Bias and variance contributing to total prediction error in two improvement methods

Data augmentation

In idealisation, a smaller bias exists and the variance changes negligibly in the generalisation results after data augmentation. To demonstrate the error components in the original model, the total error and bias curve are displayed in dashed lines. After data enhancement, prediction errors are shown with solid lines. This approach does not alter the model's complexity (in Figure 5.8(a)). This reduction in inaccuracy is mostly attributable to the fact that data boundary effects are reduced with data augmentation (i.e. it minimises prediction bias by increasing the number of trainable data points), and data augmentation supports the model with more data references. However, the variance of the prediction results from the additional data remains unchanged. Hence the variance of the model does not improve significantly. In the three generalisation test datasets, the prediction results at $L = 7.9m$ and $L = 4.5m$ were better, while the prediction results at $L = 2.3m$ were relatively poor. So the approaches to improve the generalisation results is not only to improve the prediction results at $L = 2.3m$, but also guarantee the good prediction performance at $L = 4.5m$ and $L = 7.9m$. Deep neural networks require a substantial amount of training data in order to produce more accurate results. When the data is accessible, data augmentation can be used to strengthen the model's robustness and prevent overfitting. In particular, when the embedment depth is 2.3m, the generalisation ability is constrained due to the small amount of data available. Therefore, the 19200 training data at $L = 1.5m$ and $L = 2.5m$ are added to improve the generalisation ability. After the data enhancement, the generalisation capability of the network was not significantly improved at $L = 2.3m$. Meanwhile the prediction accuracy at both $L = 4.5m$ and $L = 7.9m$ has decreased. Because the mechanisms in the small burial depths change so rapidly, data augmentation at small burial depths just hinder the model's ability to capture the specific mechanisms and mechanism change. Therefore, the data augmentation failed, as evidenced by the decline in predictions across all three test datasets.



(a) Impact of model complexity on generalisations at $L = 2.3m$

(b) Impact of model complexity on generalisations at $L = 4.5m$

(c) Impact of model complexity on generalisations at $L = 7.9m$

Figure 5.9: Impact of model complexity on generalisations at three embedment depths

Model complexity change

In contrast, after model complexity increase, the generalisation results are less biased but more variance. This approach increases the number of the neurons or the layers of hidden and the complexity of input and output relationships at the same time. When the model is too complex, its fitting capability improves, and model training becomes more difficult. Each training epoch involves more occasionally. As a direct consequence, the prediction accuracy of the model has improved and the robustness of the model is reduced correspondingly, resulting in a reduced bias and more variance (in Figure 5.8(b)). There are two possible scenarios for this approach; too high or too low a model complexity will result in a large prediction error, so it is necessary to first experiment to determine where the model complexity lies. Then, the most suitable model complexity is chosen to improve the generalisation ability.

The experiments in this chapter were all implemented with a fully-connected neural network with 128 neurons in the first hidden layer and 256 neurons in the second hidden layer, indicated by (128,256). Eight controlled experiments were conducted by increasing and decreasing the neurons in multiples, adjusting the model complexity. The simplest of these neural network structures was (64, 64), and the most complex was (512, 1024), and the results were calculated as shown in Figure 5.9. The three error evaluation metrics show similar trends. The model structure (128, 256) is already very close to the optimal solution. As the model's complexity increases, the model's prediction error shows a trend of first decreasing and then increasing. Of the three evaluation metrics, R^2 provides an excellent response to changes in bias. Since RMSE is more sensitive to variance, the difference between RMSE and MAE also provides an excellent response to changes in variance. The bias and variance illustrate a generally consistent trend with Figure 5.8. Also the variation of RMSE, R^2, and MAE with depth is consistent with Figure 5.1. However, the optimal network structure differs for different burial depths. At $L = 7.9m$, the optimal network structure is (256, 256), and as the depth of burial reduces, the failure mechanism changes more rapidly, necessitating a neural network with a higher degree of complexity to learn it. At $L = 2.3m$, the optimal network structure is (256, 512). The prediction results of the optimal network structure at $L = 2.3m$ in each direction are shown in Table 5.3. In the table, the generalisation ability has been improved in all directions. It shows that increasing the complexity of the model is the most fundamental way to improve the generalisation ability of the model.

Table 5.3: Generalisation abilities at $L = 2.3m$ after training by more complex model (Better performance is viewed in green and same performance is viewed in light green)

Data	Direction	θ=37°	θ=127°	θ=217°	θ=307°
Original model	ϕ=40°	RMSE: 0.012 R^2: 0.470	RMSE: 0.015 R^2: 0.883	RMSE: 0.015 R^2: 0.765	RMSE: 0.019 R^2: 0.644
	ϕ=70°	RMSE: 0.016 R^2: 0.959	RMSE: 0.025 R^2: 0.985	RMSE: 0.028 R^2: 0.962	RMSE: 0.015 R^2: 0.979
	ϕ=83°	RMSE: 0.026 R^2: 0.443	RMSE: 0.023 R^2: 0.996	RMSE: 0.030 R^2: 0.853	RMSE: 0.023 R^2: 0.977
Complexer model	ϕ=40°	RMSE: 0.012 R^2: 0.916	RMSE: 0.009 R^2: 0.971	RMSE: 0.013 R^2: 0.933	RMSE: 0.014 R^2: 0.954
	ϕ=70°	RMSE: 0.014 R^2: 0.978	RMSE: 0.013 R^2: 0.993	RMSE: 0.015 R^2: 0.975	RMSE: 0.016 R^2: 0.986
	ϕ=83°	RMSE: 0.017 R^2: 0.987	RMSE: 0.014 R^2: 0.996	RMSE: 0.017 R^2: 0.955	RMSE: 0.018 R^2: 0.995

5.6 Application of generalisation ability

One of the project's primary objectives is to train the model with existing data and produce a deep learning model capable of predicting the foundation's response under new load combinations. This objective necessitates that the trained deep learning model not only can predict with high accuracy but also the ability to generalise, as it is impossible to have the foundation's response under every potential load combination. The performance of the FC-NN based model has been optimised by the improved generalisation ability in Section 5.5. For a given burial depth and soil strength profile, non-linear regression prediction for the displacement-load relationship can predict the complete load response, including the failure envelope. A caisson foundation with a burial depth of 3m in homogeneous soil is used as a showcase. To better simulate the conditions in practice, $L = 3m$ data is excluded from the training set of the model. The data points on each loading path were not intercepted to obtain the complete loading process. A total of 109,032 data points were obtained for the remaining nine burial depths and were divided into 80% training set and 20% test set. After training, the interpolation prediction error on the test set was $RMSE = 0.004$, $R^2 = 1.000$, $MAE = 0.002$. The generalisation (i.e. extrapolation test) error at $L = 3m$ is $RMSE = 0.004$, $R^2 = 0.998$, $MAE = 0.002$, showing excellent accuracy and generalisation ability. At $L = 3m$, additional 120 θ values and 25 ϕ values were set, making up 3000 new loading directions. After calculating the spherical coordinate equation, the ultimate displacement values at 3000 directions were determined and inputted into the trained model. The predictions were imported into Origin for surface fitting and compared with the true values obtained from Abaqus calculations for 96 directions in Figure 5.10. The grey surface in the figure is the predicted envelope fitted by 3000 points, the true values essentially fall on the surface, indicating that the predictions are very accurate. This demonstrates that the model can fit the foundation behaviour through the foundation configuration and soil strength configuration. The data set of the neural network can also be continuously updated and the model structure is easily adapted. This approach is more precise, adaptable and comprehensive than only concentrating on the foundations' ultimate bearing capacity.

