Design Optimization of Pile Foundations

UG Project Report submitted in the partial fulfillment for the award of

Bachelor of Technology in Civil Engineering

By

Rahul Singh

20065080

Under the guidance of

Dr. Vishwajit Anand

Department of Civil Engineering

INDIAN INSTITUTE OF TECHNOLOGY

(BANARAS HINDU UNIVERSITY)

VARANASI – 221005

November, 2023

UG PROJECT CERTIFICATE

This is to certify that the thesis entitled **Design Optimization of Pile Foundations** submitted by **Rahul Singh** (Roll No. **20065080**), to the Indian Institute of Technology (Banaras Hindu University), Varanasi, in the partial fulfillment for the award of **Bachelor of Technology in Civil Engineering**, is a bona fide record of work done by him/her under my/our supervision. It is certified that the statement made by the student in his/her declaration is correct to the best of my/our knowledge.

Date of Submission: **22 November 2023**

Dr. Vishwajit Anand

Assistant Professor

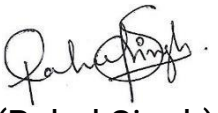
Department of Civil Engineering

IIT (BHU), Varanasi – 221005

DECLARATION BY THE CANDIDATE

I, **Rahul Singh,** certify that the work embodied in this thesis is my own bona fide work carried out by me under the supervision of **Dr. Vishwajit Anand**, from **01 January 2023** to **20 November 2023** at the Department of Civil Engineering, Indian Institute of Technology (BHU), Varanasi. The matter embodied in this thesis has not been submitted for the award of any other degree/diploma. I declare that I have faithfully acknowledged and/or cited to the researchers wherever their works have been utilized in this thesis. I further declare that I have not willfully copied any other's work, paragraphs, text, data, results, etc., reported in journals, books, magazines, reports dissertation, thesis, etc., or available at websites.

Date: **22 November 2023**



Place: **Varanasi** (Rahul Singh)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported me throughout my B.Tech project.

Firstly, I would like to thank my project guide, **Dr. Vishwajit Anand**, for their invaluable guidance, encouragement, and support throughout the project. Their expertise, patience, and willingness to share their knowledge have been instrumental in shaping this project.

I would also like to thank the HOD of the Civil Engineering Department IIT (BHU), Varanasi **Dr. Sasankasekhar Mandal,** and other faculty members of the civil engineering department, for their valuable suggestions and feedback that helped me to refine my project work.

I am grateful to my friends and classmates, who provided me with their moral support and constructive criticism whenever required. Their encouragement and motivation kept me going throughout the project.

Finally, I would like to express my gratitude to my family members, who provided me with the necessary resources and unwavering support during the project. Their love and encouragement have been the driving force behind my success.

Once again, I extend my heartfelt thanks to all those who have supported me in completing this project successfully.

ABSTRACT

Design optimization of pile foundation is a critical aspect of civil engineering, aimed at enhancing the performance of structures built on weak and compressible soil. This abstract introduces an advanced model for the design optimization of concrete pile foundations, a critical component in various civil engineering applications. In this project, we have used a dataset of 2,00,000 testcases to train the model to predict the most optimized pile foundation section with area of reinforcement for both longitudinal as well as lateral direction.

This report includes the implementation a neural network model using TensorFlow and Keras for the design optimization of concrete pile foundations. The model aims to predict key parameters related to pile design, including the diameter of the pile, area of steel reinforcement, and pitch.

The neural network architecture consists of an input layer, a hidden layer with rectified linear unit (ReLU) activation, a dropout layer for regularization, an additional hidden layer, and output layers for each predicted parameter. The model is compiled using the Adam optimizer and mean squared error loss function. Training is performed on a portion of the dataset, with cross-validation to monitor performance.

After training, the model's predictions on the test set are evaluated using various metrics, including R2 scores, explained variances, and accuracy levels within specified tolerances. The code includes visualizations such as pie charts illustrating the distribution of predictions within tolerance, histograms comparing actual and predicted values, scatter plots for each output parameter, and a bar chart depicting individual R2 scores.

This abstract introduces an advanced artificial neural network model tailored for the design optimization of concrete pile foundations. The model showcases the integration of machine learning techniques to predict critical design parameters, providing a valuable tool for engineers in the construction industry. The evaluation metrics and visualizations presented in the code offer insights into the model's performance and its ability to meet accuracy requirements within specified tolerances.

Design of the pile foundation is done considering the live load, spacing of the pile by referring the code book IS456-2000 (plain and reinforced concrete), code book IS2911:2010(Design and constructions of pile foundations), and Design Aids For Reinforced Concrete to IS: 4564978

In conclusion, this project contributes to the development of a robust and efficient design optimization model for pile foundations, which can lead to significant improvements in the performance and cost-effectiveness of structures built on weak and compressible soils.

CONTENTS

I. INTRODUCTION

The function of a Foundation is to transfer the structural loads from a building safely into the ground. The structural loads include the dead, superimposed, and wind loads. To perform the function, the foundation must be properly designed and constructed. Its stability depends upon the behavior under load of the soil on which it resists, and this is affected partly by the design of the foundation and partly by the characteristics of the soil.
It is necessary for the design and construction of the foundation to pay attention to the nature and strength of the materials to be used for the foundations as well as the likely behavior under load of the soils on which the foundation rests.

I.1. Foundations Classifications

On the selection of a suitable foundation system for a building, various factors must be taken into consideration. Among them are soil conditions, load transfer patterns, shape and size of the building, site constraints, underground tunnels and services, environmental issues, etc. There are two basic types of foundations:

(1) Shallow Foundation: Those that transfer loads in bearing close to the surface. They either form individual spread footings or mat foundations, which combine the individual footings to support an entire building or part of it. The two systems may also act in combination with each other, for example, where a service core is seated on a large mat while the columns are founded on pad footings. In shallow foundation, the below criteria is satisfied.

$$D_f \leq B_f$$

where D_f means the depth of the foundation

B_f means the width of the foundation

(2) Deep Foundations: The type of foundations distinguished from shallow foundations by the depth they are embedded into the ground. A deep foundation is a type of foundation that transfers building loads to the earth farther down from the surface than a shallow foundation does to a subsurface layer or a range of depths. There are many reasons a civil engineer would recommend a deep foundation over a shallow

foundation, but some of the common reasons are very large design loads, a poor soil at shallow depth, or site constraints (like property lines). There are different terms used to describe different types of deep foundations including piles, drilled shafts, caissons and piers. In a nut shell, a Foundation is deep if its depth is greater than the width.

$$D_f > B_f$$

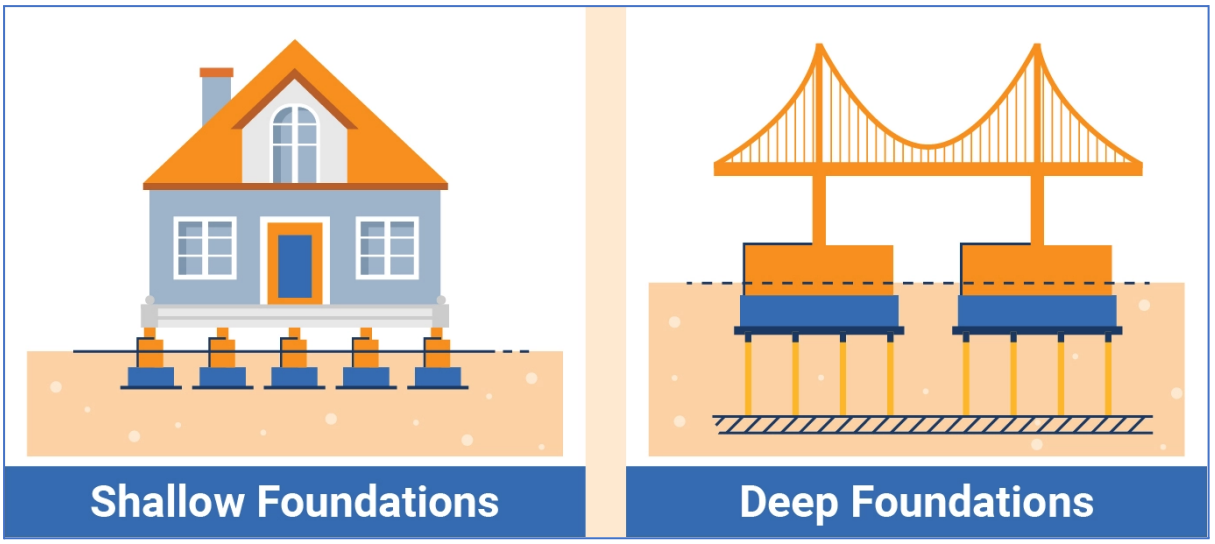


Fig-1. Shallow and Deep foundation diagram

The deep foundations are of three types: -

-> Well foundation

-> Caisson foundation

-> Pile foundation

I.2. Pile Foundation

The word 'pile' is used to describe columns, usually of reinforced concrete, driven into or cast in the ground in order to carry foundation loads to some deep underlying firm stratum or to transmit loads to the subsoil by the friction of their surfaces in contact with the subsoil. The main function of a pile is to transmit loads to lower levels of ground by a combination of friction along their sides and end bearing at the pile point or base as show in the figure below.

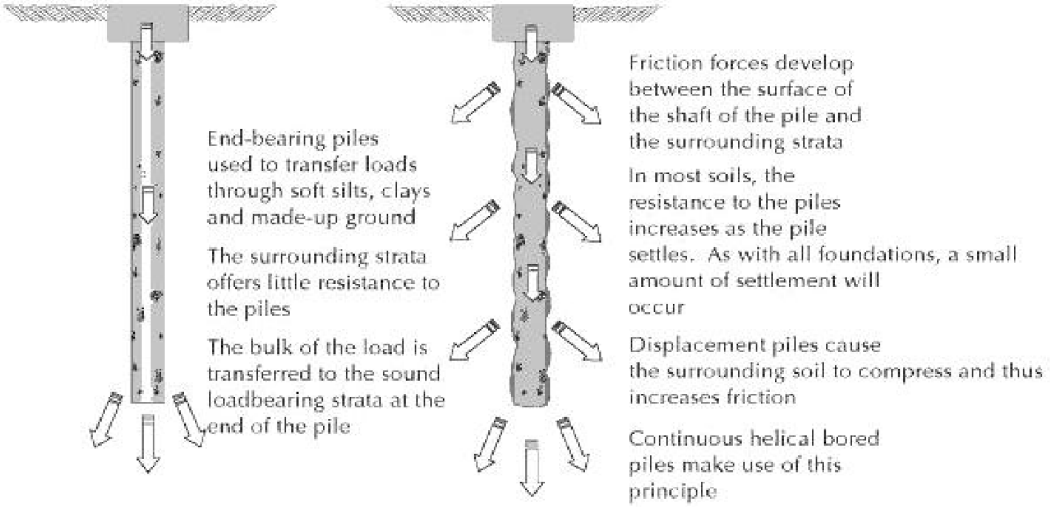


Fig-2 Diagram showing the type of loads acting on a pile

I.2.i. Purpose of Pile Foundation

1. The main purpose of a pile foundation is to transfer the loads into a strong stratum.
2. Generally, pile foundations are used when the bearing capacity of soil is very low and the structural load is heavy.
3. Compressible soil and waterlogged soil is ideal for this type of foundations.
4. Piles are mostly suitable for the foundation of high-rise buildings, bridges, piers, docks etc.

I.2. ii. Classification of Pile Foundation

Depending upon the function, material and method of installation pile foundation can be further classified into following categories;

-> Classification of piles based on load bearing mechanism

The method of load-bearing arrangement is considered here to separate the types of deep foundations. Mainly, there are two methods that we can use to classify the piles.

-> End Bearing Pile: Most types of piles we construct are end-bearing piles. The toe of the pile act to resist the vertical movement of the pile. The penetration resistance of the pile toe is considered the end bearing. Pile is rested on the soil, weather rock, or fresh rock. Depending on the condition of the resting material and toe area, the end-bearing

capacity is varying. The rock produces the highest-end bearing pressure whereas the soil produces the low-end bearing pressure.

