TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO.: PUL080MSGtE004

Development and Experimental Validation of the Physics Informed Neural Network for Predicting Axial Pile Load Capacity

by

Bipul Mainali

A THESIS

SUBMITTED TO THE DEPARTMENT OF CIVIL ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTERS OF SCIENCE IN

GEOTECHNICAL ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

LALITPUR, NEPAL

APRIL, 2026

COPYRIGHT

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF CIVIL ENGINEERING

The thesis titled “Development and Experimental Validation of the Physics Informed Neural Network for Predicting Axial Pile Load Capacity" prepared and submitted by Bipul Mainali in partial fulfilment of the requirements for the degree of Master of Science (M. Sc.) in Geotechnical Engineering. has been examined by us and is accepted for the award of M. Sc. in Geotechnical Engineering by Tribhuvan University.

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a thesis report entitled “Development and Experimental Validation of the Physics Informed Neural Network for Predicting Axial Pile Load Capacity" submitted by Bipul Mainali in partial fulfillment of the requirements for the degree of Master in Geotechnical Engineering.

Supervisor:
Prof. Dr. Indra Prasad Acharya
Pulchowk Campus,
Pulchowk, Lalitpur, Nepal

External Examiner:
Dr. Mohan Prasad Acharya
Senior Geotechnical Engineer,
NEA Engineering Company

Program Coordinator:
Assist. Prof. Dr. Bhim Kumar Dahal
MSc in Geotechnical Engineering
Pulchowk Campus,
Lalitpur, Nepal

Date: April 26, 2026

DECLARATION

I hereby declare that this study titled “Development and Experimental Validation of the Physics Informed Neural Network for Predicting Axial Pile Load Capacity" is based on my original research work. Related works on the topic by other researchers have been duly acknowledged. I owe all the liabilities relating to the accuracy and authenticity of the data and any other information included hereunder.

Bipul Mainali
(PUL080MSGtE004)
MSc in Geotechnical Engineering
Date: April 26, 2026

ABSTRACT

Accurate prediction of axial pile load capacity in layered soil conditions is a main challenge in geotechnical engineering, where conventional and empirical methods has oversimplify soil–pile interaction and also purely data-driven machine learning models lack physical consistency. This study has developed and experimentally validated a Physics-Informed Neural Network (PINN) framework for predicting axial pile load capacity. In Phase 1, a preliminary single-output PINN using Meyerhof end bearing and β-method shaft resistance was compared against Random Forest, achieving R^2 = 0.940 for internal validation and R^2 = 0.938 for external validation, confirming the feasibility of physics-informed machine learning for axial pile capacity prediction. The Phase 2 framework introduced a multi-output architecture simultaneously predicting shaft resistance, base resistance, and ultimate capacity; a ten-layer stratified soil representation; trainable physics weighting parameters; and an ensemble of three models trained under different random seeds. The Phase 2 PINN ensemble achieved R^2 = 0.977, outperforming Artificial Neural Network (R^2 = 0.901), Random Forest (R^2 = 0.934), and XGBoost (R^2 = 0.958). Experimental validation of seven laboratory pile load tests conducted at IIT Patna was scaled to prototype dimensions using a geometric scale factor of 50, and it was observed that PINN produces load–settlement curves in closer agreement than benchmark models. This study shows physical consistency of PINN model for prediction of Axial Pile Load capacity and further recommends using different geotechnical cases wise study to increase the reliability of PINN framework in the context of Geotechnical Engineering.

Keywords: *Physics Informed Neural Networks; Pile Foundations; Axial Capacity; Load Settlement Behavior; Machine Learning; Geotechnical Engineering*

ACKNOWLEDGEMENT

I would like to express my deepest gratitude and appreciation to my respected supervisor, Prof. Dr. Indra Prasad Acharya, for his valuable guidance, support, and continuous encouragement throughout the this research. His knowledge, patience, and constructive suggestions have always been instrumental in shaping both the direction and quality of this research work. I remain truly grateful to him for his mentorship.

I am also grateful to Assist. Prof. Dr. Bhim Kumar Dahal, Program Coordinator of the MSc in Geotechnical Engineering at Pulchowk Campus, for his constant encouragement, academic guidance, and support throughout my graduate studies. His motivation has greatly contributed to my learning experience.

I am thankful to Assoc. Prof. Arvind Kumar Jha of IIT Patna for his guidance and support during my research visit to IIT Patna. I am especially grateful for his insights and assistance during the one-month period of conducting the static pile load test, which significantly helped the experimental component of this study.

I am really grateful to IIT Patna for providing research facility, learning environment and accommodations during my visit.

I would like to the Bridge Sector, Department of Roads, for kindly providing the pile load test data.

At last, I am very grateful to my parents who always supported and motivated me in doing research work and do scientific contributions.

TABLE OF CONTENTS

COPYRIGHT i

APPROVAL SHEET ii

DECLARATION iii

ABSTRACT iv

ACKNOWLEDGEMENT v

TABLE OF CONTENTS viii

LIST OF FIGURES x

LIST OF TABLES xi

LIST OF SYMBOLS xii

LIST OF ABBREVIATION xv

1 INTRODUCTION 1
- 1.1 Background 1
- 1.2 Statement of Problem 1
- 1.3 Research Questions 2
- 1.4 Research Objectives 2
- 1.5 Limitation of Study 2
- 1.6 Organization of Thesis 3

2 LITERATURE REVIEW 4
- 2.1 Pile Foundations and Load–Settlement Behavior 4
- 2.2 Pile–Soil Interaction Mechanisms 4
- 2.3 Theoretical Framework for Axial Pile Behavior 6
- 2.4 Conventional Methods for Pile Capacity Prediction 8
- 2.5 Machine Learning in Geotechnical Engineering 8
- 2.6 Limitations of Data-Driven Models 9
- 2.7 Physics-Informed Neural Networks (PINN) 9
- 2.8 Research Gap 10

3 METHODOLOGY 16
- 3.1 Data Generation 16

3.2 Input Parameters and Dataset Preparation 18
3.3 Machine Learning Models 21
3.4 Development of Physics-Informed Neural Network 23
3.5 Phase 1: Preliminary PINN Framework 23
3.6 Phase 2: Enhanced Multi-Output PINN Framework 27
3.7 Model Training and Validation Strategy 35

4 EXPERIMENTAL PROGRAM 39
4.1 Overview of Experimental Program 39
4.2 Soil Characterization 39
4.3 Sand Bed Preparation 41
4.4 Model Pile and Instrumentation 42
4.5 Experimental Setup 42
4.6 Test Matrix 44

5 RESULTS AND DISCUSSION 45
5.1 Experimental Results 45
5.2 Model Performance Evaluation 49
5.3 Experimental Validation: Load–Settlement Curves 52
5.4 Physics-Informed Neural Network Behaviour 53
5.5 External Validation of Models 60

6 CONCLUSION 64
6.1 Limitations of the Study 64
6.2 Recommendations for Future Work 65

REFERENCES 67

Appendix **68**

A SOURCE CODE 68
A.1 Synthetic Pile Bearing Capacity Dataset Generation 68
A.2 PLAXIS 3D Finite Element Analysis Automation 71
A.3 Physics-Informed Neural Network Ensemble Framework 74
A.4 Load–Settlement Curve Comparison with Laboratory Data 80
A.5 Summary of Key Symbols 84

B Research Visit to IIT PATNA 85
B.1 Visit and Academic Activities at IIT Patna 85
B.2 Experimental Photographs 88

B.3 Data Processing and Supporting Records 92
B.4 Additional Results 93

LIST OF FIGURES

2.1 Conceptual axial pile load–settlement curve showing the progressive mobilization of shaft resistance at small settlements and base resistance at larger settlements 4
2.2 Schematic illustration of axial pile resistance components under an applied axial load. 5
2.3 Schematic diagram of improved physics-informed approach with measured data (Yuan et al., 2025) 10

3.1 Research framework adopted in this study 17
3.2 PLAXIS 3D Automation using Python Script using Academic License 18
3.3 Quantitative performance comparison of the Phase 1 PINN and Random Forest on internal test and external validation datasets. ... 25
3.4 Phase 1 PINN training loss convergence showing stable progressive enforcement of physics constraints. 25
3.5 Convergence of the trainable β parameter during Phase 1 training ($0.5 \rightarrow 0.43$). 26
3.6 Phase 1 parity plot: PINN predictions cluster more tightly around the $y = x$ line than Random Forest, especially at higher capacities. 26
3.7 Phase 1 external validation parity plot ($R^2 = 0.938$). 27
3.8 Workflow of the developed physics-informed neural network framework for pile axial capacity prediction. 36

4.1 Major stages of the experimental program 39
4.2 Particle size distribution of sand 39
4.3 Direct shear test results 40
4.4 Triaxial test results 40
4.5 Calibration of height of fall for density control 41
4.6 Pluviation method for sand deposition 41
4.7 Static pile load test setup 42
4.8 Laboratory setup at IIT Patna 43

5.1 Load–settlement curve obtained from laboratory scale static pile load tests. 45
5.2 Parity plot for the PINN model 50
5.3 Parity plot for the ANN model 50
5.4 Parity plot for the Random Forest model 51
5.5 Parity plot for the XGBoost model 51

5.6 Test 1: $D = 1.00$ m, $L = 7.50$ m, uniform loose sand ($D_r = 40\%$). 53
5.7 Test 2: $D = 1.00$ m, $L = 10.00$ m, uniform medium dense ($D_r = 60\%$)..... 54
5.8 Test 3: $D = 1.00$ m, $L = 12.50$ m, uniform dense ($D_r = 80\%$)..... 54
5.9 Test 4: $D = 1.50$ m, $L = 7.50$ m, uniform medium dense ($D_r = 60\%$). 55
5.10 Test 5: $D = 1.50$ m, $L = 10.00$ m, loose over dense ($D_r = 40\%$ over 80%). 55
5.11 Test 6: $D = 1.50$ m, $L = 12.50$ m, dense over loose ($D_r = 80\%$ over 40%). 56
5.12 Test 7: $D = 2.00$ m, $L = 10.00$ m, uniform medium dense ($D_r = 60\%$)..... 56
5.13 Evolution of PINN loss components during training..... 57
5.14 Validation RMSE evolution during PINN training. 58
5.15 Distribution of equilibrium residuals for the PINN model. 59
5.16 Evolution of trainable shaft resistance parameter β during training. 59
5.17 Evolution of trainable loss weights during training. 60
5.18 External parity plot for the PINN model 61
5.19 External parity plot for the ANN model 62
5.20 External parity plot for the Random Forest model 62
5.21 External parity plot for the XGBoost model 63

B.1 Research Visit at IIT Patna 85
B.2 Presentation of the research work at IIT Patna..... 86
B.3 With Assoc. Prof. Dr. Arvind Kumar Jha and his PhD & M Tech Students 86
B.4 Infront of Civil Engineering Workshop at IIT Patna 87
B.5 Infront of Geotechnical Engineering Lab at IIT Patna 87
B.6 Infront of Department of Civil and Environmental Engineering, IIT Patna 88
B.7 Specific gravity test of sand. 88
B.8 Relative density test of sand. 89
B.9 Direct shear test of sand..... 89
B.10 Sample preparation for XRD analysis. 89
B.11 Instrumentation setup for XRD analysis. 90
B.12 Sample preparation for triaxial testing and tested sample. 90
B.13 Triaxial setup for UU test of sand sample. 90
B.14 Leveling of sand bed and installation of pressure cell. 91
B.15 Steel pile installation..... 91
B.16 Data observation using the DAQ system..... 92
B.17 Sample of raw data processing. 92
B.18 XRD result of the sand sample..... 93
B.19 SEM images of the sand sample..... 93

LIST OF TABLES

2.1 Summary of Relevant Literature 11

3.1 Parameter ranges used in synthetic dataset generation (5,000 cases, 10 soil layers per case, random seed = 42) 21
3.2 Hyperparameter settings used for PINN, ANN, Random Forest, and XGBoost models..... 30

4.1 Scaling factors adopted for small-scale physical modeling (after Moosavian et al. (2021)) 43
4.2 Test matrix for laboratory-scale static pile load tests 44

5.1 Load Settlement data recorded from laboratory scale static pile load tests 46
5.2 Comparison of model performance based on statistical error metrics .. 52

A.1 Key symbols used in the source code and their physical meaning. 84

LIST OF SYMBOLS

c	Cohesion of soil (kPa)
ϕ	Angle of internal friction of soil (degrees)
γ	Unit weight of soil (kN/m^3)
γ_d	Dry unit weight of soil (kN/m^3)
γ_{sat}	Saturated unit weight of soil (kN/m^3)
w	Water content or moisture content (%)
G_s	Specific gravity of soil solids
e	Void ratio
e_{max}	Maximum void ratio
e_{min}	Minimum void ratio
D_r	Relative density (%)
q	Applied load or end bearing pressure (kPa)
q_b	Unit base resistance or toe resistance (kPa)
q_s	Unit shaft resistance (kPa)
t	Shaft resistance or skin friction (kPa)
Q	Axial load on pile (kN)
Q_u	Ultimate pile load capacity (kN)
Q_s	Shaft resistance component of pile capacity (kN)
Q_b	Base resistance component of pile capacity (kN)
Q_{ult}	Ultimate load capacity of pile (kN)
P	Applied axial load (kN)
P_u	Ultimate axial load (kN)
L	Length of the pile (m)
D	Diameter of the pile (m)
L/D	Length-to-diameter ratio of pile
A_p	Cross-sectional area of pile (m^2)
A_b	Base area of pile (m^2)
U_p	Perimeter of pile (m)

z	Depth below ground surface or axial coordinate along pile (m)
$w(z)$	Axial displacement of pile at depth z
s	Settlement of pile or foundation (mm or m)
Δ	Settlement or displacement increment (mm or m)
δ	Displacement or local deformation
ϵ	Strain
σ	Total stress (kPa)
σ'	Effective stress (kPa)
σ'_v	Vertical effective stress (kPa)
σ'_h	Horizontal effective stress (kPa)
u	Pore water pressure (kPa)
h_w	Groundwater table depth (m)
E	Young's modulus or elastic modulus (kPa)
E_s	Young's modulus of soil (kPa)
E_p	Young's modulus of pile material (kPa)
G	Shear modulus of soil (kPa)
ν	Poisson's ratio
K_0	Coefficient of earth pressure at rest
α	Adhesion factor or empirical shaft resistance factor
β	Shaft resistance scaling factor or trainable physics parameter
λ	Weight associated with loss components in PINN
λ_p	Physics loss weight
λ_b	Boundary loss weight
λ_{global}	Global consistency loss weight
$\lambda_{consistency}$	Output consistency loss weight
N_c	Bearing capacity factor related to cohesion
N_q	Bearing capacity factor related to surcharge
N_γ	Bearing capacity factor related to unit weight
N	SPT blow count
c_u	Undrained shear strength (kPa)

τ	Shear stress (kPa)
θ	Rotation angle
ρ	Mass density (kg/m^3)
$\hat{y}$	Predicted output
y	Observed or target output
$\mathscr{L}$	Total loss function
$\mathscr{L}_{data}$	Data loss term
$\mathscr{L}_{physics}$	Physics residual loss term
$\mathscr{L}_{bc}$	Boundary condition loss term
$\mathscr{L}_{cons}$	Consistency loss term
R^2	Coefficient of determination

LIST OF ABBREVIATION

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	American Petroleum Institute
ASTM	American Society for Testing and Materials
BNWF	Beam on Winkler Foundation
CNN	Convolutional Neural Network
CPT	Cone Penetration Test
DAQ	Data Acquisition System
DHM	Department of Hydrology and Meteorology
DL	Deep Learning
DNN	Deep Neural Network
EBM	Explainable Boosting Machine
FEA	Finite Element Analysis
FEM	Finite Element Method
GBM	Gradient Boosting Machine
GP	Gaussian Process
GPR	Gaussian Process Regression
GUI	Graphical User Interface
IS	Indian Standard
KNN	K-Nearest Neighbors
LBFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno
LVDT	Linear Variable Differential Transformer
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MARE	Mean Absolute Relative Error
MC	Mohr–Coulomb
ML	Machine Learning
MSE	Mean Squared Error

OMC	Optimum Moisture Content
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
PIML	Physics-Informed Machine Learning
PLAXIS	Finite element software for geotechnical analysis
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SHAP	SHapley Additive exPlanations
SPT	Standard Penetration Test
SVR	Support Vector Regression
SVM	Support Vector Machine
TU	Tribhuvan University
USCS	Unified Soil Classification System
XGBoost	Extreme Gradient Boosting

1 INTRODUCTION

1.1 Background

Pile foundations transfer structural loads to deeper and more competent soil strata when near-surface materials are insufficient. Their axial behavior is governed by shaft resistance which is mobilized progressively along the pile–soil interface and end bearing resistance at the pile tip. Prediction of this behavior accurately is essential for safe foundation design, yet it remains challenging due to the nonlinear, depth-dependent nature of soil–pile interaction and also the variability of soil stratigraphy. Full-scale pile load testing is expensive and site-specific and classical analytical methods rely on simplified assumptions that break down in layered conditions and do not capture the full load–settlement response (Coyle and Reese, 1966; Meyerhof, 1976).

Machine learning has recently been powerful alternative, with Artificial Neural Networks, Random Forest, and Extreme Gradient Boosting demonstrating strong predictive accuracy for pile capacity (Ardakani et al., 2020; Benbouras et al., 2021; Gomes et al., 2021; Lee et al., 2018). However, purely data-driven models function acts as black-box systems that may violate physical equilibrium conditions, also generalize poorly outside the training domain, and provide limited interpretability for engineering judgment.

Physics-Informed Neural Networks (PINNs) has been addressing this limitation by embedding governing physical equations and constitutive relationships directly into the neural network loss function (Raissi et al., 2019), making predictions to be consistent with both observed data and established mechanics. PINNs has recently been applied to seepage, consolidation, and soil–structure interaction problems (Gao et al., 2023; Ito et al., 2024; Yuan et al., 2025), now with initial efforts for axial pile capacity by (Al Atroush, 2025; Madianos et al., 2023). However, existing applications considers simplified soil profiles, predict only ultimate capacity as a single scalar output, and do not validate against physical load test data. A framework addressing layered soil stratigraphy, multi-output shaft and base resistance, trainable physics weighting, and experimental validation has not been reported.

1.2 Statement of Problem

Conventional analytical methods require simplified assumptions about soil uniformity that break down stratified soil profiles. Purely data-driven models do not have physical consistency and may fail beyond the training domain. Existing PINN applications to pile foundations do not effectively predict shaft and base resistance in layered soil profiles, and do not validate predictions against controlled laboratory tests. Therefore there is a need for a physics-informed framework that incorporates geotechnical

principle in layered soil profiles and validate against experimental database.

1.3 Research Questions

1. How can a PINN be establish for the feasibility of embedded geotechnical load transfer mechanisms, and how can it be compare with a Random Forest baseline?
2. How can the preliminary PINN into a multi-output framework predicting Q_s, Q_b, and Q_u along in layered soil profiles?
3. How does the enhanced PINN be compared with ANN, Random Forest, and XGBoost with accuracy, physical consistency, and generalization?
4. How can PINN-predicted load–settlement curves agree with laboratory pile load test observations after geometric scaling?

1.4 Research Objectives

The primary objective of this study is to develop and validate a physics-informed neural network for predicting axial pile load capacity integrating geotechnical load transfer mechanisms and data-driven machine learning. The specific objectives are:

1. **To develop a initial PINN framework** embedding Meyerhof's end bearing formulation, the β-method for shaft resistance, and effective stress principles, and compare it against Random Forest to establish feasibility and identify limitations of initial framework.
2. **To develop a multi-output PINN** with ten-layer soil profile representation, along with prediction of Q_s, Q_b, and Q_u, trainable physics weighting, and ensemble learning, addressing the limitations of the preliminary study.
3. **To compare the enhanced PINN model** with ANN, Random Forest, and XGBoost using statistical metrics, parity plots, physics residual analysis, and external validation.
4. **To validate the PINN framework** against laboratory pile load tests using geometric scaling ($L_r = 50$) to translate model-scale measurements to prototype dimensions.

1.5 Limitation of Study

1. **Dry sand only:** Cohesive soils, saturated conditions, and groundwater effects are not represented.
2. **1g laboratory testing:** Stress-level effects, prototype stresses are lower than actual field conditions at equivalent geometry.
3. **Axial loading only:** Lateral, moment, and dynamic loading are not within scope.

4. **Mohr–Coulomb model:** The PLAXIS 3D numerical dataset does not capture stress-path dependence, anisotropy, or time-dependent behavior.
5. **Synthetic training data:** The 5000-sample dataset is generated using governing equations.

1.6 Organization of Thesis

1. **Chapter 1: Introduction** presents the background, problem statement, objectives, and limitations.
2. **Chapter 2: Literature Review** reviews pile–soil interaction, conventional prediction methods, machine learning applications, and PINNs, and identifies the research gap.
3. **Chapter 3: Methodology** describes the two-phase methodology: Phase 1 (preliminary PINN vs Random Forest) along with its results and Phase 2 (multi-output PINN with layered soil, trainable physics, and ensemble learning).
4. **Chapter 4: Experimental Program** presents the experimental program at IIT Patna, including soil characterization, pile testing, and geometric scaling.
5. **Chapter 5: Results and Discussion** presents results from Experimental Program and Phase 2, model performance, PINN training behaviour, experimental load–settlement validation, and external validation.
6. **Chapter 6: Conclusion** gives summary of findings, limitations, and recommendations.

2 LITERATURE REVIEW

This chapter reviews pile–soil interaction principles, conventional pile capacity prediction methods, current machine learning applications in geotechnical engineering, and development of Physics-Informed Neural Networks. It also identifies the research gap.

2.1 Pile Foundations and Load–Settlement Behavior

The axial load–settlement curve of a pile is nonlinear because shaft resistance and end bearing are mobilized progressively (Coyle and Reese, 1966). Meyerhof (1976) said that pile design must address both ultimate capacity and settlement, as two piles with similar capacities may show significantly different responses. Full-scale load testing is reliable but expensive, motivating analytical, numerical, and data-driven prediction approaches.

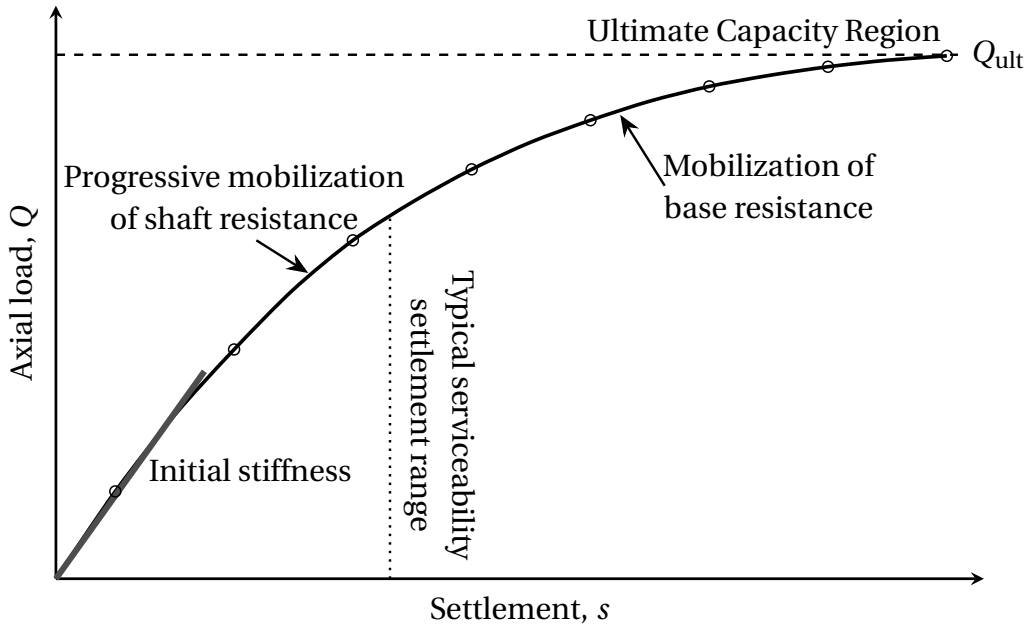

Figure 2.1: Conceptual axial pile load–settlement curve showing the progressive mobilization of shaft resistance at small settlements and base resistance at larger settlements

2.2 Pile–Soil Interaction Mechanisms

2.2.1 Shaft Resistance

Shaft resistance is mobilized along the pile shaft due to relative displacement between the pile surface and the surrounding soil. It shows the shear stress developed at the pile–soil interface and is influenced by soil type, effective stress level, pile roughness, installation method, and drainage condition. In cohesive soils, shaft resistance is often interpreted using adhesion-based or total stress approaches, whereas in cohesionless

soils it is commonly related to lateral earth pressure and interface friction (Coyle and Reese, 1966; Meyerhof, 1976).

Mobilization is progressive, also only part of the resistance activates at small displacements, with full resistance reached after sufficient movement. It explains why load transfer modeling is necessary rather than simply adding ultimate capacity.

2.2.2 End Bearing

End bearing (base resistance) is developed at the pile tip when compressive stress is transferred to the soil below the pile base. In contrast to shaft resistance, full toe resistance generally requires larger displacement for complete mobilization. End bearing depends on pile diameter, tip condition, the stiffness and strength of the bearing layer, and local stress distribution below the pile tip (Meyerhof, 1976).

In layered profiles, toe resistance is governed by the underlying bearing stratum, making stratigraphic soil profile representation essential.

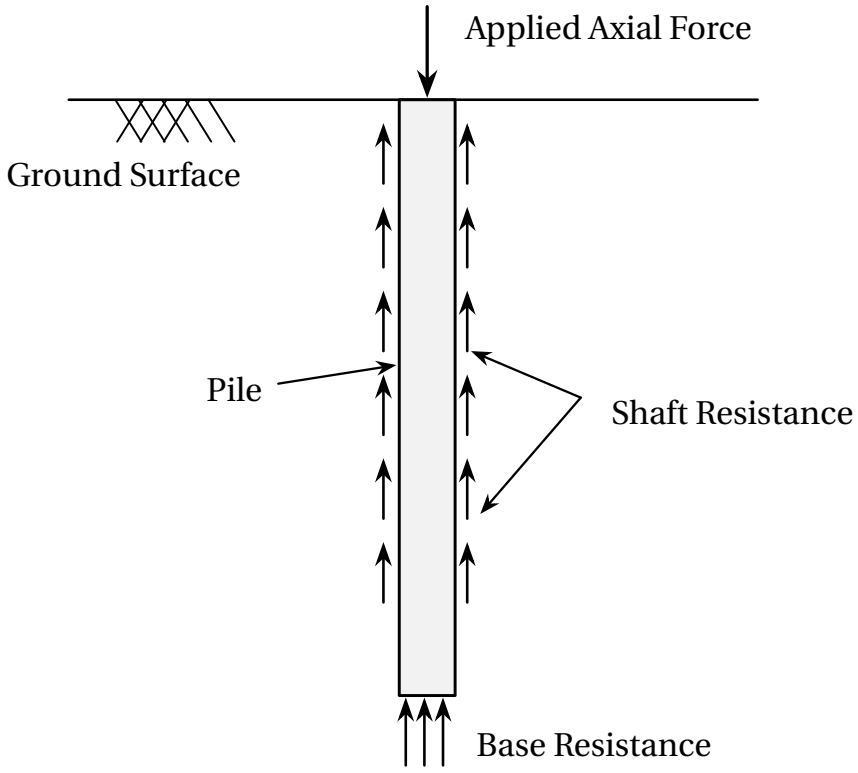

Figure 2.2: Schematic illustration of axial pile resistance components under an applied axial load.

2.2.3 Load Transfer Methods (t–z and q–z)

Load transfer methods relate unit shaft resistance t and toe resistance q to local displacement at each depth. By discretizing the pile into segments, the complete load–settlement response is calculated using equilibrium and compatibility (Coyle and

Reese, 1966), explicitly representing progressive resistance mobilization and bridging classical geotechnical principle with physics-informed modeling.

2.3 Theoretical Framework for Axial Pile Behavior

2.3.1 Axial Force Equilibrium

Force equilibrium with a differential pile element gives the governing equation for axial pile behavior. Let $P(z)$ denote the axial force, $w(z)$ the displacement, E_p the elastic modulus, A_p the cross-sectional area, U_p the perimeter, and $t(z)$ the unit shaft resistance at depth z. The axial force–deformation relationship is

$$P(z) = E_p A_p \frac{dw}{dz} \tag{2.1}$$

which represents the force–deformation relationship of the pile shaft. Taking a differential pile segment of length dz, equilibrium of forces requires that the reduction in axial force over the segment be balanced by the shaft resistance acting along the pile surface. This condition gives

$$\frac{dP}{dz} + U_p\, t(z) = 0 \tag{2.2}$$

Equation (2.2) shows that the rate of change of axial force with depth is controlled by the local shaft resistance. Substituting Equation (2.1) into Equation (2.2) gives the governing differential form for axial pile behavior:

$$\frac{d}{dz}\left(E_p A_p \frac{dw}{dz}\right) + U_p\, t(z) = 0 \tag{2.3}$$

If the pile stiffness $E_p A_p$ is assumed constant along the pile length, Equation (2.3) may be written in simplified form as

$$E_p A_p \frac{d^2 w}{dz^2} + U_p\, t(z) = 0 \tag{2.4}$$

Equation (2.4) shows that load is redistributed continuously along the pile rather than being a single capacity value, consistent with t–z and q–z concepts and the PINN physics constraint in Chapter 3.

2.3.2 Load Transfer Representation

The governing equilibrium equation requires constitutive relationships linking resistance to displacement. Unit shaft resistance and base resistance are expressed as

$$t(z) = f\big(w(z)\big) \tag{2.5}$$

$$q_b = g(w_b) \tag{2.6}$$

These displacement-dependent functions close the governing equation, enabling computation of the full load–settlement response. They are embedded directly as physical constraints in the PINN loss function in Chapter 3.

2.3.3 Bearing Capacity Components

The axial bearing capacity of a pile is composed of two primary components: the resistance mobilized along the pile shaft and the resistance developed at the pile base. These components correspond to shaft resistance and end bearing, respectively, and together define the total load-carrying capacity of the pile.

The total axial capacity Q of a pile can be expressed as

$$Q = Q_s + Q_b \tag{2.7}$$

where Q_s is the total shaft resistance and Q_b is the base resistance.

The shaft resistance is obtained by integrating the unit shaft resistance $t(z)$ along the embedded length of the pile:

$$Q_s = \int_0^L U_p \, t(z) \, dz \tag{2.8}$$

where U_p is the perimeter of the pile and L is the embedded length. This expression reflects the cumulative contribution of shear resistance mobilized along the pile–soil interface.

The base resistance is given by

$$Q_b = A_p \, q_b \tag{2.9}$$

where A_p is the cross-sectional area of the pile tip and q_b is the unit base resistance at the pile toe. While shaft resistance, which is mobilized progressively along the pile length, base resistance is primarily mobilized at larger displacements and depends strongly on the properties of the bearing stratum beneath the pile tip.

Combining Equations (2.8) and (2.9), the total axial capacity may be written as

$$Q = \int_0^L U_p \, t(z) \, dz + A_p \, q_b \tag{2.10}$$

This equation is the physical basis for the multi-output PINN in this study.