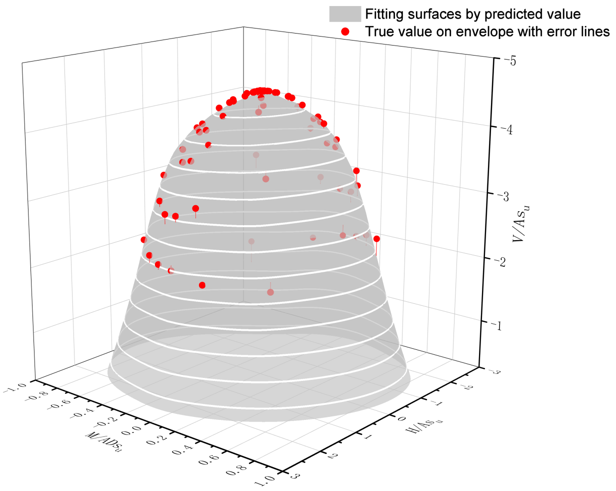


Figure 5.10: Comparison of predicted envelope and true value at $L = 3m$

5.7 Section conclusion

This chapter was an extension of Chapter 4. Further, it analysed the prediction and generalisation ability of the FC-NN model for point-to-point mapping prediction by interpolation and extrapolation experiments. The three main components of the model error were analysed through interpolation experiments on the model. By performing extrapolation tests on multiple model test sets, the results once again proved that the failure mechanism of the foundation changed with the depth of burial. Attempts were also made to enhance the model's generalisation ability by various approaches. To improve the model's generalisation in all loading directions on the new test set, altering the model's complexity is preferable to data augmentation. Finally, the improved model was utilised to accurately predict the failure envelope at a given depth of burial.

Conclusions and recommendations 6

6.1 Conclusions

This study provides two new solutions by developing artificial intelligence (AI)-based data-driven model to better investigate the three-dimensional response of different suction caissons under 3D loads. This two methods predict the mechanical response of the foundation from point-to-point mapping and line-to-line mapping respectively. During research process, geotechical and deep learning findings can be divided into the following points.

Failure envelope and mechanism
To train the neural network for deep learning, Abaqus was used to generate a significant amount of data. The distribution of the H-V-M bearing capacity and the trend of the envelope shape and size were used to validate the simulation results. This modeling's reference point was established at the mudline rather at the skirted tips. This deviation generated from the location of reference points has led to a major modification in the foundation's failure mechanism. A comparison with experimental data from Bransby reveals that this skirted foundation's failure mechanism is more similar to that of a solid foundation. At different embedment depths, the foundation was dominated by the scoop-slide failure mechanism, with few internal scoop mechanism observed. Also, the failure mechanism changes depending on the load combination. The failure envelopes of M-V and H-V do not differ significantly, consistent with existing findings. In the study of the failure process under H-M dimension, the failure mechanism of the foundation near M_{max} and H_{max} is close to a double scoop failure mechanism. Its upper scoop is controlled by the clockwise overturning moment of the foundation, and the lower scoop is generated by the reverse horizontal force, and the two scoops are mutually suppressed. At $\theta = 165°$, the two scoops balance each other, producing a slide-dominated failure mechanism. And the position of the centre of scoop changes as the load combination changes.

Displacement-load relationship prediction
In the prediction of the displacement-load relationship , a FC-NN based surrogate modelling approach is proposed to predict the foundation behaviour, making it possible to acquire a complete loading process by inputting foundation configuration and soil profile. Prediction of the displacement and load relationship by neural networks is more accurate and flexible compared to the traditional general formulation. By training on Zhang's data, this modelling approach has higher computational accuracy and efficiency than the LSTM surrogate modelling method in sand. The robustness and generalisation ability of the model were also evaluated, demonstrating that it possesses high reproductivity, high stability, and good generalisation ability. Even very "shallow" FC neural network model can learn the intrinsic failure mechanisms of the caissons from raw data and predict their nonlinear mechanical responses under complex loading effectively.

Temporal prediction for the loading path

By predicting the loading paths for different load combinations, the behaviour of various foundations is obtained. In the prediction of each loading path, it is feasible to predict the complete path using 10% data points. This kind of time-series prediction has promising applications for foundation monitoring and maintenance. It is possible that workers anticipate the future behavior of the foundation by using the collected data. The TCN model has the highest accuracy in predicting H and V, but the LSTM model excels in predicting M. This indicates that TCN model has great potential for application in the field of time series prediction.

Interpolation and extrapolation tests

Analysing prediction errors at different burial depths by interpolation experiments proves that the total prediction error can be decomposed into mechanism change error, size effect error and data boundary effect error. Changes in the intrinsic mechanism can be inferred between $L = 5m$ and $L = 6m$ by extrapolation experiments based on the performance of the generalisation. Moreover, the FC-NN model can better capture the failure mechanism at large burial depths than at small burial depths. The prediction of failure mechanisms and foundation behaviour at small burial depths cannot be optimised by splitting the deep learning model or data augmentation. The prediction of generalisations in different burial depths and loading directions can only be optimised by modifying the model's complexity. Moreover, the foundation behaviour at small burial depths requires a more complex model structure to predict, as the intrinsic failure mechanisms change faster at small burial depths.

In this research, the generalisation ability analysis is employed to infer the transition point of the intrinsic failure mechanisms. This feasible strategy demonstrates that deep learning not only has the ability to simulate the relationship between inputs and outputs adaptively, but also to mine the intrinsic patterns of data through the generalisation ability. This method provides a new direction of exploration in the geotechnical field combined with AI.

Comparison of deep learning models

In Section 4.4, the performance of different models is evaluated. The FC-NN model is the simplest model and has most computationally efficient. But FC neural network is plagued by temporal issues which can only be alleviated by adding additional neurons and making the model more complex. This approach can't capture long-term memory of the data. The 1D-CNN has the same temporal issue, and its memory term can only be modified by adjusting the size of the convolution kernel. The LSTM model can capture temporal relationships very well. However, in the repetitive experiments, the LSTM prediction results exhibited considerable fluctuations, showing that the model is less robust. 1D-CNN+LSTM model extracts local features using CNN and global features via LSTM, which have good performance in the tests of prediction accuracy. Complexity and difficulty in locating optimum hyperparameters are the only disadvantages of the model. TCN enhances the field of vision by incorporating dilation convolution and decreases model degradation with residual blocks. TCN permits long-term memory and has the greatest prediction accuracy and robustness, challenging the LSTM model in the temporal domain. The models' merits and limitations are summarized in Table 6.1 below. In summary, FC-NNs are already competent for the displacement-load regression tasks, while 1D-CNNs and LSTMs are redundant and make the model's training more challenging. In the time series task, a two-layer LSTM or 1D-CNN model structure is sufficient, while models with more layers are difficult to train. TCNs can avoid the problem of model degradation and can be configured with deeper layers to obtain higher accuracy. Therefore, the TCN model is favored as a convenient method to investigate the load response of foundations.