The pile end bearing can be calculated as,

$$F_b = f_b A_b$$

F_b – Pile end bearing resistance. A suitable factor of safety shall be applied.
f_b – allowable end bearing, not be obtained from laboratory investigation of soil or rock that rests the pile.
A_b – Pile toe area

Friction Pile: Most of the driven pile types are friction piles. The friction between the soil and the pile creates the penetration resistance of the pile. This is called *friction. The* surface area of the pile and soil overburden, type of soil, etc. are affecting the skin friction of the pile. The skin friction of the pile can be calculated as

$$F_s = f_s A_s$$

F_s – allowable skin friction of the soil/rock
f_s – allowable skin friction stress (from recommendations)
A_s – pile surface area of that section considered

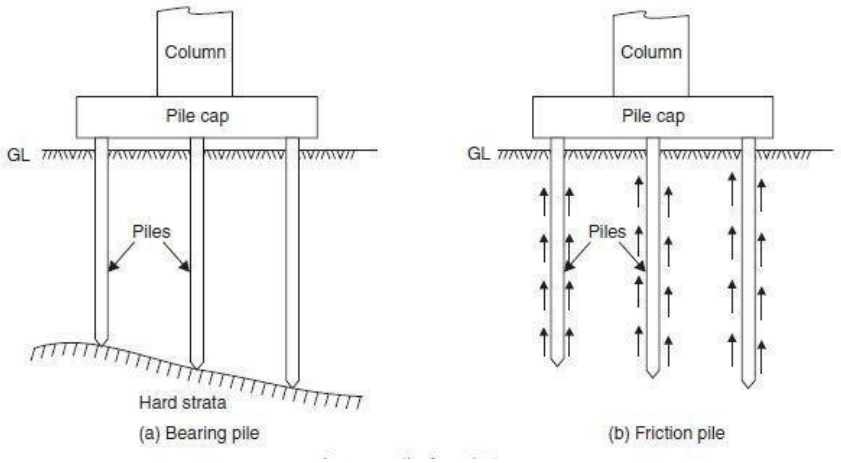


Fig-3. Classification of pile on the basis of load mechanism

Classification of piles based on construction material

-> **Timber pile:** One of the oldest types of piles is timber piles. They were constructed in ancient times. Not only the construction on the land but also the construction of a structure on water, and timber piles are used. Types of wood that have resistance against deterioration timber due to environmental conditions are used for the construction.
-> Concrete pile: One of the most frequently used types of pile foundations is concrete piles. It has a higher load-bearing capacity when

compared to the other types of piles as the concrete is very strong in compression. Generally, the concrete pile is constructed as friction piles and end-bearing piles. The precast pile is the friction pile and cast in situ bored piles are classified as end-bearing piles. Further, precast piles are also called driven piles as they cost about the ground and then they later driven into a hard soil layer. Mechanical methods or vibration hammers are used to drive the pile.

-> Sheet piles: Sheet piles are not the types of pile foundations used for load-bearing purposes like concrete piles. They are mostly used to resist lateral loads. In shoring construction, permanent sheet pile walls to resist ground movement, etc. are the main uses of sheet piles. The nature of the pile has increased its lateral stiffness. Further, they are tied together by locking each other.

-> Steel piles: Steel piles are used on special occasions, and they are not commonly used due to corrosion issues. If the pile is corroded, it will affect structural stability. It is common practice to check the ground condition and groundwater for harmful chemicals that could cause corrosion. Having those will reduce the life span of the structure. Therefore, steel pile construction shall be done with much care.

-> Composite piles: Generally, composite piles are a combination of steel and concrete. When we constructed a steel pile covered with concrete it also can be considered a composite pile. However, when we construct the steel pile, we used concrete cover around the pile to protect it against durability issues.

->Secant piles: Secant piles are the types of foundations mostly popular in deep basement excavations. When it is required to construct a deep basement without affecting the surrounding structures, we mostly used these types of pile foundations. If the groundwater table is lowered, it will directly affect the structures built on the shallow foundation. In addition, continuous dewatering in a deep excavation is almost impossible without affecting the construction.

I.3. Static Formula for calculation of bearing force of soil

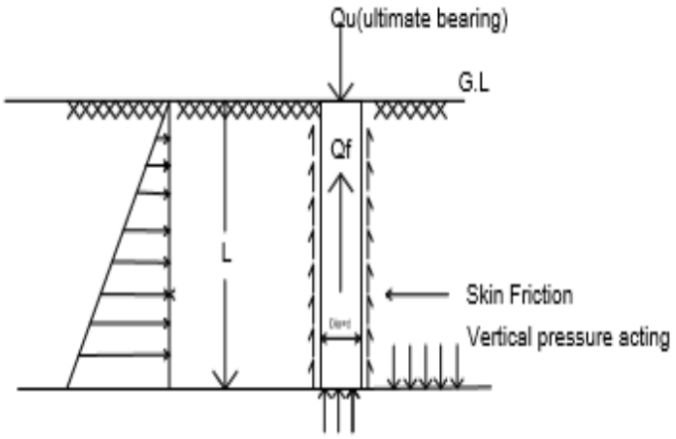


Fig-4. Analysis of pile foundation

$Qu = Q_b + Q_f$

$Q_u = q_b \times A_b + q_{sf} \times A_f$

where Qu – Ultimate bearing capacity of soil

Q_b – Resistance offered by hard strata

Q_f – Frictional resistance offered by friction

q_b – unit point or toe resistance

A_b – Area of cross-section of pile on which bearing resistance acts

q_{sf} – average skin friction

A_f – surface area of pile upon which the skin friction acts

$q_b = CN_c + 0.3\gamma BN_\gamma + \sigma' N_q$

where $\sigma' = \gamma L$

(B is very small, so Neglected)

$q_b = C + \sigma' N_q$

$A_b = \pi d^2 / 4$

$Q_f = (\alpha C + \sigma \tan \delta)\, \pi dl$

For sandy soil, C=0,

$Q_U = \gamma L N_q \times \pi d^2 / 4 + K\gamma L / 2 \tan \delta \times \pi dl$

For clay soil, $\varphi = 0$

$N_q = 1$; $N_c = 9$ (Skempton's theory)

For concrete pile, $\delta = 3/4\, \varphi$ ($\delta = 0$)

$Q_U = 9C \times \pi d2 / 4 + \alpha C\, \pi dl$

II. Literature Review

Pile foundations are commonly used to support structures on weak and compressible soil. The design of pile foundations involves the determination of the appropriate size, shape, and number of piles required to support the loads from the structure and to minimize the potential for settlement. Design optimization of pile foundation is an essential aspect of civil engineering that aims to enhance the performance of the foundation system while minimizing the cost and construction time.

In recent years, there has been a significant interest in the development of design optimization models for pile foundations. These models typically employ a combination of analytical and numerical techniques, such as finite element analysis and genetic algorithms, to optimize the design parameters of the pile foundation system. Several studies have focused on the development of such models for different types of structures and soil conditions.

One of the primary challenges in pile foundation design optimization is the consideration of multiple design criteria, such as load capacity, settlement, and constructability. Several researchers have proposed various approaches to address this challenge.

Wang et al. (2020) proposed a multi-objective optimization model that considers the load capacity, settlement, and construction cost of the pile foundation. The model employs the non-dominated sorting genetic algorithm (NSGA-II) to generate a set of Pareto-optimal solutions that balance the competing objectives.

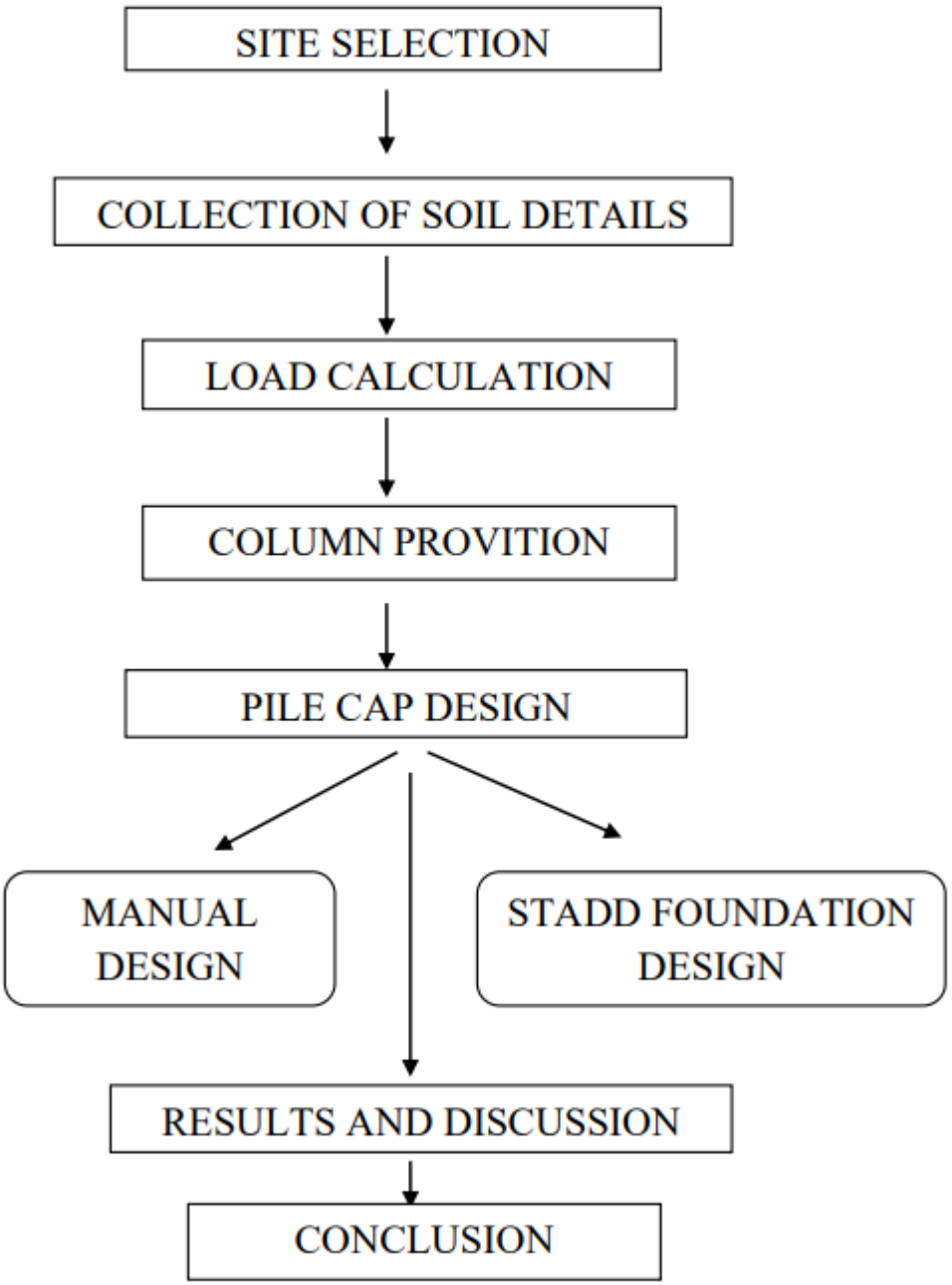
In India, the Bureau of Indian Standards (BIS) provides guidelines for the design of pile foundations in different soil conditions. These guidelines include the use of various methods, such as static load tests, dynamic load tests, and pile driving formulae, to estimate the load capacity and settlement of pile foundations. BIS also recommends the use of safety factors to account for uncertainties in soil properties and loading conditions.

Thakur et al. (2018) proposed a design optimization model for pile foundations using finite element analysis and response surface methodology. The model considers the load capacity, settlement, and

construction cost of the pile foundation and optimizes the design parameters to minimize the total cost.

In conclusion, design optimization of pile foundation is a critical aspect of geotechnical engineering that has attracted significant attention in recent years. The development of robust and efficient design optimization models that consider multiple design criteria and uncertainties can lead to significant improvements in the performance and cost-effectiveness of pile foundation systems.

III. METHODOLOGY



The design of pile foundations has a lot of aspects. In this report, we are mainly going to see the structural design aspects of pile foundation of concrete piles. The structural design of the pile typically involves determining the required cross-sectional dimensions, such as the diameter and wall thickness, area of lateral and longitudinal reinforcement require to ensure that the pile can resist the applied loads

and moments. A single pile can be designed considering it as a RCC column subjected to axial load and biaxial bending.

III.1. Structural design of pile doing manual calculation

The structural design of a pile foundation involves several steps, including determining the required pile size, estimating the load capacity of the pile, and designing the pile to resist the applied loads. These calculations can be done using manual methods, which involve using various equations and formulas to estimate the required parameters.

Manual calculations can be time-consuming and prone to errors, but they are still used in some cases where the design is relatively simple or when there is limited access to computer software. However, the use of software and computer-based tools has become increasingly common in recent years, allowing for faster and more accurate pile design calculations.