The governing equations and load transfer mechanism in this section — axial equilibrium, t–z/q–z functions, and $Q = Q_s + Q_b$ are used as physical constraints in the PINN loss function in Chapter 3.

2.4 Conventional Methods for Pile Capacity Prediction

Conventional methods estimate pile capacity as the sum of shaft resistance and end bearing using static analytical formula (Meyerhof, 1976) or empirical correlation of SPT/CPT. They are insufficient for layered profiles and do not predict the full load–settlement response, which motivates to use data-driven alternatives.

2.5 Machine Learning in Geotechnical Engineering

Goh (1995) mentioned that neural networks can model nonlinear geotechnical relationships, showing a wide range of data-driven applications.

2.5.1 Artificial Neural Networks

ANNs was among the first methods applied to pile capacity prediction and can perform better along with conventional empirical approaches with representative datasets (Lee et al., 2018; Pham et al., 2021), but are sensitive to data quality and offer limited interpretation capability.

2.5.2 Support Vector Machines

SVR offers competitive accuracy for smaller datasets (Udengaard et al., 2024) but remains totally data-driven with no embedded geotechnical principle.

2.5.3 Random Forest, XGBoost, and Ensemble Learning

In recent years, tree-based ensemble models such as Random Forest (RF), Gradient Boosting, and XGBoost have gained popularity solving geotechnical problems. These models are appealing because they can capture nonlinear interactions, are generally robust to overfitting, and provide some level of variable importance assessment. Ensemble methods have been applied successfully to pile bearing capacity prediction, producing higher accuracy than single-model approaches. (Ardakani et al., 2020; Benbouras et al., 2021; Dehghanbanadaki et al., 2021).

Gomes et al. (2021) proposed the use of machine learning models with semi-empirical method outputs as input features for driven pile bearing capacity in tropical soils. This study is important because it shows a hybrid strategy in which traditional geotechnical principle was incorporated indirectly into the data-driven framework.

Dehghanbanadaki et al. (2021) showed pile geometry and soil properties dominate predictive performance; Rizk et al. (2023) used ML to open-ended pipe piles.

2.5.4 Recent Developments in Advanced Machine Learning

Recent studies has shown more advanced machine learning frameworks. Umar et al. (2025) has studied multiple advanced models for geotechnical axial pile load capacity estimation and reported strong performance from an enhanced Gaussian Process Regression (GPR) framework. Their study is notable because it moved beyond raw prediction accuracy and emphasized model characterization and interpretability.

Across all methods, prediction targets ultimate capacity as a scalar; few models encode shaft resistance, end bearing, or depth-dependent load transfer mechanisms.

2.6 Limitations of Data-Driven Models

Despite strong predictive accuracy, data-driven models share four limitations in geotechnical engineering (Benbouras et al., 2021; Umar et al., 2025): (1) performance depends on training data representatives; (2) black-box operation offers no physical insight; (3) equilibrium and realistic load transfer behavior are not automatically enforced; and (4) extrapolation beyond the training domain is unreliable, especially for layered profiles.

2.7 Physics-Informed Neural Networks (PINN)

Physics-Informed Neural Networks was introduced as a learning framework in which neural networks are trained not only on observed data but also on the governing physical laws of the problem (Raissi et al., 2019).The network output is constrained by a loss function which includes both data mismatch term and one or more physics-based residual terms. The main idea is a model should not only fit the available data but also remain consistent with the physical principle which governs the system.

In general form, the PINN loss function can be expressed as

$$\mathscr{L} = \lambda_d \mathscr{L}_{data} + \lambda_p \mathscr{L}_{physics} + \lambda_b \mathscr{L}_{boundary}, \tag{2.11}$$

where $\mathscr{L}_{data}$ represents the error between predicted and observed values, $\mathscr{L}_{physics}$ represents the residual of governing equations or constitutive constraints, and $\mathscr{L}_{boundary}$ represents the satisfaction of boundary or initial conditions. The weighting parameters λ_d, λ_p, and λ_b control the relative contribution of each term.

The PINN concept can be required for geotechnical engineering because experimental data are often limited. Problems involving seepage, consolidation, deformation, and soil–structure interaction has already benefit from the integration of known physical relationships into the learning process. Gao et al. (2023) has shown this potential in seepage analysis through an improved PINN algorithm for phreatic line

prediction. Also, Yuan et al. (2025) discussed the growing importance of physics-informed machine learning in geotechnical engineering and highlighted its potential for a broad range of applications.

For pile related problems, Al Atroush (2025) has presented a deep learning PINN framework for predicting drilled shaft axial capacity. This study is more relevant because it has moved beyond conventional black-box learning by embedding geotechnical knowledge within the predictive model. The results suggested that PINN can show better balance between predictive accuracy and physical consistency in comparison with purely data-driven ANN models. Although this is an important step, the recent available literature on PINN applications to axial pile capacity remains limited in use and study, including for problems involving layered soils, load–settlement response, and also the combined use of laboratory and numerical datasets.

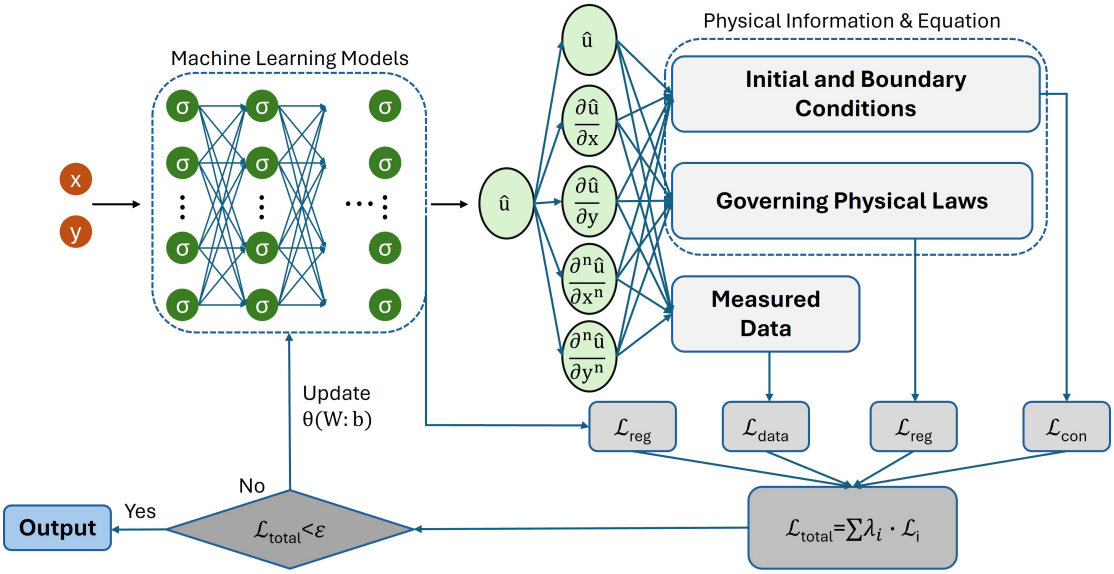

Figure 2.3: Schematic diagram of improved physics-informed approach with measured data (Yuan et al., 2025)

2.8 Research Gap

From recent literature, the key research gap can be stated as follows: there is a need for a predictive framework for axial pile capacity and load–settlement behavior that combines the accuracy of machine learning with the physical consistency of geotechnical theory, while accounting for soil stratigraphy, shaft resistance, end bearing, and displacement-dependent load transfer. This gap directly motivates the development of a Physics-Informed Neural Network model for predicting axial pile load capacity using experimental and numerical datasets.

Table 2.1: Summary of Relevant Literature

Author(s)	Focus Area	Method / Model	Input / Data Basis	Key Findings
Coyle and Reese (1966)	Axial pile response	Load-transfer concept for axially loaded piles in clay	Pile shaft behavior and displacement-dependent mobilization of resistance	Established that pile behavior should be modeled using progressive load transfer rather than only ultimate capacity
Meyerhof (1976)	Pile capacity and settlement	Classical static analytical approach	Soil strength parameters, pile geometry, shaft resistance, end bearing	Provided one of the most influential frameworks for estimating pile capacity and settlement using geotechnical principles
Goh (1995)	Geotechnical prediction	Artificial Neural Network (ANN)	Empirical geotechnical datasets	Demonstrated that neural networks can effectively model nonlinear geotechnical relationships
Lee et al. (2018)	Pile load capacity prediction	Comparative machine learning models	Pile database with capacity-related input features	Showed that multiple machine learning techniques can predict pile capacity with improved accuracy compared with traditional approaches

Author(s)	Focus Area	Method / Model	Input / Data Basis	Key Findings
Ardakani et al. (2020)	Driven pile bearing capacity	Ensemble machine learning techniques	Pile geometry and soil-related parameters	Found that ensemble learning improves prediction accuracy and robustness for pile capacity estimation
Gomes et al. (2021)	Driven piles in tropical soils	Machine learning with semi-empirical inputs	Semi-empirical design outputs combined with pile and soil parameters	Showed that integrating conventional geotechnical methods into ML inputs improves predictive capability
Benbouras et al. (2021)	Driven pile bearing capacity	Advanced ML models including deep learning	Database of driven pile cases	Reported strong predictive performance of advanced machine learning models and highlighted their engineering applicability
Pham et al. (2021)	Bored pile bearing capacity	Machine learning techniques	Bored pile dataset with geotechnical and geometric inputs	Demonstrated that machine learning can effectively predict bored pile capacity under varying conditions
Dehghanbanadaki et al. (2021)	Pile bearing capacity estimation	Machine learning with sensitivity analysis	Soil and pile parameters	Showed that some variables, especially pile dimensions and soil properties, have dominant influence on model performance

Author(s)	Focus Area	Method / Model	Input / Data Basis	Key Findings
Rizk et al. (2023)	Open-ended pipe piles	Machine learning models	Geometry and soil-related input features	Demonstrated successful prediction of load capacity and highlighted the influence of pile geometry
Udengaard et al. (2024)	Pile capacity prediction	Support Vector Regression (SVR)	Pile dataset and capacity-related features	Indicated that SVR can provide competitive accuracy in pile capacity estimation
Umar et al. (2025)	Axial pile load capacity estimation	Advanced ML models with enhanced GPR approach	Geotechnical axial pile load dataset	Reported very high prediction accuracy and emphasized model characterization, interpretability, and uncertainty awareness
Raissi et al. (2019)	Physics-informed learning	Physics-Informed Neural Network (PINN)	Data plus governing physical equations and boundary conditions	Established the general PINN framework for combining observations with physical residuals in neural network training
Gao et al. (2023)	Geotechnical/ civil physics-informed prediction	Improved PINN algorithm	Seepage-related governing physics and observational data	Demonstrated that PINN can improve physical realism and predictive stability in civil engineering problems

Author(s)	Focus Area	Method / Model	Input / Data Basis	Key Findings
Al Atroush (2025)	Drilled shaft axial capacity	Deep learning PINN	Drilled shaft data with embedded geotechnical knowledge	Showed that PINN can improve the balance between prediction accuracy and physical consistency for axial capacity problems
Yuan et al. (2025)	Geotechnical engineering	Direction paper on physics-informed machine learning	Broad review of geotechnical applications	Identified physics-informed ML as a promising direction for geotechnical problems with limited data and strong prior knowledge
Ito et al. (2024)	Geotechnical parameter estimation through inverse analysis	Physics-Informed Neural Network (PINN) for inverse analysis of consolidation and unsaturated seepage	Numerical data for 1D consolidation and laboratory soil-column measurements for unsaturated seepage; estimation of coefficient of consolidation and unsaturated hydraulic parameters	Showed that PINN could accurately reproduce excess pore-pressure dissipation and unsaturated seepage behavior, and could estimate the coefficient of consolidation, saturated hydraulic conductivity, and soil-water characteristic curve parameters with sufficient accuracy

Author(s)	Focus Area	Method / Model	Input / Data Basis	Key Findings
Madianos et al. (2023)	Axial soil–pile interaction in inhomogeneous soil	Physics-Informed Neural Network for Sturm–Liouville eigenvalue/eigenfunction problem in soil–pile interaction	Semi-analytical continuum model for an axially loaded end-bearing pile embedded in Gibson-type inhomogeneous soil; PINN trained to recover soil modes and eigenvalues	The proposed PINN achieved excellent agreement with the analytical/semi-analytical solution for eigenvalues and eigenfunctions, demonstrating accurate solution of the governing boundary-value problem without external data
Wu et al. (2026)	Dynamic pile–soil interaction	Transfer learning enhanced progressive multi-physics-informed neural networks (TLP-mPINNs)	Coupled pile and soil PDEs solved using separate DNNs for pile and soil, progressive alternating training, transfer learning, and verification against existing solutions	Demonstrated that TLP-mPINNs improved training stability, accelerated convergence, and produced more accurate dynamic pile and soil responses

3 METHODOLOGY

The research framework that is used in this study has integrated experimental investigation, numerical simulation, and data-driven modeling within a same workflow.

The main objective is to develop physically consistent predictive model for axial pile load capacity by combining geotechnical principles with machine learning techniques.

This framework consists of four main components:

- Experimental data using laboratory pile load tests
- Numerical data using PLAXIS 3D.
- Data-driven modeling using machine learning and PINNs
- Model evaluation and validation using statistical and physical criteria

The overall workflow is illustrated in Figure 3.1.

Key portions of the Python scripts developed in this study are provided in Appendix A.

3.1 Data Generation

3.1.1 Laboratory Pile Load Tests

Laboratory-scale static pile load tests were done as a key data source for model validation. The detailed experimental program are presented in Chapter 4.

Model piles (mild steel, diameters 20–40 mm, lengths 150–250 mm) were installed in sand beds at relative densities of 40%, 60%, and 80%, including both uniform and layered profiles. Axial load was applied incrementally and measured with a load cell; settlement was recorded with an LVDT.

The resulting load–settlement curves provide ground-truth data for PINN validation after geometric scaling to prototype dimensions (Chapter 4).

3.1.2 Numerical Simulation using PLAXIS 3D

Numerical simulations was carried out using PLAXIS 3D to generate synthetic data for training and validating the developed models. The simulations was designed to capture bearing capacity of different soil pile cases.

A single vertical pile of different length/ diameter in 3D soil domain was modeled with Mohr–Coulomb soil (c, ϕ, γ, E_s, ν). Boundary conditions: base fixed, lateral faces horizontally restrained, surface free. Ultimate capacity was extracted using automated Python scripting for batch execution across the full parameter space.

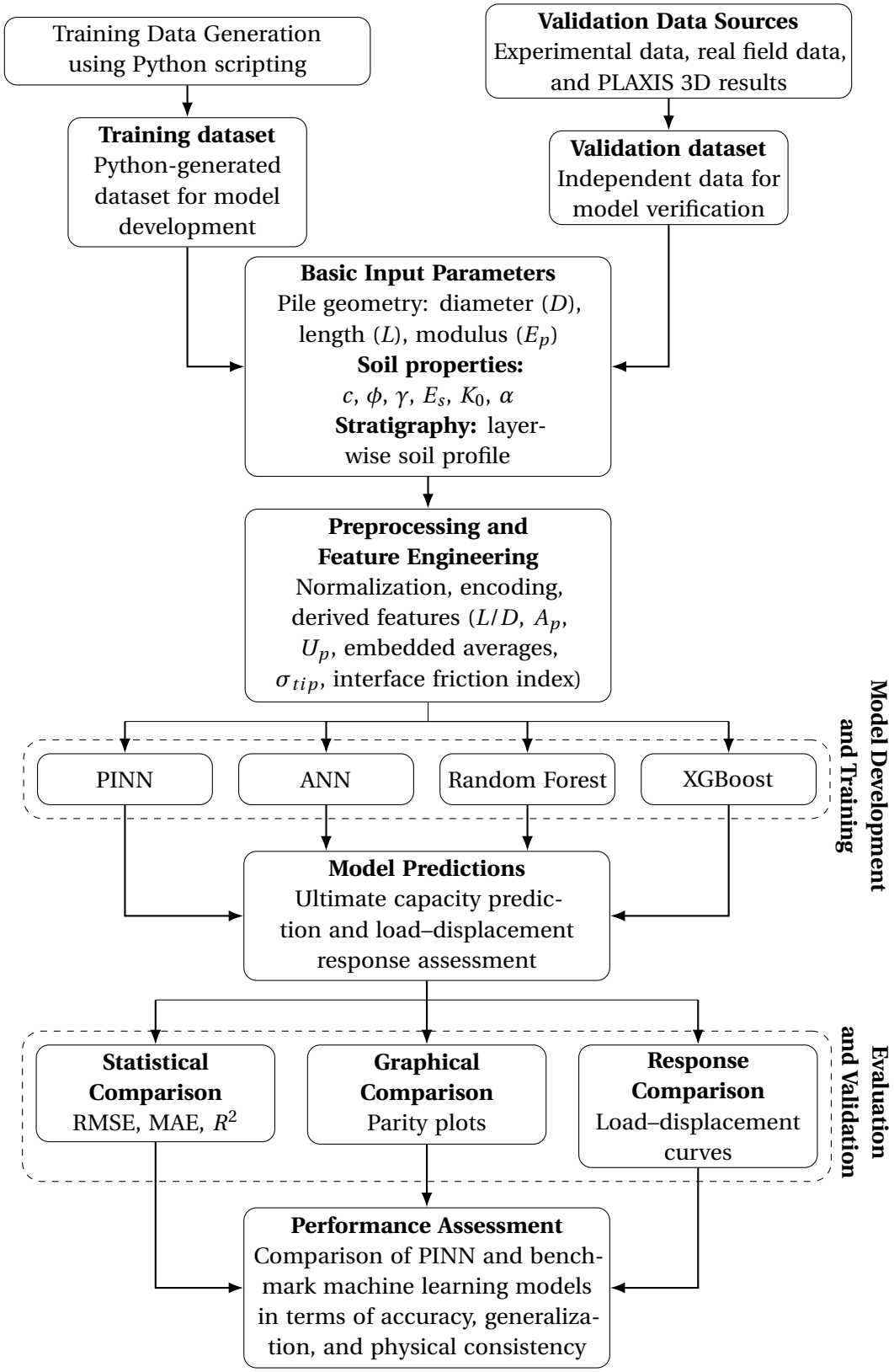

Figure 3.1: Research framework adopted in this study

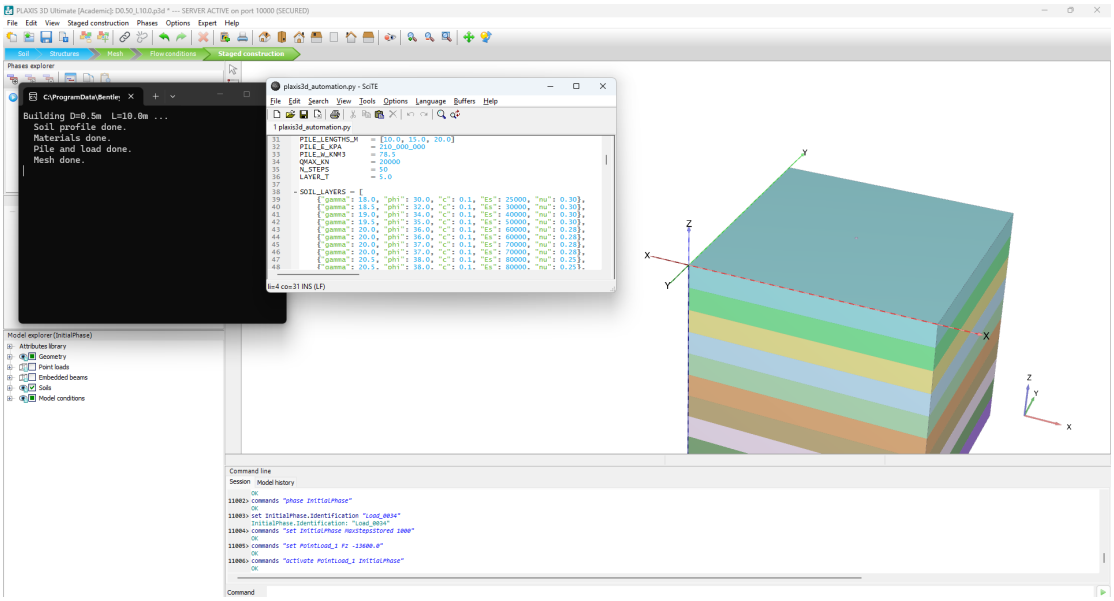

Figure 3.2: PLAXIS 3D Automation using Python Script using Academic License

3.2 Input Parameters and Dataset Preparation

3.2.1 Pile Parameters

The pile response is governed by its geometric and material properties, which directly influence load transfer and deformation behavior. In this study, the following pile parameters were considered:

- Pile length (L)
- Pile diameter (D)
- Cross-sectional area ($A_p = \pi D^2/4$)
- Perimeter ($U_p = \pi D$)
- Young's modulus of pile material (E_p)

These parameters are essential in defining the axial stiffness of the pile and its interaction with the surrounding soil. In particular, the axial force in the pile is expressed as

$$P(z) = E_p A_p \frac{dw}{dz} \tag{3.1}$$

The pile parameters are varied within realistic engineering ranges to generate a diverse dataset suitable for training machine learning and physics-informed models.

3.2.2 Soil Properties

Soil behavior plays important role in determining pile capacity and load–settlement response. In this study, soil properties were defined at each depth layer and included:

- Cohesion (c)
- Angle of internal friction (ϕ)
- Unit weight (γ)
- Elastic modulus (E_s)
- Coefficient of earth pressure (K_0)
- Adhesion factor (α)

The shaft resistance was calculated based on soil type using classical geotechnical relationships. For cohesive soils, shaft resistance was expressed as

$$t(z) = \alpha\, c \tag{3.2}$$

where α is the adhesion factor.

For cohesionless soils, shaft resistance is expressed as

$$t(z) = K_0\, \sigma'_v \tan(\delta) \tag{3.3}$$

where σ'_v is the effective vertical stress and δ is the interface friction angle.

The base resistance is computed using a bearing capacity formula:

$$q_b = cN_c + \sigma'_v N_q + 0.5\gamma D N_\gamma \tag{3.4}$$

where N_q and N_γ are bearing capacity factors dependent on ϕ.

These formula makes sure that the generated dataset remains consistent with established geotechnical principles.

3.2.3 Stratigraphy Representation

The soil profile was modeled using a layered system with a total depth of 50 m. The soil stratigraphy was discretized into layers of 5 m thickness, which results in 10 layers per profile.

Each soil layer was assigned independent soil properties, allowing variation in soil type and parameters with depth. Consecutive soil layers may have identical properties; however, each layer was treated as a separate entity to preserve depth-dependent representation.

The shaft resistance for each layer was calculated and summed to obtain total shaft resistance:

$$Q_s = \sum_{i=1}^{n} U_p\, t_i\, \Delta z \tag{3.5}$$

where $\Delta z = 5$ m is the layer thickness.

The total pile capacity is then calculated as

$$Q = Q_s + A_p q_b \tag{3.6}$$

This layered soil representation lets the model to capture the effect of soil stratigraphy, which is very h ard for realistic prediction of pile behavior.

3.2.4 Dataset Generation

A synthetic dataset of 5,000 cases was generated using a Python based simulation framework implementing the governing geotechnical equations for shaft resistance, base resistance, and total axial pile capacity. Each case represents a unique combination of pile geometry, pile type (for Young's modulus of Elasticity), and a ten-layer soil stratigraphy independently sampled from eight soil classes spanning clays and sands. The parameter ranges are shown in Table 3.1.

Pile geometry and type Pile diameter D was sampled uniformly in [0.40, 1.50] m and embedded length L in [8.0, 30.0] m. Three pile types were included: bored concrete ($E_p = 27$ GPa), driven concrete ($E_p = 30$ GPa), and steel pipe ($E_p = 200$ GPa), introducing variation in axial stiffness.

Soil stratigraphy The profile was discretised into ten layers of 5 m each (total depth 50 m). Each layer was independently assigned one of eight soil classes by uniform random selection and the layer properties were sampled uniformly within the class-specific bounds in Table 3.1. This produces heterogeneous layered profiles representative of variable depositional sequences.

Capacity calculation Shaft resistance was computed layer by layer:

$$f_s = \alpha\, c + K_0\, \sigma'_v \tan\delta, \qquad \delta = 0.75\,\phi \tag{3.7}$$

where α is the adhesion factor, σ'_v is the effective vertical stress at layer mid-depth, and δ is the interface friction angle. Base resistance used Meyerhof bearing capacity factors:

$$q_b = c\, N_c + \sigma'_{v,\text{tip}}\, N_q + 0.5\,\gamma\, D\, N_\gamma \tag{3.8}$$

where $N_q = e^{\pi \tan\phi} \tan^2(45^\circ + \phi/2)$, $N_c = (N_q - 1)\cot\phi$, and $N_\gamma = (N_q - 1)\tan(1.4\phi)$. Total capacity is $Q_u = Q_s + Q_b$.

3.2.5 Data Normalization

To improve the stability and convergence during training of machine learning models, all input features were normalized prior to training. Min–max normalization

Table 3.1: Parameter ranges used in synthetic dataset generation (5,000 cases, 10 soil layers per case, random seed = 42)

Category	Parameter	Symbol	Range	Unit
Pile geometry	Diameter	D	0.40 – 1.50	m
	Embedded length	L	8.0 – 30.0	m
	Elastic modulus	E_p	27,000 – 200,000	MPa
Soil properties (per layer)	Cohesion	c	0 – 100	kPa
	Friction angle	ϕ	3 – 45	°
	Unit weight	γ	16.0 – 21.0	kN/m^3
	Elastic modulus	E_s	4,000 – 110,000	kPa
	Earth pressure coefficient	K_0	0.30 – 1.10	–
	Adhesion factor	α	0.0 – 1.0	–
Soil classification	Clay types: Very Soft Clay, Soft Clay, Stiff Clay			
	Sand types : Loose Sand, Medium Sand, Dense Sand, Silty Sand, Gravelly Sand			
	Profile: 10 layers × 5 m = 50 m total depth			
Outputs	Shaft resistance	Q_s	Computed from Eq. (3.7)	
	Base resistance	Q_b	Computed from Eq. (3.8)	
	Ultimate capacity	Q_u	$Q_s + Q_b$	

technique was adopted:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.9}$$

This transformation scales all input variables to a common range, preventing dominance of variables with large numerical values and improving model performance.

3.3 Machine Learning Models

In order to evaluate the performance of the proposed Physics-Informed Neural Network (PINN), conventional machine learning models were developed as benchmark approaches. These models are trained using the same dataset and input parameters, enabling a direct and consistent comparison with the PINN framework.

The selected models include Artificial Neural Networks (ANN), Random Forest (RF), and Extreme Gradient Boosting (XGBoost), which are widely used for regression problems involving nonlinear relationships.

3.3.1 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) are used to model the relationship between input parameters and pile response through nonlinear transformations. The ANN used in this study is a feed forward network which consists of an input layer, one or more hidden layers, and an output layer.

The ANN model can be expressed as

$$y = \mathcal{N}(\mathbf{x}; \boldsymbol{\theta}) \tag{3.10}$$

where $\mathbf{x}$ represents the input features (pile geometry, soil properties, and stratigraphy), y is the predicted output (load), and $\boldsymbol{\theta}$ denotes the network parameters (weights and biases).

The network parameters are optimized by minimizing the prediction error using a loss function of the form

$$\mathscr{L} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i^{pred} - y_i^{obs} \right)^2 \tag{3.11}$$

3.3.2 Random Forest (RF)

Random Forest is an ensemble learning method based on multiple decision trees. Each tree is trained using a randomly sampled subset of the dataset, and the final prediction is obtained by averaging the outputs of all trees.

The prediction from the Random Forest model is given by

$$y = \frac{1}{M} \sum_{j=1}^{M} T_j(\mathbf{x}) \tag{3.12}$$

where $T_j(\mathbf{x})$ represents the prediction from the j^{th} decision tree and M is the total number of trees.

Random Forest is used in this study as a robust nonlinear regression model for comparison with PINN predictions.

3.3.3 Extreme Gradient Boosting (XGBoost)

XGBoost is a gradient boosting algorithm that builds an ensemble of decision trees sequentially. Each tree is trained to minimize the residual error of the previous model.

The prediction is expressed as

$$y = \sum_{k=1}^{K} f_k(\mathbf{x}) \tag{3.13}$$

where $f_k(\mathbf{x})$ represents the k^{th} decision tree and K is the total number of boosting iterations.

The objective function minimized in XGBoost includes both prediction error and regularization:

$$\mathscr{L} = \sum_{i=1}^{N} \ell \left(y_i^{obs}, y_i^{pred} \right) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.14}$$

where ℓ is the loss function (typically squared error) and $\Omega(f_k)$ is a regularization term controlling model complexity.

3.4 Development of Physics-Informed Neural Network

The Physics-Informed Neural Network (PINN) framework developed in this study is designed to predict the ultimate axial capacity of pile foundations by integrating geotechnical mechanics with data-driven learning. In comparison with conventional method, PINN rely solely on governing differential equations, the present model incorporates both load-transfer mechanisms and empirical geotechnical relationships to ensure physical consistency and improved generalization.

The model is formulated as a hybrid learning system which combines:

- Data-driven learning (synthetic and experimental datasets)
- Physics based constraints (geotechnical theory)
- Feature engineering to embed domain knowledge into inputs

This approach is aligned with modern PINN methodologies where physical laws are embedded directly into the loss function in order to improve robustness and interpretability (Raissi et al., 2019; Yuan et al., 2025).

The PINN development is proceeds in two phases. Section 3.5 describes the preliminary Phase 1 framework, its results, and its limitations that motivated the enhanced Phase 2 architecture which is described in the remaining subsections.

3.5 Phase 1: Preliminary PINN Framework

A preliminary single-output PINN is developed and was compared against a Random Forest baseline model to confirm that embedding geotechnical physics into the loss function produces measurable improvements over a purely data-driven approach, and also to identify the specific limitations that would guide Phase 2 design.

3.5.1 Governing Equations

The Phase 1 PINN has incorporated three standard geotechnical relationships into its loss function.

Effective Vertical Stress

$$\sigma'_v(z) = \gamma' z \tag{3.15}$$

where γ' is the effective unit weight of the soil.

End Bearing (Meyerhof)

$$Q_b = N_q \, \sigma'_v(L) \, A_b, \qquad A_b = \frac{\pi D^2}{4} \tag{3.16}$$

$$N_q = e^{(\pi \tan\phi)} \tan^2\left(45^\circ + \frac{\phi}{2}\right) \tag{3.17}$$

where $\sigma'_v(L)$ is the effective vertical stress at the pile tip depth L, A_b is the pile base area, and ϕ is the friction angle at the pile tip.

Shaft Resistance (β-method)

$$Q_s = \sum_i \sigma'_v(z_i) \tan\phi \, p \, \Delta z, \qquad p = \pi D \tag{3.18}$$

where z_i is the midpoint depth of layer i, Δz is the layer thickness, and $p = \pi D$ is the pile perimeter.