Table 6.1: Pros and cons of different models(Minus sign means poor performance; Plus sign indicates good performance)

Model	Efficiency	Accuracy	Robustness	Memory	Complexity
FC-NN	++	+	+	−−	++
LSTM	−−	++	−	++	−
1D-CNN	+	++	+	−	+
1D-CNN+LSTM	−	++	+	++	−−
TCN	+	++	++	++	−−

6.2 Limitations

Although this deep learning-based model has demonstrated excellent prediction accuracy, applicability, robustness and generalization, there are still some limitations to the model.

- In Section 4.3.6, the ideal uniform and linear soil profiles are used for training the model. The model performs poorly when trying to predict the foundation's response in the non-linear soil with this trained model. This phenomenon exposes the model's limitations in practical use because the soil strength profile is full of randomness in reality.

- In Section 5.3, extrapolation experiments were performed to examine the intrinsic mechanism changes at different burial depths. In Figure 5.7, when the large and small burial depth models are extrapolated separately to predict other burial depths, the prediction accuracy gradually decreases as the predicted burial depths are far from the training set range. This is the second limitation of deep learning models, i.e., the deep learning model cannot predict the foundation behaviour outside the training range well.

Both of these limitations illustrate that deep learning is highly dependent on the validity and scope of the training set data. In practical problems, obtaining very comprehensive and practical training data is difficult, which limits further learning of the model. Moreover, there are still many areas where the model can be improved and enhanced.

6.3 Recommendations

There are some ideas in the research that have not been realised due to time constraints. These future works will make the model more comprehensive and adaptable to a broader variety of circumstances. These recommendations are given below.

- For the first limitation, a more complex nonlinear soil strength profile can be considered by introducing the parameter of coefficient of variation and CPT data in the model. The second limitation is the common problem with the deep learning method. A feasible way to alleviate the problem is simulating the foundation geometry with embedment ratios greater than 1 to increase the training set range.

- In this Abaqus simulation, only the Mohr-Coulomb constitutive model is used for modeling, however nonlinear models such as the NGI-ADP model may also be employed. Different constitutive models' predictions will be refined by more comparative experimentation data. The soil strength heterogeneity index may be utilized as a learnable parameter for the neural network, rather than simply as a discriminative parameter for whether the soil is homogeneous or not. The dataset may be enhanced. The trained model will be more general and widely adaptable.

- The experiments in laboratory or in-site may detect either displacements or forces as data. Consequently, it is expected that the network will be able to forecast the associated displacements after a given load. In this non-linear prediction, the force may correspond to more than one displacement because the loading path contains the response after foundation failure. This phenomena causes the model to predict force-induced displacements with significant inaccuracy. This issue may be effectively alleviated by truncating the data once the foundation fails.

- In time series prediction, the unequal distribution of data might lead the neural network to focus on the elastic deformation section and ultimate state data, while neglecting the response force developing and yielding. The loading path would be optimized via linear interpolation, resulting in a more uniform data distribution and more precise prediction. The neural network will also pay more attention to the general trend of the path rather than just the beginning and endpoints.

- After the model has been trained, the neural network's parameters may be visualized and fitted formulas in a neural network can be extracted.These fitted formulas will significant assistance in solving geotechnical problems and identifying potential internal relationships between parameters.

Appendix A

A.1 FEM Data

A.1.1 Code for batch production of inp files

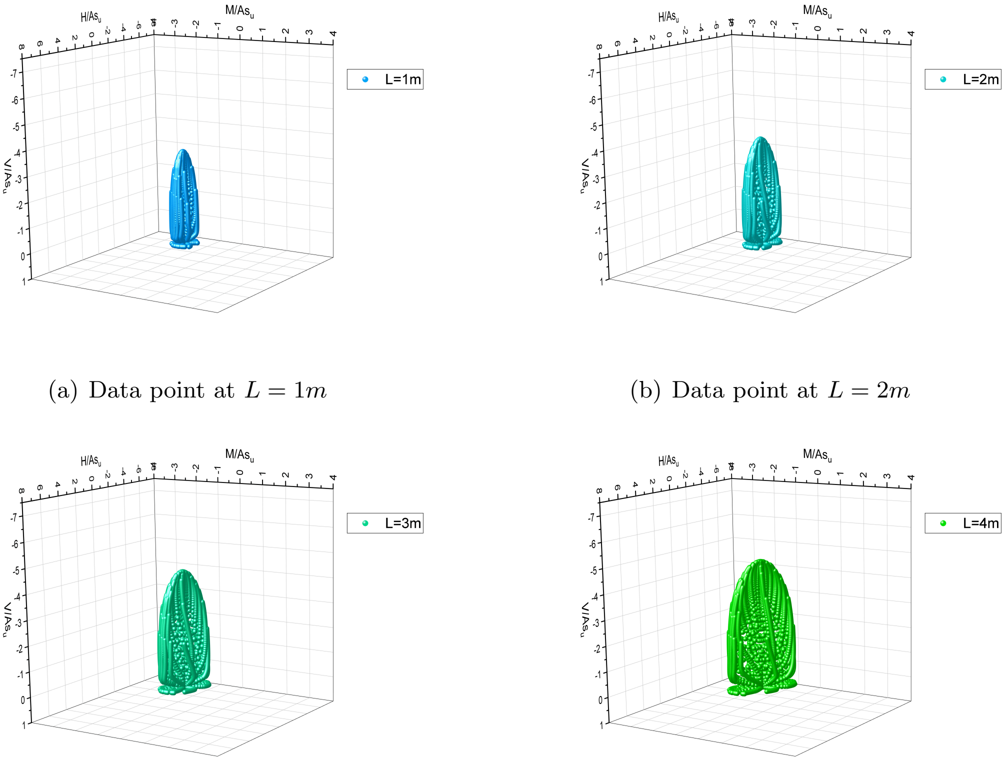
```
function inp_generate3d
clear;
clc;
close all;
phi = [15,30,60,75,80,85,88,90];
theta = 15:30:345;
for i = 1:12
    for j = 1:8
        filename = ['Caisson_1D.inp'];
        filename_new = ['Caisson_1D','theta',num2str(theta(i)),'phi',
           num2str(phi(j)),'.inp'];
        fid_o = fopen(filename);
        tline = fgets(fid_o);
        file_to_open=[filename_new];
        fid_temp= fopen(file_to_open,'w+');
        m = 1;
        n = 1;
        while ischar(tline)
            if m == 100821%H
                tline=['_PickedSet22, 1, 1, ',num2str(0.2*sin(phi(j)/180*
                    pi)*cos(theta(i)/180*pi)),',','\n'];
            end
            if m == 100823%V
                tline=['_PickedSet22, 3, 3, ',num2str((-1)*cos(phi(j)
                    /180*pi)),',','\n'];
            end
            if m == 100825%M
                tline=['_PickedSet22, 5, 5, ',num2str(0.02*sin(phi(j)
                    /180*pi)*sin(theta(i)/180*pi)),',','\n'];
            end
            fprintf(fid_temp,tline,'\n\r');
            tline = fgets(fid_o);
            m=m+1;
        end
        fclose(fid_temp);
        fclose(fid_o);
    end
end
```