Problem: Design a pile under a column transmitting an axial load of 400 kN and moment in x as 120 KNm and moment in y as 130 kNm. The pile is to be driven to a hard stratum available at a depth of 6m.

Solution:

Step-1.

Given
Load: 400 kN
Moment in x = 120 kNm
Moment in y = 130 kNm

Factored Load: 1.5 x 400 = 600 kN
Factored Moment in x: 1.5 x 150 = 180 kNm
Factored Moment in y: 1.5 x 130 = 195 kNm

Use M25 concrete and Fe415 steel
Let the length of the pile above ground, including pile capacity = 0.6 m
Total length of pile = 6.6 m
Let us take the diameter of pile as 600 mm X 600 mm

Slenderness ratio (λ) = $Leff/LLD$

λ = (0.8*6.6)/0.6 = 8.8 $\leq$ 12

Step-2. Minimum eccectrity

As per IS 456: 2000, clause 25.4, all columns shall be designed for minimum eccentricity, equal to the summation of the unsupported length of column divided by 500 and lateral dimensions divided by 30, subject to a minimum of 20 mm.

e_{min} = max(l/500 + D/30 or 20mm)

$e_{min\ x-x}$ = $e_{min\ y-y}$ = max(6600/500 + 600/30, 20) = 33.2 mm

Step-3. Minimum Bending Moment

$M_{x-x} = M_{y-y} = 600 \times \frac{33.2}{1000}$ = 19.92 kNm

$M_{ux} \geq M_{x-x}$

$M_{uy} \geq M_{y-y}$

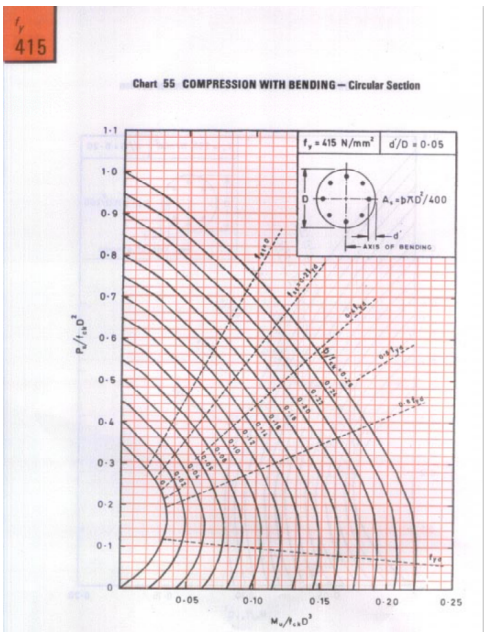
Hence, we have to design it for axial load with biaxial bending moment

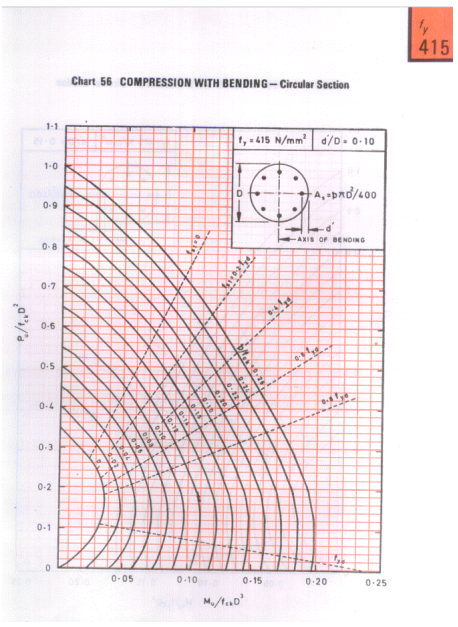
Step-4. Finding M_{ux} and M_{uy}

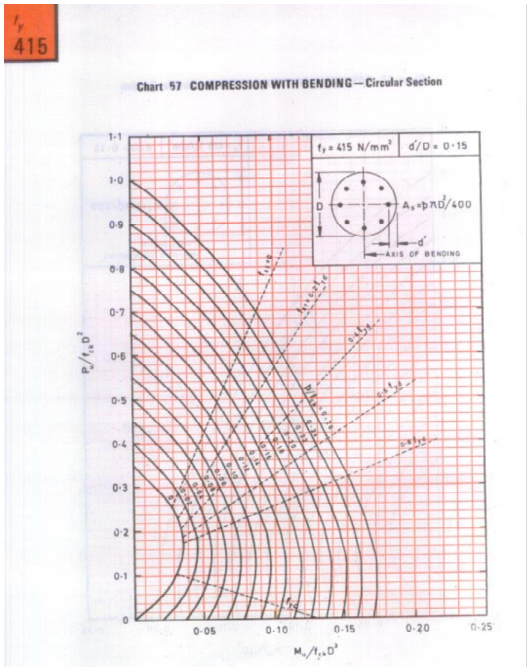
Lets assume an effective cover of 60mm

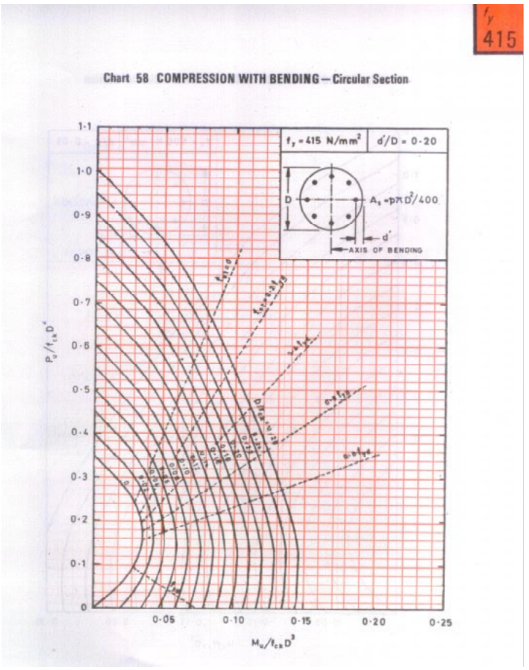
$$\frac{d}{D} = \frac{60}{600} = 0.1$$

Now, we are going to use charts-55,56,57,58 DESIGN AIDS FOR REINFORCED CONCRETE TO IS : 456- 1978









As d/D = 0.1 we are going to use chart - 56

y-value = $\frac{P}{fckD*D}$ = (600 X 10^3) / (25 X 600^2) = 0.0667

Lets assume %p be 1.4%.

k-value = (%p)/fck = 1.4/25 = 0.056

Now using interpolation x-value = 0.07

$M_{ux}/fckD^3$ = 0.07

M_{ux} = 378kNm

M_{uy} = 378kNm

39.6 Members Subjected to Combined Axial Load and Biaxial Bending

The resistance of a member subjected to axial force and biaxial bending shall be obtained on the basis of assumptions given in **39.1** and **39.2** with neutral axis so chosen as to satisfy the equilibrium of load and moments about two axes. Alternatively such members may be designed by the following equation:

$$\left[\frac{M_{ux}}{M_{ux1}}\right]^{\alpha_n} + \left[\frac{M_{uy}}{M_{uy1}}\right]^{\alpha_n} \leq 1.0$$

where

M_{ux}, M_{uy} = moments about x and y axes due to design loads,

M_{ux1}, M_{uy1} = maximum uniaxial moment capacity for an axial load of P_u, bending about x and y axes respectively, and

α_n is related to P_u/P_{uz}

where $P_{uz} = 0.45 f_{ck} \cdot A_c + 0.75 f_y \cdot A_{sc}$

For values of P_u/P_{uz} = 0.2 to 0.8, the values of α_n vary linearly from 1.0 to 2.0. For values less than 0.2, α_n is 1.0; for values greater than 0.8, α_n is 2.0.

$A_s = (p*\pi*D^2) /400 = 3958.4\ mm^2$

$P_z = 0.45fckA_c + 0.75fyA_s$
$P_z = 0.45 * 25 * (\pi*600^2 - 3958.4) + 0.75 * 415 * 3958.4 = 13910.97kN$

$P_u/P_z = 600/13910.97 = 0.04$

Value of $\alpha_n = 1$

Now, lets check for the condition

$(180/378 + 175/378) = 0.93 \leq 1$

Hence, design is safe

Now, provide 8 bars of 25mm diameter as main bar

Step-5: Design for pitch

39.4 Compression Members with Helical Reinforcement

The strength of compression members with helical reinforcement satisfying the requirement of **39.4.1** shall be taken as 1.05 times the strength of similar member with lateral ties.

39.4.1 The ratio of the volume of helical reinforcement to the volume of the core shall not be less than $0.36 (A_g/A_c - 1) f_{ck}/f_y$

where

A_g = gross area of the section,

A_c = area of the core of the helically reinforced column measured to the outside diameter of the helix,

f_{ck} = characteristic compressive strength of the concrete, and

f_y = characteristic strength of the helical reinforcement but not exceeding 415 N/mm².

Lets assume the diameter of spiral reinforcement as 10mm
For pitch the following condition should be satisfied

$0.36 * (fck/fy) * (\frac{Ag}{Ac} - 1) \leq \frac{Vh}{Vc}$

where $Ag = \pi * 600 * 600/4$

Ac = $\pi * dc * dc/4$ where dc = D - 2*d = 600 - 2*60 = 480

Vc = Ac * 1000

Vh = $\pi * 10 * 10/4$ * (1000/p) * π * (480 - 10)

After solving we get p ≥ 33.62mm

d) *Helical reinforcement*

1) *Pitch*—Helical reinforcement shall be of regular formation with the turns of the helix spaced evenly and its ends shall be anchored properly by providing one and a half extra turns of the spiral bar. Where an increased load on the column on the strength of the helical reinforcement is allowed for, the pitch of helical turns shall be not more than 75 mm, nor more than one-sixth of the core diameter of the column, nor less than 25 mm, nor less than three times the diameter of the steel bar forming the helix. In other cases, the requirements of **26.5.3.2** shall be complied with.

2) The diameter of the helical reinforcement shall be in accordance with **26.5.3.2** (c) (2).

Max pitch = min(75mm, dc/6) = 75mm

Min pitch max(25mm, 3*10) = 25mm

Hence, taking spiral reinforcement of 10 mm diameter having a pitch of 35mm.

III.2. Dataset generation

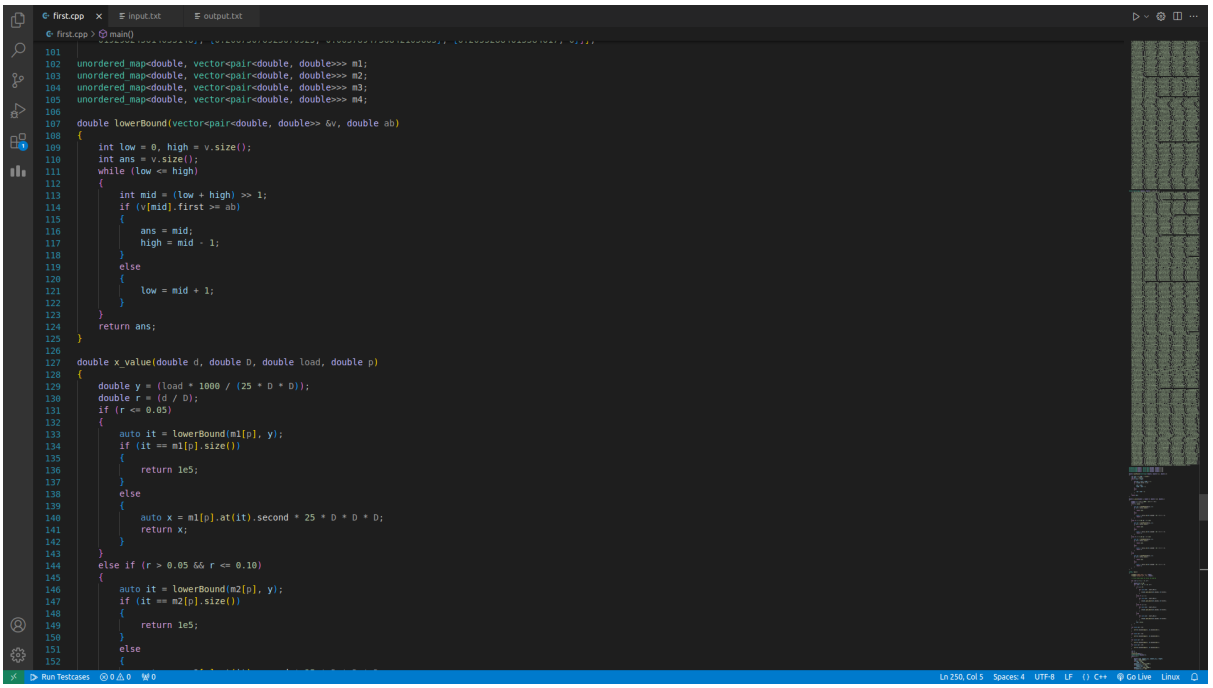
In the design process of a pile foundation, accurate and efficient calculations are essential to ensure that the foundation system can safely and effectively support the applied loads.