A trainable shaft calibration factor β is introduced, and constrained to $(0, 1)$ using sigmoid activation and scaling unit shaft resistance as:

$$t(z) = \beta \, \sigma'_v(z) \tan\phi \tag{3.19}$$

Ultimate Capacity

$$Q_u = Q_b + Q_s \tag{3.20}$$

3.5.2 Loss Function

$$\mathscr{L} = \lambda_{data}\mathscr{L}_{data} + \lambda_{physics}\mathscr{L}_{physics} + \lambda_{con}\mathscr{L}_{con} + \lambda_{reg}\mathscr{L}_{reg} \tag{3.21}$$

$\mathscr{L}_{data}$ is the MSE between predicted and observed Q_u. $\mathscr{L}_{physics}$ enforces the governing geotechnical equations. $\mathscr{L}_{con}$ enforces $\widehat{Q}_u \approx \widehat{Q}_b + \widehat{Q}_s$. $\mathscr{L}_{reg}$ penalises large deviations of trainable parameters from initial values. All four λ weights are fixed constants in Phase 1.

3.5.3 Benchmark Model

A Random Forest regressor (300 trees, maximum depth = 12) served as the conventional baseline, which is selected for its consistent performance on geotechnical datasets and robust feature importance measures (Ardakani et al., 2020; Gomes et al., 2021).

3.5.4 *Results*

The Phase 1 PINN was trained for 4000 epochs using the Adam optimizer. The trainable β converged from 0.5 to approximately 0.43 within the first 1500 epochs, which shows stable adaptive physics weighting without manual calibration.

Quantitative performance is shown in Figure 3.3. The PINN has achieved $R^2 = 0.940$ (RMSE = 1441 kN, MAE = 1010 kN) versus Random Forest $R^2 = 0.919$. External validation gave $R^2 = 0.938$, which conforms robust generalisation to data from PLAXIS 3D simulations, bridge sector foundations, and laboratory tests.

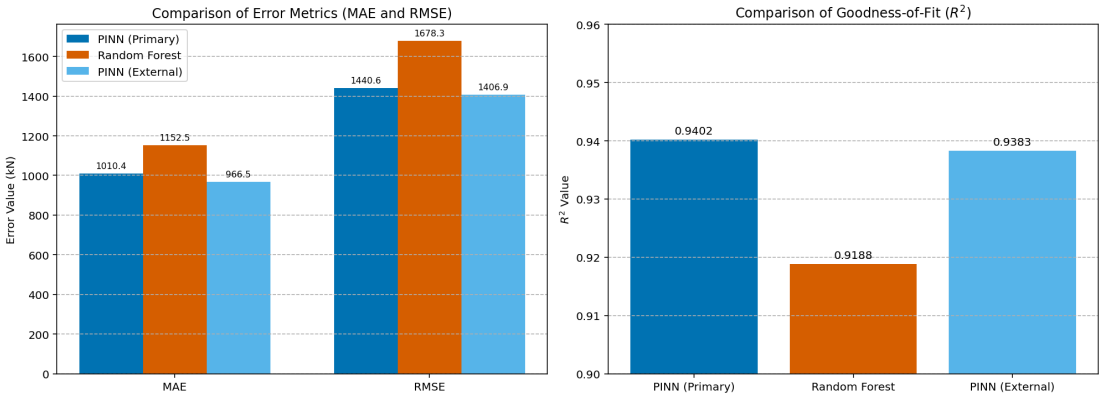

Figure 3.3: Quantitative performance comparison of the Phase 1 PINN and Random Forest on internal test and external validation datasets.

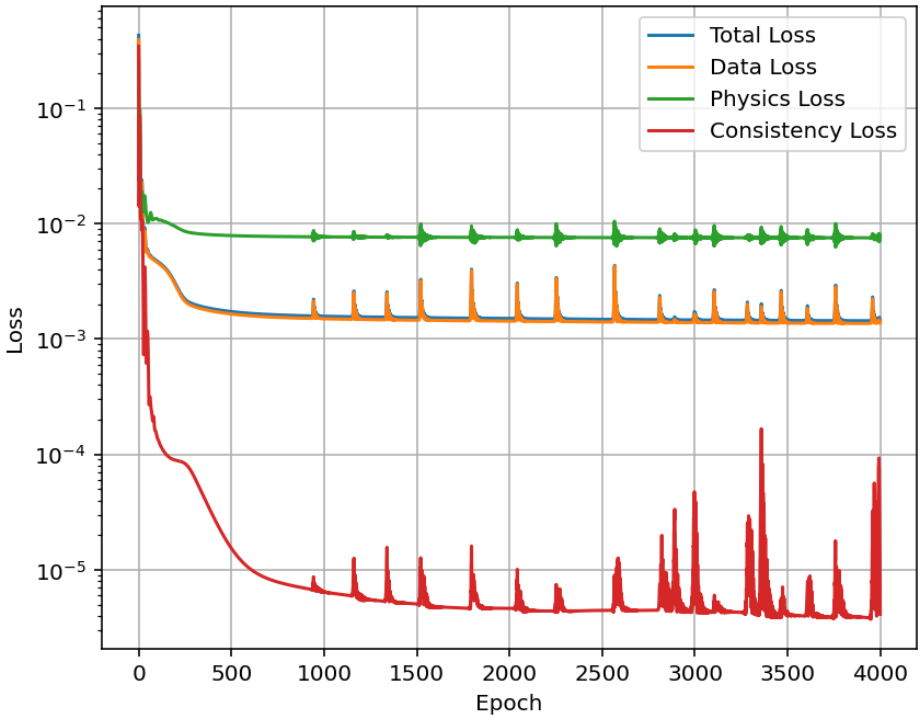

Figure 3.4: Phase 1 PINN training loss convergence showing stable progressive enforcement of physics constraints.

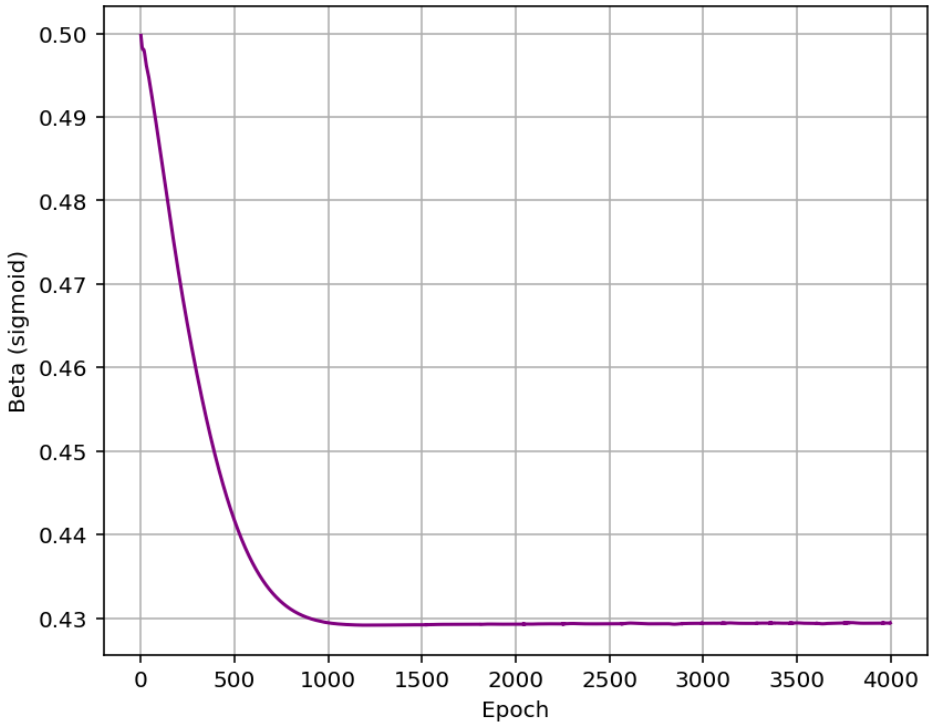

Figure 3.5: Convergence of the trainable β parameter during Phase 1 training (0.5 → 0.43).

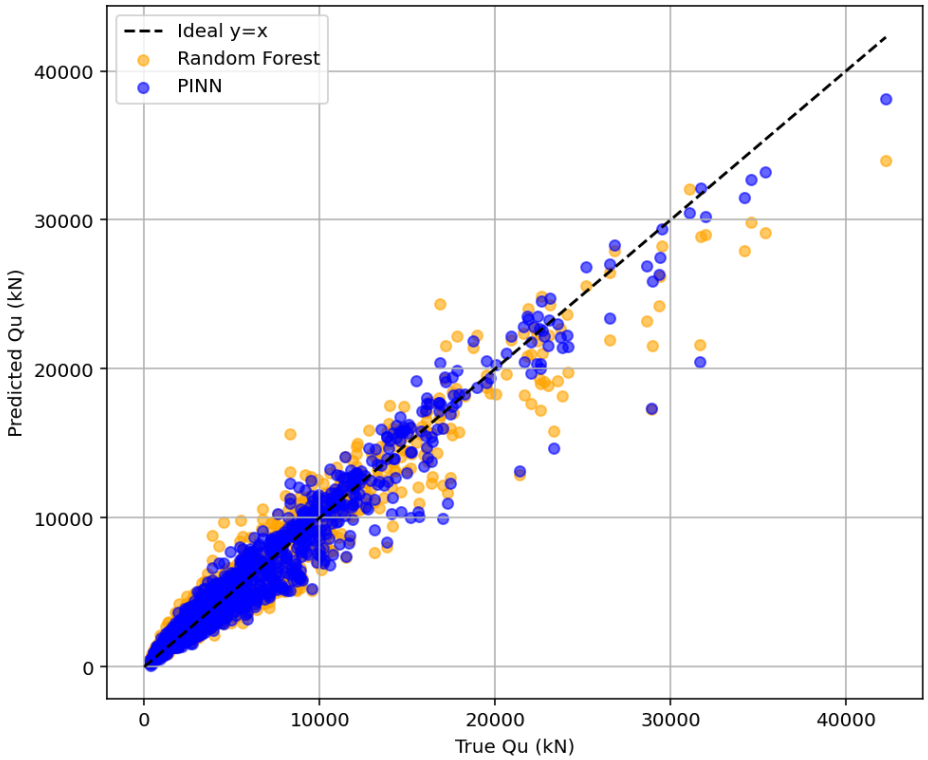

Figure 3.6: Phase 1 parity plot: PINN predictions cluster more tightly around the y = x line than Random Forest, especially at higher capacities.

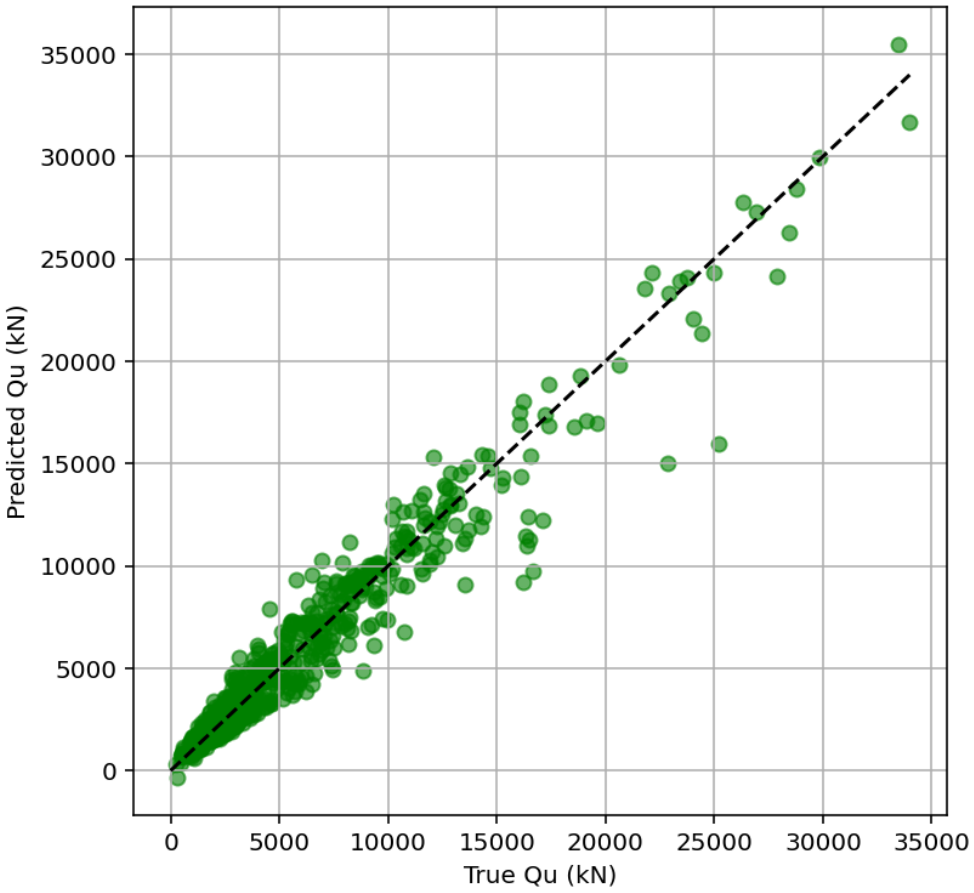

Figure 3.7: Phase 1 external validation parity plot ($R^2 = 0.938$).

3.5.5 Limitations Identified

Three limitations motivated the Phase 2 framework:

1. **Single-output prediction.** Only Q_u was predicted; Q_s and Q_b could not be decomposed separately, which are needed for load–settlement analysis.
2. **Simplified soil representation.** A single-layer effective stress profile was used, preventing representation of layered stratigraphy.
3. **Fixed physics loss weights.** Fixed λ values caused training instability when applied to the more complex multi-output layered-soil problem, requiring trainable λ parameters in Phase 2.

3.6 Phase 2: Enhanced Multi-Output PINN Framework

The enhanced framework addresses the three Phase 1 limitations: single-output prediction is replaced by simultaneous $Q_s/Q_b/Q_u$ prediction; single-layer soil is replaced by a ten-layer stratified representation; and fixed λ weights are replaced by trainable parameters. The following subsections describe the governing equations, architecture, loss formulation, and training strategy.

3.6.1 Governing Equation of Axial Pile Behavior

The axial response of a pile is governed by force equilibrium along its length:

$$\frac{d}{dz}\left(E_p A_p \frac{dw}{dz}\right) + U_p t(z) = 0 \tag{3.22}$$

where:

- E_p = pile elastic modulus
- A_p = cross-sectional area
- U_p = pile perimeter
- $t(z)$ = unit shaft resistance

This equation is embedded into the PINN as a physics constraint.

3.6.2 Geotechnical Load Transfer Formulation

Shaft Resistance (β-method)

$$t(z) = \beta(z)\,\sigma'_v(z) \tag{3.23}$$

where $\beta(z)$ is treated as a **trainable parameter**.

End Bearing Resistance

$$q_b = N_q\,\sigma'_{v,tip} \tag{3.24}$$

Total axial capacity is computed as:

$$Q_u = Q_s + Q_b = \int_0^L U_p\,t(z)\,dz + A_p q_b \tag{3.25}$$

This formula ensures consistency with classical geotechnical theory.

3.6.3 Neural Network Formulation

The PINN is designed as a multi-output model:

$$(Q_s, Q_b, Q_u) = \mathcal{N}(\mathbf{x};\boldsymbol{\theta}) \tag{3.26}$$

where **x** includes:

- Pile geometry (L, D, A_p, U_p)
- Soil properties (c, ϕ, γ, E_s)
- Engineered features (e.g., L/D, σ_{tip})

3.6.4 Feature Engineering

To enhance learning efficiency, physically meaningful features as below are introduced:

$$\frac{L}{D}, \quad A_p = \frac{\pi D^2}{4}, \quad U_p = \pi D \tag{3.27}$$

$$\sigma_{tip} = \gamma L \tag{3.28}$$

These features help the model learn geotechnical relationships more effectively.

3.6.5 Physics Residual Formulation

The equilibrium residual is defined as:

$$R(z) = \frac{d}{dz}\left(E_p A_p \frac{d\hat{w}}{dz}\right) + U_p t(z) \tag{3.29}$$

Additionally, global consistency is enforced as below:

$$R_{global} = Q_u - (Q_s + Q_b) \tag{3.30}$$

3.6.6 Loss Function Formulation

The total loss function is defined as:

$$\mathscr{L} = \mathscr{L}_{data} + \lambda_p \mathscr{L}_{physics} + \lambda_b \mathscr{L}_{boundary} + \lambda_g \mathscr{L}_{global} \tag{3.31}$$

Data Loss

$$\mathscr{L}_{data} = \text{Huber}(Q_u^{pred}, Q_u^{true}) \tag{3.32}$$

Physics Loss

$$\mathscr{L}_{physics} = \frac{1}{N}\sum R(z_i)^2 \tag{3.33}$$

Global Consistency Loss

$$\mathscr{L}_{global} = (Q_u - (Q_s + Q_b))^2 \tag{3.34}$$

This multi-term loss ensures that, both data accuracy and physical consistency, which is a key advantage of PINNs over traditional neural networks.

3.6.7 Training Strategy

Two stage training strategy was adopted:

- **Stage 1:** Data-driven pretraining (physics disabled)
- **Stage 2:** Physics-informed training (full loss)

Optimization is performed using:

- Adam optimizer (initial training)
- L-BFGS optimizer (fine-tuning)

3.6.8 Bearing Capacity Prediction Framework

The final prediction combines learned components:

$$Q_u^{PINN} = Q_s^{PINN} + Q_b^{PINN} \tag{3.35}$$

This ensures:

- Physical interpretability
- Reduced overfitting
- Better extrapolation capability

3.6.9 Hyperparameter Settings

The final PINN and benchmark models were trained using the hyperparameter settings which is summarized in Table 3.2. These values were selected to balance predictive accuracy, training stability, and computational efficiency. The PINN employed a staged optimization strategy using Adam followed by L-BFGS refinement, along with trainable physics weights and a gradual physics warm-up schedule. Such dynamic balancing of data and physics losses is consistent with recent PINN developments in geotechnical and scientific machine learning applications.

Table 3.2: Hyperparameter settings used for PINN, ANN, Random Forest, and XGBoost models.

Category	Hyperparameter	Value	Description
General configuration			
Reproducibility	Random seed	42	Base random seed used for reproducibility.
Reproducibility	Ensemble seeds	(42, 52, 62)	Seeds used for training three PINN ensemble members.

Category	Hyperparameter	Value	Description
Data split	Test size	0.25	Fraction of total dataset reserved for testing.
Data split	Validation fraction within training	0.15	Fraction of training data reserved for validation.
Data files	Training dataset	5000 datasets	Main training dataset file.
Data files	External validation dataset	500 datasets	Independent dataset used for external validation.
Input representation and feature engineering			
Feature setting	Pile type one-hot encoding	True	Enables categorical encoding of pile type when available.
Feature setting	Include E_s	True	Includes soil elastic modulus in the feature set.
Feature setting	Engineered features	True	Enables derived geotechnical features such as L/D, A_p, U_p, and stress proxies.
Profile discretization	Total profile depth	50 m	Total soil depth considered in the input profile.
Profile discretization	Layer thickness	5 m	Thickness of each soil layer.
Profile discretization	Number of layers	10	Total number of layers in the soil profile.
PINN architecture			
PINN structure	Activation function	tanh	Activation used in PINN hidden layers.
PINN structure	Q-network hidden layers	(256, 256, 128)	Hidden-layer sizes for multi-output capacity network predicting Q_s, Q_b, and Q_u.

Category	Hyperparameter	Value	Description
PINN structure	W-network hidden layers	(256, 256, 256, 128)	Hidden-layer sizes for displacement field network used in physics enforcement.
PINN training			
Training	Epochs	500	Maximum number of Adam optimization epochs.
Training	Batch size	64	Mini-batch size used during training.
Training	Learning rate	1×10^{-3}	Initial learning rate for Adam optimizer.
Training	Early stopping patience	70	Number of epochs without improvement before stopping training.
Training	Collocation points	48	Number of collocation points per sample used in physics residual evaluation.
Training	Use L-BFGS refinement	True	Enables second-stage optimization after Adam training.
Training	L-BFGS maximum iterations	150	Maximum iterations for L-BFGS refinement.
Training	L-BFGS learning rate	0.8	Step size for L-BFGS optimizer.
Loss formulation and physics control			
Data loss	Huber loss	True	Uses Huber loss instead of MSE for improved robustness to noisy targets.
Data loss	Huber delta (scaled)	0.03	Threshold parameter for Huber loss in normalized space.

Category	Hyperparameter	Value	Description
Trainable physics	Trainable β	True	Enables trainable shaft resistance calibration factor.
Trainable physics	Initial β logit	0.0	Initial value of raw β parameter; corresponds to $\beta = 0.5$ after sigmoid mapping.
Trainable physics	Trainable λ weights	True	Enables adaptive learning of physics-loss weights.
Loss weights	$\lambda_{p,\max}$	1.0	Maximum weight for PDE residual loss.
Loss weights	$\lambda_{b,\max}$	1.0	Maximum weight for boundary-condition loss.
Loss weights	$\lambda_{g,\max}$	1.0	Maximum weight for global physics-consistency loss.
Loss weights	$\lambda_{c,\max}$	1.0	Maximum weight for multi-output consistency loss.
Regularization	$\lambda_{\beta,\text{reg}}$	1×10^{-4}	Regularization weight for trainable β.
Regularization	$\lambda_{\lambda,\text{reg}}$	1×10^{-4}	Regularization weight for adaptive loss weights.
Physics warm-up and mobilization			
Warm-up	Physics warm-up start epoch	10	Epoch at which physics losses start contributing.
Warm-up	Physics warm-up end epoch	140	Epoch at which full physics weight is activated.
Mobilization	Shaft mobilization factor η_s	0.01	Controls shaft resistance mobilization with displacement.

Category	Hyperparameter	Value	Description
Mobilization	Base mobilization factor η_b	0.05	Controls base resistance mobilization with displacement.
Plotting and output generation			
Output	Figure resolution	300 dpi	Resolution used for saved figures.
Output	Save PDF figures	True	Enables saving figures in PDF format.
Output	Save PNG figures	True	Enables saving figures in PNG format.
Output	Settlement points	60	Number of points used to generate synthetic load–settlement curves.
ANN benchmark			
ANN structure	Epochs	500	Maximum number of training epochs.
ANN structure	Batch size	64	Mini-batch size for ANN training.
ANN structure	Learning rate	8×10^{-4}	Initial Adam learning rate.
ANN structure	Early stopping patience	45	Number of epochs without improvement before stopping.
ANN structure	Activation function	relu	Hidden-layer activation for ANN.
ANN structure	Hidden layers	(512, 256, 128, 64)	Fully connected ANN architecture.
Random Forest benchmark			
RF tuning	Number of random search iterations	20	Number of random hyperparameter combinations evaluated.
RF tuning	Cross-validation folds	5	Number of folds in cross-validation.

Category	Hyperparameter	Value	Description
XGBoost benchmark			
XGBoost tuning	Number of random search iterations	20	Number of random hyperparameter combinations evaluated.
XGBoost tuning	Cross-validation folds	5	Number of folds in cross-validation.

Among the above settings, the most critical parameters for the PINN were the adaptive physics weights, the trainable shaft calibration factor β, and the physics warm-up schedule. The warm-up strategy allowed the model to first learn the dominant data patterns before progressively enforcing geotechnical consistency, which helped stabilize training and reduce premature convergence to poor local minima. This training philosophy is consistent with recent observations that PINNs benefit from careful balancing of data, boundary, and physics residual terms.

3.6.10 PINN Workflow

Figure 3.8 illustrates the overall workflow of the developed PINN framework, including data preprocessing, feature engineering, multi-output prediction, physics-based loss evaluation, optimization, and model validation. The model combines engineered geotechnical features, multi-output neural prediction, and physics-based residual constraints within a unified training framework.

3.7 Model Training and Validation Strategy

3.7.1 Dataset Division

To ensure robust model training and reliable performance evaluation, the dataset was divided into three subsets: training, validation, and testing. The dataset consisted of synthetic and numerical observations which represents a wide range of pile geometries and layered soil conditions.

A three-way data split was adopted as follows:

- Training set: 75% of the data
- Validation set: 15% of the training data
- Testing set: 25% of the total data

The validation set was used for hyperparameter tuning and early stopping, while the testing set was reserved for final performance evaluation on unseen data.

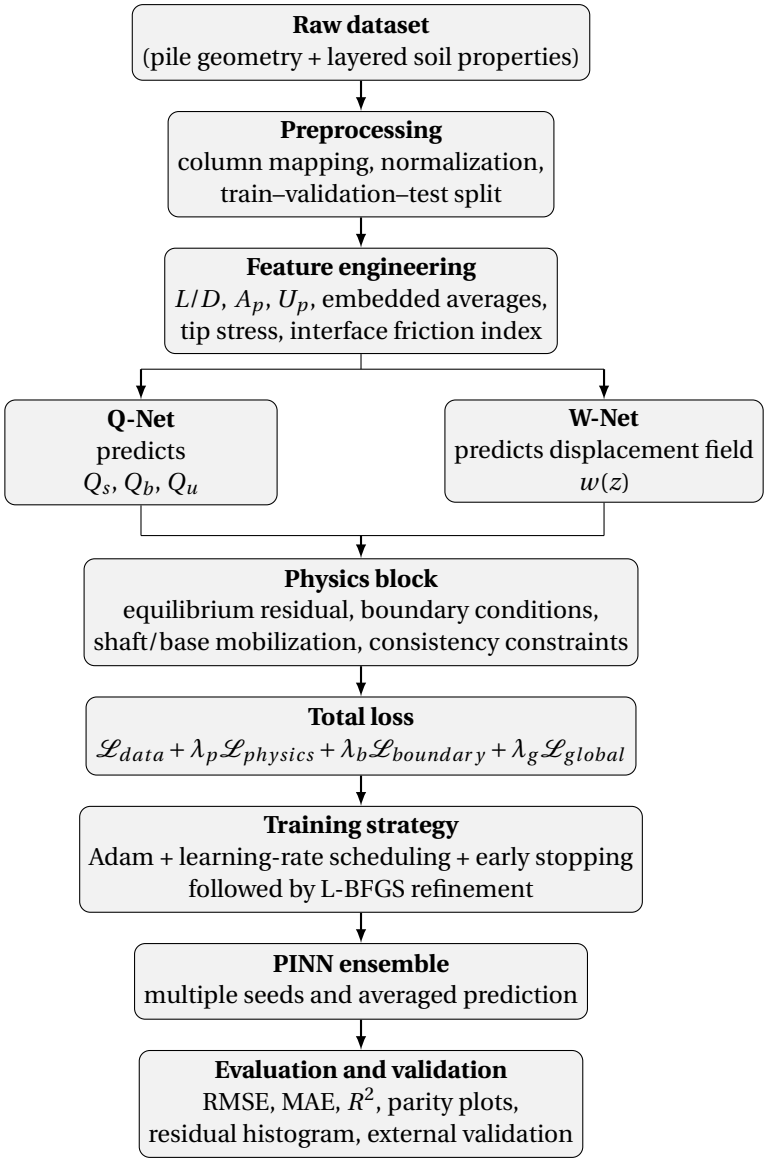

Figure 3.8: Workflow of the developed physics-informed neural network framework for pile axial capacity prediction.

The data split was done in random way to ensure representative sampling of soil stratigraphy, pile dimensions, and material properties across all subsets.

For the Physics-Informed Neural Network (PINN), additional collocation points were generated along the pile depth so that we can enforce the governing differential equation. These collocation points enables the model to satisfy physical laws throughout the domain rather than only at observed data locations.

3.7.2 Data Preprocessing

All input features and target variables were normalized using Min-Max scaling:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3.36}$$

This normalization improves numerical stability and accelerates convergence of neural network training.

3.7.3 Training Procedure

Training procedures for all models are described in full in Section 3.5 (Phase 1) and Section 3.6.9 (Phase 2). Benchmark models (ANN, RF, XGBoost) minimized prediction error with cross-validation hyperparameter tuning. The ensemble prediction averages $M = 3$ PINN members trained on seeds 42, 52, and 62:

$$Q_u^{ensemble} = \frac{1}{M}\sum_{i=1}^{M} Q_u^{(i)} \tag{3.37}$$

3.7.4 Performance Evaluation Metrics

Three complementary metrics were used:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum (y^{pred} - y^{obs})^2} \tag{3.38}$$

$$\text{MAE} = \frac{1}{N}\sum |y^{pred} - y^{obs}| \tag{3.39}$$

$$R^2 = 1 - \frac{\sum (y^{pred} - y^{obs})^2}{\sum (y^{obs} - \bar{y})^2} \tag{3.40}$$

RMSE penalizes large errors; MAE gives average absolute deviation and is more robust to outliers; $R^2 = 1$ indicates perfect fit. All three are reported together for a complete accuracy assessment.

3.7.5 Validation Strategy

Model validation was performed using both statistical and physics-based criteria.

3.7.5.1 Statistical Validation

RMSE, MAE, R^2, and parity plots were used for all models.

3.7.5.2 Physics-Based Validation (PINN)

Additional PINN criteria: equilibrium residual distribution, consistency $Q_u \approx Q_s + Q_b$, and realistic shaft/base mobilization.

3.7.5.3 Load–Settlement Curve Validation

To evaluate the physical consistency of the predicted pile response, synthetic load–settlement curves was generated using formula:

$$P(s) = Q_s \tanh\left(\frac{s}{\eta_s D}\right) + Q_b \tanh\left(\frac{s}{\eta_b D}\right) \tag{3.41}$$

where:

- $P(s)$ = axial load corresponding to pile head settlement s
- Q_s = predicted ultimate shaft resistance
- Q_b = predicted ultimate base resistance
- s = pile head settlement
- D = pile diameter
- η_s = shaft mobilization parameter controlling the rate of shaft resistance mobilization
- η_b = base mobilization parameter controlling the rate of end bearing mobilization

The tanh form is consistent with t–z and q–z load transfer relationships; $\eta_s < \eta_b$ reflects earlier mobilization of shaft resistance. Curves were compared against the seven scaled experimental cases in Chapter 5.

3.7.5.4 External Validation

To evaluate the generalization capability of the developed models, an independent external dataset was used for validation which was not involved in model training or hyperparameter tuning, ensuring an unbiased evaluation of model performance.

The external dataset was generated and compiled from multiple sources, including:

- **Numerical simulations:** Data generated using Python scripting integrated with PLAXIS 3D.
- **Bridge sector data:** Field data obtained from pile foundation systems used in bridge engineering applications.
- **Experimental data:** Laboratory-scale pile load test data.

4 EXPERIMENTAL PROGRAM

4.1 Overview of Experimental Program

The experimental program is done to investigate axial load–settlement behavior of model piles in dry sand under controlled laboratory conditions. Dry sand was chosen because its relative density can be controlled precisely using pluviation technique, allowing uniform and reproducible specimen preparation without time-dependent effects. The workflow followed four stages, as illustrated in Figure 4.1.

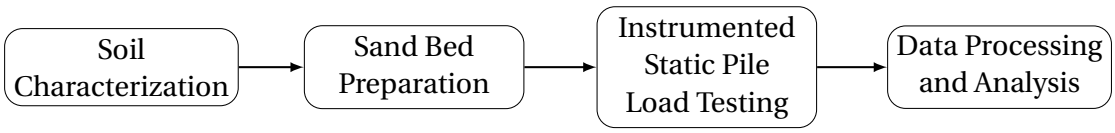

Figure 4.1: Major stages of the experimental program

Initially, detailed soil characterization was done to determine the physical and mechanical properties of the sand. Simultaneously, sand beds were prepared at specified relative densities using a calibrated pluviation technique. Model piles was installed, then static axial loading tests were conducted using an instrumented loading system.

4.2 Soil Characterization

4.2.1 Index Properties

The particle size distribution is shown in Figure 4.2.