A.1.2 Code for Abaqus automatical running

```
clc;
clear;
phi = [15,30,60,75,80,85,88,90];

for i=15:30:345
    for j = 1:8
       temp=num2str(i);
       temp1=num2str(phi(j));
       filename=['Caisson_1_5Dtheta',temp,'phi',temp1];
       disp(filename)
      eval(['dos(''abaqus job=',filename,' cpus=8 int'')']);
      pause(1);
    end
end
```

A.1.3 Data point in Homogeneous soil

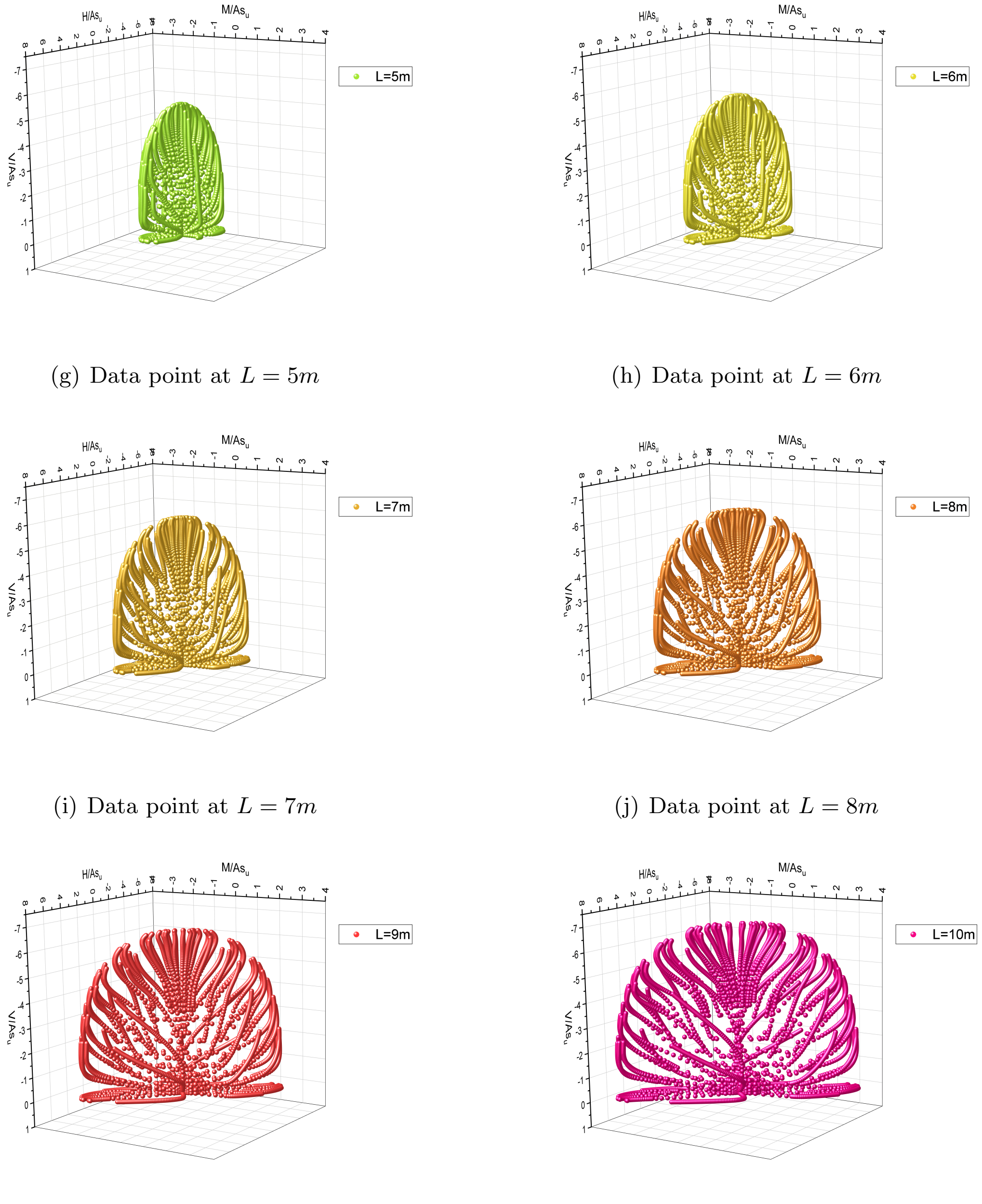


(a) Data point at $L = 1m$

(b) Data point at $L = 2m$

(c) Data point at $L = 3m$

(d) Data point at $L = 4m$



(g) Data point at $L = 5m$

(h) Data point at $L = 6m$

(i) Data point at $L = 7m$

(j) Data point at $L = 8m$

(k) Data point at $L = 9m$

(l) Data point at $L = 10m$

Figure A.0: 3D Data point at $L = 1 - 10m$ calculated by FE analysis

A.2 Deep learning model Code

```
# -*- coding: utf-8 -*-
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from numpy import concatenate
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from keras.models import Sequential
from keras.layers import Dropout
from keras.layers import Dense, Activation
from keras.layers import LSTM
from keras.layers import LeakyReLU
from keras.layers.core import Flatten
import keras.backend as K
from keras.callbacks import LearningRateScheduler
from keras.models import load_model
import torch
import torch.nn as nn
import torch.optim as optim
from math import sqrt
from math import sin
from math import cos
from math import pi
from sklearn.metrics import mean_squared_error,r2_score
import tensorflow as tf
from sklearn.model_selection import train_test_split
from google.colab import drive
drive.mount('/content/drive')
np.random.seed(7)
split = 96000
split1 = 97200
finish = 98400
# IMPORTING DATASET
dataset = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/flexdirection.
                                       csv',usecols=[0,1,2,3,4,5,6])
dataset = dataset.astype('float32')
scaler = MinMaxScaler(feature_range=(-1, 1)) #          scalar
dataset = scaler.fit_transform(dataset) #1
X = dataset[0:split, 0:4]
Y = dataset[0:split, 4:7]
train_X, test_X, train_y, test_y = train_test_split(X, Y, test_size = 0.2,
                                       random_state = 8)
train_X.shape,test_X.shape,train_y.shape,test_y.shape,train_X,train_y
#####Build model#######
model = Sequential()
model.add(Dense(units=256,input_dim=4))
model.add(Activation('relu'))
model.add(Dense(units=256,activation='relu'))
model.add(Dense(3))
model.compile(loss='mse', optimizer=tf.keras.optimizers.Adam(learning_rate = 0
                                       .001),metrics=['accuracy'])
model.summary()#Show model structure
history = model.fit(train_X, train_y, epochs=100, batch_size=128,