In the pursuit of advancing engineering solutions in the realm of civil infrastructure, a meticulously crafted dataset of 2 lakh random inputs for pile foundation design has been generated using a sophisticated C++ codebase.

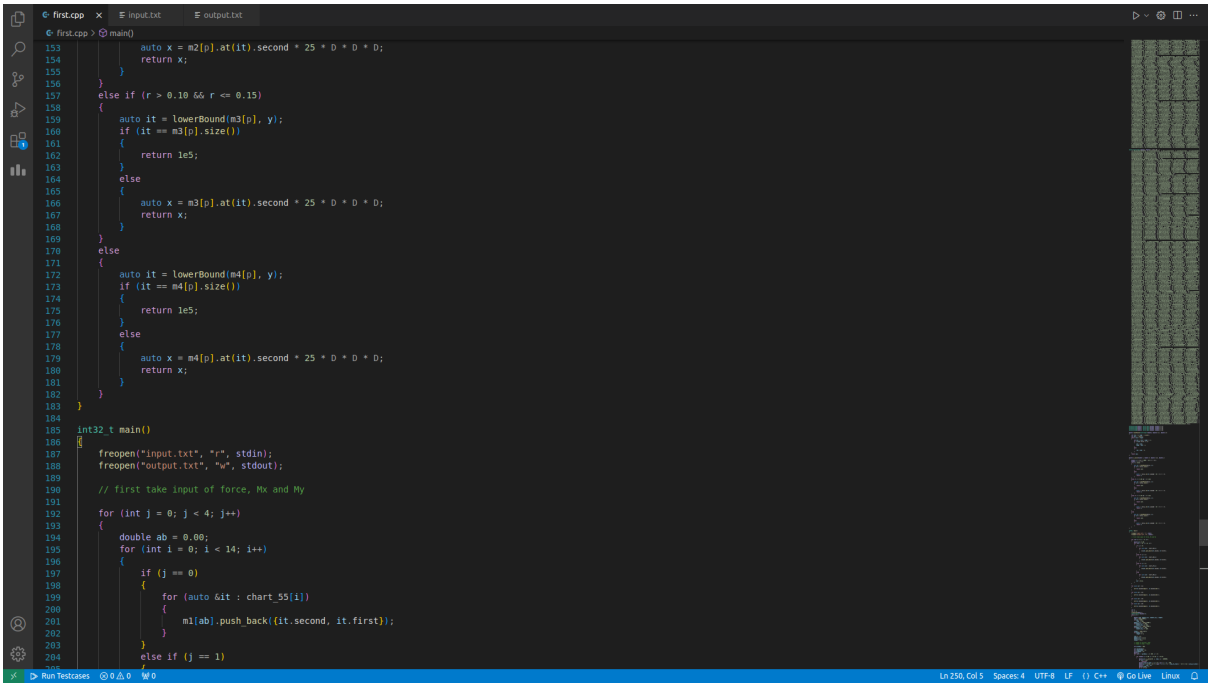
The C++ code intelligently simulates various scenarios and conditions that mimic real-world situations encountered in the construction of pile foundations. The dataset encompasses a comprehensive set of

parameters essential for pile design, meticulously curated to provide a holistic representation of the factors influencing foundation behavior.

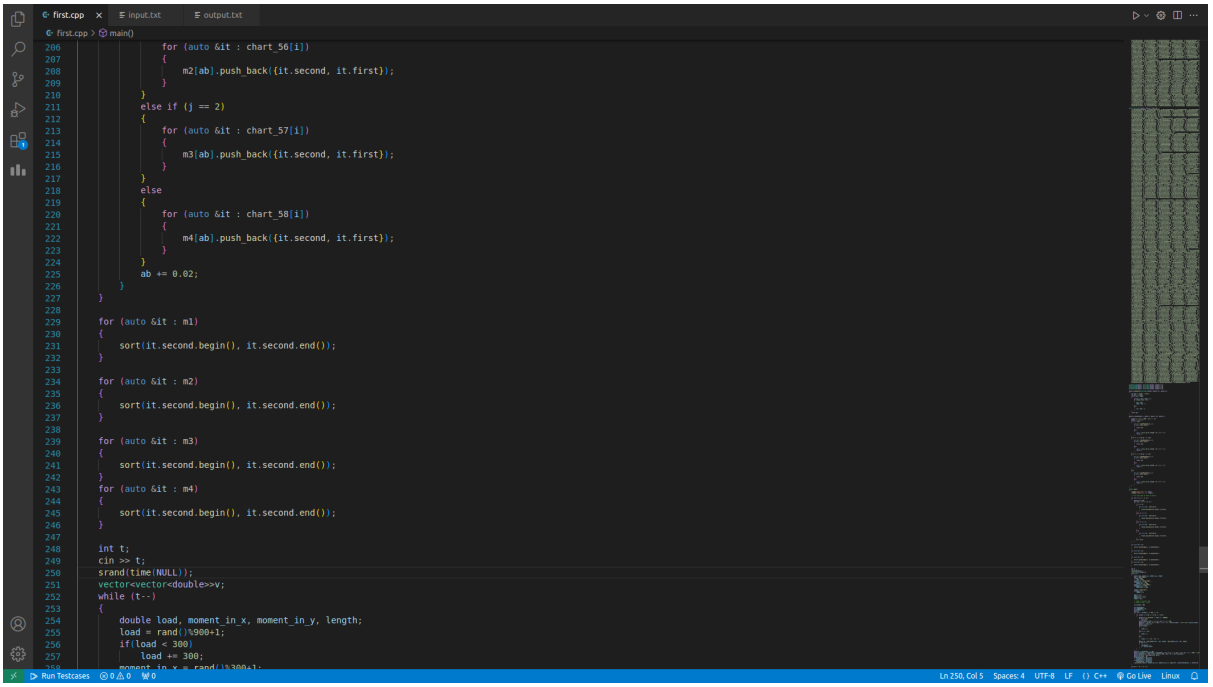
This part involves the digitization of charts of SP - 16. Using them to get outputs for randomly generated inputs. The input fields are Load, Moment in x (in kNm), Moment in y (int kNm) and the output fields are Diameter of pile (in m), Area of steel (in mm^2), and pitch of spiral reinforcement (in mm).



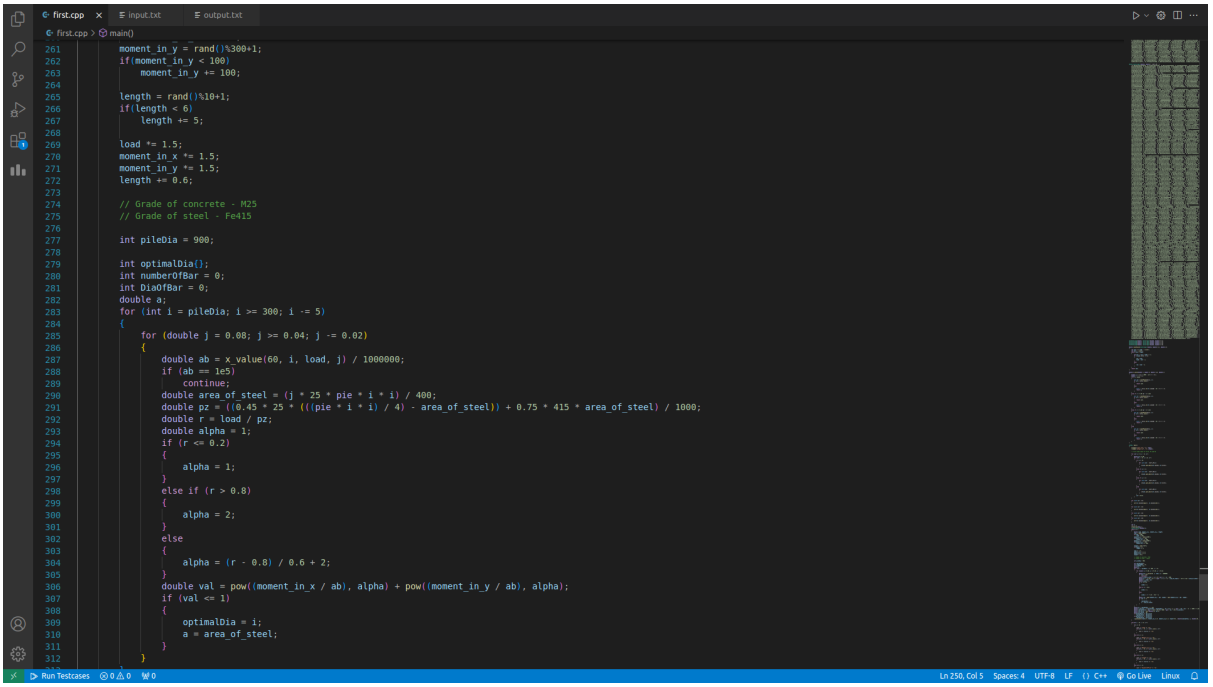
```cpp
unordered_map<double, vector<pair<double, double>>> m1;
unordered_map<double, vector<pair<double, double>>> m2;
unordered_map<double, vector<pair<double, double>>> m3;
unordered_map<double, vector<pair<double, double>>> m4;

double lowerBound(vector<pair<double, double>> &v, double ab)
{
    int low = 0, high = v.size();
    int ans = v.size();
    while (low <= high)
    {
        int mid = (low + high) >> 1;
        if (v[mid].first >= ab)
        {
            ans = mid;
            high = mid - 1;
        }
        else
        {
            low = mid + 1;
        }
    }
    return ans;
}

double x_value(double d, double D, double load, double p)
{
    double y = (load * 1000 / (25 * D * D));
    double r = (d / D);
    if (r <= 0.05)
    {
        auto it = lowerBound(m1[p], y);
        if (it == m1[p].size())
        {
            return 1e5;
        }
        else
        {
            auto x = m1[p].at(it).second * 25 * D * D * D;
            return x;
        }
    }
    else if (r > 0.05 && r <= 0.10)
    {
        auto it = lowerBound(m2[p], y);
        if (it == m2[p].size())
        {
            return 1e5;
        }
        else
        {
```