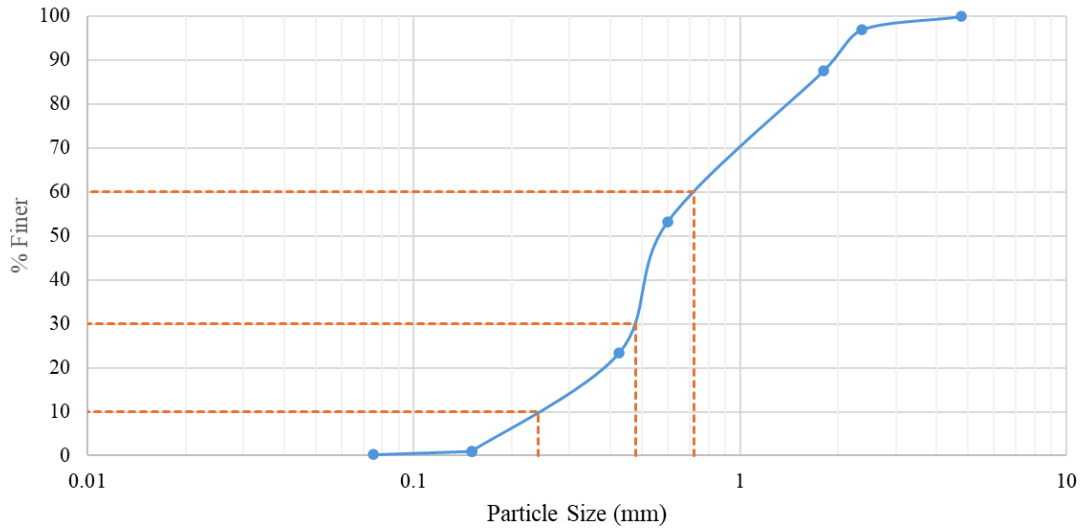

Figure 4.2: Particle size distribution of sand

Key properties:

- Sand fraction: 99.73%

- Fines: 0.27%
- $D_{10} = 0.24$ mm, $D_{30} = 0.48$ mm, $D_{60} = 0.72$ mm
- $C_u = 3.0$, $C_c = 2.78$
- $G_s = 2.62$

4.2.2 Shear Strength Properties

Direct Shear Test:

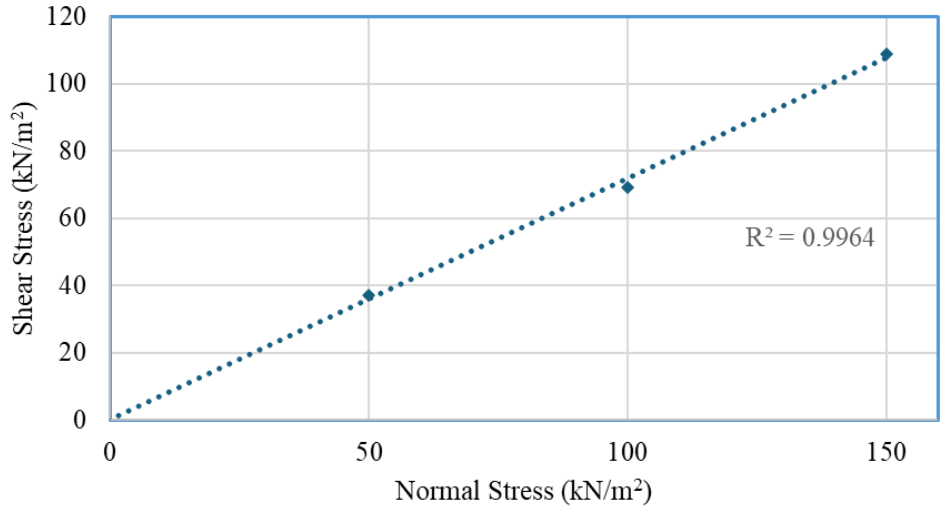

Figure 4.3: Direct shear test results

$\phi = 35.64°$

Triaxial Test:

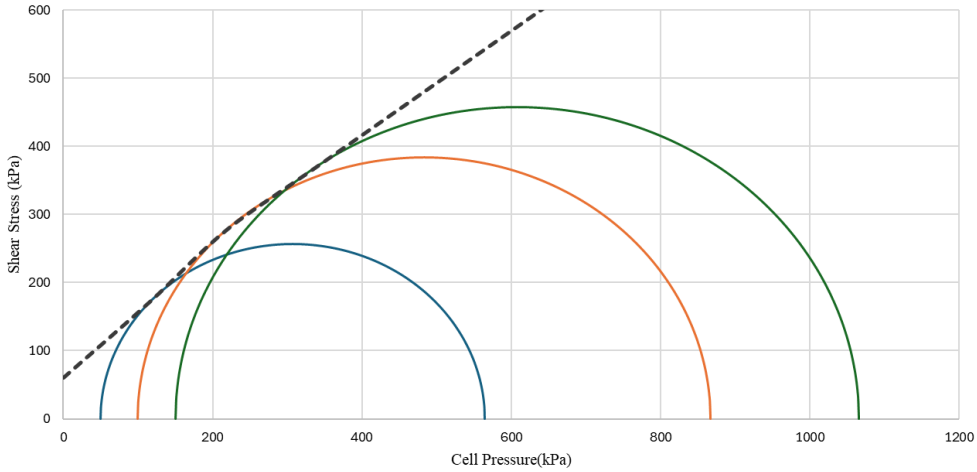

Figure 4.4: Triaxial test results

$\phi = 47.53°$

4.3 Sand Bed Preparation

Sand beds were prepared at relative densities of 40%, 60%, and 80% using the pluviation method.

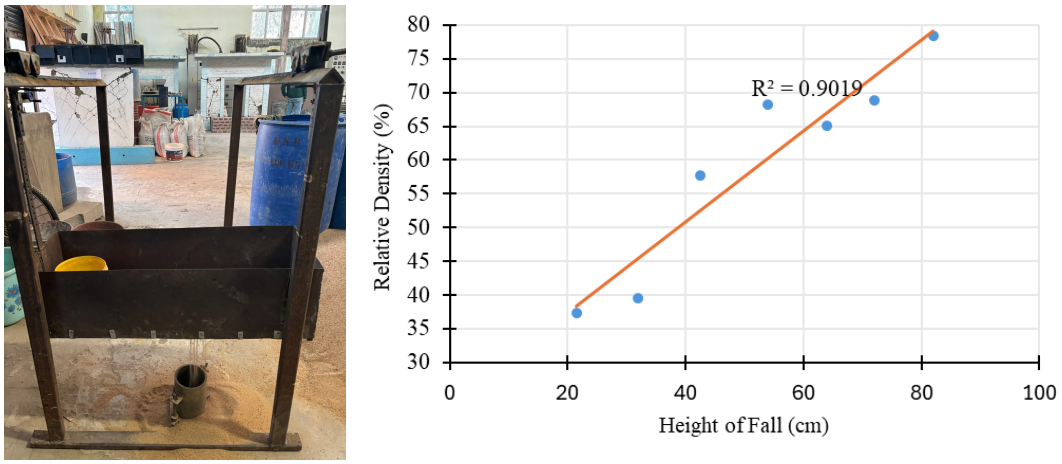

Figure 4.5: Calibration of height of fall for density control

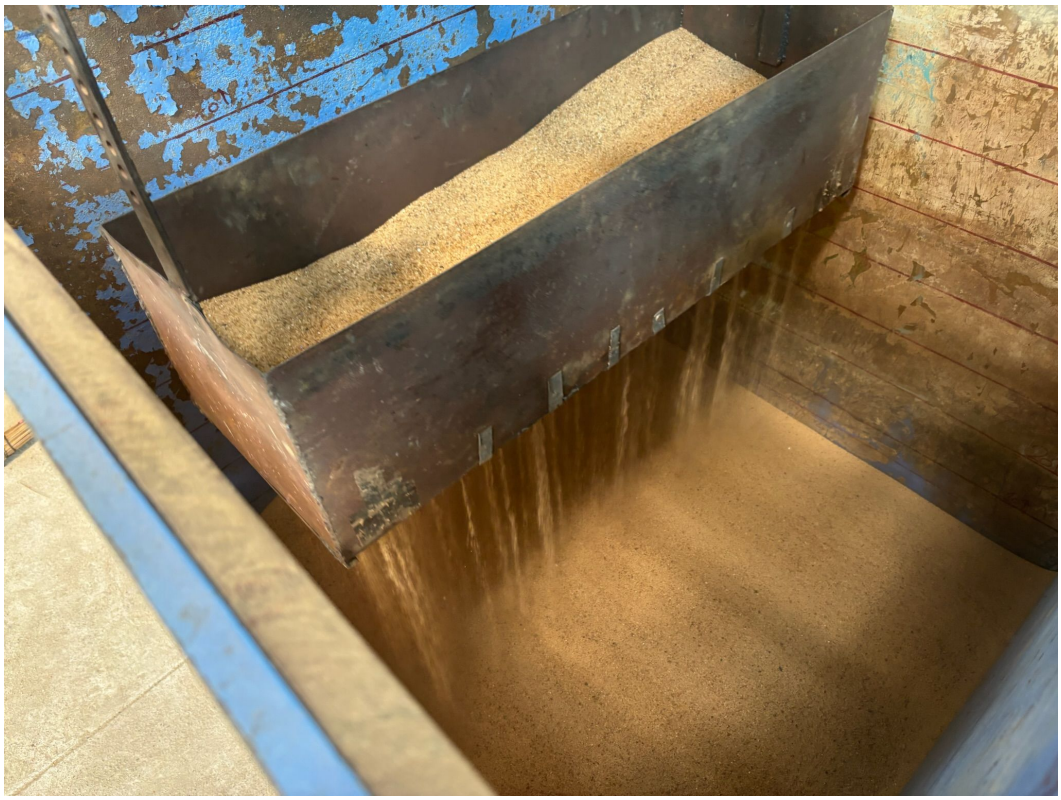

Figure 4.6: Pluviation method for sand deposition

Layered soil profiles was prepared using sequential deposition of different density layers.

4.4 Model Pile and Instrumentation

4.4.1 Model Pile

- Material: Mild steel
- Diameter: 20, 30, 40 mm
- Length: 150, 200, 250 mm

4.4.2 Instrumentation

- Load cell
- LVDT
- Soil pressure sensors
- DAQ system

4.5 Experimental Setup

The experimental setup is shown in Figures 4.7 and 4.8.

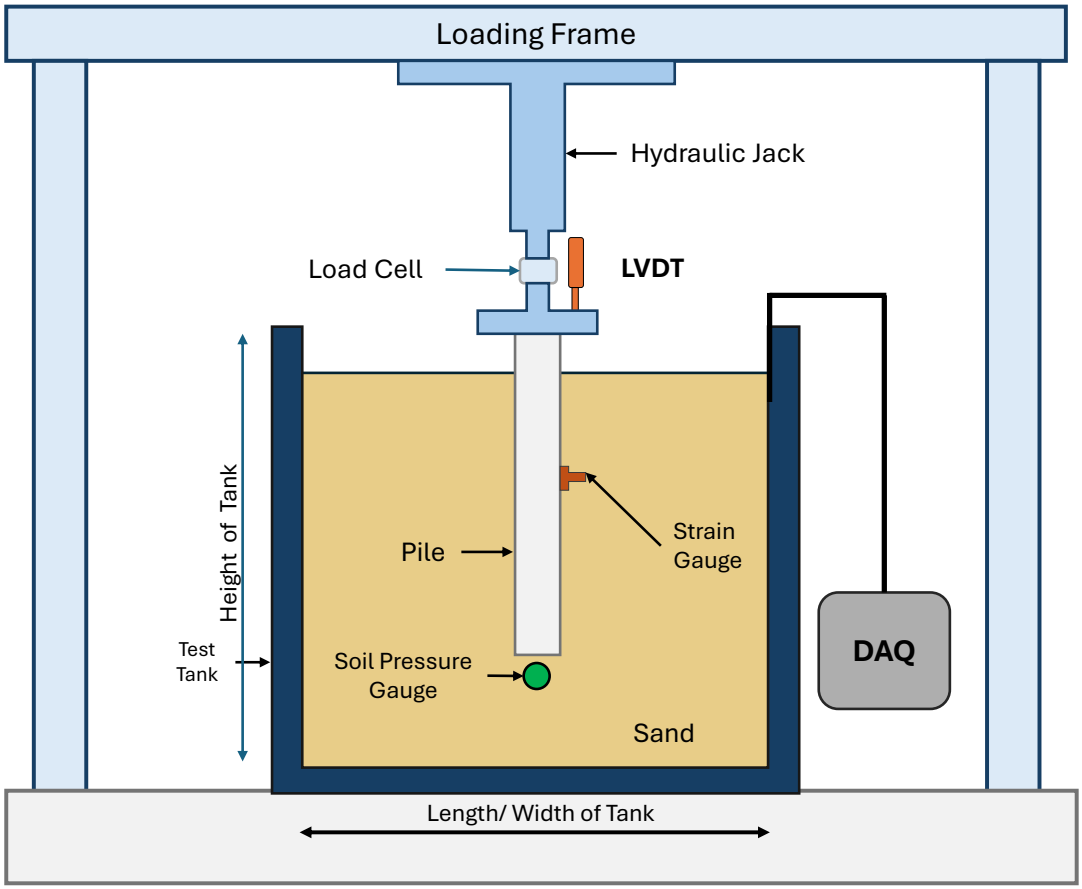

Figure 4.7: Static pile load test setup

Figure 4.8: Laboratory setup at IIT Patna

4.5.1 Scaling and Model Similarity

Table 4.1: Scaling factors adopted for small-scale physical modeling (after Moosavian et al. (2021))

Quantity	**Prototype Scale**	**Model Scale**
Linear dimensions (length, displacement)	L_r	1
Acceleration (gravity)	1	1
Stress	L_r	1
Force	L_r^3	1
Mass	L_r^3	1
Strain	1	1
Moment	L_r^4	1
Void ratio	$e_p = e_m + \lambda \ln(L_r)$	e_m
Factor of safety (FS_v)	1	1

The scaling relationships used in this study are based on dimensional analysis principles for 1g physical modeling, where the geometric scale factor L_r governs the transformation between prototype and model quantities. In these type of models, stresses are reduced due to lower self-weight, so it requires adjustment of soil state

parameters such as void ratio to maintain similitude. As highlighted by Moosavian et al. (2021), soil behavior depends on effective stress and void ratio, and therefore the model soil must be relatively looser than the prototype to compensate for reduced stress levels.

In the present study, a geometric scale factor of $L_r = 50$ is adopted to represent prototype pile foundations of engineering significance.

4.6 Test Matrix

A test matrix was developed so that we can systematically investigate the influence of pile geometry, relative density, and soil layering on the axial load–settlement behavior of piles in sand. The test program was designed to see the effect of individual parameters while maintaining control over other variables.

The details of the experimental program are summarized in Table 4.2.

Table 4.2: Test matrix for laboratory-scale static pile load tests

Test No	Pile Material	Diameter (mm)	Length (mm)	L/D	Relative Density (%)	Sand Condition	Soil Layering
1	Steel	20	150	7.5	40	Uniform loose	Single
2	Steel	20	200	10.0	60	Uniform medium	Single
3	Steel	20	250	12.5	80	Uniform dense	Single
4	Steel	30	150	5.0	60	Uniform medium	Single
5	Steel	30	200	6.7	40 over 80	Layered	Two layers
6	Steel	30	250	8.3	80 over 40	Layered	Two layers
7	Steel	40	200	5.0	60	Uniform medium	Single

4.6.1 Rationale of Test Program

The test matrix was designed to capture the effects of key parameters influencing pile behavior in sand.

- **Effect of Relative Density:** Tests 1, 2, and 3 was designed to investigate the influence of loose, medium dense, and dense sand conditions on pile response.
- **Effect of Pile Length:** Variation in pile length allows assessment of shaft resistance mobilization and its contribution to overall capacity.
- **Effect of Pile Diameter:** Tests with different diameters evaluate scale effects and changes in stress distribution around the pile.
- **Effect of Soil Layering:** Tests 5 and 6 simulate layered soil conditions to study the influence of soil stratification on load transfer behavior.
- **Validation Case:** Test 7 uses a larger diameter ($D = 40$ mm) at medium density to assess the effect of diameter independently of slenderness.

5 RESULTS AND DISCUSSION

This chapter presents results from the experimental investigation, model evaluation, and validation, and is organized to reflect the two-phase development structure. Section 5.1 presents laboratory load–settlement observations. Section 5.2 shows predictive accuracy. Section 5.4 analyses PINN training behaviour and trainable parameter evolution. Section 5.3 compares predictions against the seven scaled experimental cases. Section 5.5 examines generalization on the external dataset.

5.1 Experimental Results

This section presents the key observations from the laboratory-scale static pile load tests conducted on model piles installed in sand beds of varying density and layering conditions. Detailed descriptions of the experimental setup, materials, and procedures have already been presented in Chapter 4; therefore, this section mainly focuses on the interpretation of load–settlement response and the influence of governing parameters.

5.1.1 Load–Settlement Behaviour of Model Piles

The load–settlement curves obtained from the laboratory tests shows a typical nonlinear response characteristic of pile foundations in granular soils. At initial stages of loading, the response is relatively stiff, indicating elastic behavior with limited mobilization of shaft resistance. As the load increases, the response becomes progressively nonlinear, showing gradual mobilization of shaft friction followed by contribution from end bearing.

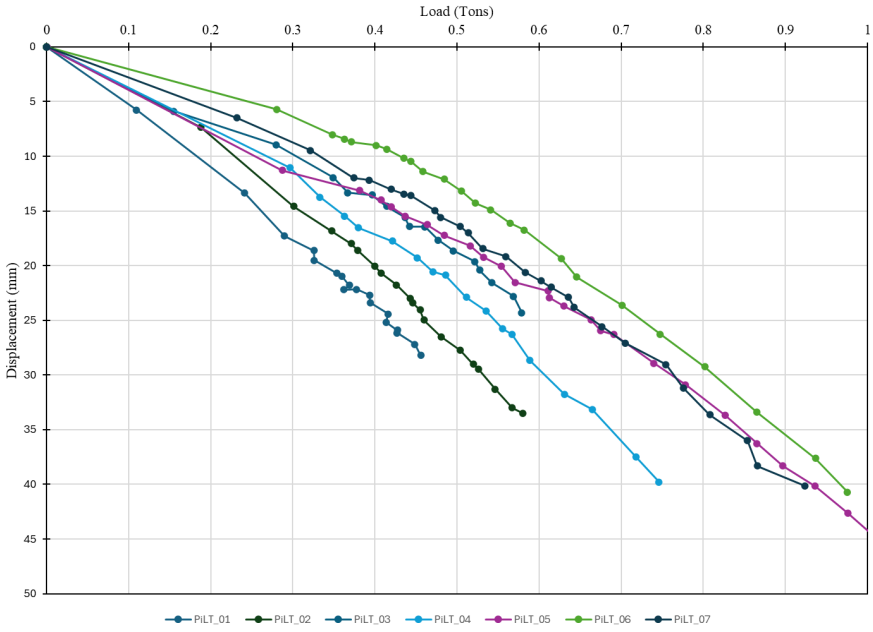

Figure 5.1: Load–settlement curve obtained from laboratory scale static pile load tests.

Table 5.1: Load Settlement data recorded from laboratory scale static pile load tests

Test 1		Test 2		Test 3		Test 4		Test 5		Test 6		Test 7	
S (mm)	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**
0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.76	0.11	7.35	0.19	5.89	0.15	11.05	0.3	11.28	0.29	5.7	0.28	6.47	0.23
13.38	0.24	14.58	0.3	8.95	0.28	13.75	0.33	13.14	0.38	8.01	0.35	9.47	0.32
17.28	0.29	16.82	0.35	11.94	0.35	15.49	0.36	14.01	0.41	8.42	0.36	11.96	0.37
18.63	0.33	17.97	0.37	13.33	0.37	16.53	0.38	14.62	0.42	8.67	0.37	12.18	0.39
19.53	0.33	18.61	0.38	13.54	0.4	17.75	0.42	15.46	0.44	9	0.4	13	0.42
20.7	0.35	20.04	0.4	14.58	0.41	19.27	0.45	16.24	0.46	9.36	0.41	13.45	0.44
20.97	0.36	20.69	0.41	15.58	0.44	20.55	0.47	17.23	0.48	10.18	0.44	13.58	0.44
21.8	0.37	21.77	0.43	16.42	0.44	20.86	0.49	18.19	0.52	10.46	0.44	14.94	0.47
22.19	0.36	22.98	0.44	16.47	0.46	22.88	0.51	19.21	0.53	11.4	0.46	15.58	0.48
22.17	0.38	23.39	0.45	17.67	0.48	24.14	0.54	20.07	0.55	12.09	0.48	16.43	0.5
22.69	0.39	24.04	0.46	18.66	0.5	25.8	0.56	21.53	0.57	13.18	0.51	16.99	0.51
23.39	0.39	24.97	0.46	19.64	0.52	26.27	0.57	22.31	0.61	14.29	0.52	18.44	0.53

Test 1		Test 2		Test 3		Test 4		Test 5		Test 6		Test 7	
S (mm)	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**	**S (mm)**	**P (kN)**
24.41	0.42	26.49	0.48	20.4	0.53	28.66	0.59	22.93	0.61	14.88	0.54	19.15	0.56
25.19	0.41	27.76	0.5	21.57	0.54	31.76	0.63	23.69	0.63	16.1	0.56	20.62	0.58
25.89	0.43	29.03	0.52	22.8	0.57	33.16	0.66	24.95	0.66	16.73	0.58	21.38	0.6
26.15	0.43	29.44	0.53	24.3	0.58	37.48	0.72	25.97	0.67	19.33	0.63	21.98	0.62
27.21	0.45	31.29	0.55	-	-	39.78	0.75	26.31	0.69	21.05	0.65	22.89	0.64
28.2	0.46	33	0.57	-	-	-	-	28.94	0.74	23.6	0.7	23.82	0.64
-	-	33.51	0.58	-	-	-	-	30.89	0.78	26.27	0.75	25.57	0.68
-	-	-	-	-	-	-	-	33.7	0.83	29.23	0.8	27.07	0.71
-	-	-	-	-	-	-	-	36.26	0.87	33.41	0.87	29.03	0.75
-	-	-	-	-	-	-	-	38.3	0.9	37.61	0.94	31.18	0.78
-	-	-	-	-	-	-	-	40.17	0.94	40.71	0.98	33.63	0.81
-	-	-	-	-	-	-	-	42.65	0.98	-	-	36	0.85
-	-	-	-	-	-	-	-	45.28	1.02	-	-	38.32	0.87
-	-	-	-	-	-	-	-	-	-	-	-	40.14	0.92

The experimentally obtained load–settlement response from laboratory testing is shown in Figure 5.1. and data are presented in Table 5.1

All tests shows a smooth transition from initial stiffness to nonlinear deformation without any abrupt failure. This behavior confirms that load transfer in sand occurs progressively through mobilization of frictional resistance along the pile shaft and stress development at the pile base.

5.1.2 Effect of Relative Density on Pile Response

The experimental results clearly shows that relative density has a significant influence on pile response. Tests conducted at different relative densities of 40%, 60%, and 80% indicates a systematic increase in stiffness and load-carrying capacity with increasing density.

In loose sand ($D_r = 40\%$), the load–settlement response is relatively soft, with larger settlements occurring at lower load levels. This behavior is attributed to lower confinement and reduced interparticle contact forces.

While, medium dense ($D_r = 60\%$) and dense sand ($D_r = 80\%$) exhibit significantly higher stiffness. Dense sand shows the steepest load–settlement curve, indicating higher resistance to deformation and greater mobilization of shaft friction and end bearing.

The effect of density is also reflected in reduced settlement for the same applied load and increased ultimate capacity in dense conditions. These observations are consistent with the fundamental behavior of granular soils, while increased density leads to higher interparticle friction and improved stress transfer.

5.1.3 Effect of Pile Geometry (Length and Diameter)

Pile geometry was found to have a significant effect on axial pile behavior. Variations in pile length and diameter influences both the magnitude of load capacity and the shape of the load–settlement curve.

An increase in pile length has resulted in higher load-carrying capacity due to the increased surface area available for mobilization of shaft resistance. Longer piles has shown enhanced frictional contribution, particularly in medium and dense sand conditions.

Similarly, an increase in pile diameter increased in both shaft resistance and end bearing. Larger diameter piles showed greater load capacity, especially at higher settlement levels where base resistance was dominant.

The variation in length-to-diameter (L/D) ratio influenced the relative contribution

of shaft and base resistance. Higher L/D ratios contributed shaft resistance, while lower L/D ratios showed relatively stronger influence of end bearing.

These results confirm that pile geometry plays a critical role in governing load transfer mechanisms and must be explicitly incorporated in predictive models.

5.1.4 Effect of Soil Layering on Pile Behaviour

The influence of soil layering was observed through tests involving loose-over-dense and dense-over-loose sand profiles.

In the loose-over-dense condition, the initial response is relatively soft due to the weaker upper layer. However, as the pile penetrates deeper and the dense layer becomes engaged, the stiffness had increased significantly. This results in improved load resistance at higher settlements.

While, the dense-over-loose condition exhibits higher initial stiffness due to the stronger upper layer, but shows increased settlement at higher load levels because of the weaker underlying layer. This indicates that the lower layer plays a critical role in controlling the ultimate behavior of the pile.

All these results highlight that pile response is strongly influenced by the sequence of soil layers, and not merely by average soil properties. The experimental observations emphasize the importance of incorporating soil stratigraphy explicitly in predictive modeling frameworks such as PINN.

5.2 Model Performance Evaluation

5.2.1 Comparison of Prediction Accuracy

All models was trained on the same dataset and evaluated on an independent test set. Quantitative results are in Table 5.2; parity plots are in Figures 5.2–5.5. The PINN achieves the best accuracy while maintaining physical consistency in comparison to ANN is competitive statistically but lacks interpretability; tree-based models show higher variability and cannot enforce equilibrium (Al Atroush, 2025).

5.2.2 Parity Plot Analysis

Parity plots is used to visually evaluate the agreement between predicted and observed pile capacities for different models. A perfect prediction corresponds to data points on the 45-degree reference line.

Figure 5.2 shows the parity plot for the PINN model. The data points are tightly clustered around the reference line, which indicates excellent agreement with observed values and minimal prediction error. The distribution of points also suggests

consistent performance across both low and high capacity ranges.

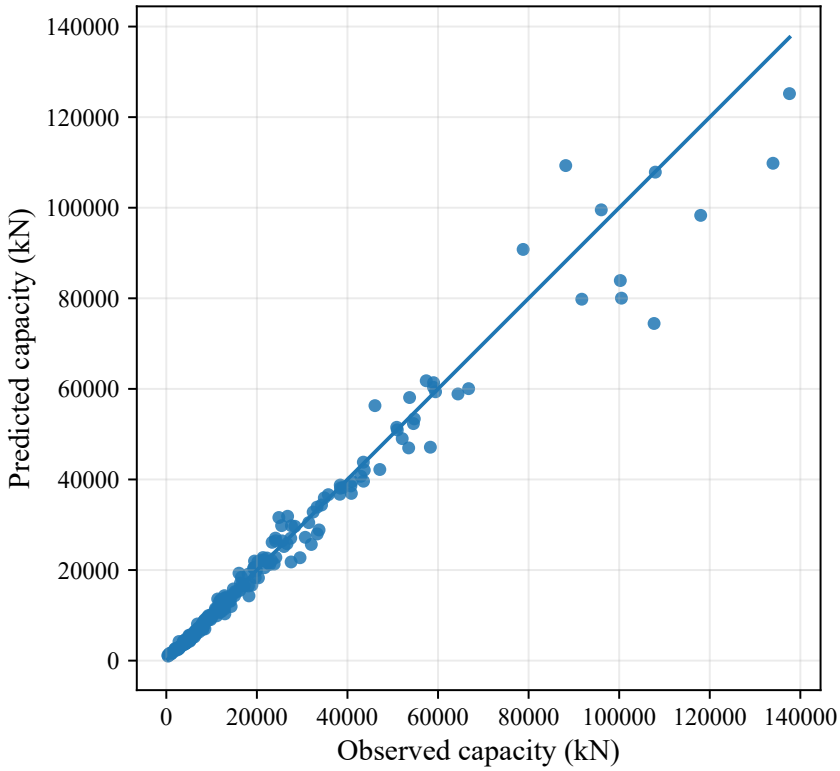

Figure 5.2: Parity plot for the PINN model

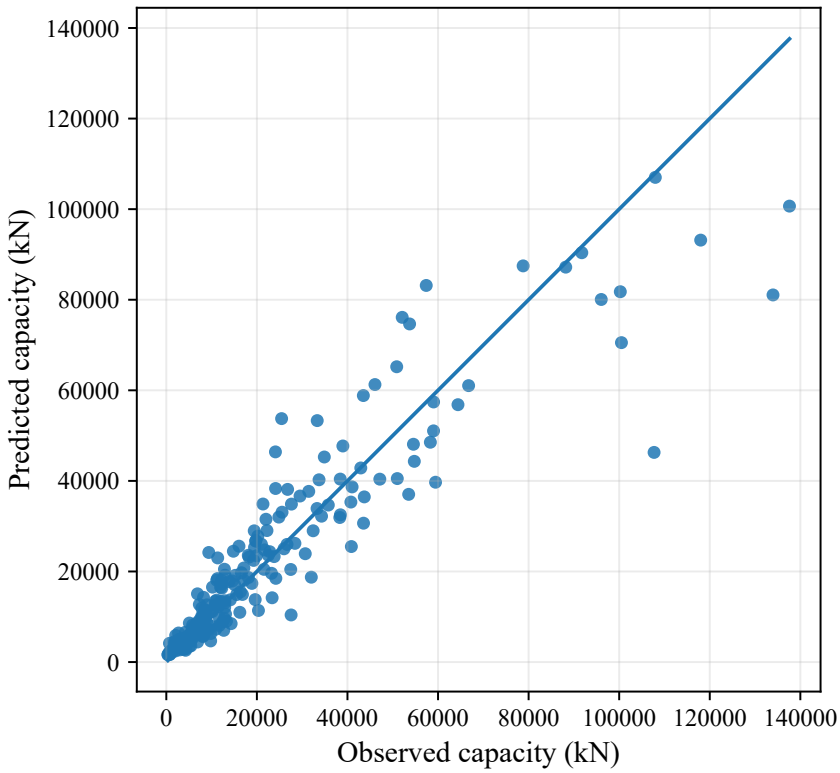

Figure 5.3: Parity plot for the ANN model

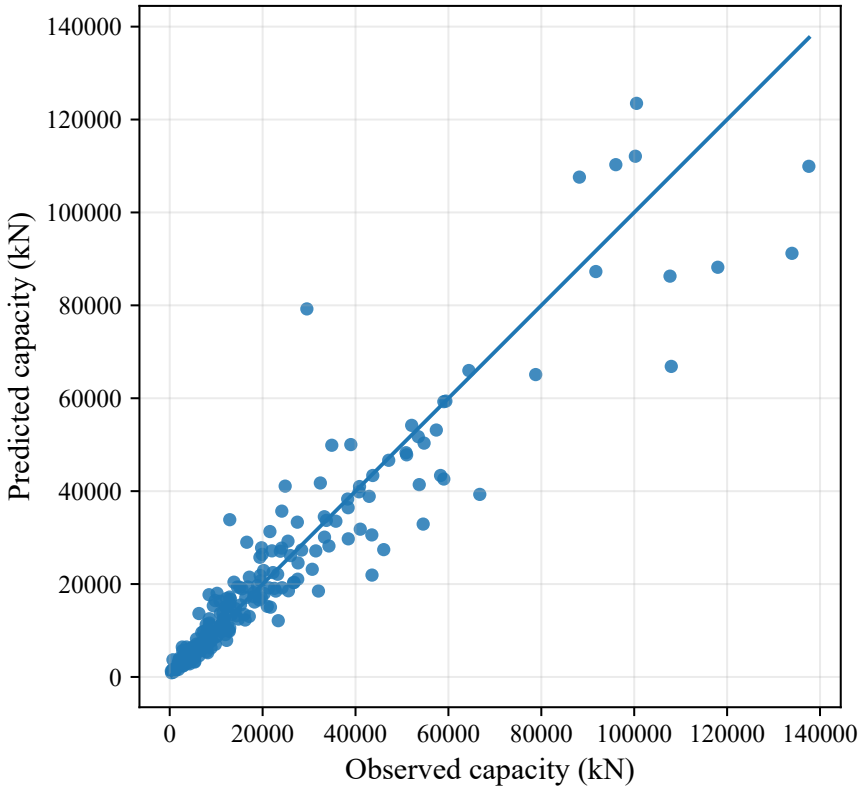

Figure 5.4: Parity plot for the Random Forest model

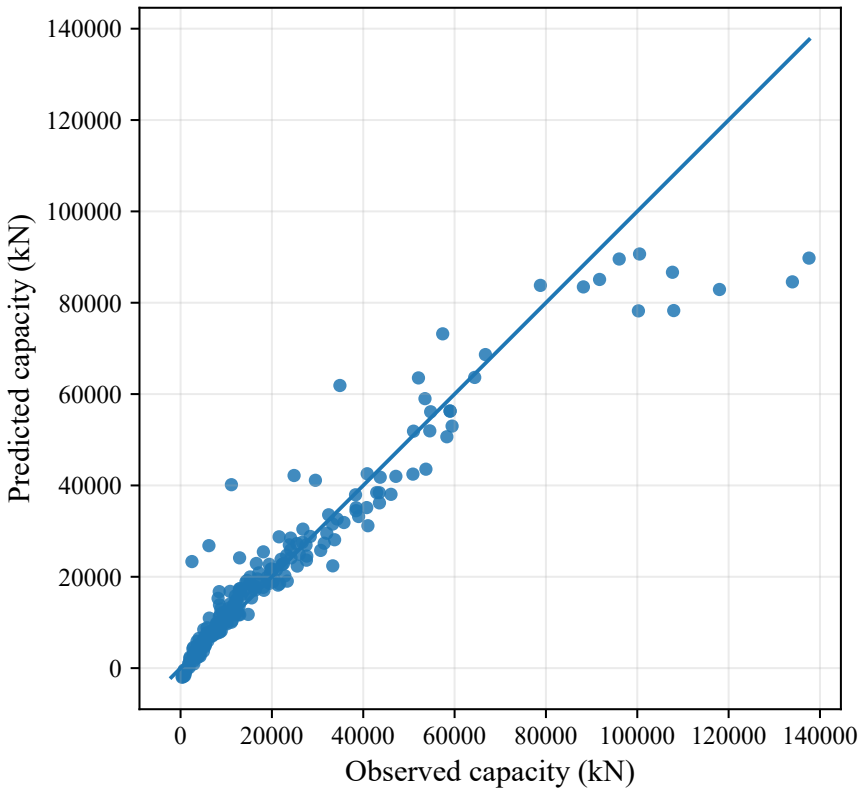

Figure 5.5: Parity plot for the XGBoost model

The ANN model (Figure 5.3) shows greater dispersion, mainly at higher capacity values. This indicates reduced consistency and potential overfitting to certain regions of the dataset.