                                       validation_split=0.1, verbose=1,
                                       shuffle=True)
hist = pd.DataFrame(history.history)
hist['epoch'] = history.epoch
hist.tail()#Record training history data
#####plot loss curve######
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('MSE Loss')
plt.xlabel('Epoch')
plt.legend(['Train set', 'Validation set'], loc='upper right')
plt.show()
#######Model Prediction######
yhat = model.predict(test_X)
yhat.shape,test_X.shape
######Convert to true value######
inv_yhat = concatenate((test_X[:, :],yhat ), axis=1)
testPredict_out = scaler.inverse_transform(inv_yhat)
inv_yhat = testPredict_out[:,4:]
inv_y = concatenate((test_X[:, :],test_y ), axis=1)
testY = scaler.inverse_transform(inv_y)
inv_y = testY[:,4:]

######Model Evaluation######
rmse = sqrt(mean_squared_error(inv_y, inv_yhat))#RMSE
r2=r2_score(inv_y, inv_yhat)# R^2
mae=mean_absolute_error(inv_y, inv_yhat)# MAE
print(inv_yhat.shape )
print(inv_y.shape )
print('Test RMSE: %.3f' % rmse)
print('Test  R 2  %.3f' % r2)
print('Test  M A E %.3f' % mae)
######Plot every loading path in H, V, and M#######

plt.rcParams['figure.figsize'] = (4.0, 4.0)
inv_yhat1 = inv_yhat[:,0]
inv_y1 = inv_y[:,0]
plt.scatter(inv_yhat1,inv_y1, s=2 ,color='', marker='o', edgecolors='r',label=
                                            'true')
plt.plot([-4.0, 4.0],[-4.0, 4.0],'#000000')
plt.xlim(-4.0, 4.0)
plt.ylim(-4.0, 4.0)

#plt.title('Model loss')
plt.ylabel('Abaqus dimensionless calculation')
plt.xlabel('FC-NN dimensionless prediction')
#plt.text(0.1, 0.1, s="R2=0.712 MAE=0.157")
#plt.legend(['R20.712', 'MAE0.157'], loc='upper right')
plt.figure(figsize=(900, 900))
plt.show()
inv_yhat1 = inv_yhat[:,1]
inv_y1 = inv_y[:,1]
plt.scatter(inv_yhat1,inv_y1, s=2 ,color='', marker='o', edgecolors='r',label=
                                            'true')
plt.plot([-6.0, 0],[-6.0, 0],'#000000')
plt.xlim(-6.0, 0)
plt.ylim(-6.0, 0)

#plt.title('Model loss')
plt.ylabel('Abaqus dimensionless calculation')
plt.xlabel('FC-NN dimensionless prediction')
#plt.text(0.1, 0.1, s="R2=0.712 MAE=0.157")
#plt.legend(['R20.712', 'MAE0.157'], loc='upper right')
plt.figure(figsize=(900, 900))
plt.show()
```

```
inv_yhat1 = inv_yhat[:,2]
inv_y1 = inv_y[:,2]
plt.scatter(inv_yhat1,inv_y1, s=2 ,color='', marker='o', edgecolors='r',label='true')
plt.plot([-2.0, 2.0],[-2.0, 2.0],'#000000')
plt.xlim(-2.0, 2.0)
plt.ylim(-2.0, 2.0)

#plt.title('Model loss')
plt.ylabel('Abaqus dimensionless calculation')
plt.xlabel('FC-NN dimensionless prediction')
#plt.text(0.1, 0.1, s="R2=0.712 MAE=0.157")
#plt.legend(['R20.712', 'MAE0.157'], loc='upper right')
plt.figure(figsize=(900, 900))
plt.show()
```