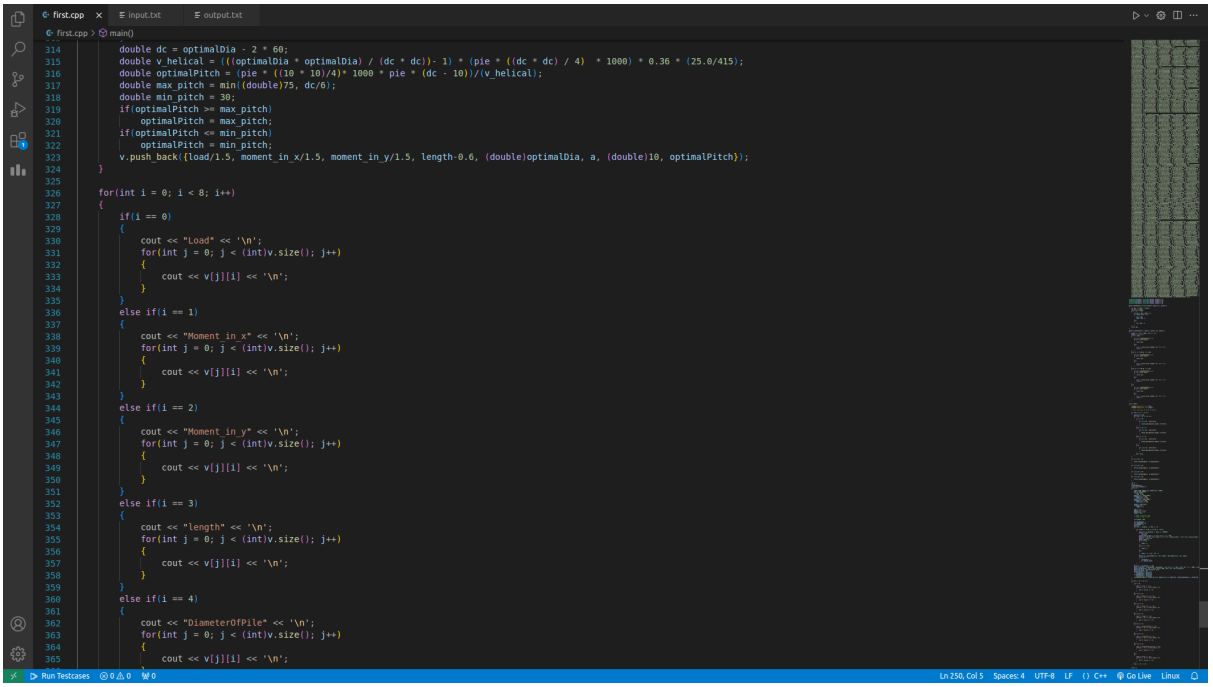
```cpp
            auto x = m2[p].at(it).second * 25 * D * D * D;
            return x;
        }
    }
    else if (r > 0.10 && r <= 0.15)
    {
        auto it = lowerBound(m3[p], y);
        if (it == m3[p].size())
        {
            return 1e5;
        }
        else
        {
            auto x = m3[p].at(it).second * 25 * D * D * D;
            return x;
        }
    }
    else
    {
        auto it = lowerBound(m4[p], y);
        if (it == m4[p].size())
        {
            return 1e5;
        }
        else
        {
            auto x = m4[p].at(it).second * 25 * D * D * D;
            return x;
        }
    }
}

int32_t main()
{
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    // first take input of force, Mx and My

    for (int j = 0; j < 4; j++)
    {
        double ab = 0.00;
        for (int i = 0; i < 14; i++)
        {
            if (j == 0)
            {
                for (auto &it : chart_55[i])
                {
                    m1[ab].push_back({it.second, it.first});
                }
            }
            else if (j == 1)
```



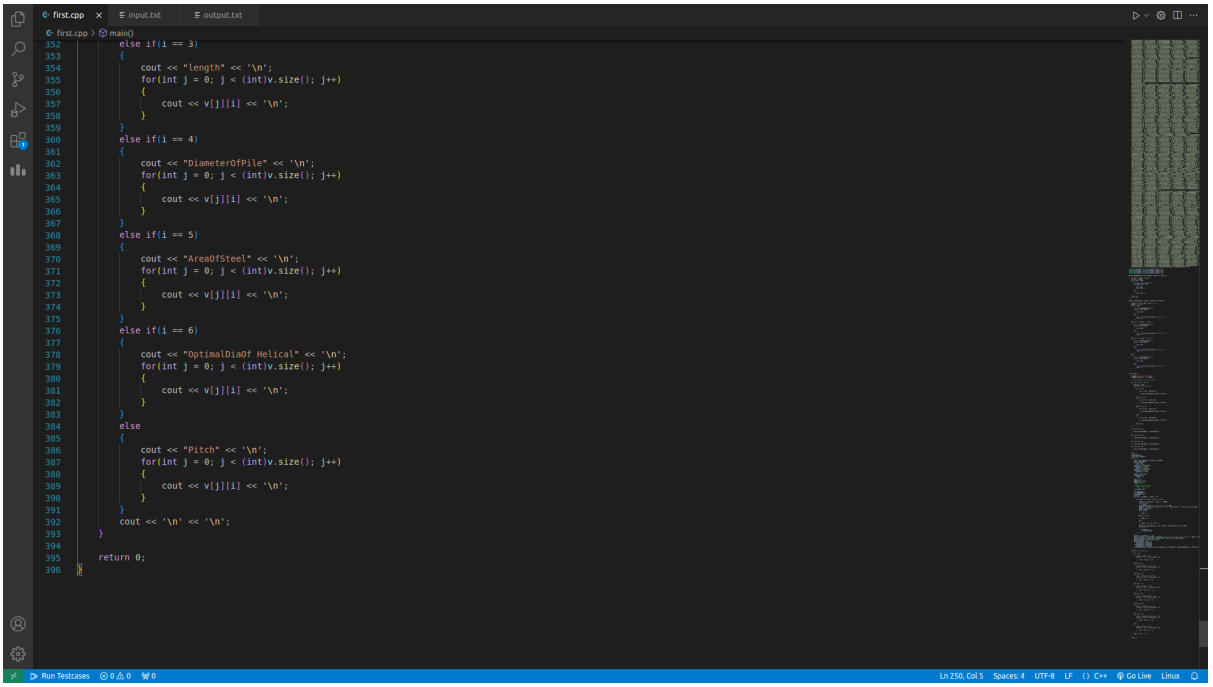
```
                for (auto &it : chart_56[i])
                {
                    m2[ab].push_back({it.second, it.first});
                }
            }
            else if (j == 2)
            {
                for (auto &it : chart_57[i])
                {
                    m3[ab].push_back({it.second, it.first});
                }
            }
            else
            {
                for (auto &it : chart_58[i])
                {
                    m4[ab].push_back({it.second, it.first});
                }
            }
            ab += 0.02;
        }
    }

    for (auto &it : m1)
    {
        sort(it.second.begin(), it.second.end());
    }

    for (auto &it : m2)
    {
        sort(it.second.begin(), it.second.end());
    }

    for (auto &it : m3)
    {
        sort(it.second.begin(), it.second.end());
    }
    for (auto &it : m4)
    {
        sort(it.second.begin(), it.second.end());
    }

    int t;
    cin >> t;
    srand(time(NULL));
    vector<vector<double>>v;
    while (t--)
    {
        double load, moment_in_x, moment_in_y, length;
        load = rand()%900+1;
        if(load < 300)
            load += 300;
        moment_in_x = rand()%300+1;
```



```
        moment_in_y = rand()%300+1;
        if(moment_in_y < 100)
            moment_in_y += 100;

        length = rand()%10+1;
        if(length < 6)
            length += 5;

        load *= 1.5;
        moment_in_x *= 1.5;
        moment_in_y *= 1.5;
        length += 0.6;

        // Grade of concrete - M25
        // Grade of steel - Fe415

        int pileDia = 900;

        int optimalDia{};
        int numberOfBar = 0;
        int DiaOfBar = 0;
        double a;
        for (int i = pileDia; i >= 300; i -= 5)
        {
            for (double j = 0.08; j >= 0.04; j -= 0.02)
            {
                double ab = x_value(60, i, load, j) / 1000000;
                if (ab == 1e5)
                    continue;
                double area_of_steel = (j * 25 * pie * i * i) / 400;
                double pz = ((0.45 * 25 * (((pie * i * i) / 4) - area_of_steel)) + 0.75 * 415 * area_of_steel) / 1000;
                double r = load / pz;
                double alpha = 1;
                if (r <= 0.2)
                {
                    alpha = 1;
                }
                else if (r > 0.8)
                {
                    alpha = 2;
                }
                else
                {
                    alpha = (r - 0.8) / 0.6 + 2;
                }
                double val = pow((moment_in_x / ab), alpha) + pow((moment_in_y / ab), alpha);
                if (val <= 1)
                {
                    optimalDia = i;
                    a = area_of_steel;
                }
            }
```



```
        double dc = optimalDia - 2 * 60;
        double v_helical = (((optimalDia * optimalDia) / (dc * dc))- 1) * (pie * ((dc * dc) / 4)  * 1000) * 0.36 * (25.0/415);
        double optimalPitch = (pie * ((10 * 10)/4)* 1000 * pie * (dc - 10))/(v_helical);
        double max_pitch = min((double)75, dc/6);
        double min_pitch = 30;
        if(optimalPitch >= max_pitch)
            optimalPitch = max_pitch;
        if(optimalPitch <= min_pitch)
            optimalPitch = min_pitch;
        v.push_back({load/1.5, moment_in_x/1.5, moment_in_y/1.5, length-0.6, (double)optimalDia, a, (double)10, optimalPitch});
    }

    for(int i = 0; i < 8; i++)
    {
        if(i == 0)
        {
            cout << "Load" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        else if(i == 1)
        {
            cout << "Moment_in_x" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        else if(i == 2)
        {
            cout << "Moment_in_y" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        else if(i == 3)
        {
            cout << "length" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        else if(i == 4)
        {
            cout << "DiameterOfPile" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
```

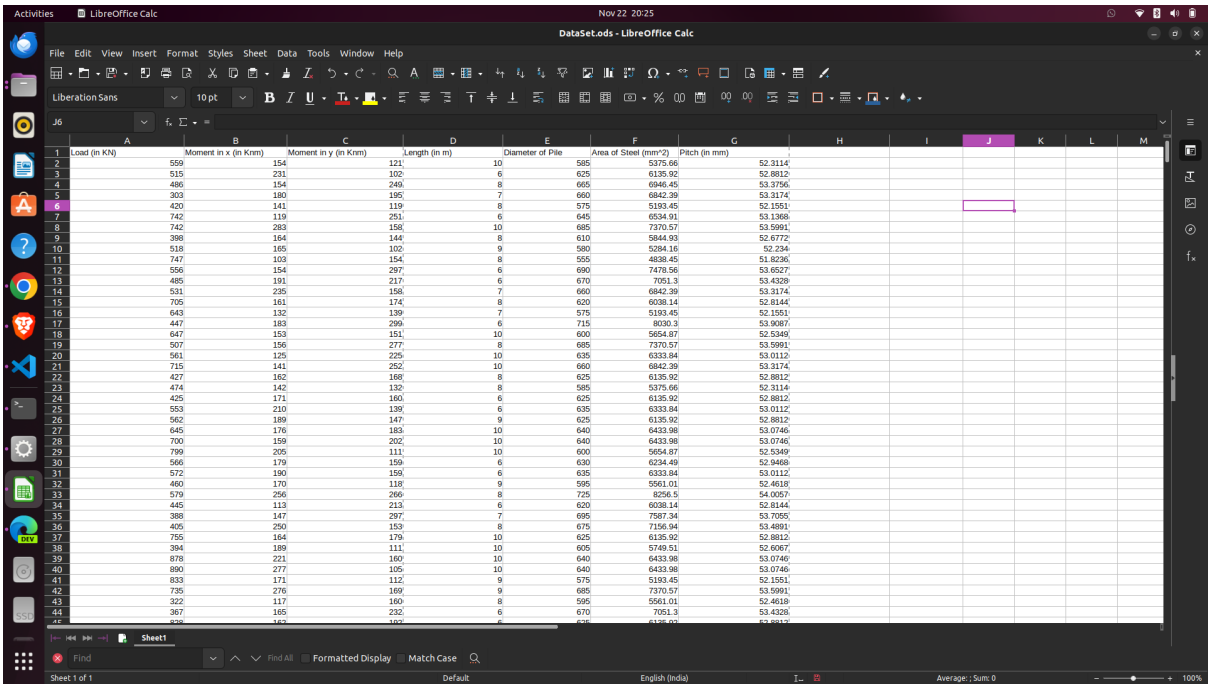


```
        else if(i == 3)
        {
            cout << "length" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        else if(i == 4)
        {
            cout << "DiameterOfPile" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        else if(i == 5)
        {
            cout << "AreaOfSteel" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        else if(i == 6)
        {
            cout << "OptimalDiaOf Helical" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        else
        {
            cout << "Pitch" << '\n';
            for(int j = 0; j < (int)v.size(); j++)
            {
                cout << v[j][i] << '\n';
            }
        }
        cout << '\n' << '\n';
    }

    return 0;
}
```

Once the test cases have been generated, the dataset is used to test the design criteria under various conditions. The test results can be analysed to determine if the design criteria are accurate and if any changes need to be made to improve the design process. The dataset can also be used to perform statistical analyses, such as regression analyses, to determine the sensitivity of the design criteria to the input parameters.