The Random Forest model (Figure 5.4) shows noticeable scatter across the entire range, which reflects limited capability in capturing complex nonlinear interactions inherent in pile–soil behavior.

The XGBoost model (Figure 5.5) performs better than Random Forest, with reduced scatter; however, its predictions remain less consistent compared to the PINN model.

These graphical comparisons clearly indicate that the PINN model provides the most reliable and consistent predictions among all models considered.

5.2.3 Comparison of Error Metrics (RMSE, MAE, and R^2)

The quantitative performance of the models was observed using RMSE, MAE, and the coefficient of determination (R^2), which has provided a comprehensive measure of prediction accuracy and reliability.

Table 5.2: Comparison of model performance based on statistical error metrics

Model	RMSE (kN)		MAE (kN)		R^2	
	Train	Test	Train	Test	Train	Test
PINN Ensemble	4518.27	2988.55	1923.82	1165.89	0.966	0.977
XGBoost	7679.55	4065.43	3771.69	1646.41	0.902	0.958
Random Forest	8451.51	5087.78	4510.35	2307.84	0.881	0.934
ANN	9411.61	6220.85	5381.68	3336.39	0.853	0.901

The PINN ensemble achieves the lowest test RMSE (2989 kN) and highest R^2 (0.977), followed by XGBoost ($R^2 = 0.958$), Random Forest ($R^2 = 0.934$), and ANN ($R^2 = 0.901$). Physics-based constraints has improved both accuracy and physical reliability, in compared to recent literature of higher capacities and complex soil profiles (Al Atroush, 2025).

5.3 Experimental Validation: Load–Settlement Curves

The seven laboratory pile load tests described in Chapter 4 provide the only source of physical ground-truth data in this study. After scaling to prototype dimensions using $L_r = 50$, PINN ensemble predictions and those of ANN, Random Forest, and XGBoost were compared against the observed load–settlement response. The observed data were smoothed using the hyperbolic Chin–Kondner interpretation method (Chin, 1970) to produce a monotonically increasing reference curve. Mobilisation parameters

η_s and η_b were calibrated from observed data so the comparison reflects differences in predicted capacity rather than mobilisation form.

Across all seven tests, the PINN ensemble has provided the closest agreement with observed behaviour while other machine learning model are not uniformly consistent. The dataset is limited to seven tests in dry sand under 1g conditions; capacity values are prototype-scale estimates derived through geometric scaling, so the comparison is indicative rather than definitive as it is also strongly supportive by Parity Plot and Error Metrics (RMSE, MAE and R^2).

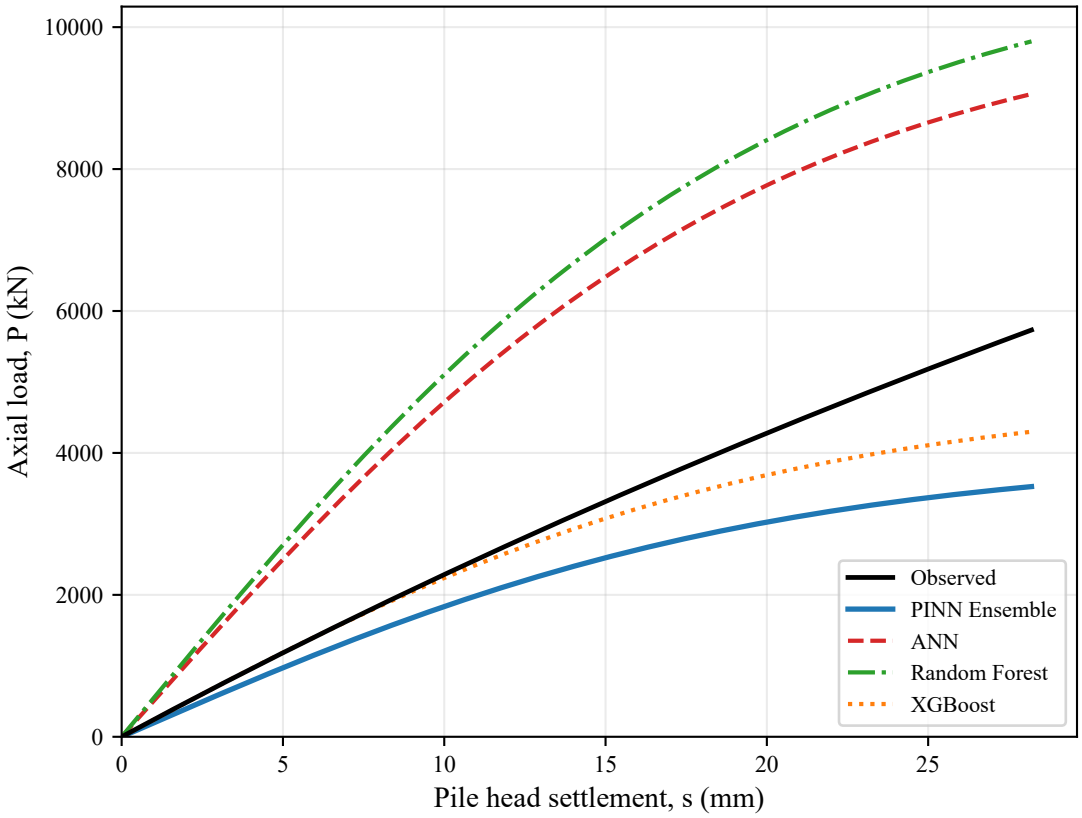

Figure 5.6: Test 1: $D = 1.00$ m, $L = 7.50$ m, uniform loose sand ($D_r = 40\%$).

5.4 Physics-Informed Neural Network Behaviour

This section presents a analysis of the internal behaviour of the Physics-Informed Neural Network (PINN) during training and validation. Unlike conventional machine learning models, the PINN framework has integrated governing physical equations into the learning process. Therefore, evaluation of the PINN model requires not only assessment of prediction accuracy but also analysis of convergence behaviour, physics satisfaction, and evolution of trainable physical parameters.

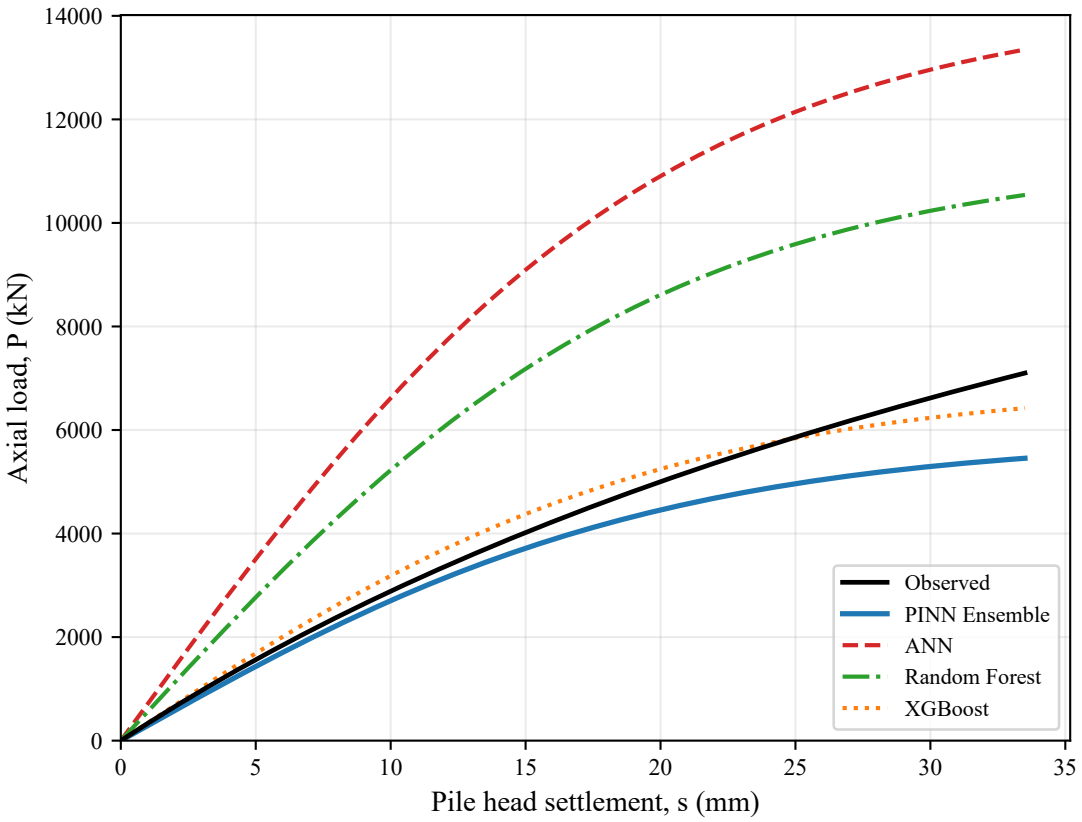

Figure 5.7: Test 2: $D = 1.00$ m, $L = 10.00$ m, uniform medium dense ($D_r = 60\%$).

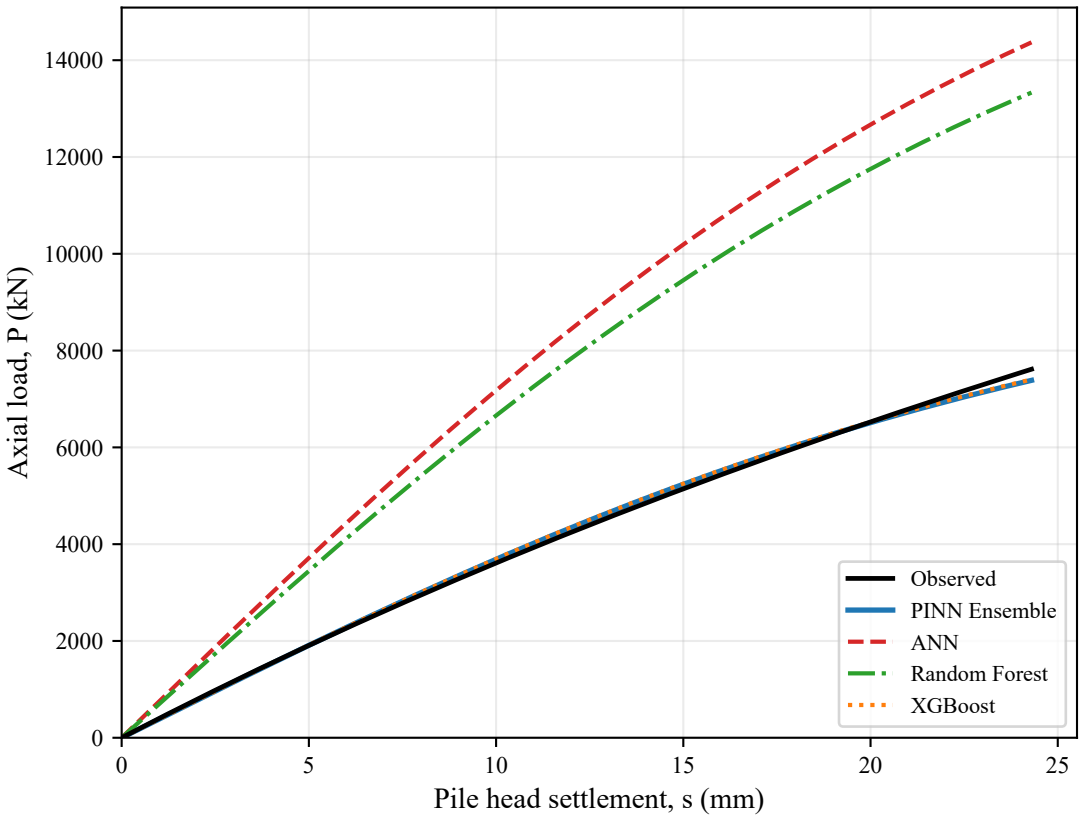

Figure 5.8: Test 3: $D = 1.00$ m, $L = 12.50$ m, uniform dense ($D_r = 80\%$).

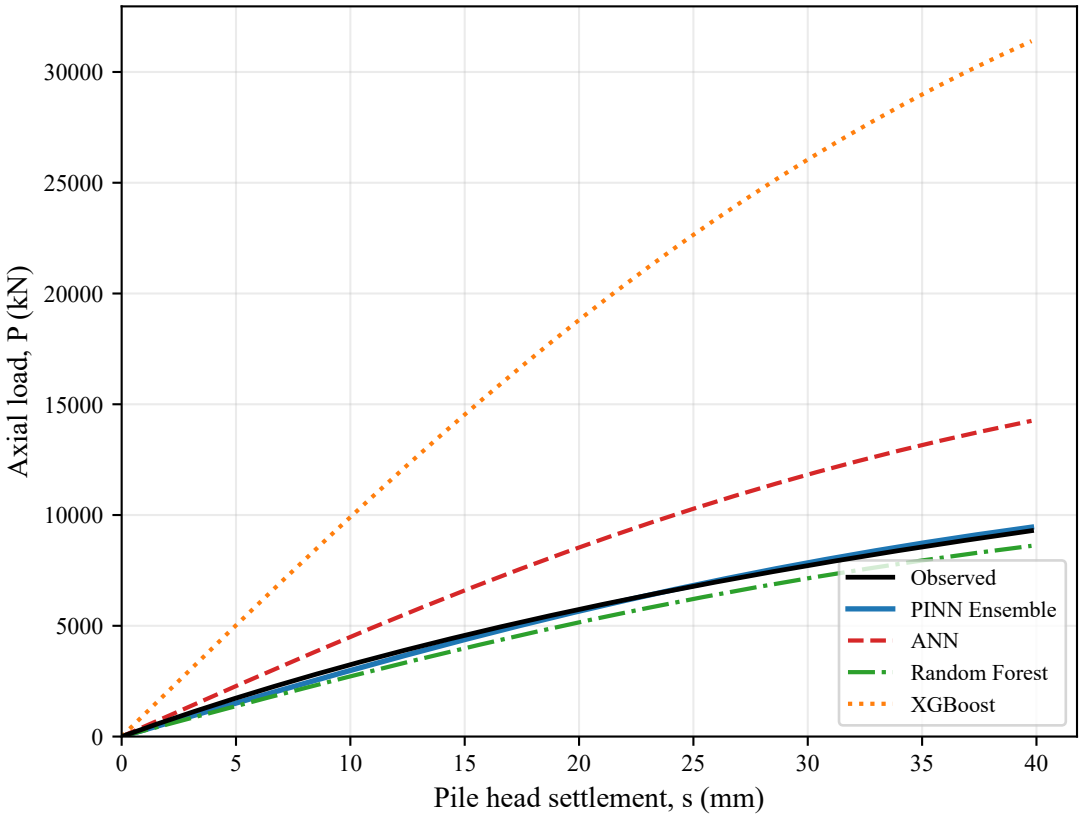

Figure 5.9: Test 4: $D = 1.50$ m, $L = 7.50$ m, uniform medium dense ($D_r = 60\%$).

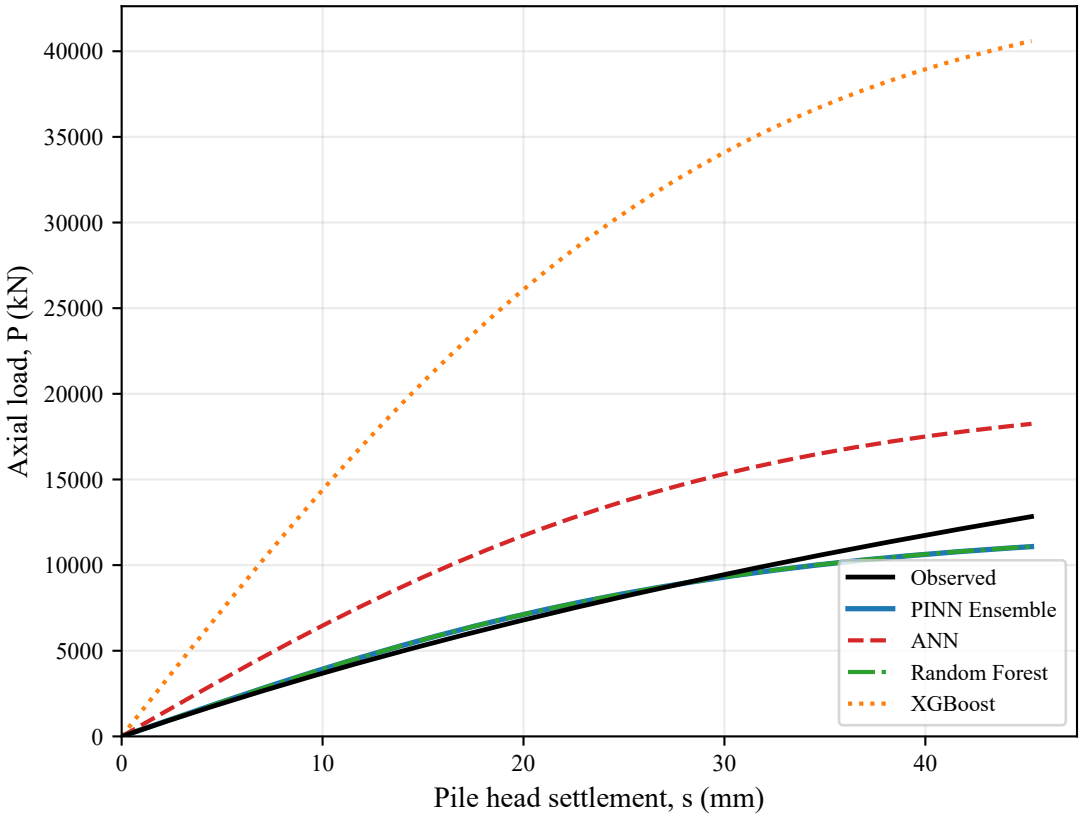

Figure 5.10: Test 5: $D = 1.50$ m, $L = 10.00$ m, loose over dense ($D_r = 40\%$ over 80%).

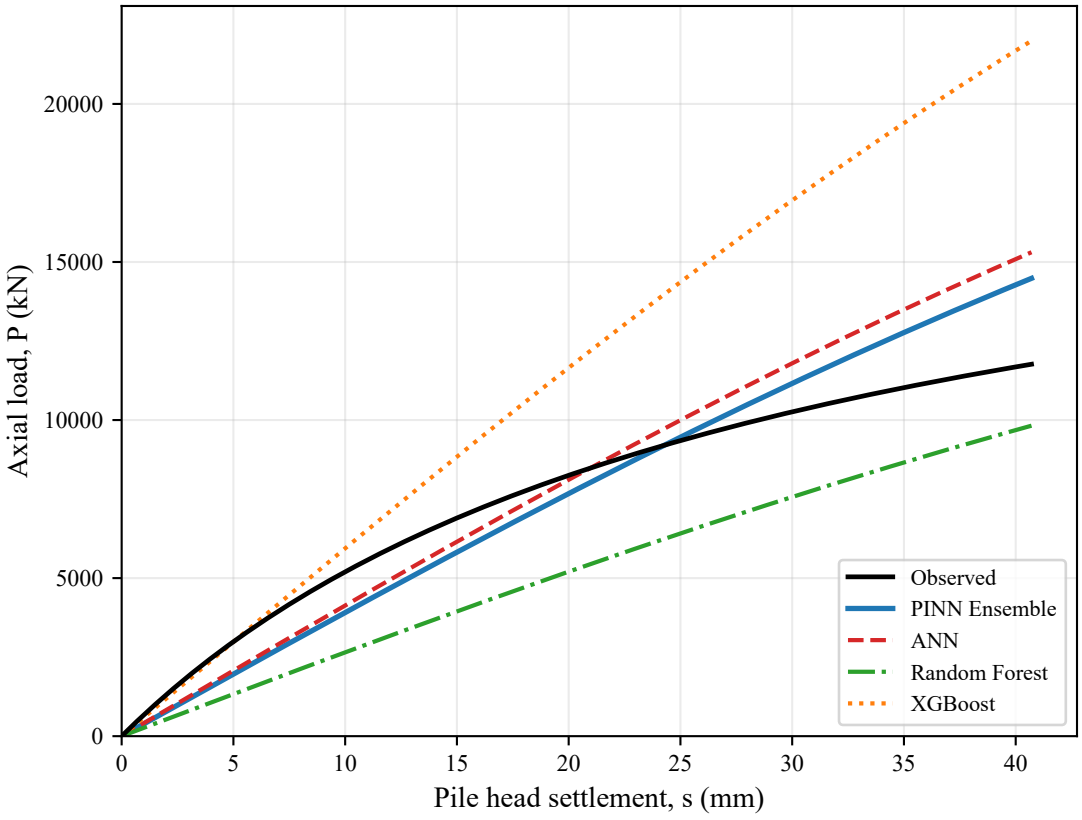

Figure 5.11: Test 6: $D = 1.50$ m, $L = 12.50$ m, dense over loose ($D_r = 80\%$ over 40%).

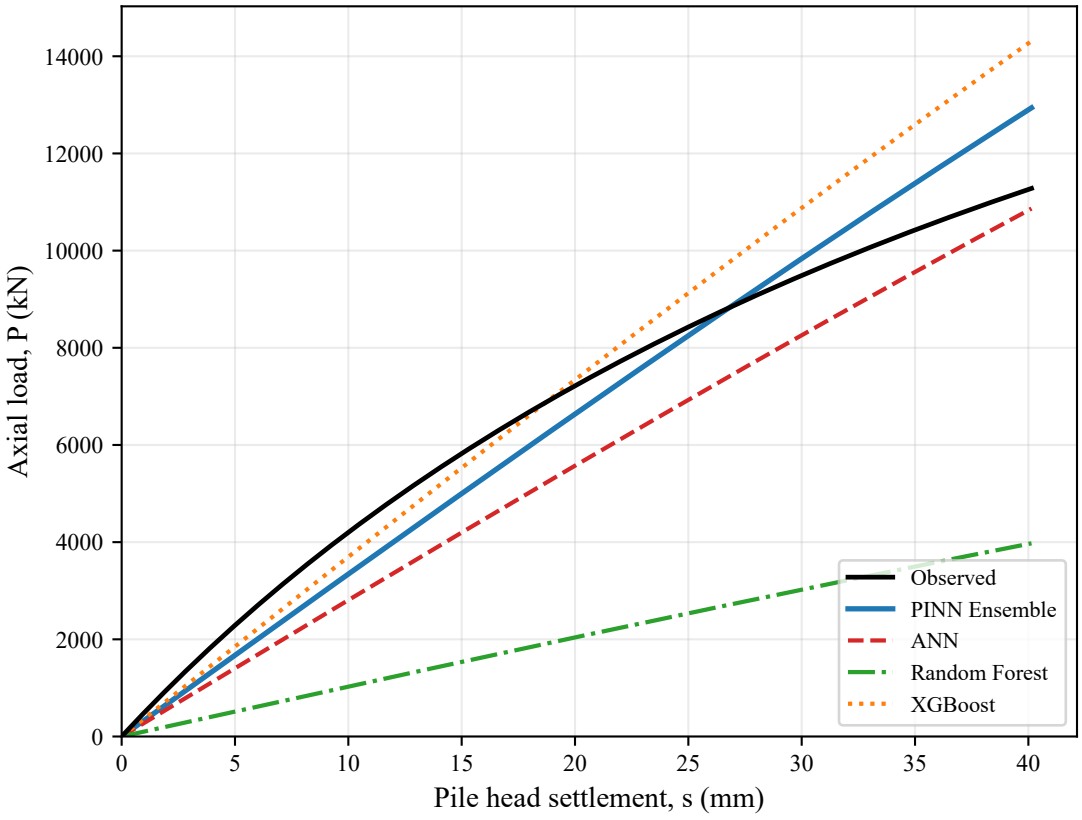

Figure 5.12: Test 7: $D = 2.00$ m, $L = 10.00$ m, uniform medium dense ($D_r = 60\%$).

5.4.1 Training Convergence and Loss Evolution

The training process of the PINN model involves minimization of a composite loss function consisting of data loss, physics residual loss, boundary condition loss, and consistency constraints. The evolution of these loss components during training is as shown in Figure 5.13.

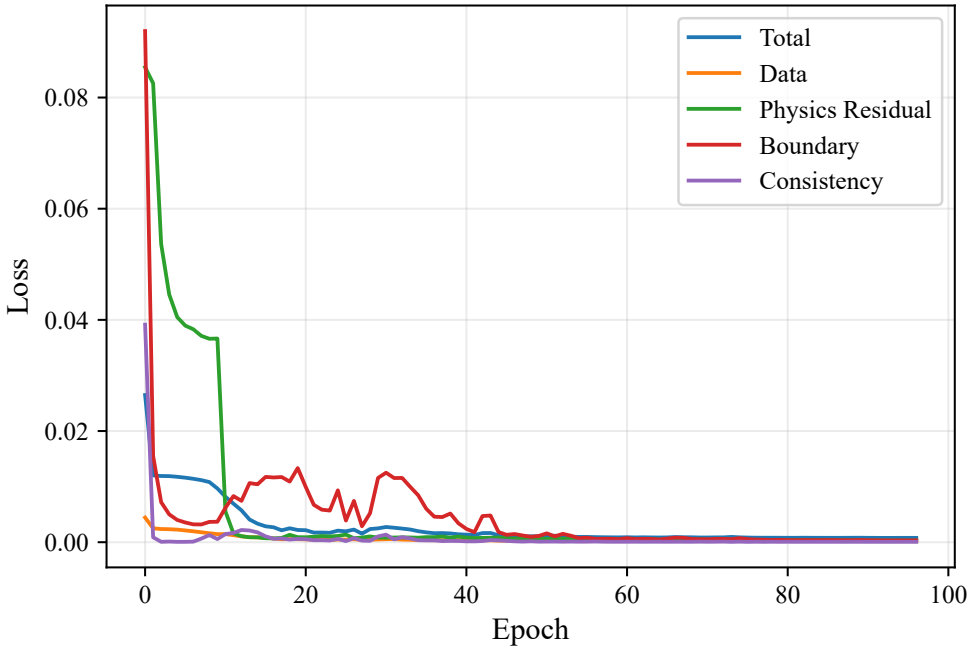

Figure 5.13: Evolution of PINN loss components during training.

It can be observed that the total loss decreases steadily along with training epochs, which indicates stable convergence of the optimization process. The data loss component reduces rapidly during the initial epochs, showing effective learning of input–output relationships.

The physics residual and boundary condition losses decrease more gradually, which is expected since these components enforce compliance with governing equations and physical constraints. The gradual reduction of physics-related losses shows that the network progressively learns to satisfy equilibrium conditions along the pile length.

The use of a physics warm up strategy has played a critical role in stabilizing training. Initially, the model prioritizes data fitting, and then the influence of physics-based constraints increases. This prevents premature domination of physics loss, which could otherwise hinder convergence.

5.4.2 Validation Error Evolution

The evolution of validation error during training is shown in Figure 5.14.