A.3 Deep learning model structure and training procedure

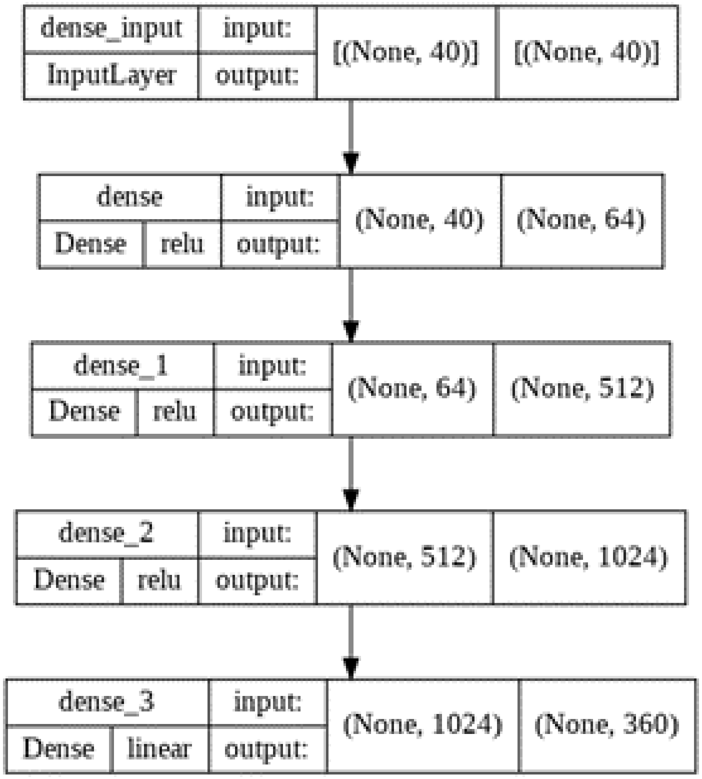


Figure A.1: FC neural network structure

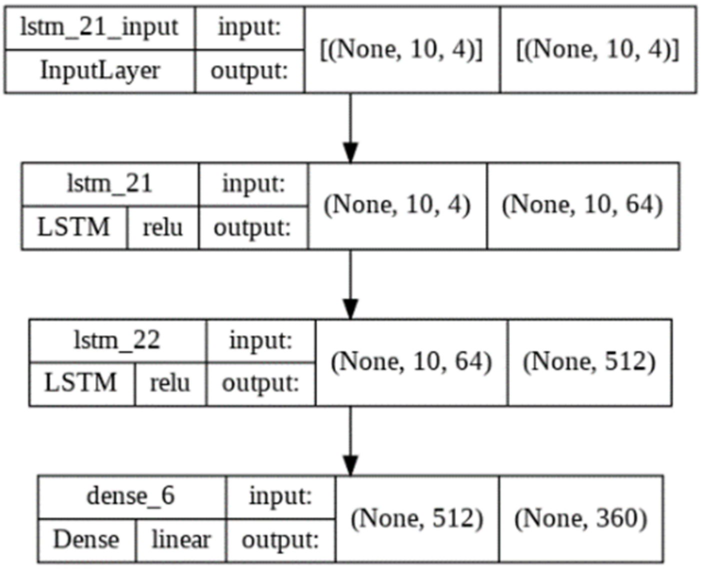


Figure A.2: LSTM model network structure

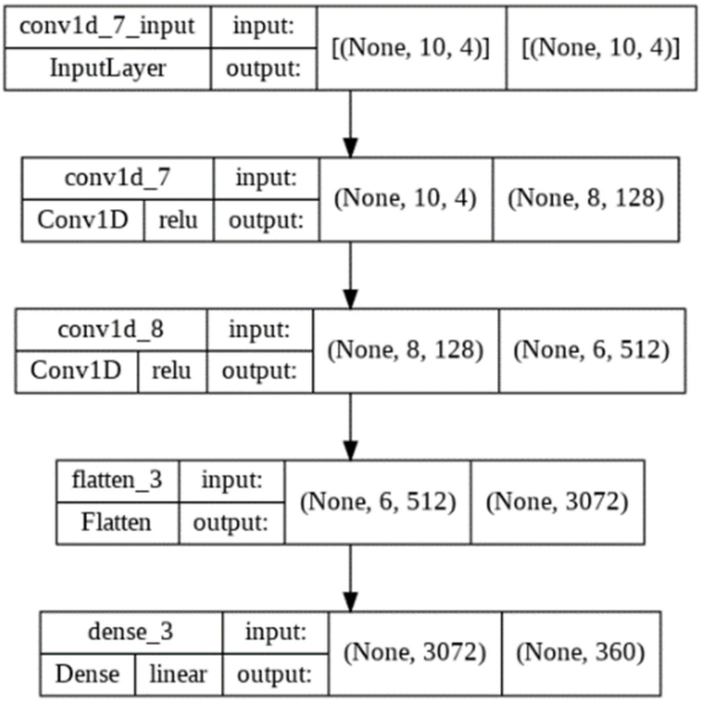


Figure A.3: 1D-CNN model network structure

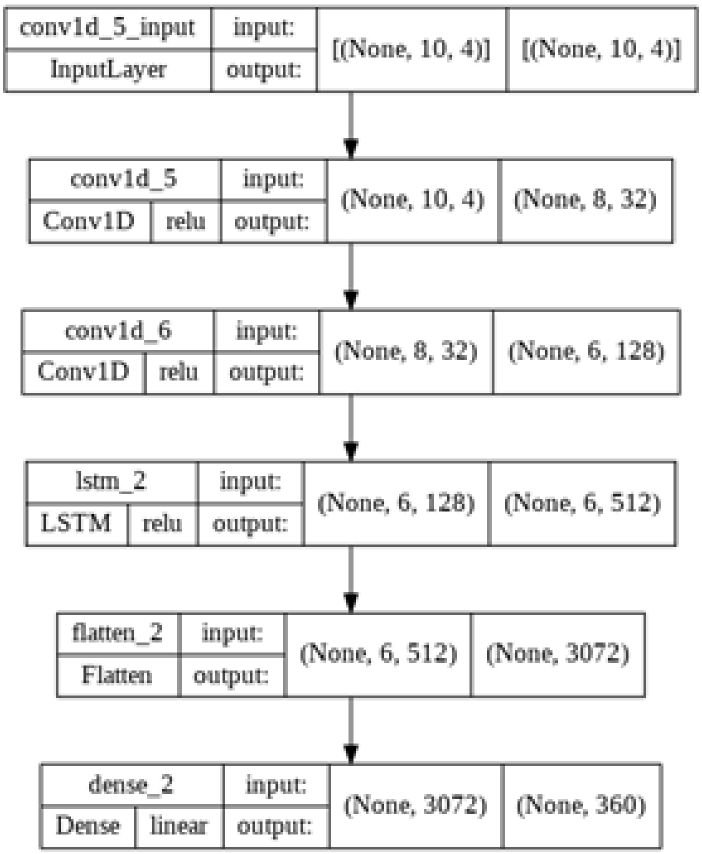


Figure A.4: 1D-CNN+LSTM model network structure

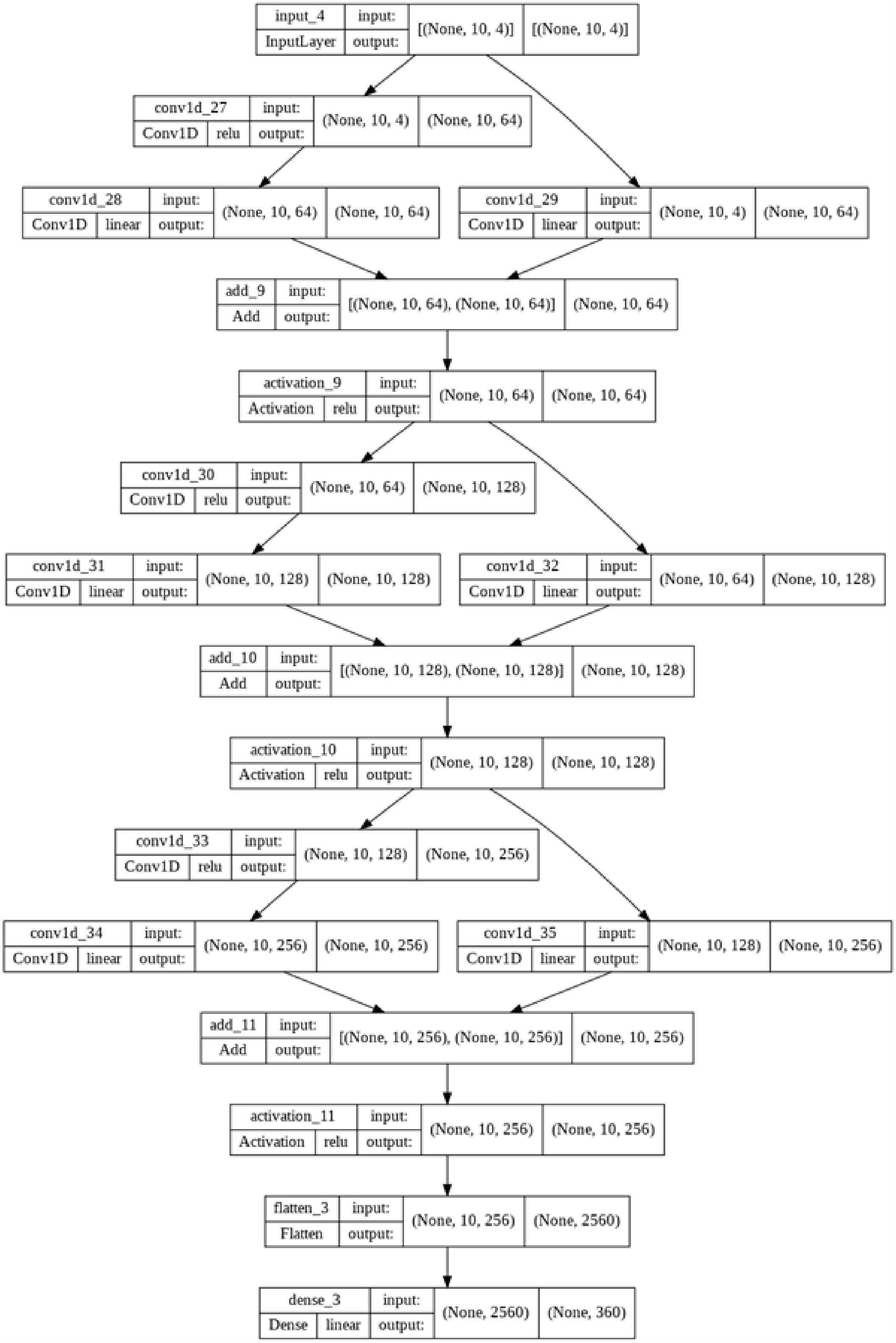

input_4 | InputLayer | input: | output: | [(None, 10, 4)] | [(None, 10, 4)]
conv1d_27 | Conv1D | relu | input: | output: | (None, 10, 4) | (None, 10, 64)
conv1d_28 | Conv1D | linear | input: | output: | (None, 10, 64) | (None, 10, 64)
conv1d_29 | Conv1D | linear | input: | output: | (None, 10, 4) | (None, 10, 64)
add_9 | Add | input: | output: | [(None, 10, 64), (None, 10, 64)] | (None, 10, 64)
activation_9 | Activation | relu | input: | output: | (None, 10, 64) | (None, 10, 64)
conv1d_30 | Conv1D | relu | input: | output: | (None, 10, 64) | (None, 10, 128)
conv1d_31 | Conv1D | linear | input: | output: | (None, 10, 128) | (None, 10, 128)
conv1d_32 | Conv1D | linear | input: | output: | (None, 10, 64) | (None, 10, 128)
add_10 | Add | input: | output: | [(None, 10, 128), (None, 10, 128)] | (None, 10, 128)
activation_10 | Activation | relu | input: | output: | (None, 10, 128) | (None, 10, 128)
conv1d_33 | Conv1D | relu | input: | output: | (None, 10, 128) | (None, 10, 256)
conv1d_34 | Conv1D | linear | input: | output: | (None, 10, 256) | (None, 10, 256)
conv1d_35 | Conv1D | linear | input: | output: | (None, 10, 128) | (None, 10, 256)
add_11 | Add | input: | output: | [(None, 10, 256), (None, 10, 256)] | (None, 10, 256)
activation_11 | Activation | relu | input: | output: | (None, 10, 256) | (None, 10, 256)
flatten_3 | Flatten | input: | output: | (None, 10, 256) | (None, 2560)
dense_3 | Dense | linear | input: | output: | (None, 2560) | (None, 360)

Figure A.5: TCN model network structure

Bibliography

Abdel-Rahman, K. and Achmus, M. (2005). Finite element modelling of horizontally loaded monopile foundations for offshore wind energy converters in germany. In *Proceedings of the international symposium on frontiers in offshore geotechnics. Taylor and Francis, Perth*, pages 391–396.

Atkinson, J. (2000). Non-linear soil stiffness in routine design. *Géotechnique*, 50(5):487–508.

Baghbani, A. (2022). Application of artificial intelligence in geotechnical engineering: A state-of-the-art review. page 26.

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv:1803.01271 [cs]*. arXiv: 1803.01271.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bhattacharya, S., Nikitas, N., Garnsey, J., Alexander, N., Cox, J., Lombardi, D., Wood, D. M., and Nash, D. F. (2013). Observed dynamic soil–structure interaction in scale testing of offshore wind turbine foundations. *Soil Dynamics and Earthquake Engineering*, 54:47–60.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Bransby, M. and Randolph, M. (1998). Combined loading of skirted foundations. *Géotechnique*, 48(5):637–655.