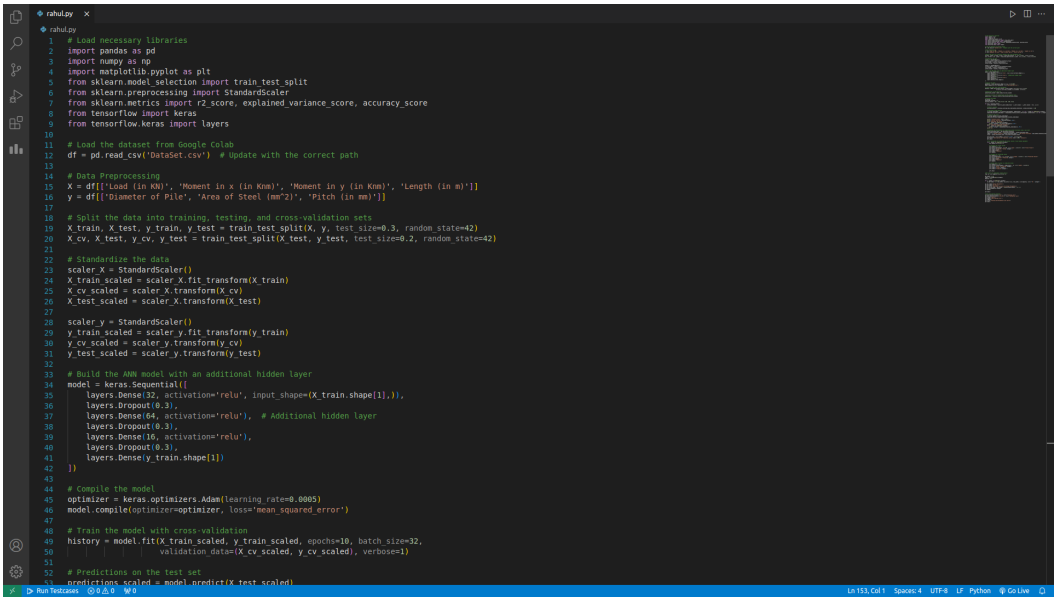
By testing the design criteria using a large dataset of test cases, the accuracy and reliability of the design process can be improved. This can help to ensure that the pile foundation can safely and effectively support the applied loads and provide long-lasting stability to the structure.

After generation of the test cases, it is converted to .csv file to train our ML model.

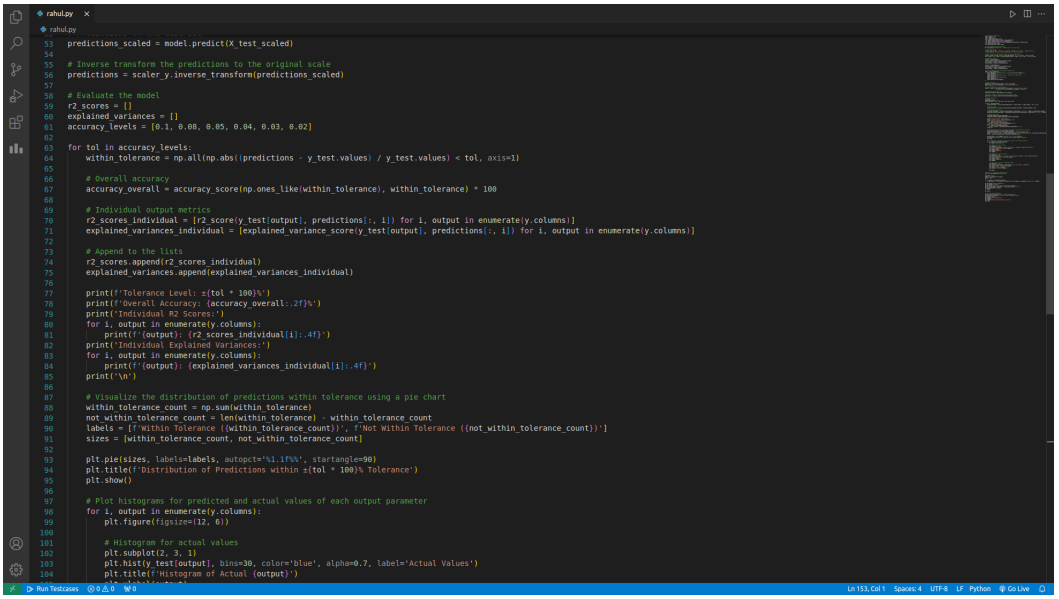


III.3. Training of ML model

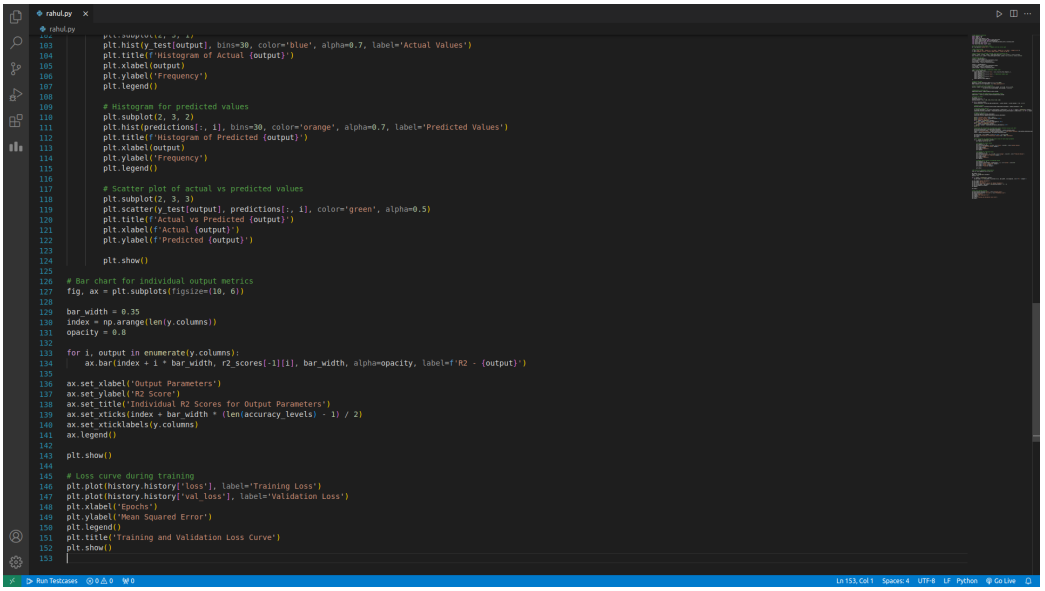
The ML model is Artificial Neural Network (ANN) model which uses the TensorFlow and Keras libraries to optimize the design of concrete pile foundations. The objective is to predict key parameters crucial for pile design, including the diameter of the pile, area of steel reinforcement, and pitch. This analysis will delve into the various components of the code, from data preprocessing to model evaluation.



```python
# Load necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, explained_variance_score, accuracy_score
from tensorflow import keras
from tensorflow.keras import layers

# Load the dataset from Google Colab
df = pd.read_csv('DataSet.csv')  # Update with the correct path

# Data Preprocessing
X = df[['Load (in KN)', 'Moment in x (in Knm)', 'Moment in y (in Knm)', 'Length (in m)']]
y = df[['Diameter of Pile', 'Area of Steel (mm^2)', 'Pitch (in mm)']]

# Split the data into training, testing, and cross-validation sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
X_cv, X_test, y_cv, y_test = train_test_split(X_test, y_test, test_size=0.2, random_state=42)

# Standardize the data
scaler_X = StandardScaler()
X_train_scaled = scaler_X.fit_transform(X_train)
X_cv_scaled = scaler_X.transform(X_cv)
X_test_scaled = scaler_X.transform(X_test)

scaler_y = StandardScaler()
y_train_scaled = scaler_y.fit_transform(y_train)
y_cv_scaled = scaler_y.transform(y_cv)
y_test_scaled = scaler_y.transform(y_test)

# Build the ANN model with an additional hidden layer
model = keras.Sequential([
    layers.Dense(32, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dropout(0.3),
    layers.Dense(64, activation='relu'),  # Additional hidden layer
    layers.Dropout(0.3),
    layers.Dense(16, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(y_train.shape[1])
])

# Compile the model
optimizer = keras.optimizers.Adam(learning_rate=0.0005)
model.compile(optimizer=optimizer, loss='mean_squared_error')

# Train the model with cross-validation
history = model.fit(X_train_scaled, y_train_scaled, epochs=10, batch_size=32,
                    validation_data=(X_cv_scaled, y_cv_scaled), verbose=1)

# Predictions on the test set
predictions_scaled = model.predict(X_test_scaled)
```



```python
predictions_scaled = model.predict(X_test_scaled)

# Inverse transform the predictions to the original scale
predictions = scaler_y.inverse_transform(predictions_scaled)

# Evaluate the model
r2_scores = []
explained_variances = []
accuracy_levels = [0.1, 0.08, 0.05, 0.04, 0.03, 0.02]

for tol in accuracy_levels:
    within_tolerance = np.all(np.abs((predictions - y_test.values) / y_test.values) < tol, axis=1)

    # Overall accuracy
    accuracy_overall = accuracy_score(np.ones_like(within_tolerance), within_tolerance) * 100

    # Individual output metrics
    r2_scores_individual = [r2_score(y_test[output], predictions[:, i]) for i, output in enumerate(y.columns)]
    explained_variances_individual = [explained_variance_score(y_test[output], predictions[:, i]) for i, output in enumerate(y.columns)]

    # Append to the lists
    r2_scores.append(r2_scores_individual)
    explained_variances.append(explained_variances_individual)

    print(f'Tolerance Level: ±{tol * 100}%')
    print(f'Overall Accuracy: {accuracy_overall:.2f}%')
    print('Individual R2 Scores:')
    for i, output in enumerate(y.columns):
        print(f'{output}: {r2_scores_individual[i]:.4f}')
    print('Individual Explained Variances:')
    for i, output in enumerate(y.columns):
        print(f'{output}: {explained_variances_individual[i]:.4f}')
    print('\n')

    # Visualize the distribution of predictions within tolerance using a pie chart
    within_tolerance_count = np.sum(within_tolerance)
    not_within_tolerance_count = len(within_tolerance) - within_tolerance_count
    labels = ['Within Tolerance ({within_tolerance_count})', f'Not Within Tolerance ({not_within_tolerance_count})']
    sizes = [within_tolerance_count, not_within_tolerance_count]

    plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
    plt.title(f'Distribution of Predictions within ±{tol * 100}% Tolerance')
    plt.show()

    # Plot histograms for predicted and actual values of each output parameter
    for i, output in enumerate(y.columns):
        plt.figure(figsize=(12, 6))

        # Histogram for actual values
        plt.subplot(2, 3, 1)
        plt.hist(y_test[output], bins=30, color='blue', alpha=0.7, label='Actual Values')
        plt.title(f'Histogram of Actual {output}')
```



Data Preprocessing:

1. Data Loading: The code starts by loading a dataset from a CSV file using the Pandas library. The dataset presumably contains information related to applied loads, moments, lengths, and design parameters of concrete piles.

2. Feature Selection: The input features (`X`) and output targets (`y`) are then defined. Input features include load, moments in x and y directions, and pile length. Output targets consist of diameter of pile, area of steel, and pitch.

3. Data Splitting: The dataset is split into training, testing, and cross-validation sets using the `train_test_split` function from scikit-learn.

4. Standardization: Both input features and output targets are standardized using `StandardScaler` to ensure that each feature contributes proportionally to the model training.

Neural Network Architecture:

1. Model Architecture: The neural network is constructed using Keras Sequential API. It consists of an input layer, a hidden layer with 32 neurons and ReLU activation, a dropout layer for regularization, an

additional hidden layer with 64 neurons, another dropout layer, a third hidden layer with 16 neurons, and a final output layer.

2. Compilation:The model is compiled using the Adam optimizer and mean squared error loss function, suitable for regression tasks.

Model Training:

The model is trained using the training set and validated on the cross-validation set for 500 epochs with a batch size of 32. Training history is stored in the `history` variable.

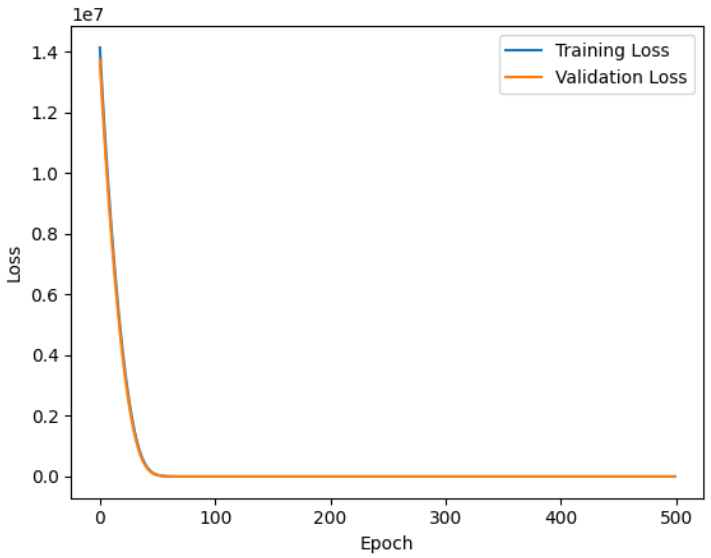
Model Evaluation:

1. Prediction: The trained model is used to predict the output targets on the test set.

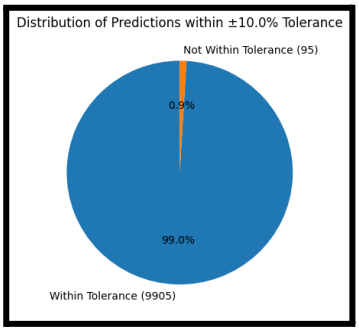
2. Inverse Transformation: Predictions are inverse-transformed to the original scale using the `StandardScaler` to facilitate result interpretation.