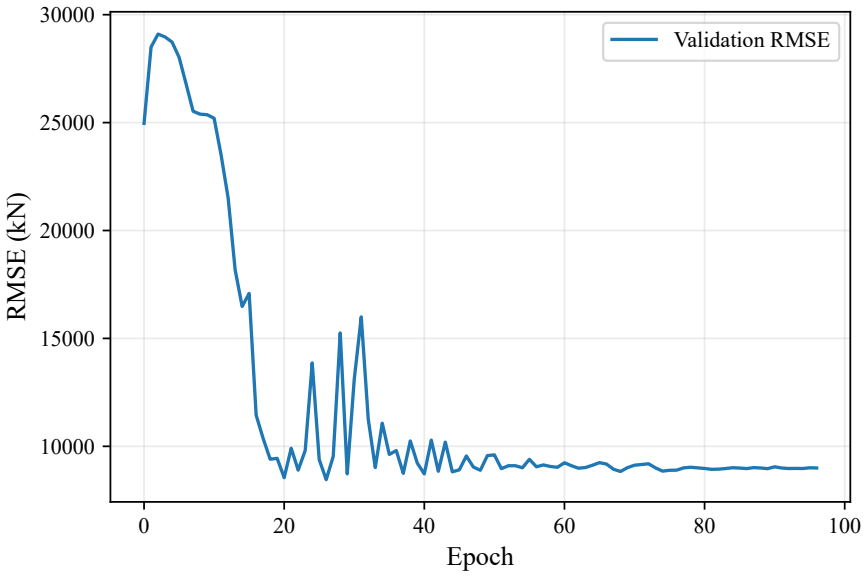

Figure 5.14: Validation RMSE evolution during PINN training.

The validation RMSE decreases consistently during the initial training phase, indicating improved generalization performance. As training progresses, the validation error stabilizes, which suggest that the model has reached an optimal balance between underfitting and overfitting.

The absence of significant oscillations in validation error shows the stability of the training process.

Also, the refinement stage using the L-BFGS optimizer has contributed to fine-tuning the model parameters, which leads to improved convergence and reduced prediction error.

5.4.3 Residual Distribution and Equilibrium Satisfaction

The satisfaction of the governing equilibrium equation is a key requirement of the PINN framework. The distribution of the residual of the governing equation is shown in Figure 5.15.

The residual values are concentrated around zero, which indicates that the governing equation is satisfied across the pile length. The narrow distribution of residuals demonstrates that the PINN model has successfully enforces equilibrium conditions during training.

This behaviour has shown that PINN model in compared to conventional machine learning approaches, which has not enforce physical laws and produce predictions that may violate equilibrium conditions.

The residual analysis confirms that the PINN model provides physically consistent

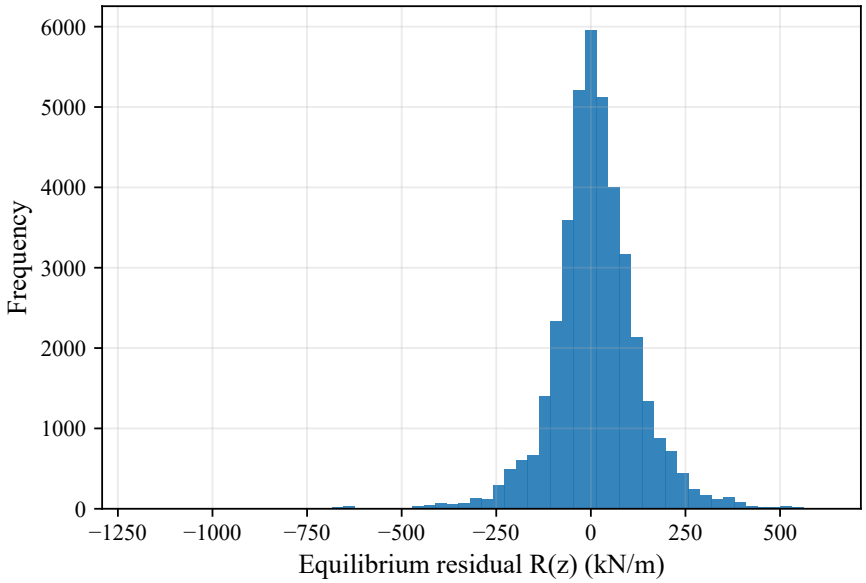

Figure 5.15: Distribution of equilibrium residuals for the PINN model.

predictions, ensuring that the computed load transfer mechanism is aligned with fundamental geotechnical principles.

5.4.4 Interpretation of Trainable Parameters (β and λ)

A key feature of the developed PINN framework is the introduction of trainable parameters, including the shaft resistance scaling factor (β) and the weights associated with different loss components (λ).

The evolution of the parameter β during training is shown in Figure 5.16.

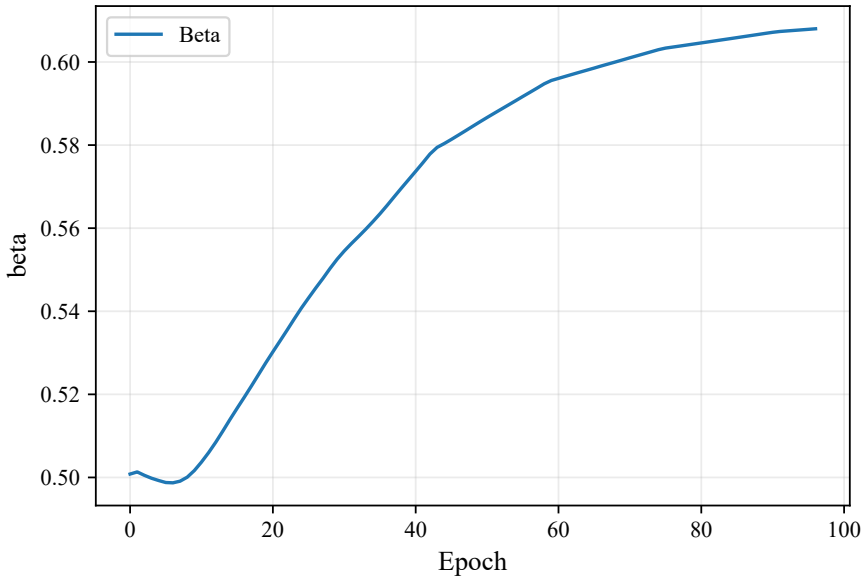

Figure 5.16: Evolution of trainable shaft resistance parameter β during training.

The parameter β has converged to a stable value, indicating that the model is able to calibrate shaft resistance in a data-driven model in a physically consistent manner. This shows the ability of the PINN to adapt classical geotechnical formulations to match observed behaviour.

Similarly, the evolution of trainable loss weights (λ) is presented in Figure 5.17.

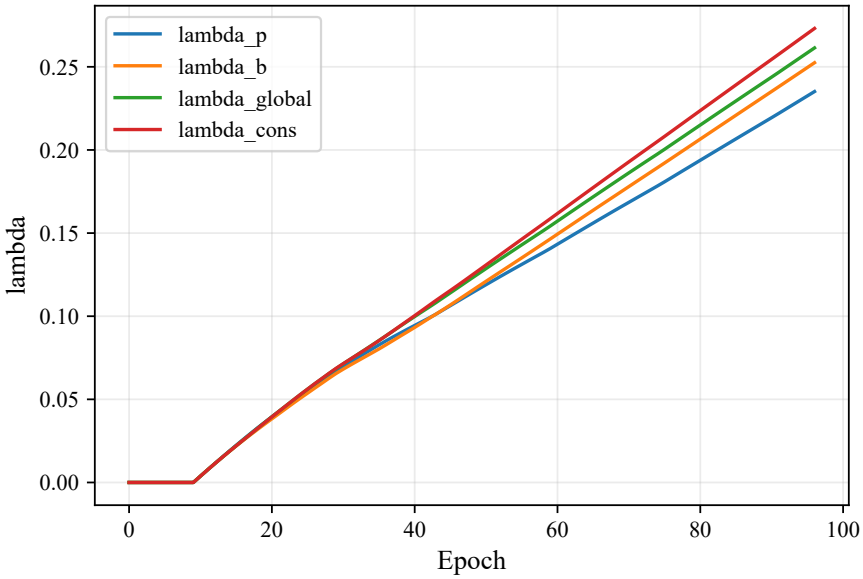

Figure 5.17: Evolution of trainable loss weights during training.

The adaptive adjustment of these weights allows the model to dynamically balance the contribution of data loss and physics-based constraints. This dynamic weighting mechanism shows that neither data fitting nor physics enforcement dominates excessively during training.

Such behaviour is consistent with recent advancements in physics-informed learning (Al Atroush, 2025), where adaptive weighting strategies have been shown to improve model stability and performance.

Overall, the convergence of β and λ shows that the PINN model not only learns predictive relationships but also calibrates physically meaningful parameters, enhancing both interpretability and reliability.

5.5 External Validation of Models

In order to evaluate the prediction capability of the developed models, an independent external dataset was used for validation. This dataset was not included during training and consists of data generated from PLAXIS 3D simulations, laboratory experiments, and field measurements obtained from bridge foundation projects.

5.5.1 External Parity Plot Analysis

Parity plots are used to visually assess the agreement between predicted and observed pile capacities for the external dataset.

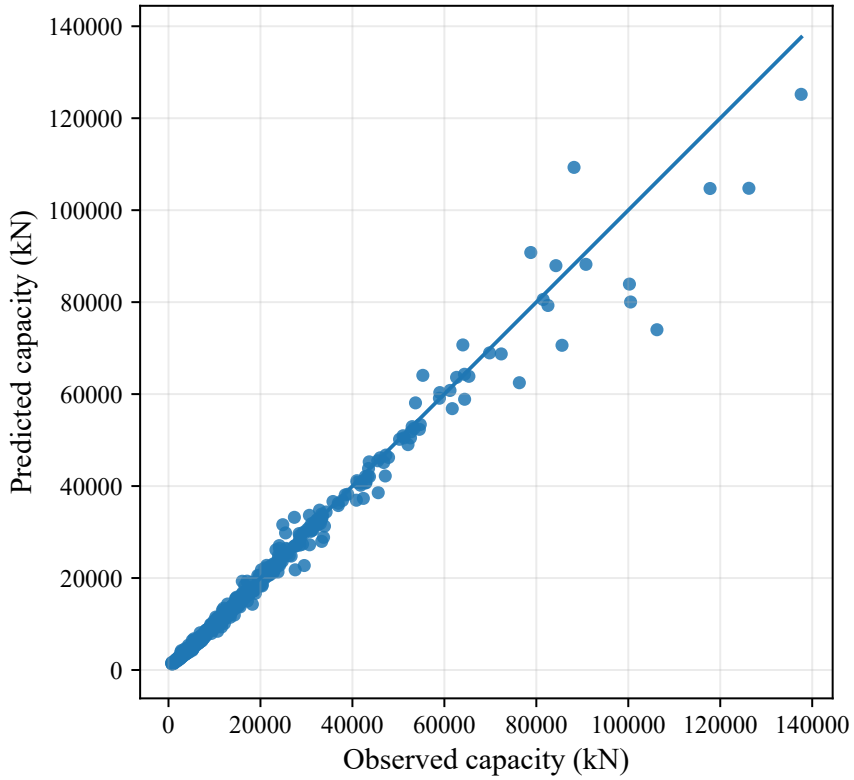

Figure 5.18: External parity plot for the PINN model

Figure 5.18 shows the parity plot for the PINN model. The data points are closely aligned along the 45-degree reference line, indicating strong agreement with observed values and consistent predictive performance.

ANN model (Figure 5.19) shows increased scatter and deviation from the reference line, particularly at higher capacity values, indicating reduced generalization under unseen conditions.

The Random Forest model (Figure 5.20) shows noticeable dispersion across the entire range of predictions, reflecting its limited ability to generalize complex soil–pile interactions beyond the training dataset.

The XGBoost model (Figure 5.21) shows improved alignment compared to Random Forest but still exhibits greater scatter compared to the PINN model, particularly for higher capacity values.

These graphical comparisons clearly shows that the PINN model provides the most consistent and reliable predictions across the external dataset.

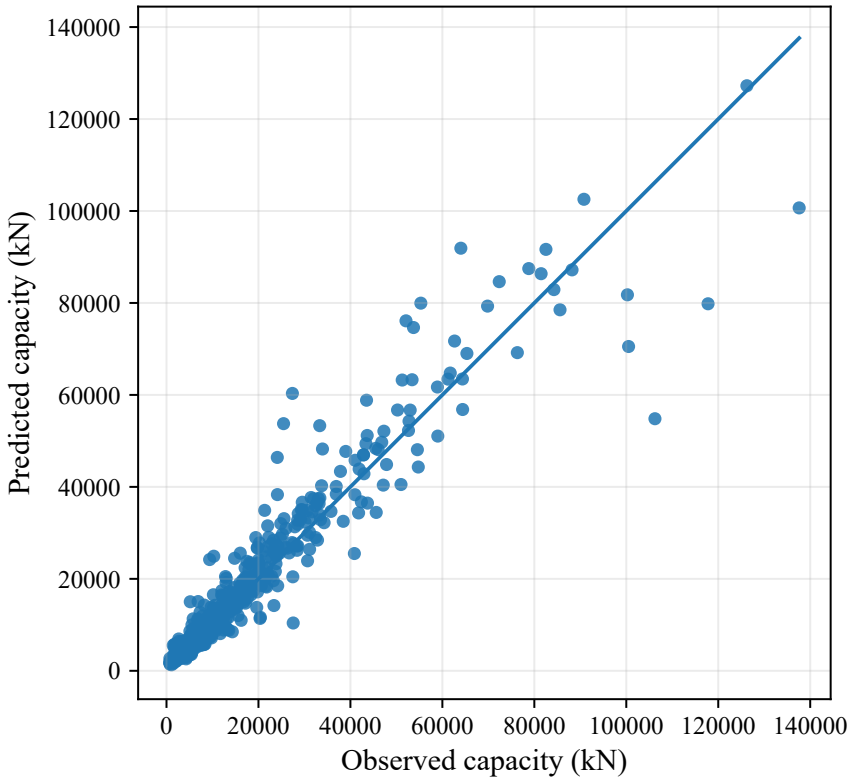

Figure 5.19: External parity plot for the ANN model

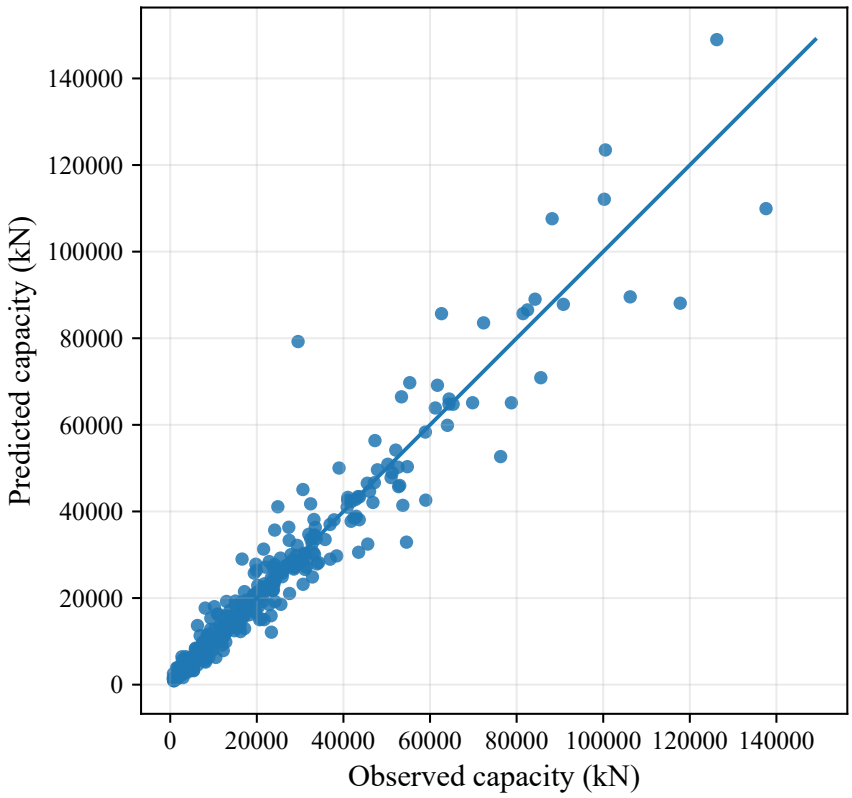

Figure 5.20: External parity plot for the Random Forest model

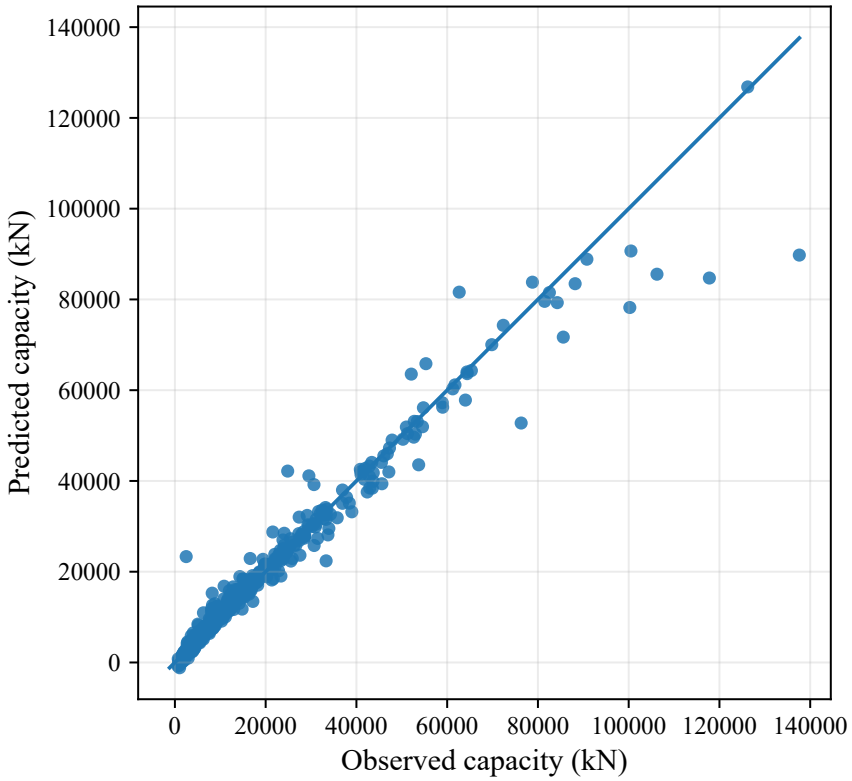

Figure 5.21: External parity plot for the XGBoost model

6 CONCLUSION

This study has developed and validated a Physics-Informed Neural Network framework for predicting axial pile load capacity and load–settlement behaviour in layered soil conditions. The research used a two-phase approach, firstly single-output PINN benchmarked against Random Forest (Phase 1) and second is enhanced multi-output framework with ten-layer soil stratigraphy, trainable physics parameters, and ensemble learning (Phase 2). For experimental validation, seven laboratory pile load tests at IIT Patna, it was scaled to prototype dimensions using $L_r = 50$.

Following conclusions are drawn from this study:

1. The preliminary Phase 1 PINN achieved $R^2 = 0.940$ on internal validation and $R^2 = 0.938$ on external validation, defeating Random Forest ($R^2 = 0.919$), and confirmed the feasibility of embedding geotechnical physics into a neural network loss function. Three limitations were identified initially: single-output prediction, simplified soil representation, and fixed physics loss weights.
2. The Phase 2 enhanced PINN has successfully got $R^2 = 0.977$ on the test set, defeating ANN ($R^2 = 0.901$), Random Forest ($R^2 = 0.934$), and XGBoost ($R^2 = 0.958$), showing systematic improvement through the targeted enhancements introduced in Phase 2.
3. The multi-output architecture simultaneously predicted shaft resistance Q_s, base resistance Q_b, and ultimate capacity Q_u while maintaining physical consistency $\hat{Q}_u \approx \hat{Q}_s + \hat{Q}_b$, that single-output models cannot supply.
4. The equilibrium residual distribution was concentrated around zero across test cases, which confirms that the governing differential equation was satisfied throughout the pile domain.
5. Experimental validation against the seven scaled load–settlement curves has shown that the PINN is able to show the closest agreement with observed behaviour, particularly for the layered cases (Tests 5 and 6), directly confirming the value of the ten-layer stratigraphy representation introduced in Phase 2.
6. External validation on an independent dataset confirmed superior generalisation of the PINN over all benchmark models.

6.1 Limitations of the Study

1. The training dataset was generated from governing equations rather than field measurements, which may limit soil representation for site-specific conditions.
2. PLAXIS 3D simulations with Mohr–Coulomb constitutive model was used which does not capture stress-path dependence, anisotropy, or time-dependent soil behaviour.

3. Experimental validation was conducted under 1g conditions in dry sand only, and 1g scaling uncertainties remain despite the applied corrections.
4. The mobilisation parameters η_s and η_b was calibrated using observed data rather than derived directly from the PINN displacement field, representing a disconnection between the trained physics and the output curves.
5. The framework addresses axial pile load capacity only.

6.2 Recommendations for Future Work

1. Fine tuning the framework on field pile load test databases can be done to address the synthetic data limitation.
2. Direct load–settlement generation from the WNet displacement field can be done to produce fully physics-derived output curves without post-hoc calibration.
3. Experimental programme can be done in a geotechnical centrifuge to eliminate 1g scaling uncertainties.
4. Framework can be extended to cohesive soils, saturated conditions, and combined loading to broaden application for general foundation design.

REFERENCES

Al Atroush, M. E. (2025). A deep learning physics-informed neural network (pinn) for predicting drilled shaft axial capacity. *Applied Computing and Geosciences*, 26:100246.

Ardakani, H., Soroush, A., and Karimpour, A. (2020). Prediction of pile bearing capacity using ensemble machine learning techniques. *Soils and Foundations*, 60(6):1516–1530.

Benbouras, M. A., Petrisor, A. I., Zedira, H., Ghelani, L., and Lefilef, L. (2021). Forecasting the bearing capacity of the driven piles using advanced machine-learning techniques. *Applied Sciences*, 11(22):10908.

Chin, F. K. (1970). Estimation of the ultimate load of piles not carried to failure. In *Proceedings of the 2nd Southeast Asian Conference on Soil Engineering*, pages 81–90, Singapore.

Coyle, H. M. and Reese, L. C. (1966). Load transfer for axially loaded piles in clay. *Journal of the Soil Mechanics and Foundations Division, ASCE*, 92(SM2):1–26.

Dehghanbanadaki, H. et al. (2021). Machine learning techniques for estimation of pile bearing capacity based on sensitivity analysis. *International Journal of Civil Engineering*.

Gao, Y., Qian, L., Yao, T., Mo, Z., Zhang, J., Zhang, R., Liu, E., and Li, Y. (2023). An improved physics-informed neural network algorithm for predicting the phreatic line of seepage. *Advances in Civil Engineering*, 2023:5499645.

Goh, A. T. C. (1995). Empirical design in geotechnics using neural networks. *Geotechnique*, 45(4):709–714.

Gomes, I. M. et al. (2021). Use of machine learning techniques for predicting the bearing capacity of piles. *Soils and Rocks*, 44(4):e2021074921.

Ito, S., Fukunaga, R., and Sako, K. (2024). Inverse analysis for estimating geotechnical parameters using physics-informed neural networks. *Soils and Foundations*, 64:101533.

Lee, J. H., Kim, Y., and Yoon, S. (2018). Comparative study of machine learning models for pile load capacity prediction. *Computers and Geotechnics*, 103:1–11.

Madianos, M., Lykourgias, P., Loupas, D., Papanikolaou, M., François, S., Anoyatis, G., and Tsikas, A. (2023). A physics informed neural network (pinn) approach for soil-pile interaction. In *COMPDYN 2023: 9th ECCOMAS Thematic Conference on*

Computational Methods in Structural Dynamics and Earthquake Engineering, pages 2917–2928, Athens, Greece.

Meyerhof, G. G. (1976). Bearing capacity and settlement of pile foundations. *Journal of the Geotechnical Engineering Division, ASCE*, 102(3):195–228.

Moosavian, S. M. H., Ghalandarzadeh, A., and Hoseini, A. (2021). A rehabilitation approach for rocking foundations: Vertical-monotonic and horizontal-cyclic loading. *Engineering Structures*, 245:112814.

Pham, B. T. et al. (2021). Predicting the bearing capacity of bored piles using machine learning techniques. *Neural Computing and Applications*.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Rizk, R. et al. (2023). Prediction of load capacity of open-ended pipe piles using ml. *Geotechnical and Geological Engineering*.

Udengaard, J. T. et al. (2024). Pile capacity prediction based on machine learning using support vector regression. In *19th Nordic Geotechnical Meeting*, Gothenburg, Sweden.

Umar, I. H., Salga, M. S., Lin, H., Hassan, J. I., Ahmad, A., Ibrahim, A. S., and Jechira, B. K. (2025). Performance characterisation of machine learning models for geotechnical axial pile load capacity estimation: an enhanced gpr-based approach. *Geomechanics and Geoengineering*.

Wu, J., Zhao, W., Wang, K., and El Naggar, M. H. (2026). Physics-informed neural networks (pinns) for dynamic pile-soil interaction problems. *Computers and Geotechnics*, 193:107941.

Yuan, B., Choo, C. S., Yeo, L. Y., Wang, Y., Yang, Z., Guan, Q., Suryasentana, S., Choo, J., Shen, H., Megia, M., et al. (2025). Physics-informed machine learning in geotechnical engineering: a direction paper. *Geomechanics and Geoengineering*, pages 1–32.

A SOURCE CODE

This appendix presents the complete source code developed for this thesis, arranged in the sequential order in which each component is used throughout the research workflow:

1. **Synthetic dataset generation** (Section A.1) is used to produces the 5,000-case pile bearing capacity training dataset.
2. **PLAXIS 3D automation** (Section A.2) automates finite-element axial pile load tests.
3. **PINN ensemble framework** (Section A.3) is for the core Physics-Informed Neural Network with ANN, Random Forest, and XGBoost baselines.
4. **Load–settlement curve comparison** (Section A.4) for post-processing script comparing model predictions against laboratory pile load test measurements.

All scripts are written in Python 3.10. Key dependencies are PyTorch 2.x, scikit-learn, XGBoost, NumPy, Pandas, Matplotlib, and SciPy.

A.1 Synthetic Pile Bearing Capacity Dataset Generation

The training dataset is generated entirely in Python without any finite-element solver. Each of the 5,000 cases is assigned a randomised pile geometry and a ten-layer soil profile whose parameters are sampled from class-specific statistical ranges. Shaft resistance is integrated numerically over the embedded pile length; base resistance uses the Vesic–Meyerhof bearing capacity factors.

A.1.1 Bearing Capacity Factors

Code A.1: Calculation of bearing capacity factors Nc, Nq, and Ngamma following the Vesic (1963) formulation. Limiting values Nc=9, Nq=1, Ngamma=0 are applied for purely cohesive soils.

```
    def bearing_capacity_factors(phi_deg: float):
    """ Returns Nc, Nq, Ngamma (Vesic 1963).
    Limiting case for undrained / cohesive soils applied
    when phi_deg <= 0.5 degrees.
    """
    if phi_deg <= 0.5:
    return 9.0, 1.0, 0.0
    phi    = math.radians(phi_deg)
    Nq     = (math.exp(math.pi * math.tan(phi))
    * (math.tan(math.radians(45) + phi / 2) ** 2))
    Nc     = (Nq - 1) / math.tan(phi)
    Ngamma = (Nq - 1) * math.tan(1.4 * phi)
    return Nc, Nq, Ngamma
```

A.1.2 Soil Classification and Parameter Ranges

Code A.2: Soil classification dictionary defining sampling ranges for eight geotechnical soil classes.

```
SOIL_CLASSES = {
    "Very Soft Clay": {
        "c": (12,25), "phi": (3,10), "gamma": (16.0,17.5),
        "Es": (4e3,10e3), "K0": (0.8,1.1), "alpha":
            (0.75,1.0)
    },
    "Soft Clay": {
        "c": (25,50), "phi": (8,16), "gamma": (16.5,18.0),
        "Es": (8e3,18e3), "K0": (0.7,1.0), "alpha":
            (0.65,0.9)
    },
    "Stiff Clay": {
        "c": (50,100), "phi": (15,24), "gamma": (17.0,19.0),
        "Es": (18e3,45e3), "K0": (0.55,0.9), "alpha":
            (0.45,0.75)
    },
    "Loose Sand": {
        "c": (0,5), "phi": (28,34), "gamma": (17.0,18.5),
        "Es": (12e3,30e3), "K0": (0.40,0.60), "alpha":
            (0.0,0.05)
    },
    "Medium Sand": {
        "c": (0,8), "phi": (32,38), "gamma": (17.5,19.0),
        "Es": (25e3,50e3), "K0": (0.38,0.55), "alpha":
            (0.0,0.05)
    },
    "Dense Sand": {
        "c": (0,10), "phi": (36,42), "gamma": (18.0,20.0),
        "Es": (45e3,90e3), "K0": (0.35,0.50), "alpha":
            (0.0,0.05)
    },
    "Silty Sand": {
        "c": (5,20), "phi": (26,34), "gamma": (17.0,19.0),
        "Es": (15e3,40e3), "K0": (0.45,0.70), "alpha":
            (0.10,0.25)
    },
    "Gravelly Sand": {
        "c": (0,8), "phi": (38,45), "gamma": (19.0,21.0),
        "Es": (60e3,110e3), "K0": (0.30,0.45), "alpha":
            (0.0,0.03)
    },
}
```

A.1.3 Single Pile Case Generator

Code A.3: Core case generator: pile geometry, ten-layer soil profile assignment, effective stress integration, shaft capacity summation over the embedded length, and base capacity from Vesic factors at the pile tip.

```
    def generate_case(case_id: int, rng: random.Random):
    # Pile geometry and elastic modulus by pile type
    D         = round(rng.uniform(0.4, 1.5), 3)
    L         = round(rng.uniform(8.0, 30.0), 3)
    pile_type = rng.choice(["Bored Concrete",
    "Driven Concrete",
    "Steel Pipe"])
    Ep = {"Bored Concrete": 27e6,
        "Driven Concrete": 30e6,
        "Steel Pipe":      200e6}[pile_type]      # kPa
    perimeter = math.pi * D
    area      = math.pi * D**2 / 4.0

    # Ten-layer soil profile (each layer 5 m thick)
    layers = []
    for i in range(10):
    name = rng.choice(list(SOIL_CLASSES.keys()))
    soil = sample_soil(rng, name)
    soil["layer_top"]    = i * 5.0
    soil["layer_bottom"] = (i + 1) * 5.0
    soil["thickness"]    = 5.0
    layers.append(soil)

    # Effective vertical stress at top of each layer
    sigma_top, running = [], 0.0
    for layer in layers:
    sigma_top.append(running)
    running += layer["gamma"] * layer["thickness"]

    # Shaft resistance integrated over embedded pile length
    Qs = 0.0
    for i, layer in enumerate(layers):
    z1, z2 = layer["layer_top"], layer["layer_bottom"]
    if L <= z1: break
    dz        = min(L, z2) - z1
    sigma_mid = sigma_top[i] + layer["gamma"] * dz / 2.0
    delta     = math.radians(0.75 * layer["phi"])
    ca        = layer["alpha"] * layer["c"]
    fs        = max(ca + layer["K0"]
    * sigma_mid * math.tan(delta), 0.0)
    Qs       += perimeter * dz * fs             # kN
```

```
42
43      # Base capacity using Vesic factors at pile tip layer
44      tip_index = min(int((L - 1e-9) // 5.0), 9)
45      tip       = layers[tip_index]
46      sigma_tip = (sigma_top[tip_index]
47      + tip["gamma"] * (L - tip["layer_top"]))
48      Nc, Nq, Ngamma = bearing_capacity_factors(tip["phi"])
49      q_b  = max(tip["c"]*Nc + sigma_tip*Nq
50      + 0.5*tip["gamma"]*D*Ngamma, 0.0)
51      Qb   = area * q_b
52      Qult = Qs + Qb
53      return build_record(case_id, pile_type, D, L, Ep,
54      layers, sigma_tip, Nc, Nq, Ngamma,
55      Qs, Qb, Qult)
```

A.1.4 Dataset Driver Function

Code A.4: Top-level function that generates the complete 5000-case dataset and writes it to a CSV file. The random seed ensures full reproducibility.

```
1       def generate_dataset(n=5000, seed=42,
2       output_csv="pile_bearing_capacity_dataset_5000.csv"):
3       rng  = random.Random(seed)
4       rows = [generate_case(i + 1, rng) for i in range(n)]
5       path = Path(output_csv)
6       with path.open("w", newline="", encoding="utf-8") as f:
7       writer = csv.DictWriter(f,
8       fieldnames=list(rows[0].keys()))
9       writer.writeheader()
10      writer.writerows(rows)
11      return path
12
13      if __name__ == "__main__":
14      out  = (Path(__file__).resolve().parent
15      / "pile_bearing_capacity_dataset.csv")
16      path = generate_dataset(output_csv=str(out))
17      print(f"Dataset saved to: {path}")
```

A.2 PLAXIS 3D Finite Element Analysis Automation

The PLAXIS 3D automation script drives PLAXIS 3D Connect Edition V22 via its Python remote-scripting API. It runs axial pile load tests for every diameter–length combination, builds the complete model programmatically, executes the calculation, extracts load–settlement data, and applies the Chin–Kondner extrapolation to estimate ultimate capacity. Here, the script can differ depending on the version of PLAXIS used and also using directly from Python Option inside PLAXIS or we are running from any

other editor like (Anaconda (Spyder)).