Bransby, M. and Randolph, M. (1999). The effect of embedment depth on the undrained response of skirted foundations to combined loading. *Soils and foundations*, 39(4):19–33.

Bransby, M. and Yun, G.-J. (2009). The undrained capacity of skirted strip foundations under combined loading. *Géotechnique*, 59(2):115–125.

Butterfield, R., Houlsby, G. T., and Gottardi, G. (1997). Standardized sign conventions and notation for generally loaded foundations. *Géotechnique*, 47(5):1051–1054.

Byrne, B. W. (2000). *Investigations of suction caissons in dense sand.* PhD thesis, University of Oxford Oxford, UK.

Cheng, J., Dong, L., and Lapata, M. (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733.*

Cheng, X. and Vanapalli, S. K. (2021). Prediction of the nonlinear behavior of laterally loaded piles in unsaturated soils. *Computers and Geotechnics*, 140:104480.

Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., and Campbell, J. P. (2020). Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science & Technology*, 9(2):14–14.

Feng, X., Randolph, M., Gourvenec, S., and Wallerand, R. (2014). Design approach for rectangular mudmats under fully three-dimensional loading. *Géotechnique*, 64(1):51–63.

Fortmann-Roe, S. (2012). Understanding the bias-variance tradeoff. *URL: http://scott. fortmann-roe. com/docs/BiasVariance. html (hämtad 2019-03-27).*

Fu, D., Gaudin, C., Tian, Y., Cassidy, M. J., and Bienen, B. (2017). Uniaxial capacities of skirted circular foundations in clay. *Journal of Geotechnical and Geoenvironmental Engineering*, 143(7):04017022.

Gerolymos, N., Zafeirakos, A., and Karapiperis, K. (2015). Generalized failure envelope for caisson foundations in cohesive soil: static and dynamic loading. *Soil Dynamics and Earthquake Engineering*, 78:154–174.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Gourvenec, S. (2007). Failure envelopes for offshore shallow foundations under general loading. *Géotechnique*, 57(9):715–728.

Gourvenec, S. (2008). Effect of embedment on the undrained capacity of shallow foundations under general loading. *Géotechnique*, 58(3):177–185.

Gourvenec, S. and Barnett, S. (2011). Undrained failure envelope for skirted foundations under general loading. *Géotechnique*, 61(3):263–270.

Gourvenec, S. and Cassidy, M. (2005). *Frontiers in Offshore Geotechnics: Proceedings of the International Symposium on Frontiers in Offshore Geotechnics (IS-FOG 2005), 19-21 Sept 2005, Perth, WA, Australia.* CRC Press.

Gourvenec, S. and Randolph, M. (2003). Effect of strength non-homogeneity on the shape of failure envelopes for combined loading of strip and circular foundations on clay. *Géotechnique*, 53(6):575–586.

Graine, N., Hjiaj, M., and Krabbenhoft, K. (2021). 3d failure envelope of a rigid pile embedded in a cohesive soil using finite element limit analysis. *International Journal for Numerical and Analytical Methods in Geomechanics*, 45(2):265–290.

Haykin, S. (2009). *Neural networks and learning machines, 3/E.* Pearson Education India.

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

Hochreiter, S. and Schmidhuber, J. (1997a). Flat minima. *Neural computation*, 9(1):1–42.