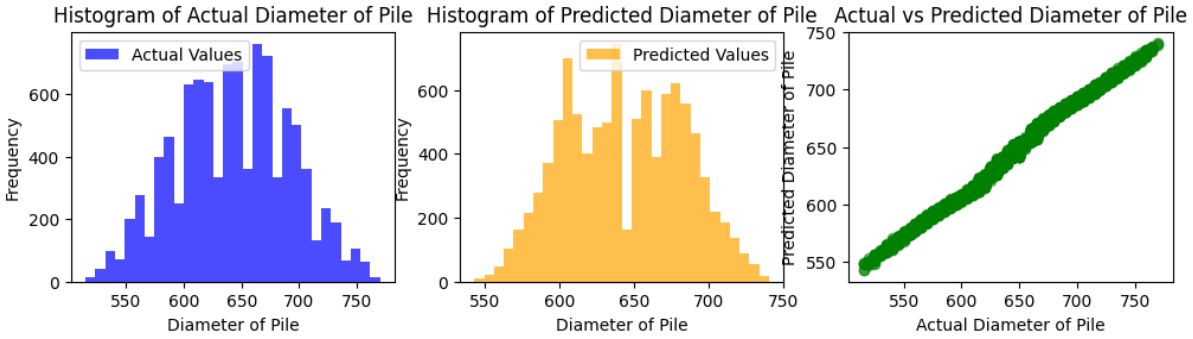
3. Evaluation Metrics: The code assesses the model's performance using various metrics, including R2 scores, explained variances, and accuracy levels within specified tolerances.

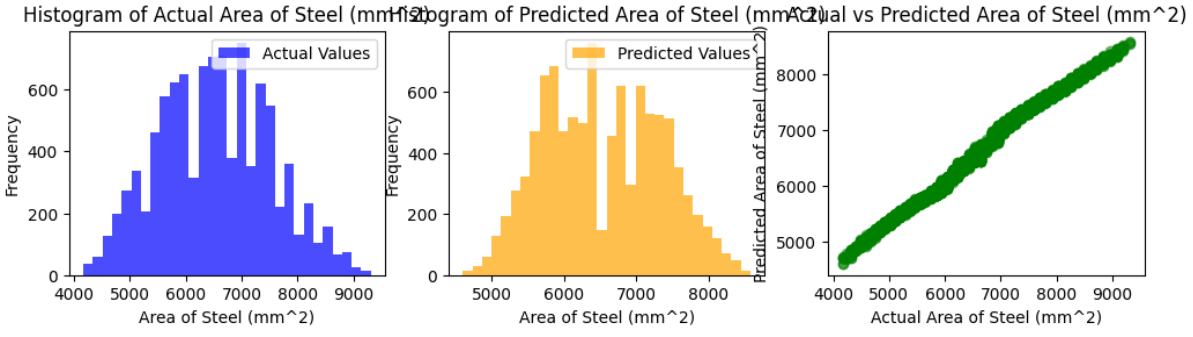
4. Visualization: Visualizations include pie charts depicting the distribution of predictions within tolerance, histograms comparing actual and predicted values, scatter plots for each output parameter, and a bar chart illustrating individual R2 scores.

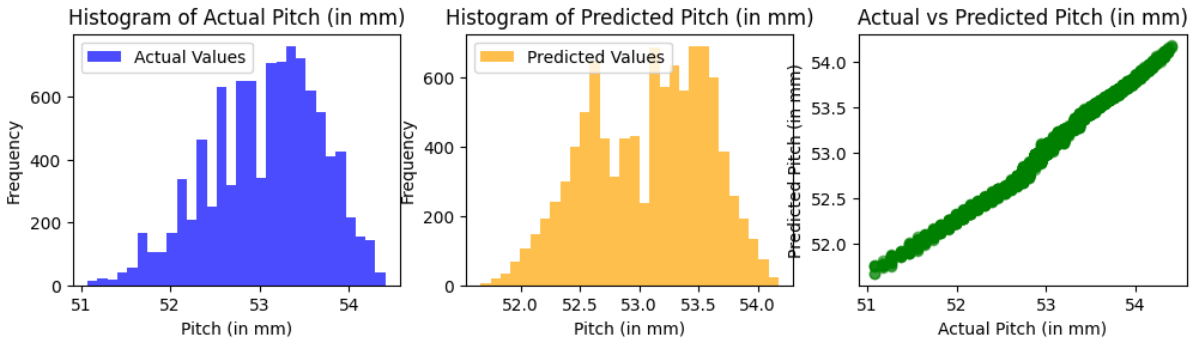


Tolerance Level: ±10.0%
Overall Accuracy: 99.05%
Individual R2 Scores:
Diameter of Pile: 0.9558
Area of Steel (mm^2): 0.9556
Pitch (in mm): 0.9524
Individual Explained Variances:
Diameter of Pile: 0.9560
Area of Steel (mm^2): 0.9557
Pitch (in mm): 0.9532

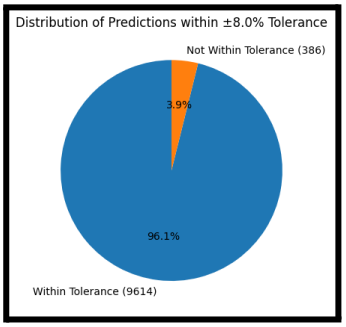


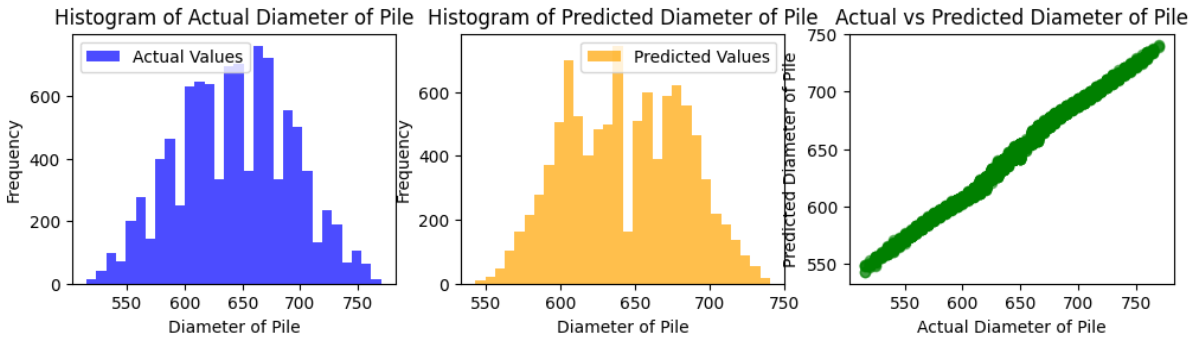


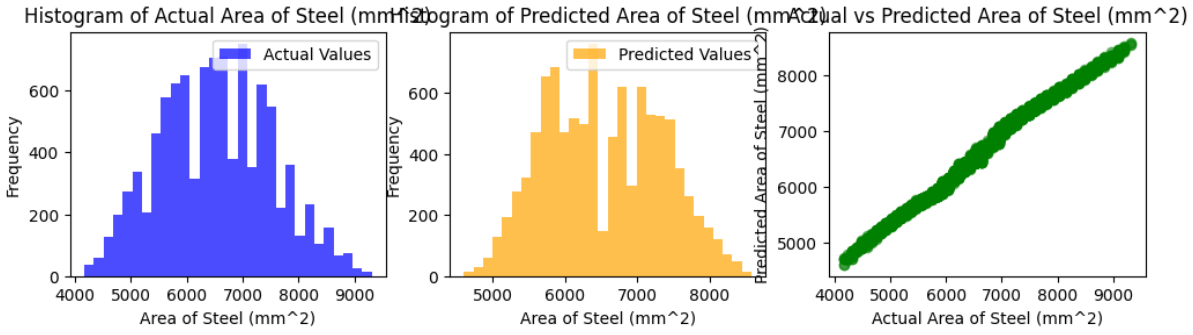


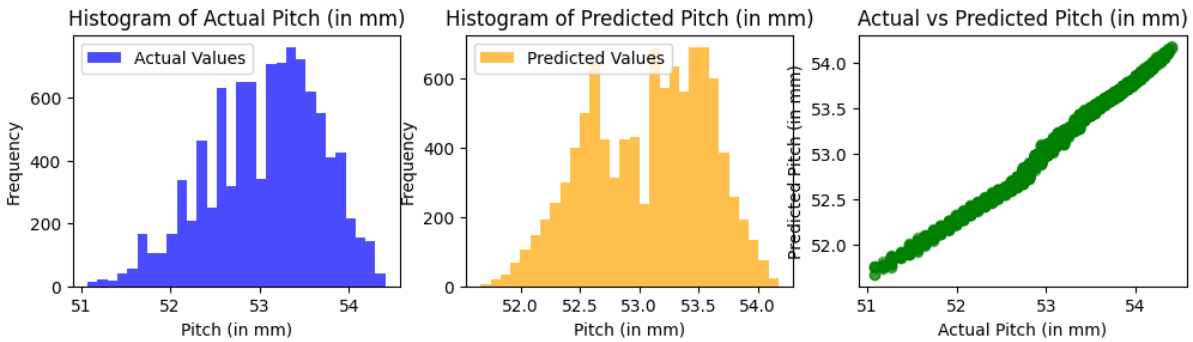


Tolerance Level: ±8.0%
Overall Accuracy: 96.14%
Individual R2 Scores:
Diameter of Pile: 0.9558
Area of Steel (mm^2): 0.9556
Pitch (in mm): 0.9524
Individual Explained Variances:
Diameter of Pile: 0.9560
Area of Steel (mm^2): 0.9557
Pitch (in mm): 0.9532