A.2.1 Analysis Settings

Code A.5: User-configurable parameters for the PLAXIS 3D parametric study: pile dimensions, material constants, maximum applied load, number of load steps, and the ten-layer Mohr–Coulomb soil profile.

```
    # Parametric combinations
    PILE_DIAMETERS_M = [0.5, 1.0, 1.5]       # m
    PILE_LENGTHS_M   = [10.0, 15.0, 20.0]    # m

    # Pile material
    PILE_E_KPA  = 210_000_000    # Young's modulus (kPa)
    PILE_W_KNM3 = 78.5           # Unit weight (kN/m3)

    # Load application
    QMAX_KN = 200_000            # Maximum applied load (kN)
    N_STEPS = 100                # Number of incremental load
       phases
    LAYER_T = 5.0                # Layer thickness (m)

    # Ten-layer Mohr-Coulomb soil profile
    SOIL_LAYERS = [
    {"gamma":18.0,"phi":30.0,"c":0.1,"Es":25000,"nu":0.30},
    {"gamma":18.5,"phi":32.0,"c":0.1,"Es":30000,"nu":0.30},
    {"gamma":19.0,"phi":34.0,"c":0.1,"Es":40000,"nu":0.30},
    {"gamma":19.5,"phi":35.0,"c":0.1,"Es":50000,"nu":0.30},
    {"gamma":20.0,"phi":36.0,"c":0.1,"Es":60000,"nu":0.28},
    {"gamma":20.0,"phi":36.0,"c":0.1,"Es":60000,"nu":0.28},
    {"gamma":20.0,"phi":37.0,"c":0.1,"Es":70000,"nu":0.28},
    {"gamma":20.0,"phi":37.0,"c":0.1,"Es":70000,"nu":0.28},
    {"gamma":20.5,"phi":38.0,"c":0.1,"Es":80000,"nu":0.25},
    {"gamma":20.5,"phi":38.0,"c":0.1,"Es":80000,"nu":0.25},
    ]
```

A.2.2 Automated Model Builder

Code A.6: PLAXIS 3D model construction via remote scripting: soil geometry, Mohr–Coulomb material assignment, embedded beam pile element, point load at the pile head, medium mesh generation, and creation of 100 incremental load phases.

```
    def build_model(D, L):
    g_in.clear(); time.sleep(2)
    g_in.gotosoil(); time.sleep(1)

    # Soil geometry and borehole
    g_in.SoilContour.initializerectangular(0, 0, 50, 50)
```

```
    bh = g_in.borehole(25, 25)
    g_in.setproperties(bh, "Head", 0.0)
    for _ in SOIL_LAYERS:
    g_in.soillayer(LAYER_T)

    # Assign Mohr-Coulomb materials
    mats = []
    for i, layer in enumerate(SOIL_LAYERS):
    mat = g_in.soilmat()
    g_in.set(mat.SoilModel, 2)          # Mohr-Coulomb
    g_in.set(mat.DrainageType, 0)       # Drained
    g_in.set(mat.gammaUnsat, layer["gamma"])
    g_in.setproperties(mat, "Eref", layer["Es"])
    g_in.setproperties(mat, "nu",   layer["nu"])
    g_in.setproperties(mat, "cref", layer["c"])
    g_in.setproperties(mat, "phi",  layer["phi"])
    g_in.setproperties(mat, "psi",
    max(0.0, layer["phi"] - 30.0))
    mats.append(mat)
    for i, mat in enumerate(mats):
    g_in.setmaterial(g_in.Soillayers[i], mat)

    # Embedded beam pile and point load
    g_in.gotostructures(); time.sleep(1)
    pm = g_in.embeddedbeammat()
    g_in.commands(f"set {pm.Name} E {PILE_E_KPA}")
    g_in.commands(f"set {pm.Name} w {PILE_W_KNM3}")
    g_in.commands(f"set {pm.Name} Diameter {D}")
    pile = g_in.embeddedbeam((25,25,0), (25,25,-L))[3]
    g_in.setmaterial(pile, pm)
    load = g_in.pointload(25, 25, 0)[-1]
    g_in.commands(f"set {load.Name} Fz -1")

    # Mesh
    g_in.gotomesh(); time.sleep(1)
    try:    g_in.commands("generateemesh medium")
    except: g_in.commands("generateemesh coarse")
    time.sleep(5)

    # Incremental load phases
    g_in.gotostages(); time.sleep(1)
    q_step, phase_names = QMAX_KN / N_STEPS, []
    prev_name = g_in.Phases[0].Name
    for i in range(1, N_STEPS + 1):
    g_in.commands(f"phase {prev_name}")
    ph = list(g_in.Phases)[-1]
    g_in.commands(f"set {load.Name} Fz {-(q_step*i)}")
```

```
    g_in.commands(f"activate {load.Name} {ph.Name}")
    phase_names.append(ph.Name)
    prev_name = ph.Name
    return phase_names
```

A.2.3 Chin–Kondner Ultimate Capacity Estimation

Code A.7: Chin–Kondner (1970) hyperbolic extrapolation of the load–settlement curve. The linearised s/Q vs s relationship gives slope a; the ultimate capacity is Qu = 1/a.

```
    def chin_kondner(results):
    pts = [(r["settlement_mm"], r["applied_kN"])
    for r in results if r["settlement_mm"] > 0]
    if len(pts) < 10:
    return None
    x  = [s     for s, q in pts]
    y  = [s / q for s, q in pts]
    n  = len(x)
    sx, sy = sum(x), sum(y)
    sxx    = sum(xi**2 for xi in x)
    sxy    = sum(xi*yi for xi, yi in zip(x, y))
    den    = n * sxx - sx**2
    if abs(den) < 1e-12:
    return None
    a = (n * sxy - sx * sy) / den
    return round(1.0 / a, 1) if a > 0 else None
```

A.3 Physics-Informed Neural Network Ensemble Framework

This section presents the key components of the PINN ensemble framework. Two neural networks are used jointly during training: `QNet` predicts the three pile capacity components (Q_s, Q_b, Q_u); `WNet` approximates the axial displacement field $w(z)$ to evaluate the governing pile equilibrium ODE as a physics residual penalty.

A.3.1 Hyperparameter Configuration

Code A.8: Centralised Config dataclass with all architecture sizes, training hyperparameters, and physics constants

```
    @dataclass
    class Config:
    seed: int = 42
    ensemble_seeds: Tuple[int, ...] = (42, 52, 62)
    # PINN architecture
    pinn_activation: str = "tanh"
    pinn_q_hidden: Tuple[int, ...] = (256, 256, 128)
    pinn_w_hidden: Tuple[int, ...] = (256, 256, 256, 128)
    # Training
```

```
    pinn_epochs: int = 500
    pinn_batch_size: int = 64
    pinn_lr: float = 1e-3
    pinn_patience: int = 70
    collocation_points: int = 48
    use_lbfgs: bool = True
    lbfgs_max_iter: int = 150
    lbfgs_lr: float = 0.8
    # Loss configuration
    use_huber_data_loss: bool = True
    huber_delta_scaled: float = 0.03
    use_trainable_beta: bool = True
    use_trainable_lambdas: bool = True
    # Loss weight upper bounds
    lambda_p_max: float = 1.0
    lambda_b_max: float = 1.0
    lambda_global_max: float = 1.0
    lambda_consistency_max: float = 1.0
    # Regularisation
    lambda_beta_reg: float = 1e-4
    lambda_weight_reg: float = 1e-4
    # Physics warm-up epochs
    physics_warmup_start: int = 10
    physics_warmup_end: int = 140
    # Mobilisation reference displacements (fractions of D)
    eta_shaft: float = 0.01
    eta_base: float = 0.05
    # ANN baseline
    ann_epochs: int = 500
    ann_lr: float = 8e-4
    ann_patience: int = 45
    ann_hidden: Tuple[int, ...] = (512, 256, 128, 64)
```

A.3.2 Geotechnical Engineering Feature

Code A.9: Derived geotechnical input features: embedded-length weighted averages of soil properties, tip-layer values, effective stress at the pile tip, and composite friction index. These supplement the raw per-layer parameters.

```
    def add_engineered_features(df, C):
    Dcol, Lcol = C["D"], C["L"]
    # Pile geometry
    df["eng_slenderness_L_over_D"] = (
    df[Lcol] / np.maximum(df[Dcol], 1e-6))
    df["eng_Ap_m2"] = np.pi * (df[Dcol] ** 2) / 4.0
    df["eng_Up_m"]  = np.pi * df[Dcol]
    # Per-layer arrays (N_samples x 10)
```

```
    c_arr   = df[[f"layer_{i}_c_kPa"
    for i in range(1,11)]].to_numpy()
    phi_arr = df[[f"layer_{i}_phi_deg"
    for i in range(1,11)]].to_numpy()
    K0_arr  = df[[f"layer_{i}_K0"
    for i in range(1,11)]].to_numpy()
    al_arr  = df[[f"layer_{i}_alpha"
    for i in range(1,11)]].to_numpy()
    ga_arr  = df[[f"layer_{i}_gamma_kN_m3"
    for i in range(1,11)]].to_numpy()

    # Weighted averages over the embedded portion only
    tops  = np.arange(0.0, 50.0, 5.0)
    Lvals = df[Lcol].to_numpy().reshape(-1, 1)
    dz    = np.clip(Lvals - tops.reshape(1,-1), 0.0, 5.0)
    denom = np.maximum(dz.sum(axis=1), 1e-8)
    wavg  = lambda arr: (arr * dz).sum(axis=1) / denom

    df["eng_mean_c_embedded"]     = wavg(c_arr)
    df["eng_mean_phi_embedded"]   = wavg(phi_arr)
    df["eng_mean_K0_embedded"]    = wavg(K0_arr)
    df["eng_mean_alpha_embedded"] = wavg(al_arr)

    # Tip-layer values
    tip_idx = np.clip(
    np.floor((df[Lcol].to_numpy()-1e-6)/5.).astype(int),
    0, 9)
    df["eng_tip_c"]   = c_arr[np.arange(len(df)), tip_idx]
    df["eng_tip_phi"] = phi_arr[np.arange(len(df)), tip_idx]

    # Effective vertical stress at pile tip
    gd = ga_arr * 5.0
    sv = np.zeros(len(df))
    for r in range(len(df)):
    ti    = tip_idx[r]
    sv[r] = (gd[r,:ti].sum()
    + ga_arr[r,ti]
    * max(df.iloc[r][Lcol] - ti*5., 0.))
    df["eng_sigma_tip_kPa"] = sv
    # Composite indices
    delta = np.deg2rad(0.75 * df["eng_mean_phi_embedded"])
    df["eng_interface_friction_index"] = (
    df["eng_mean_K0_embedded"] * np.tan(delta))
    df["eng_alpha_c_index"] = (
    df["eng_mean_alpha_embedded"]
    * df["eng_mean_c_embedded"])
    return df
```

A.3.3 Network Architectures: QNet and WNet

Code A.10: For calculation of QNet and WNet

```
    class QNet(nn.Module):
    """
    Capacity prediction network.
    Hidden layers : (256, 256, 128),  Activation : tanh
    Output        : sigmoid  =>  three values in (0,1)
    [Qs_scaled, Qb_scaled, Qu_scaled]
    """
    def __init__(self, in_dim: int):
    super().__init__()
    self.mlp = MLP(in_dim, 3,
    CFG.pinn_q_hidden,
    CFG.pinn_activation)
    def forward(self, x):
    return torch.sigmoid(self.mlp(x))


    class WNet(nn.Module):
    """
    Displacement field network (training only).
    Input  : [z_hat | features | Qu_scaled | D]
    Output : w(z) >= 0 (softplus), scaled by 0.05*D.
    Hidden layers : (256, 256, 256, 128),  activation : tanh
    """
    def __init__(self, in_dim_total: int):
    super().__init__()
    self.mlp = MLP(in_dim_total, 1,
    CFG.pinn_w_hidden,
    CFG.pinn_activation)
    def forward(self, z_hat, x, q_u_s, D):
    inp = torch.cat([z_hat, x, q_u_s], dim=1)
    return F.softplus(self.mlp(inp)) * 0.05 * D
```

A.3.4 Physics-Informed Loss Assembly

Code A.11: For Total PINN loss as a weighted sum of five terms: Huber data fidelity, ODE equilibrium residual, head and toe boundary conditions, global equilibrium consistency, and the additive constraint Qu = Qs + Qb.

```
    def pinn_loss_components(qnet, wnet, beta_raw, lambda_raws,
    Xs, Xraw, y_true_s,
    y_min, y_max, device, epoch):
    D, L, Ep, c, phi, gamma, Es, K0, alpha = unpack_raw(Xraw)
    Ap   = torch.pi * D**2 / 4.0
    Up   = torch.pi * D
```

```
    EA   = Ep * Ap
    beta = current_beta(beta_raw, device)
    ramp = ramp_factor(epoch)
    lam_p, lam_b, lam_gl, lam_c = current_lambdas(
    lambda_raws, ramp, device)
    # 1. Network forward pass
    q_pred_s = qnet(Xs)
    q_pred   = (q_pred_s * (y_max_t - y_min_t) + y_min_t)
    Qs_pred, Qb_pred, Qu_pred = (q_pred[:,0:1],
    q_pred[:,1:2],
    q_pred[:,2:3])

    # 2. Data loss (Huber)
    L_data = F.huber_loss(q_pred_s, y_true_s,
    delta=CFG.huber_delta_scaled)

    # 3. ODE residual  EA/L^2 * w_zz + Up*t(w,z) = 0
    R    = EA / (L**2) * w_zz + Up * t
    Rn   = R / (Q_ref / PROFILE_DEPTH)
    L_phys_res = torch.mean(Rn**2)

    # 4. Boundary conditions
    #    Head z=0: P(0) = Qu  |  Toe z=L: P(L) = Qb
    L_bc = (torch.mean(((P0 - Qu_pred) / Q_ref)**2)
    + torch.mean(((PL - Qb_pred) / Q_ref)**2))

    # 5. Global equilibrium: predicted vs physics-derived
    L_global = (torch.mean(((Qs_pred-Qs_phys)/Q_ref)**2)
    + torch.mean(((Qb_pred-Qb_phys)/Q_ref)**2)
    + lam_gl
    * torch.mean(((Qu_pred-Qu_phys)/Q_ref)**2))

    # 6. Additive consistency  Qu = Qs + Qb
    L_cons = torch.mean(
    ((Qu_pred - (Qs_pred + Qb_pred)) / Q_ref)**2)
    # Total weighted loss
    L_total = (L_data
    + lam_p  * L_phys_res
    + lam_b  * L_bc
    + L_global
    + lam_c  * L_cons
    + CFG.lambda_beta_reg   * L_beta_reg
    + CFG.lambda_weight_reg * L_lambda_reg)
    return {"L_total": L_total, "L_data": L_data,
        "L_phys_res": L_phys_res, "L_bc": L_bc,
        "L_global": L_global, "L_consistency": L_cons}
```

A.3.5 Two-Stage Training: Adam and L-BFGS

Code A.12: Training strategy for each ensemble member: Adam with early stopping and gradient clipping (Stage 1) followed by L-BFGS with strong Wolfe line search (Stage 2).

```
    # Stage 1: Adam
    opt = torch.optim.Adam(params, lr=CFG.pinn_lr)
    scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(
    opt, mode="min", factor=0.5, patience=15)
    best_val_rmse, patience_ctr = float("inf"), 0

    for epoch in range(1, CFG.pinn_epochs + 1):
    qnet.train(); wnet.train()
    for Xs, Xraw, ys, _ in dl_train:
    Xs, Xraw, ys = (Xs.to(device),
    Xraw.to(device), ys.to(device))
    opt.zero_grad(set_to_none=True)
    L = pinn_loss_components(qnet, wnet, beta_raw,
    lambda_raws, Xs, Xraw, ys,
    y_min, y_max, device, epoch)
    L["L_total"].backward()
    torch.nn.utils.clip_grad_norm_(params, max_norm=3.0)
    opt.step()

    val_rmse = evaluate_member_rmse(
    qnet, X_val, y_val[:,2], y_min, y_max, device)
    scheduler.step(val_rmse)

    if val_rmse < best_val_rmse - 1e-6:
    best_val_rmse = val_rmse
    best_state    = copy.deepcopy(qnet.state_dict())
    patience_ctr  = 0
    else:
    patience_ctr += 1
    if patience_ctr >= CFG.pinn_patience:
    break

    # Stage 2: L-BFGS refinement
    lbfgs = torch.optim.LBFGS(
    lbfgs_params,
    lr             = CFG.lbfgs_lr,
    max_iter       = CFG.lbfgs_max_iter,
    line_search_fn = "strong_wolfe")

    def closure():
    lbfgs.zero_grad(set_to_none=True)
    total = 0.0
```

```
    for Xs, Xraw, ys in batches:
    L = pinn_loss_components(
    qnet, wnet, beta_raw, lambda_raws,
    Xs, Xraw, ys, y_min, y_max, device,
    epoch=CFG.physics_warmup_end + 1)
    L["L_total"].backward()
    total += L["L_total"]
    return total / max(len(batches), 1)

    lbfgs.step(closure)
```

A.3.6 ANN Baseline Architecture

Code A.13: Fully connected ANN benchmark without physics constraints

```
    class ANNRegressor(nn.Module):
    """
    Data-driven ANN baseline for comparison with the PINN.
    Output: sigmoid ensures the scaled prediction stays in [0,1].
    """
    def __init__(self, in_dim: int):
    super().__init__()
    layers, prev = [], in_dim
    for h in CFG.ann_hidden:
    layers += [nn.Linear(prev, h),
    activation(CFG.ann_activation)]
    prev = h
    layers += [nn.Linear(prev, 1)]
    self.net = nn.Sequential(*layers)
    for m in self.net:
    if isinstance(m, nn.Linear):
    nn.init.kaiming_normal_(m.weight,
    nonlinearity="relu")
    nn.init.zeros_(m.bias)
    def forward(self, x):
    return torch.sigmoid(self.net(x))
```

A.4 Load–Settlement Curve Comparison with Laboratory Data

After training is complete, this post-processing script compares model-predicted capacities against laboratory pile load test measurements for seven validation cases. It loads the trained models, calibrates the hyperbolic mobilisation parameters η_s and η_b from the observed curve, builds smooth predicted curves for all models, and saves comparison figures.

A.4.1 Plot Style Configuration

Code A.14: Matplotlib style settings (Times New Roman, 300 DPI) and line styles for the four models and the observed curve, for consistently across all comparison figures.

```
matplotlib.rcParams.update({
    "font.family":      "serif",
    "font.serif":       ["Times New Roman","Times","DejaVu
       Serif"],
    "mathtext.fontset": "dejavuserif",
    "axes.grid":        True,
    "grid.alpha":       0.25,
    "axes.labelsize":   11,
    "axes.titlesize":   11,
    "legend.fontsize":  8.5,
    "pdf.fonttype":     42,
    "ps.fonttype":      42,
    "savefig.bbox":     "tight",
})

OBS_STYLE = dict(color="black", linestyle="-",
linewidth=1.8, label="Observed", zorder=10)

MODEL_STYLE = {
    "PINN_Ensemble": dict(color="#1f77b4", linestyle="-",
    linewidth=2.0, label="PINN Ensemble"),
    "ANN":           dict(color="#d62728", linestyle="--",
    linewidth=1.6, label="ANN"),
    "RF":            dict(color="#2ca02c", linestyle="-.",
    linewidth=1.6, label="Random Forest"),
    "XGBoost":       dict(color="#ff7f0e", linestyle=":",
    linewidth=1.6, label="XGBoost"),
}
```

A.4.2 Observed Curve Smoothing

Code A.15: Chin–Kondner linearisation used to smooth noisy observed load–settlement data.

```
def smooth_observed(s_raw_mm: np.ndarray, P_raw: np.ndarray,
n_pts: int = 300):
"""
Smooth observed Q-s data by Chin-Kondner linearisation.
Fits:  s/Q = a + b*s   =>   Q(s) = s / (a + b*s)
where  a = initial compliance,  b = 1/Qu (asymptote).
Returns (s_fine_mm, P_fine_kN) over the observed range.
"""
order    = np.argsort(s_raw_mm)
s, P     = s_raw_mm[order], P_raw[order]
_, idx   = np.unique(s, return_index=True)
```

```
    s, P     = s[idx], P[idx]
    mask     = (s > 0) & (P > 0)
    b_coef   = np.polyfit(s[mask], s[mask] / P[mask], 1)
    b, a     = float(b_coef[0]), max(float(b_coef[1]), 1e-8)
    b        = max(b, 1e-12)
    s_fine   = np.linspace(0., float(s[-1]), n_pts)
    P_fine   = np.clip(s_fine / (a + b * s_fine), 0, None)
    return s_fine, P_fine
```

A.4.3 Mobilisation Parameter Calibration

Code A.16: L-BFGS-B calibration of shaft and base mobilisation parameters eta_s and eta_b.

```
    def calibrate_eta_fixed_frac(s_obs_mm, P_obs, D, Qu, frac):

    s_m = s_obs_mm / 1000.0
    Qs  = frac * Qu
    Qb  = (1 - frac) * Qu

    def loss(params):
    eta_s, eta_b = params
    P_pred = (Qs * np.tanh(s_m / (eta_s * D + 1e-12)) +
    Qb * np.tanh(s_m / (eta_b * D + 1e-12)))
    return np.mean((P_pred - P_obs) ** 2)

    s_ref = float(s_m.max()) / 2.0 / D
    res   = minimize(loss,
    x0     = [s_ref * 0.5, s_ref],
    bounds = [(0.02, 5.0), (0.02, 5.0)],
    method = "L-BFGS-B",
    options= {"ftol":1e-14,"maxiter":2000})
    eta_s, eta_b = np.clip(res.x, [0.02,0.02], [5.0,5.0])
    return float(eta_s), float(eta_b)
```

A.4.4 Predicted Load–Settlement Curve

Code A.17: Construction of a predicted load–settlement curve for any model

```
    def predicted_curve(Qu: float, frac: float,
    eta_s: float, eta_b: float,
    D: float, s_max_mm: float):
    # Build the predicted P-s curve with calibrated parameters.
    s_m = np.linspace(0., s_max_mm / 1000., SETTLEMENT_POINTS)
    P   = (frac * Qu * np.tanh(s_m / (eta_s * D + 1e-12)) +
    (1-frac) * Qu * np.tanh(s_m / (eta_b * D + 1e-12)))
    return s_m * 1000., P
```

A.4.5 Main Comparison Workflow

Code A.18: Main execution workflow: reads observed load test data and pile geometry, loads trained model artefacts, calibrates eta_s and eta_b for each of the seven cases using the PINN-predicted capacity, builds predicted curves for all models, and exports individual plots, a combined grid figure, and a summary CSV file.

```
    def main():
    RESULTS_DIR.mkdir(parents=True, exist_ok=True)
    # Load observed laboratory data and pile geometry
    df_obs   = pd.read_csv(OBSERVED_CSV)
    df_cases = pd.read_csv(CASES_CSV)
    case_ids = df_obs["case"].unique().tolist()
    # Load scalers and trained models
    art  = load_artifacts(OUTPUTS_DIR)
    X    = prepare_X(df_cases, art)
    pred = predict_all(X, art)
    all_data = {}
    for cid in case_ids:
    pos   = df_cases.index[
    df_cases["case_id"] == cid][0]
    D_val = float(df_cases.loc[pos, D_col])
    sub   = (df_obs[df_obs["case"]==cid]
    .sort_values("settlement_mm"))
    s_obs = sub["settlement_mm"].to_numpy()
    P_obs = sub["axial_load_kN"].to_numpy()
    s_max = float(s_obs.max())
    # Smooth observed data
    obs_s, obs_P = smooth_observed(s_obs, P_obs)
    # Calibrate eta using PINN capacity and shaft fraction
    Qu_p   = float(pred["PINN_Ensemble"]["Qu"][pos])
    frac_p = (float(pred["PINN_Ensemble"]["Qs"][pos])
    / max(Qu_p, 1.0))
    eta_s, eta_b = calibrate_eta_fixed_frac(
    s_obs, P_obs, D_val, Qu_p, frac_p)
    # Build predicted curves for all models
    pred_curves = {}
    for model, vals in pred.items():
    Qu_m = float(vals["Qu"][pos])
    s_mm, P = predicted_curve(
    Qu_m, frac_p, eta_s, eta_b, D_val, s_max)
    pred_curves[model] = (s_mm, P)
    all_data[cid] = {"obs_s_sm": obs_s, "obs_P_sm": obs_P,
        "pred_curves": pred_curves,
        "s_max_obs": s_max, "D": D_val}
    # Save individual plots, grid figure, and CSV
    for cid, d in all_data.items():
```

```
    plot_single(cid, d["obs_s_sm"], d["obs_P_sm"],
    d["pred_curves"], d["s_max_obs"],
    plots_dir / f"load_settlement_{cid}")
    plot_grid(all_data,
    plots_dir / "load_settlement_all_cases")
    export_csv(all_data,
    RESULTS_DIR / "load_settlement_observed_vs_predicted.csv")
    if __name__ == "__main__":
    main()
```

A.5 Summary of Key Symbols

Table A.1: Key symbols used in the source code and their physical meaning.

Symbol	Code variable	Physical meaning
D	`D`	Pile diameter (m)
L	`L`	Pile embedded length (m)
E_p	`Ep`	Pile Young's modulus (kPa)
A_p	`Ap`	Pile cross-sectional area (m^2)
U_p	`Up`	Pile perimeter (m)
Q_s	`Qs_pred`	Shaft (skin friction) capacity (kN)
Q_b	`Qb_pred`	Base (tip) capacity (kN)
Q_u	`Qu_pred`	Ultimate pile capacity = $Q_s + Q_b$ (kN)
$w(z)$	`wnet(z_hat,...)`	Axial pile displacement (m)
$t(z)$	`t`	Mobilised unit shaft friction (kPa)
t_{ult}	`tult`	Ultimate unit shaft friction (kPa)
$q_{b,ult}$	`qbult`	Ultimate unit base resistance (kPa)
β	`beta_raw`	Interface friction reduction factor
λ_p	`lambda_raws["p"]`	Weight – physics residual loss
λ_b	`lambda_raws["b"]`	Weight – boundary condition loss
N_c, N_q	`Nc, Nq`	Vesic (1963) bearing capacity factors
η_s	`eta_shaft`	Shaft mobilisation reference displacement
η_b	`eta_base`	Base mobilisation reference displacement
K_0	`K0`	At-rest lateral earth pressure coefficient
α	`alpha`	Adhesion factor (API / Tomlinson method)

B Research Visit to IIT PATNA

This appendix presents selected supplementary photographs related to the experimental work carried out at IIT Patna. These images are included to document the testing process, instrumentation, data acquisition, and supporting laboratory activities. Additional personal and academic photographs related to the research visit are also included for record purposes.

B.1 Visit and Academic Activities at IIT Patna

Figure B.1: Research Visit at IIT Patna

Figure B.2: Presentation of the research work at IIT Patna.

Figure B.3: With Assoc. Prof. Dr. Arvind Kumar Jha and his PhD & M Tech Students

Figure B.4: Infront of Civil Engineering Workshop at IIT Patna

Figure B.5: Infront of Geotechnical Engineering Lab at IIT Patna

Figure B.6: Infront of Department of Civil and Environmental Engineering, IIT Patna

B.2 Experimental Photographs

Figure B.7: Specific gravity test of sand.

Figure B.8: Relative density test of sand.

Figure B.9: Direct shear test of sand.

Figure B.10: Sample preparation for XRD analysis.

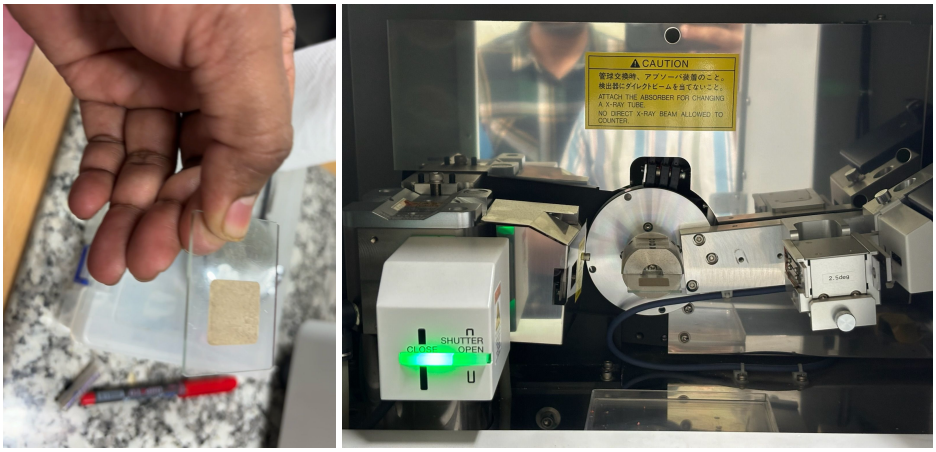

Figure B.11: Instrumentation setup for XRD analysis.

Figure B.12: Sample preparation for triaxial testing and tested sample.

Figure B.13: Triaxial setup for UU test of sand sample.