Hochreiter, S. and Schmidhuber, J. (1997b). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Houlsby, G. (2016). Interactions in offshore foundation design. *Géotechnique*, 66(10):791–825.

Houlsby, G., Kelly, R., Huxtable, J., and Byrne, B. (2006). Field trials of suction caissons in sand for offshore wind turbine foundations. *Géotechnique*, 56(1):3–10.

Hung, L. C. and Kim, S.-R. (2014). Evaluation of undrained bearing capacities of bucket foundations under combined loads. *Marine Georesources & Geotechnology*, 32(1):76–92.

Ibsen, L. B., Barari, A., and Larsen, K. A. (2014). Adaptive plasticity model for bucket foundations. *Journal of Engineering Mechanics*, 140(2):361–373.

Jonkman, J., Butterfield, S., Musial, W., and Scott, G. (2009). Definition of a 5-mw reference wind turbine for offshore system development. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Karapiperis, K. and Gerolymos, N. (2014). Combined loading of caisson foundations in cohesive soil: finite element versus winkler modeling. *Computers and Geotechnics*, 56:100–120.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kunz, F., Stellzig-Eisenhauer, A., Zeman, F., and Boldt, J. (2020). Artificial intelligence in orthodontics. *Journal of Orofacial Orthopedics / Fortschritte der Kieferorthopädie*, 81(1):52–68.

Kuo, Y., Jaksa, M., Lyamin, A., and Kaggwa, W. (2009). Ann-based model for predicting the bearing capacity of strip footing on multi-layered cohesive soil. *Computers and Geotechnics*, 36(3):503–516.

Lau, B. H. (2015). *Cyclic behaviour of monopile foundations for offshore wind turbines in clay*. PhD thesis, University of Cambridge.

Livieris, I. E., Pintelas, E., and Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32(23):17351–17360.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Luo, P., Wang, X., Shao, W., and Peng, Z. (2018). Towards understanding regularization in batch normalization. *arXiv preprint arXiv:1809.00846*.

Manual, A. U. (2020). Abaqus user manual. *Abacus*.

Mau, W., Hasselmo, M. E., and Cai, D. J. (2020). The brain in motion: How ensemble fluidity drives memory-updating and flexibility. *Elife*, 9:e63550.

Mehravar, M., Harireche, O., and Faramarzi, A. (2016). Evaluation of undrained failure envelopes of caisson foundations under combined loading. *Applied Ocean Research*, 59:129–137.

Momeni, E., Nazir, R., Armaghani, D. J., and Maizir, H. (2014). Prediction of pile bearing capacity using a hybrid genetic algorithm-based ann. *Measurement*, 57:122–131.

Murff, J. (1994). Limit analysis of multi-footing foundation systems. In *International conference on computer methods and advances in geomechanics*, pages 233–244.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Icml*.

Nebauer, C. (1998). Evaluation of convolutional neural networks for visual recognition. *IEEE transactions on neural networks*, 9(4):685–696.

Nejad, F. P. and Jaksa, M. B. (2017). Load-settlement behavior modeling of single piles using artificial neural networks and cpt data. *Computers and Geotechnics*, 89:9–21.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

OWA, O. W. A. (2019). Suction installed caisson foundations for offshore wind: Design guidelines. *Carbon Trust*.

Padmini, D., Ilamparuthi, K., and Sudheer, K. (2008). Ultimate bearing capacity prediction of shallow foundations on cohesionless soils using neurofuzzy models. *Computers and Geotechnics*, 35(1):33–46.

Pisanò, F., Flessati, L., and Di Prisco, C. (2016). A macroelement framework for shallow foundations including changes in configuration. *Géotechnique*, 66(11):910–926.

Randolph, M. and Gourvenec, S. (2017). *Offshore geotechnical engineering*. CRC press.

Reimers, C. and Requena-Mesa, C. (2020). Deep learning–an opportunity and a challenge for geo-and astrophysics. In *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, pages 251–265. Elsevier.

Roscoe, K. (1956). The stability of short pier foundations in sand. *British Welding Journal*, pages 343–354.

Rystad, J. (2021). What will the future energy system look like? *Energy News*, 39(2):8–11.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Shahin, M. A. (2014). Load–settlement modeling of axially loaded steel driven piles using cpt-based recurrent neural networks. *Soils and Foundations*, 54(3):515–522.

Shahin, M. A. (2016). State-of-the-art review of some artificial intelligence applications in pile foundations. *Geoscience Frontiers*, 7(1):33–44.

Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2001). Artificial neural network applications in geotechnical engineering. *Australian geomechanics*, 36(1):49–62.

Skau, K. S., Chen, Y., and Jostad, H. P. (2018). A numerical study of capacity and stiffness of circular skirted foundations in clay subjected to combined static and cyclic general loading. *Geotechnique*, 68(3):205–220.

Suryasentana, S. K., Dunne, H. P., Martin, C. M., Burd, H. J., Byrne, B. W., and Shonberg, A. (2020). Assessment of numerical procedures for determining shallow foundation failure envelopes. *Géotechnique*, 70(1):60–70.

Taiebat, H. and Carter, J. (2000). Numerical studies of the bearing capacity of shallow foundations on cohesive soil subjected to combined loading. *Géotechnique*, 50(4):409–418.

Villalobos Jara, F. A. (2006). *Model testing of foundations for offshore wind turbines.* PhD thesis, University of Oxford.

Vulpe, C. (2015). Design method for the undrained capacity of skirted circular foundations under combined loading: effect of deformable soil plug. *Géotechnique*, 65(8):669–683.

Vulpe, C., Gourvenec, S., and Power, M. (2014). A generalised failure envelope for undrained capacity of circular shallow foundations under general loading. *Géotechnique Letters*, 4(3):187–196.

Wang, H. (2020). *Lateral behaviour of offshore monopile and bucket foundations in sand.* PhD thesis, Zhejiang University.

Yin, Z.-Y., Teng, J.-C., Li, Z., and Zheng, Y.-Y. (2020). Modelling of suction bucket foundation in clay: From finite element analyses to macro-elements. *Ocean Engineering*, 210:107577.

Yu, L.-q., Wang, L.-Z., Guo, Z., Bhattacharya, S., Nikitas, G., Li, L.-L., and Xing, Y.-L. (2015). Long-term dynamic behavior of monopile supported offshore wind turbines in sand. *Theoretical and Applied Mechanics Letters*, 5(2):80–84.

Zhang, P., Jin, Y.-F., Yin, Z.-Y., and Yang, Y. (2020a). Random forest based artificial intelligent model for predicting failure envelopes of caisson foundations in sand. *Applied Ocean Research*, 101:102223.

Zhang, P., Yin, Z.-Y., and Jin, Y.-F. (2021). State-of-the-Art Review of Machine Learning Applications in Constitutive Modeling of Soils. *Archives of Computational Methods in Engineering*, 28(5):3661–3686.

Zhang, P., Yin, Z.-Y., Zheng, Y., and Gao, F.-P. (2020b). A lstm surrogate modelling approach for caisson foundations. *Ocean Engineering*, 204:107263.