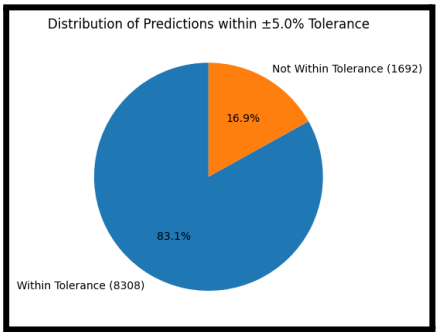


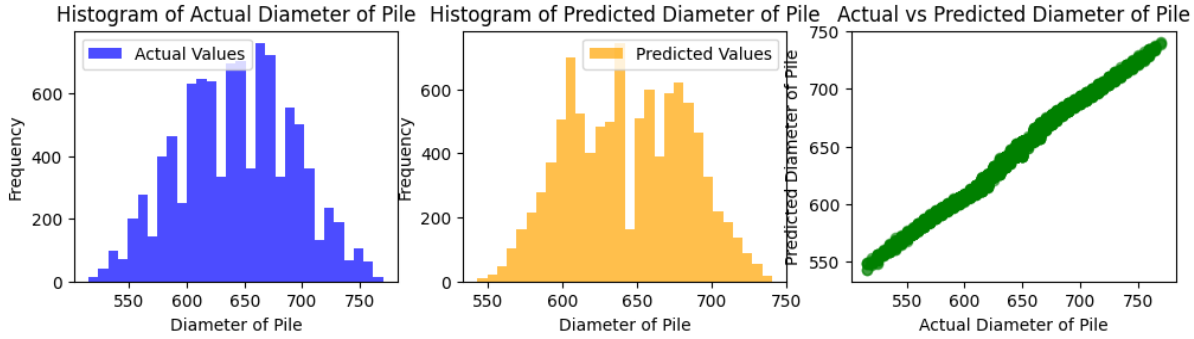


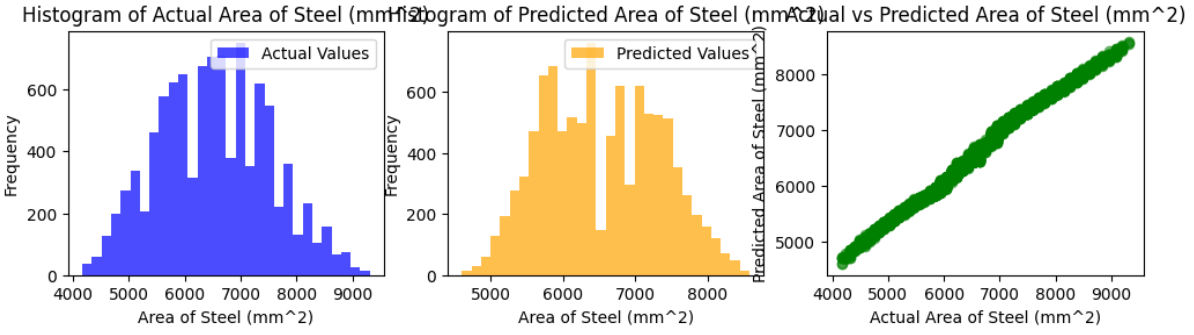


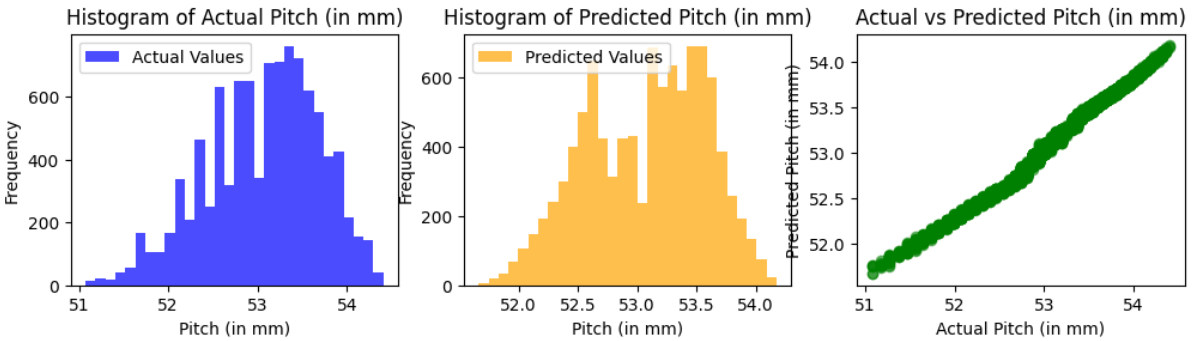


Tolerance Level: ±5.0%
Overall Accuracy: 83.08%
Individual R2 Scores:
Diameter of Pile: 0.9558
Area of Steel (mm^2): 0.9556
Pitch (in mm): 0.9524
Individual Explained Variances:
Diameter of Pile: 0.9560
Area of Steel (mm^2): 0.9557
Pitch (in mm): 0.9532

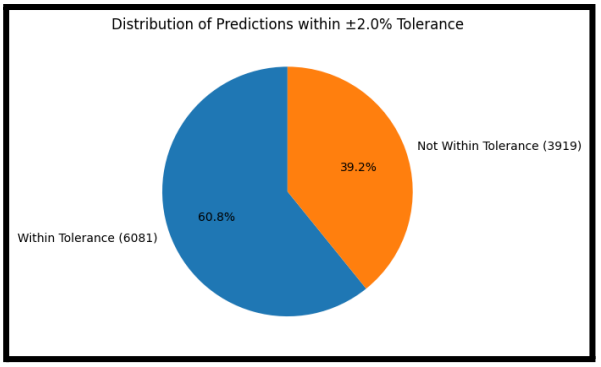


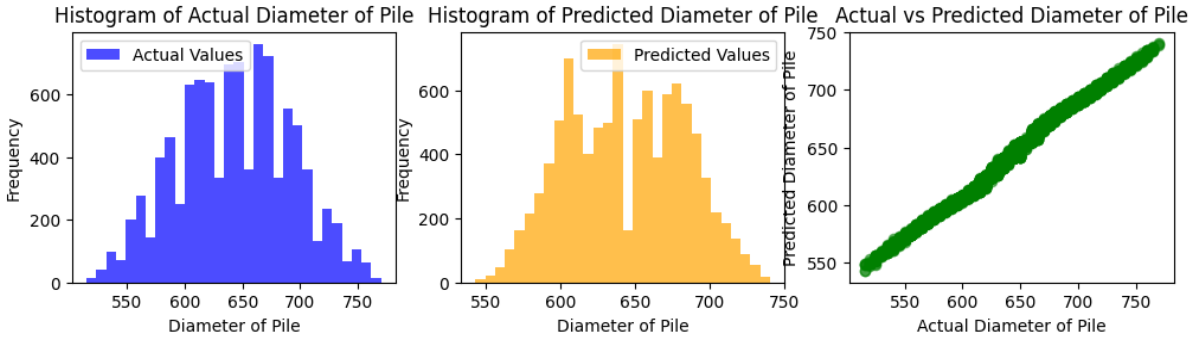


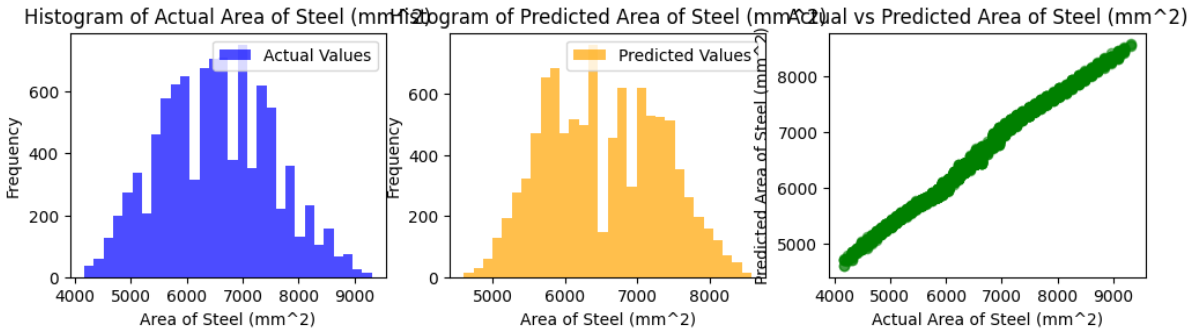


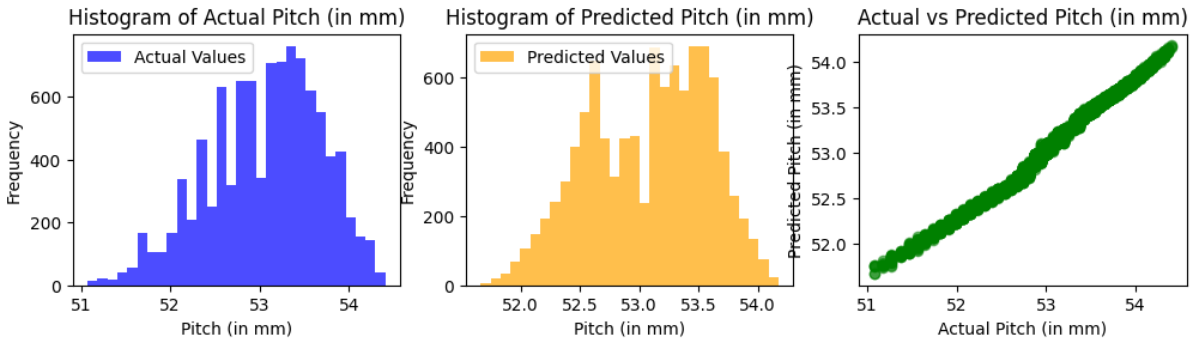


Overall Accuracy: 60.81%
Individual R2 Scores:
Diameter of Pile: 0.9558
Area of Steel (mm^2): 0.9556
Pitch (in mm): 0.9524
Individual Explained Variances:
Diameter of Pile: 0.9560
Area of Steel (mm^2): 0.9557
Pitch (in mm): 0.9532









IV. RESULTS AND DISCUSSIONS

- In this study, we have conducted a design optimization of pile foundation for concrete piles. The design was carried out using a combination of manual calculations and computer-based design tools. The design optimization process involved varying the axial load, moments, pile length to achieve the most economical design while meeting the required load capacity.

- Manual calculation is an important part of the design process for pile foundation. The use of manual calculations also allowed for a deeper understanding of the design process and the factors that influence the design of the pile foundation.

- We have successfully created the dataset of two lakh test cases for design of pile foundation which is used to train our ML model.
- The accuracy with +-10% is very good but it decreases as we decrease the tolerance to +-2%.

- In conclusion, the design optimization process used in this study was effective in achieving a more economical pile foundation design while maintaining the required load capacity. The results of this study can be useful in informing the design of pile foundations for various types of structures.

V. CONCLUSIONS AND FUTURE WORK

- The implemented machine learning (ML) model demonstrates its efficacy in optimizing the design of concrete pile foundations. The utilization of an Artificial Neural Network (ANN) facilitates the prediction of critical design parameters, including diameter, steel area, and pitch, showcasing the model's ability to capture complex relationships within the dataset.
- The evaluation metrics, such as R2 scores, explained variances, and accuracy levels within specified tolerances, provide a comprehensive assessment of the model's accuracy and reliability. The model's capacity to predict within defined tolerances underscores its practical utility in real-world scenarios.
- The visualizations accompanying the analysis, including pie charts, histograms, scatter plots, and bar charts, offer valuable insights into the distribution of predictions, model behavior, and individual output metrics. These visual aids enhance the interpretability of the model's performance.
- The incorporation of dropout layers in the neural network contributes to model robustness by preventing overfitting. This ensures that the model generalizes well to unseen data, a critical aspect for its application in diverse construction projects.
- The loss curves during training indicate the convergence of the model, suggesting that the optimization process effectively minimizes the mean squared error. This convergence enhances confidence in the stability and reliability of the trained model.
- To enhance the model's generalization capabilities, future work could focus on expanding the dataset. Including a more extensive range of scenarios, soil types, and structural configurations can improve the model's adaptability to a broader spectrum of real-world conditions.
- Further exploration of hyperparameter tuning may optimize the model's architecture and training parameters. This includes adjusting the number of neurons in each layer, dropout rates, and learning rates to identify configurations that improve predictive accuracy.
- The integration of additional geotechnical data, such as soil properties and subsurface conditions, can refine the model's

accuracy. Incorporating a more detailed understanding of the foundation's interaction with the underlying soil can lead to more accurate predictions.

- ❖ Experimenting with ensemble methods, such as combining predictions from multiple models, could enhance the overall predictive power. Ensemble techniques often mitigate the weaknesses of individual models and improve overall performance.
- ❖ Implementing a system for real-time monitoring and feedback during the construction phase can provide an opportunity for model refinement. The model could be continuously updated based on the feedback obtained from the actual performance of constructed foundations.
- ❖ Integrating the ML model with existing design codes and standards can create a powerful tool for engineers. This involves ensuring that the model's predictions align with industry standards and can be seamlessly integrated into the existing design processes.
- ❖ Exploring the applicability of the model across various construction projects and foundation types can broaden its impact. This includes adapting the model for different types of piles, foundation materials, and structural configurations.

In summary, the presented ML model serves as a promising tool for design optimization of pile foundations. Future efforts can leverage these insights to refine the model's capabilities, expand its applicability, and enhance its integration into the broader landscape of civil engineering practices.

VI. REFERENCES

1. Code

- IS Code 456:2000 (Plain and Reinforcement Concrete), IS Code 2911(Part-1/Sec-4):2010 (Design and construction of pile foundations) and SP 34:1987(Handbook on Concrete Reinforcement and Detailing).

2. Journal

- https://nitsri.ac.in/Department/Civil%20Engineering/CGE-202_7_Pile_Foundation_Design__A_Student_Guide.pdf

- https://www.youtube.com/watch?v=hzcyUbwgqqQ

- https://onlinepubs.trb.org/Onlinepubs/nchrp/nchrp_syn_42.pdf

3. Books

- Das, B.M. (2010). Principles of Foundation Engineering. Cengage Learning, Boston.

- Fellenius, B.H. (1984). "The design of pile foundations." Geotechnique, 34(4), 449-489.

- N. Krishna Raju(2008) “Reinforced Concrete Design”.