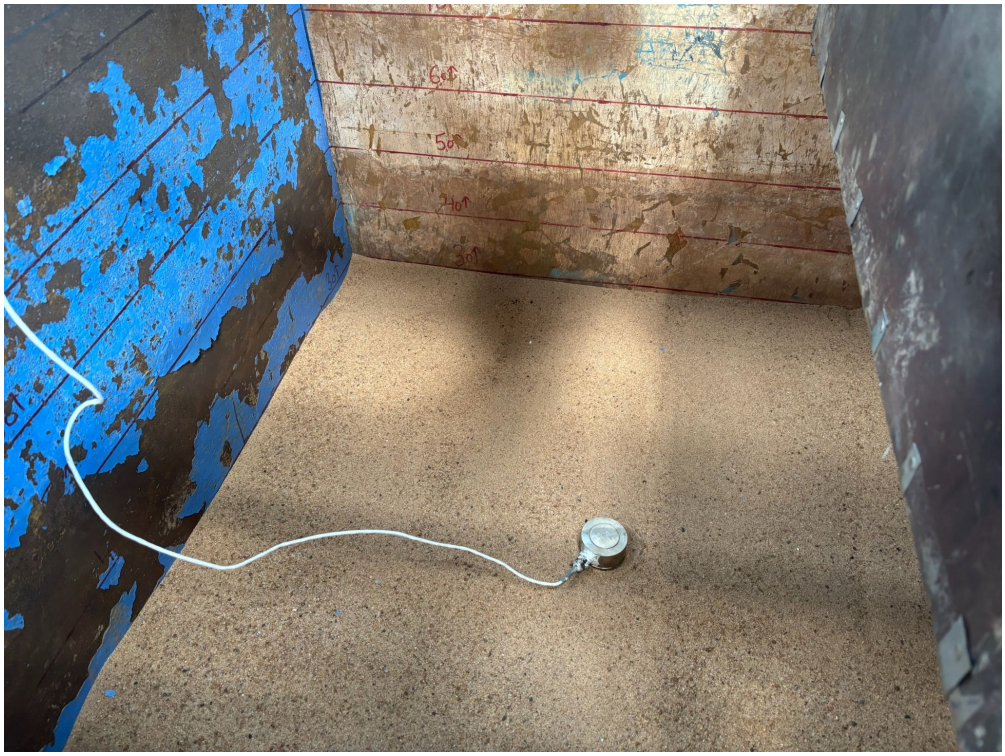

Figure B.14: Leveling of sand bed and installation of pressure cell.

Figure B.15: Steel pile installation.

Figure B.16: Data observation using the DAQ system.

B.3 Data Processing and Supporting Records

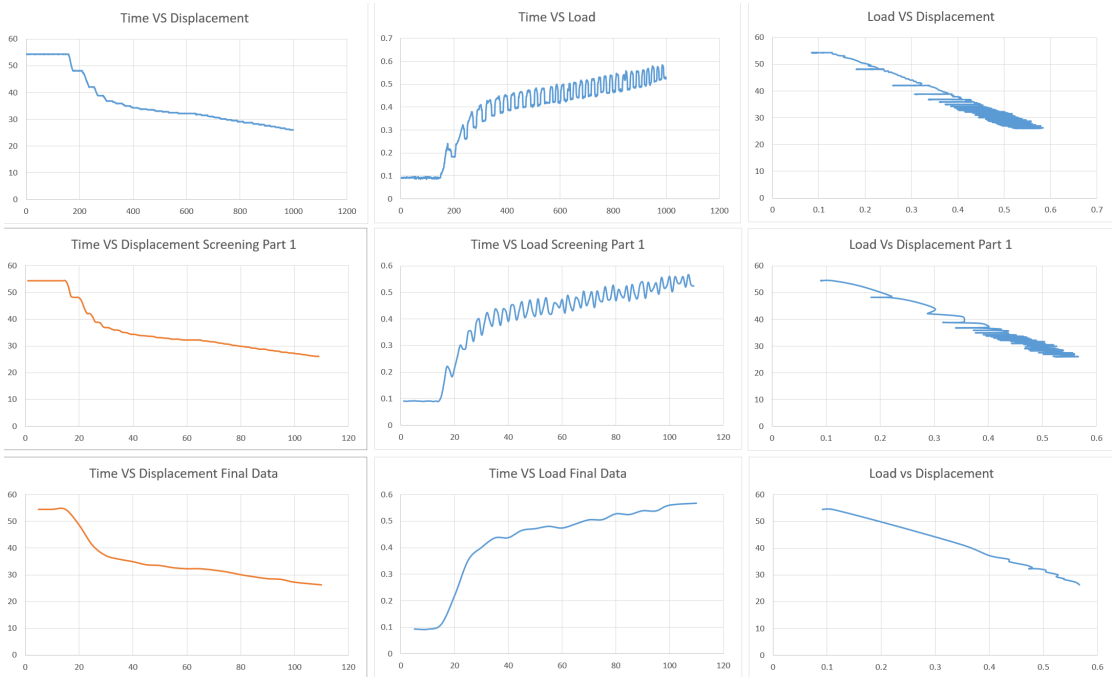

Figure B.17: Sample of raw data processing.

B.4 Additional Results

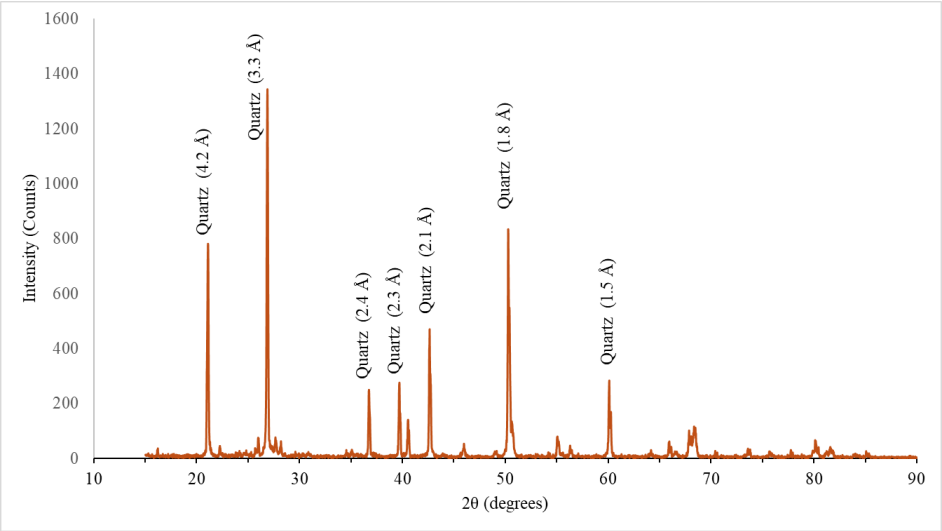

Figure B.18: XRD result of the sand sample.

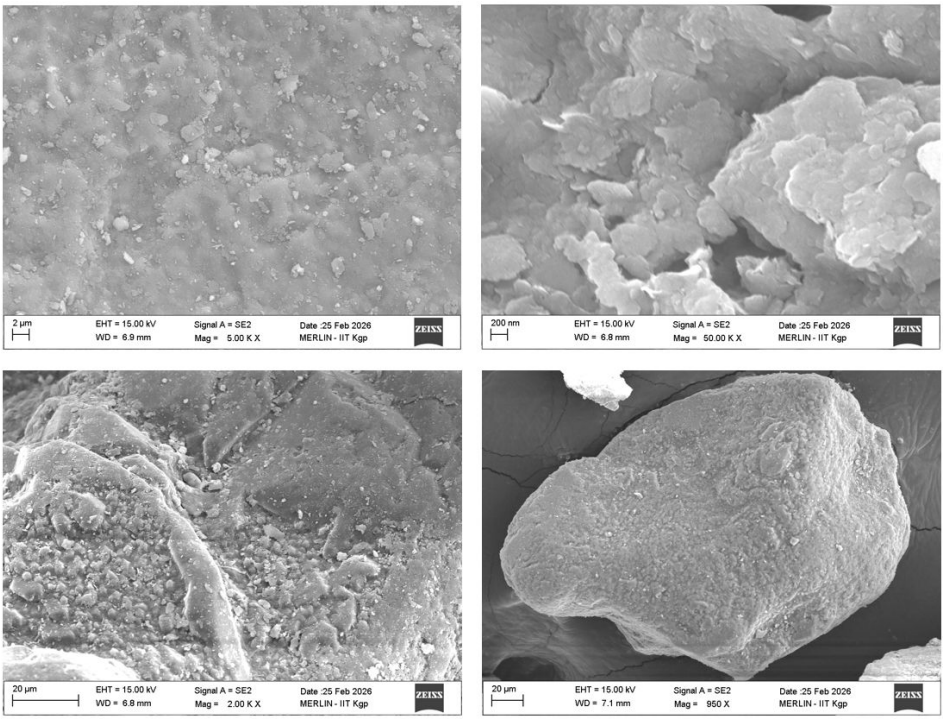

Figure B.19: SEM images of the sand sample.

17th IOE Graduate Conference

Certificate Of Participation

This certificate is awarded to

Bipul Mainali

in recognition of an invaluable contribution as

Paper Presenter

at the 17th IOE Graduate Conference

Organized by

Tribhuvan University, IOE, Dean's Office and Purwanchal Campus

held from Jan 2 – 3, 2026 at Purwanchal Campus, Dharan, Nepal.

Asst. Prof. Kaji Ram Karki
Campus Chief , Purwanchal Campus

Asso. Prof. Anu Shrestha
Conference Converner

Prof. Dr. Sushil Bahadur Bajracharya
Dean, Institute of Engineering

Physics Informed Machine Learning for Predicting Pile Load Capacity from Soil Parameters

Bipul Mainali [a], Indra Prasad Acharya [b],

[a,b] *Department of Civil Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal*
[a] 080msgte004.bipul@pcampus.edu.np , [b] indrapd@ioe.edu.np

Abstract
Accurate prediction of axial pile load capacity is critical for the safe and economical design of deep foundations. Traditional empirical and analytical approaches often oversimplify soil pile interaction, while fully data driven machine learning models lack interpretability and generalization capability. This study develops a Physics Informed Neural Network (PINN) that embeds Meyerhof's end bearing theory, the β method for shaft resistance, and effective stress principles directly into its loss function. A synthetic dataset was generated using established geotechnical formulations incorporating soil friction angle, unit weight, groundwater depth, and pile geometry. The PINN was compared against a Random Forest model using internal and external validation. Results show that the PINN achieved MAE = 1010.441 kN, RMSE = 1440.577 kN, and $R^2 = 0.9402$, outperforming Random Forest (MAE = 1152.535 kN, RMSE = 1678.251 kN, $R^2 = 0.9188$). External validation further confirmed strong generalization with MAE = 966.490 kN, RMSE = 1406.940 kN, and $R^2 = 0.9383$. These findings demonstrate that physics informed machine learning provides a transparent, data efficient, and physically consistent framework for predicting pile load capacity, offering a promising advancement for geotechnical engineering practice.

Keywords
Physics Informed Machine Learning, Pile load capacity, Soil pile interaction, Geotechnical engineering

1. Introduction

Pile foundations are essential for transferring structural loads to deeper, more competent soil or rock layers when near surface materials lack the required bearing capacity. However, accurately predicting axial pile capacity remains a longstanding challenge due to the complex, nonlinear nature of soil pile interaction. Variability in subsurface stratification, stress history, and groundwater conditions further complicates the application of conventional analytical and empirical design methods [1, 2].

Classical bearing capacity theories developed by Terzaghi and Meyerhof form the basis for estimating end bearing resistance and load transfer mechanisms. Likewise, empirical approaches such as the β method provide practical means for evaluating shaft friction, yet these methods rely on simplified assumptions regarding soil stiffness, stress distribution, and drainage behavior [2]. Such simplifications may be insufficient in heterogeneous, layered, or partially saturated soil profiles where load transfer mechanisms deviate from idealized formulations.

With the emergence of data driven techniques, machine learning (ML) has offered alternative pathways for modeling the highly nonlinear relationships that govern pile capacity. Studies employing Artificial Neural Networks, Support Vector Regression, Gaussian Process Regression, and ensemble methods have reported strong predictive performance across diverse geotechnical datasets [3, 4, 5, 6, 7, 8]. While ML models excel at approximating complex patterns, they often function as black boxes with limited interpretability and require large, high quality datasets conditions rarely attainable in geotechnical engineering practice.

Physics Informed Machine Learning (PIML) has recently emerged as a promising paradigm that mitigates these limitations by embedding governing physical laws into the model training process [9, 10]. Physics Informed Neural Networks (PINNs) enforce equation based constraints through their loss function, ensuring that predictions remain consistent with fundamental geotechnical principles such as effective stress, bearing capacity theory, and load transfer mechanisms. PINNs have been successfully applied to seepage, consolidation, and stress strain modeling [11], yet applications specifically targeting axial pile capacity remain limited.

This study aims to address this gap by developing a PINN framework that incorporates Meyerhof's bearing capacity formulation, the β method for shaft resistance, and effective stress principles. By combining physics based constraints with data driven learning, the objective is to enhance prediction accuracy, improve generalization capability, and provide transparent, physically consistent predictions for deep foundation design.

2. Literature Review

Machine learning has been increasingly applied to axial pile capacity prediction due to its ability to capture nonlinear relationships between soil properties, pile geometry, and capacity. Gomes et al. [3] evaluated multiple ML algorithms and demonstrated that Random Forest (RF) models outperform empirical and semi empirical methods across diverse soil conditions. A conducted a comparative study of twelve ML techniques using international datasets and found that deep neural networks and ensemble learning generally

produced the highest accuracy[4]. Additional studies incorporating SVMs, hybrid neural networks, Gaussian Processes, and soft computing frameworks also reported strong predictive capabilities [5, 6, 7, 12].

Random Forest was selected as the baseline model for this study because:

- It consistently performs well on geotechnical datasets with nonlinear and coupled parameters.
- It is robust against overfitting in small to medium sized datasets.
- It provides meaningful measures of feature importance.
- Prior studies have validated its effectiveness for pile capacity prediction [8, 3].

Recent advancements in geotechnical computing have explored Physics Informed Machine Learning (PIML) as a pathway to enhance model interpretability and robustness. A PINN based framework was developed for modeling drilled shaft behavior, demonstrating that embedding constitutive soil behavior and load transfer equations can significantly improve prediction consistency [13]. Meanwhile, comprehensive "direction papers" on PIML in geomechanics highlight its potential for solving problems governed by PDEs, stress equilibrium, and soil structure interaction [11, 9, 10].

This study builds on these developments by integrating a geotechnical physics based PINN and benchmarking its performance against the Random Forest model. Through this comparison, the goal is to assess whether embedding physics into the model training process yields improved accuracy, generalization, and interpretability over traditional ML methods.

3. Methodology

The study employs a Physics Informed Neural Network (PINN) framework to model a complex physical system, such as the ultimate bearing capacity (Q_u) of a foundation, integrating experimental or field data with governing physical laws. The methodology is structured around three core components: the Neural Network architecture, the Physics Informed Loss Function, and the training strategy.

3.1 Governing Equations

The PINN incorporates standard geotechnical equations:

3.1.1 Effective Stress

$$\sigma'_v(z) = \gamma' z, \tag{1}$$

where γ' is the effective unit weight.

3.1.2 End Bearing Capacity

$$Q_b = N_q \sigma'_v(L)\, A_b, \qquad A_b = \frac{\pi D^2}{4}, \tag{2}$$

$$N_q = e^{(\pi \tan\phi)} \tan^2\left(45^\circ + \frac{\phi}{2}\right). \tag{3}$$

3.1.3 Skin Friction

$$Q_s = \sum_i \sigma'_v(z_i) \tan\phi\, p\, \Delta z, \qquad p = \pi D. \tag{4}$$

3.1.4 Ultimate Capacity

$$Q_u = Q_b + Q_s. \tag{5}$$

3.2 PINN Loss Function

The training of the PINN is driven by a total loss function ($\mathscr{L}_{\text{total}}$) that is a weighted sum of several components:

$$\mathscr{L}_{\text{total}} = \lambda_{\text{data}}\mathscr{L}_{\text{data}} + \lambda_{\text{physics}}\mathscr{L}_{\text{physics}} + \lambda_{\text{con}}\mathscr{L}_{\text{con}} + \lambda_{\text{reg}}\mathscr{L}_{\text{reg}}$$

- **Data Loss ($\mathscr{L}_{\text{data}}$):** Quantifies the error between the model's prediction ($\hat{u}$) and the measured data points. This is typically a Mean Squared Error (MSE).
- **Physics Loss ($\mathscr{L}_{\text{physics}}$):** Enforces the adherence of the model's output to the known Governing Physical Laws (Partial Differential Equations or physical formulas) and Initial/Boundary Conditions. This is calculated using automatic differentiation on the NN's output.
- **Consistency Loss ($\mathscr{L}_{\text{con}}$):** This term is included to enforce specific physical constraints or consistency across the domain.
- **Total Loss Minimization:** The objective is to minimize $\mathscr{L}_{\text{total}}$ using an optimization algorithm (e.g., Adam or L BFGS), simultaneously fitting the data and satisfying the physics constraints.

Data loss uses standard mean squared error, the physics term enforces the governing equations, and consistency loss ensures:

$$\widehat{Q}_u \approx \widehat{Q}_b + \widehat{Q}_s. \tag{6}$$

A trainable friction coefficient β is constrained to $(0, 1)$ through a sigmoid activation.

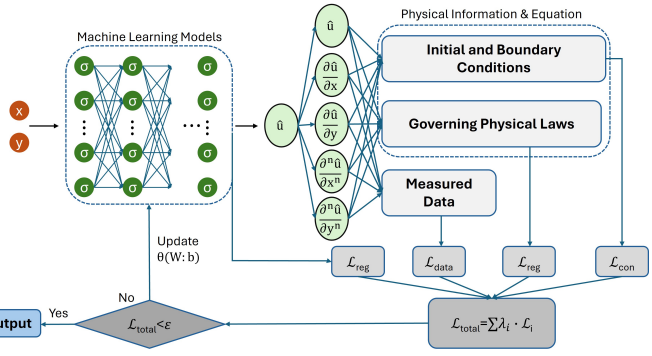

Figure 1: Schematic diagram of improved physics informed approach with measured data[11].

3.3 Benchmark Machine Learning Model

A Random Forest Regressor with tuned hyperparameters (300 trees, max depth = 12) serves as the traditional machine learning comparison for performance benchmarking against the PINN model.

3.4 Evaluation Metrics

To quantitatively evaluate the predictive performance of data driven and physics informed machine learning models, several statistical error metrics are commonly employed. Among them, the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2) are the most widely used indicators in regression based problems [14, 15].

3.4.1 Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) measures the average magnitude of prediction errors without considering their direction. It is defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i\hat{y}_i|, \tag{7}$$

where y_i denotes the observed value, $\hat{y}_i$ is the predicted value, and N is the total number of samples. MAE is robust to outliers relative to squared error metrics and provides an interpretable measure of average prediction deviation [16].

3.4.2 Root Mean Squared Error (RMSE)

The Root Mean Squared Error (RMSE) represents the square root of the average squared differences between predictions and observations:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i\hat{y}_i)^2}. \tag{8}$$

RMSE penalizes large errors more heavily than MAE due to the squaring operation and is therefore sensitive to outliers. It is commonly used in engineering prediction tasks and physical modeling because it aligns with Gaussian error assumptions [17].

3.4.3 Coefficient of Determination (R^2)

The Coefficient of Determination (R^2) assesses how well the model explains the variance in the observed data:

$$R^2 = 1\frac{\sum_{i=1}^{N}(y_i\hat{y}_i)^2}{\sum_{i=1}^{N}(y_i\bar{y})^2}, \tag{9}$$

where $\bar{y}$ is the mean of observed values. An R^2 value close to 1 indicates a strong explanatory power, while values near 0 imply weak predictive performance. Despite its interpretability, R^2 may be misleading in nonlinear and non stationary systems, and thus is often used in conjunction with MAE and RMSE [18].

4. Findings

4.1 Model Performance Comparison

A comparison with a traditional machine learning technique, Random Forest, demonstrated the superiority of the PINN approach. The Parity Plot comparing PINN vs. Random Forest shows that the blue PINN data points are generally closer to the $y = x$ line across the entire range of Q_u values compared to the orange Random Forest points. This indicates that the inclusion of physical constraints in the PINN leads to a more robust, less scattered, and more accurate prediction model, especially at higher Q_u values.

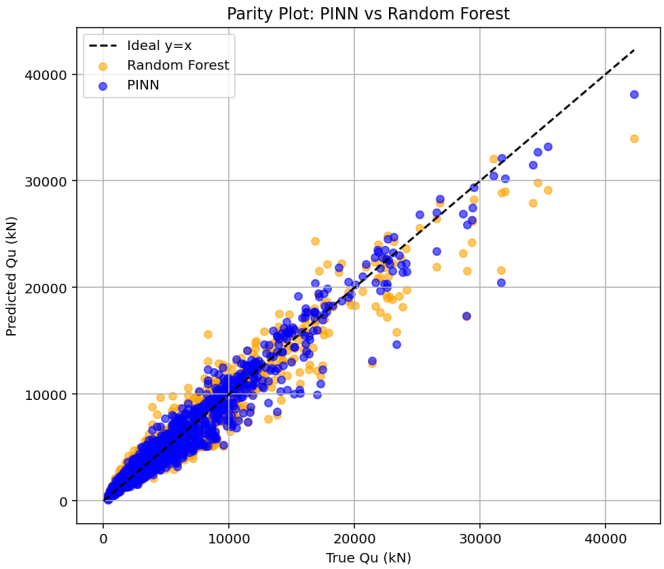

Figure 2: Model performance comparison (PINN vs RF).

4.2 Training Convergence

The PINN Training Losses graph illustrates a successful and stable training process:

- The Total Loss and Data Loss decrease significantly and quickly stabilize, demonstrating a successful fit to the measured data.
- The Physics Loss remains low and relatively constant, confirming that the model effectively respects the underlying physical laws throughout the training duration.
- The Consistency Loss drops sharply and remains at a very low magnitude (10^5 to 10^4), indicating that the physical constraints are being strongly enforced.
- The convergence of the learned β to ≈ 0.43 suggests that the physics constraints required slightly less weight than the data component to achieve the optimal solution.

The model was trained for 4,000 epochs. A key feature of this methodology is the use of a dynamic or learned weighting parameter, β, which influences the balance of the loss components. The Learned Beta History chart indicates that the β value, initialized near 0.5, rapidly converges to an optimal value around 0.43 within the first 1,500 epochs, demonstrating the effectiveness of the adaptive weighting strategy in stabilizing the training process and achieving an optimal physics data balance.

4.3 Sensitivity Analysis

The model was successfully used to conduct a sensitivity analysis across the key input parameters: Depth (h), Internal Friction Angle (ϕ), Diameter, and γ (unit weight).

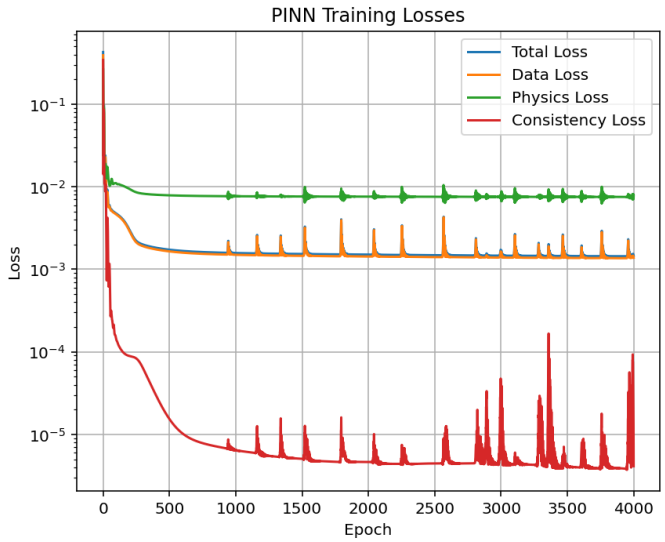

Figure 3: Training Loss Convergence

Figure 4: β Learning History

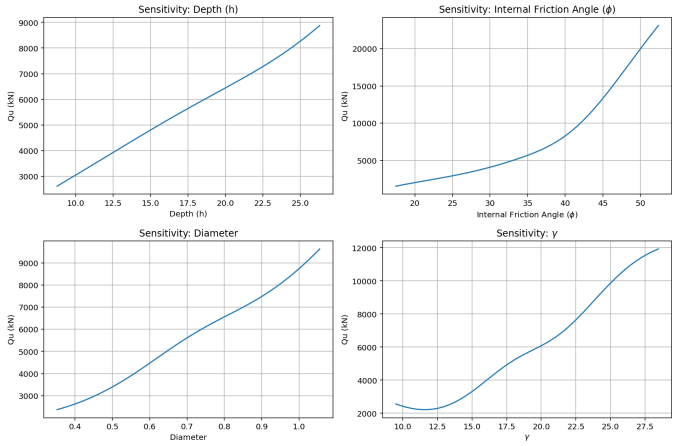

Figure 5: Sensitivity analysis of input parameters.

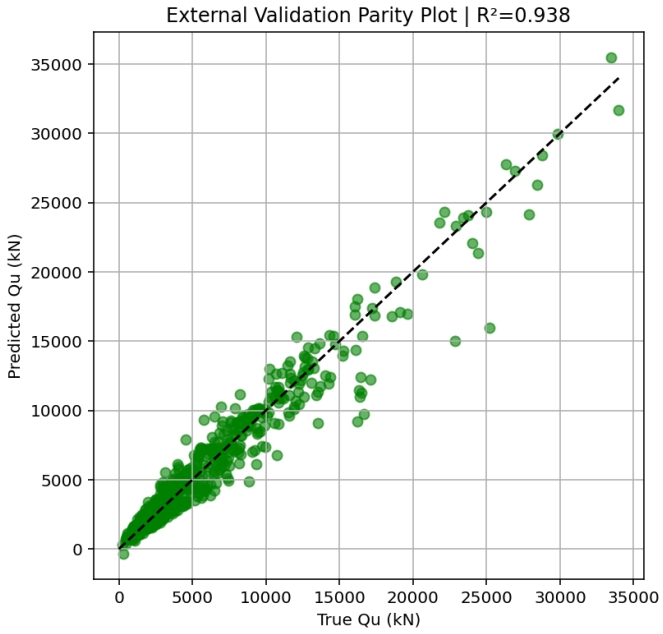

Figure 6: External Validation Plot

- Q_u exhibits a nearly linear positive relationship with Depth (h) across the observed range.
- Q_u demonstrates a strong, exponential positive correlation with the Internal Friction Angle (ϕ) and Diameter.
- The relationship with γ is complex and non linear, showing a minimum Q_u value before increasing significantly, highlighting the ability of the PINN to capture intricate non linear physical interactions.

4.4 Model Accuracy and Validation

The PINN model demonstrated high predictive accuracy on external validation data, achieving an R^2 value of 0.9383. The key external validation metrics are MAE = 966.490 kN, RMSE = 1406.940 kN, and R^2 = 0.9383. The External Validation Parity Plot shows that the predicted Q_u values cluster tightly around the ideal $y = x$ dashed line, confirming that the PINN generalizes robustly to unseen data.

5. Conclusion

This study demonstrates that the integration of geotechnical physics within a machine learning framework implemented through a Physics Informed Neural Network substantially enhances both predictive accuracy and physical reliability in estimating axial pile capacity. By embedding Meyerhof's end bearing formulation, the β method for shaft friction, and effective stress concepts directly into the learning process, the PINN is guided toward solutions that remain consistent with fundamental soil pile interaction mechanisms. This physics guided structure enables the PINN to generalize more effectively across varying soil conditions and pile geometries, capture sensitivity trends that align with established soil mechanics theory, and reduce prediction uncertainty compared with purely data driven methods. In contrast, the Random Forest model, despite performing well, lacks the ability to enforce physical constraints and therefore exhibits weaker extrapolation capability. The findings underscore that physics informed machine learning offers a powerful and transparent alternative for geotechnical prediction tasks,

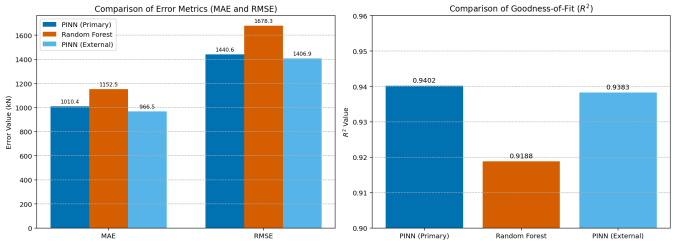

Figure 7: Quantitative Performance Comparison of PINN and Random Forest Regression

especially in applications where governing equations are well established. As such, PIML provides a promising pathway toward developing reliable, data efficient, and theoretically grounded models for deep foundation design and broader geotechnical engineering applications.

References

[1] K Terzaghi. *Theoretical Soil Mechanics*. John Wiley & Sons, 1943.

[2] GG Meyerhof. *The ultimate bearing capacity of foundations*. Geotechnique, 1951.

[3] RC Gomes, A Ramos, and A Costa. Application of machine learning techniques for predicting pile capacity. *Engineering Geology*, 293:106298, 2021.

[4] M Benbouras, M Rahman, and M Elshafie. A comparative study of machine learning methods for predicting axial pile capacity. *Computers and Geotechnics*, 136:104259, 2021.

[5] BT Pham, I Prakash, MD Nguyen, and DT Bui. Hybrid soft computing models for predicting the bearing capacity of bored piles. *Geotechnical and Geological Engineering*, 39(3):2387–2404, 2021.

[6] R Rizk, A Moussa, and M Elkasaby. Machine learning models for predicting the capacity of open-ended pipe piles. *Soils and Foundations*, 63(1):101208, 2023.

[7] A Dehghanbanadaki, A Hooshmand, and A Kord. Evaluating the performance of ml and ensemble methods for pile capacity prediction with sensitivity analysis. *Measurement*, 176:109193, 2021.

[8] K Udengaard, Y Li, and G Chen. A simple decision-tree model for predicting pile capacity using limited datasets. *International Journal of Civil Engineering*, 18(10):1089–1102, 2020.

[9] M Raissi, P Perdikaris, and GE Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear pdes. *Journal of Computational Physics*, 378:686–707, 2019.

[10] GE Karniadakis, IG Kevrekidis, L Lu, et al. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021.

[11] Biao Yuan, Chung Siung Choo, Lit Yen Yeo, Yu Wang, Zhongxuan Yang, Qingzheng Guan, Stephen Suryasentana, Jinhyun Choo, Hao Shen, Maria Megia, et al. Physics-informed machine learning in geotechnical engineering: a direction paper. *Geomechanics and Geoengineering*, pages 1–32, 2025.

[12] A Umar, S Yusuf, and A Musa. Uncertainty-informed ml models for predicting pile capacity: Bart, gpr, mars and interpretable boosting. *International Journal of Geotechnical Engineering*, 2025. In press.

[13] R Al Atroush, H El Naggar, and M Elkasabgy. Physics-informed neural network framework for modeling drilled shaft behavior. *Computers and Geotechnics*, 2025. In press.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[16] Cort J. Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30:79–82, 2005.

[17] Tae Chung. Rmse and model performance evaluation. *Journal of Hydrologic Engineering*, 2002.

[18] Nico J. D. Nagelkerke. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, 1991.

Bipul Mainali

Development and Experimental Validation of the Physics Informed Neural Network for Predicting Axial Pile Load Capac...

Tribhuvan University

Document Details

Submission ID
trn:oid:::3117:583568793

Submission Date
Apr 27, 2026, 11:31 AM GMT+5:45

Download Date
Apr 27, 2026, 11:35 AM GMT+5:45

File Name
Development and Experimental Validation of the Physics Informed Neural Network for Predictin....pdf

File Size
4.3 MB

66 Pages

13,686 Words

74,120 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

Match Groups

48 Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
2% Publications
0% Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Replaced Characters
31 suspect characters on 8 pages
Letters are swapped with similar characters from another alphabet.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.